IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 16 August 2023, accepted 5 September 2023, date of publication 8 September 2023,
date of current version 13 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3313264

RESEARCH ARTICLE

Multi-UAV Cooperative Mission Assignment Based on Hybrid WAFC-RRAS Algorithm

GANG HUANG, MIN HU, XUEYING YANG, AND FEIYAO HUANG
Department of Aerospace Science and Technology, Space Engineering University, Beijing 101416, China
Corresponding author: Min Hu (jlhm09@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 61403416.

ABSTRACT Multi-UAV cooperative mission assignment is an important research direction in the field of UAV research. In the planning process, the assignment conflict between UAVs and mission points and the solution efficiency are the key difficulties in multi-UAV cooperative mission assignment. Based on the Weighted Approximate Flight Cost (WAFC) method and the Reduced Redundant Assignment Scheme (RRAS) algorithm, a multi-UAV cooperative mission problem considering different assignment models is proposed. The contributions of this study are: firstly, the approximate flight cost matrix is constructed based on the vertical distance between the UAV and the mission point, and the matrix is applied to calculate the initial flight distance of the UAV, which fully takes into account the effects of mountain type and radar threat on the mission assignment. Secondly, the auxiliary flight cost method and the spatial mapping matrix are constructed, which maps the discrete approximation of the flight cost to the continuous space and solves the UAV's local conflict problem in the assignment process. Finally, an adaptive selection mutation strategy based on iterative individual fitness values is proposed to reduce the redundancy of candidate assignment schemes and improve the efficiency of the planning system. The simulation results verify that the algorithm has high cooperative capability, good robustness and fast solution speed when dealing with cooperative multi-UAV mission assignment planning.

INDEX TERMS Vertical cut method, approximate flight cost method, mapping strategies, assignment conflicts, multi-UAV cooperative mission assignment.

I. INTRODUCTION

Multi-UAV cooperative mission assignment (MUCMA) refers to the mission scheduling and division of multiple UAVs according to complex mission environment and different mission requirements, so as to achieve the highest mission efficiency and the lowest combat cost of UAVs [1], [2], [3]. With the increasing variety of UAVs performing missions and the mission requirements becoming more and more complex, it is difficult for a single UAV to complete the reconnaissance and saturation attack in the enemy area by itself with its limited load. However, MUCMA can handle various complex missions through cooperation among multiple UAs to improve the efficiency of mission completion. However, MUCMA also has many difficulties, for example, in the assignment process, MUCMA not only needs to consider the differences in performance between UAVs, the complexity of the simulation environment and the priority of mission execution, but also to avoid problems such as UAV assignment conflicts and long planning times [4]. Therefore, the research of MUCMA still faces great challenges.

In the last decade, researchers have proposed many excellent mission assignment algorithms, which are divided into two main categories, centralized mission planning and distributed mission planning, depending on the different structures of cooperative control of multiple UAVs [5], [6].

Centralized algorithm (CA)-based cooperative mission assignment adopts a centralized algorithm that can centrally control and coordinate the assignment relationship between UAVs and mission points. Tang et al. [7] proposed a heterogeneous UAV cooperative multi-tasking assignment problem with objective priority constraints based on fuzzy c-mean

The associate editor coordinating the review of this manuscript and approving it for publication was Guillermo Valencia-Palomo.

clustering idea and ant colony optimisation algorithm, which fully considered the performance evaluation of the original algorithm in a dynamic environment as well as the extension of the algorithm, which is suitable for dealing with more realistic dynamic simulation environments. Zhen et al. [8] presented a hybrid algorithm based on artificial potential field ant colony optimization for UAV cooperative planning. The algorithm uses the artificial potential field method to construct the simulation environment and the centralized ant colony algorithm to improve the local search capability of each UAV with high assignment efficiency under different simulation environments. Wang et al [9] proposed a unified modeling approach for cooperative mission assignment of complex multi-constrained UAVs, using the spatial vertical cut to calculate the approximate flight cost of the UAV, and optimizing the mission assignment algorithm using the flight cost matrix. It is simple and flexible when used in cases with low dimensionality, but it oversimplifies the complex simulation environment and over-relies on the control center; hence, the assignment model has poor scalability.

Distributed algorithm (DA)-based cooperative mission assignment is computationally flexible, highly scalable, and can handle different levels of assignment separately, but the system's robustness decreases as the number of UAVs increases [10]. Liu et al. [11] designed a mechanism to avoid UAV assignment conflicts and proposed a distributed adaptive algorithm based on individual and group decision making. Individual decision-making is applied in UAV intelligence and adjustment, and group decision-making is utilized in the leadership mechanism and joint decision-making. Shirani et al. [12] designed a distributed controller based on suboptimal LQR-PID extended system control laws for coordinated mission assignment involving the cargo transportation of multiple UAVs. The controller prioritizes loads of the UAVs and optimizes the overall mission planning while avoiding conflicts and collision obstacles. Qiu et al. [13] proposed a distributed optimal control framework to transform the UAVs cooperative assignment problem into a multi-objective optimization problem. In addition, a multi-objective pigeon heuristic algorithm was proposed to coordinate the local interaction information of UAVs in a complex environment, effectively avoiding mission assignment conflicts.

Although the above methods are able to solve MUCMA in different environments, with the increasing demand for cooperative multi-UAV assignment systems, the number of UAVs and mission points has increased dramatically, making it difficult for the planning system to make assignment decisions in an effective time. At the same time, the inconsistency in the number of UAVs and mission points increases the blindness of the assignment mechanism and leads to assignment conflicts in the planning system [14], [15], [16]. The specific difficulties are described below:

- MUCMA is divided into various assignment models in accordance with the relationship between the numbers of UAVs and mission points, and different assignment models have different assignment mechanisms and strategies. In a case with UAVs and mission points, if the same spatial conversion and mapping mechanism is chosen, the UAVs and mission points will cause assignment conflicts, which will greatly reduce the efficiency of the assignment system. Therefore, how to construct reasonable mapping strategies in accordance with different assignment models is crucial to avoid assignment failures.
- With the dramatic increase in the number of UAVs and mission points, the computational complexity of the planning system will increase significantly. At the same time, after many iterations, a large number of candidate assignment schemes are generated, and it is difficult to solve the optimal assignment scheme in a short time. Therefore, it remains a difficult problem for MUCMA to solve the optimal assignment scheme effectively under the situation of data redundancy.

In order to solve these difficulties, this study adopts fixed-wing UAV as the research object and aims to avoid assignment conflicts, reduce redundant data and shorten planning time. The main contributions are summarized below.

- In constructing the assignment model: To reduce the complexity of the assignment system, the parameters of UAVs in some studies used the same flight distance, flight speed, and so on, but in the actual combat environment, UAV formation often consists of UAVs with different performances.. Therefore, this study focuses on the influence of heterogeneous UAVs on MUCMA. A weighted hybrid fitness function for the cooperative mission assignment of UAVs is constructed. The sum of flight cost, flight time, and constraint violation is used as the fitness function, and constraint violation is applied as the penalty function at the important evaluation stage of the evolutionary algorithm.
- In terms of resolving assignment conflicts: Different assignment models have different assignment mechanisms and mapping strategies, and reasonable mapping strategies must be considered in conjunction with the physical meaning of the actual problem. In this study, the flight cost between UAVs and mission points is used as the mapping medium, and the main order matching method and the auxiliary flight cost method are employed to map discrete UAVs and mission points to a continuous flight cost space. The method effectively avoids UAVs assignment conflicts.
- In terms of improving the performance of the algorithm: In order to reduce the redundancy of candidate assignment schemes and avoid the algorithm falling into local optimum. This paper adaptively selects different mutation strategies according to the calculation of individual fitness values in each generation, which reduces the redundancy rate of candidate assignment schemes and enhances the diversity of candidate assignment schemes.

In order to detect the performance of the improved algorithm, we present three sets of simulation experiments

and evaluate the feasibility and stability of the computer simulation results. The experimental results show that the method proposed in this paper is superior in solving multi-UAV cooperative mission assignment conflicts and planning efficiency, and is a good reference for different types of multi-agent mission planning.

The rest of the paper is organized as follows. Section II introduces three different assignment models and assignment mechanisms, constructing the MUCMA fitness function and the co-constraints for UAVs. Section III describes the construction of approximate flight costs and spatial mapping strategies. Section IV introduces the classical differential evolution algorithm and the method for reducing redundant data, and describes the MUCMA execution framework. In Section V, the WAFC-RRAS algorithm is analyzed for effectiveness and stability and its performance is compared with other algorithms. A brief summary of the conclusions is given in Section VI.

II. MULTI-UAV COOPERATIVE MISSION ASSIGNMENT MODEL

This section first introduces the different simulation scenarios of MUCMA, the assignment mechanism and the difficulties in solving each assignment model. Then, the method of experimental modeling is introduced. Finally, the method of constructing the fitness function and the co-constraints of MUCMA are presented.

A. COOPERATIVE MISSION ASSIGNMENT SCENARIOS AND ASSIGNMENT MECHANISMS

MUCMA is a multi-model, multi-constrained, computationally difficult and complex optimization *NP* problem [17], [18]. During the planning process, obstacles such as mountains and radars have to be considered, as well as various factors such as the number and performance of UAVs, the execution timing of UAVs, and the weighting of missions. This requires MUCMA to consider the complexity of the simulation environment, the synergy between heterogeneous UAVs and a clear assignment mechanism.

1) COOPERATIVE MISSION ASSIGNMENT SCENARIOS

The MUCMA simulation scenario is described in detail: in a complex 3D environment the UAVs finds a set of corresponding mission assignment sequences. According to a cooperative mission assignment fitness function and constraints. The assigned UAVs should have the shortest flight cost, the shortest flight time and the smallest constraint violation.

Figure 1 presents the four different assignment models of MUCMA, namely $N = M$, $N > M$, $N < M$, and $N \approx M$. The blue circle indicates the number of UAVs $N = [U_1, U_2, \ldots, U_n]$. The yellow pentagram indicates the number of mission points $M = [M_1, M_2, \ldots, M_m]$. The grey dashed line and grey solid line indicate the assignment sequence of UAVs that may strike and eventually strike the mission points, respectively. The purple dashed line and purple solid line indicate the assignment sequence of UAVs that may strike and eventually strike the mission points sequentially, respectively. Model $N \approx M$ combines the assignment mechanisms of the three models. The assignment mechanism of this model is extremely complex and requires consideration of not only whether the UAVs are assigned in conflict, but also whether the UAVs conform to the optimal assignment mechanism.

2) MUCMA'S MISSION ASSIGNMENT MECHANISM

This study focuses on three MUCMA assignment models, namely $N = M$, $N > M$, and $N < M$. The different assignment models have different assignment mechanisms and assignment difficulties, which are described as follows:

When $N = M$, the mission assignment mechanism of this model is relatively simple. Each UAV is assigned a mission point, each mission point is assigned a UAV, and the UAV and mission point have one-to-one correspondence. The key to solving the problem is to avoid conflicts caused by mission competition and use cooperation to reduce the probability of destruction so that the MUCMA fitness value is maximized.

When $N > M$, the mission assignment mechanism of this model is highly complex. Each mission point can be assigned multiple UAVs, each UAV must execute a mission point, and the UAVs and mission points have a many-to-one relationship. The number of UAVs in this assignment model is not balanced with the number of mission points, resulting in high computational complexity and high probability of mission competition conflict in the assignment process. The key to solving the problem is to reduce the probability of conflict in the assignment.

When $N < M$, the model is the most complex of the three models. Each UAV can handle multiple mission points, each mission point must be assigned a UAV, and the UAVs and mission points have a one-to-many relationship. This assignment model has multiple missions to be performed by one UAV, and in terms of this involves UAVs performing mission point timings. The key to solving this model is the need to use fewer resources to complete more complex missions and to be able to deal effectively with mission point weighting relationships.

B. MODELLING OF THE SIMULATION ENVIRONMENT

In this paper, two kinds of threats are considered: mountain and radar detection. Currently, the impact of mountain on UAV mission execution is rarely considered in the literatures, because the irregular simulation environment of mountain is very challenging. Radar is the most common detection threat for UAV penetration. The simulation environment is shown in Figure 2. Figure 2(a) is the unified simulation environment of the two obstacles. The right figure of Figure 2 is the schematic diagram of radar detection and mountain respectively.

- Mountain

The Digital Elevation Model (DEM) is a digital simulation of the mountain through mountain elevation data. In this paper, USGS 24kdem is used as a mountain obstacle, and

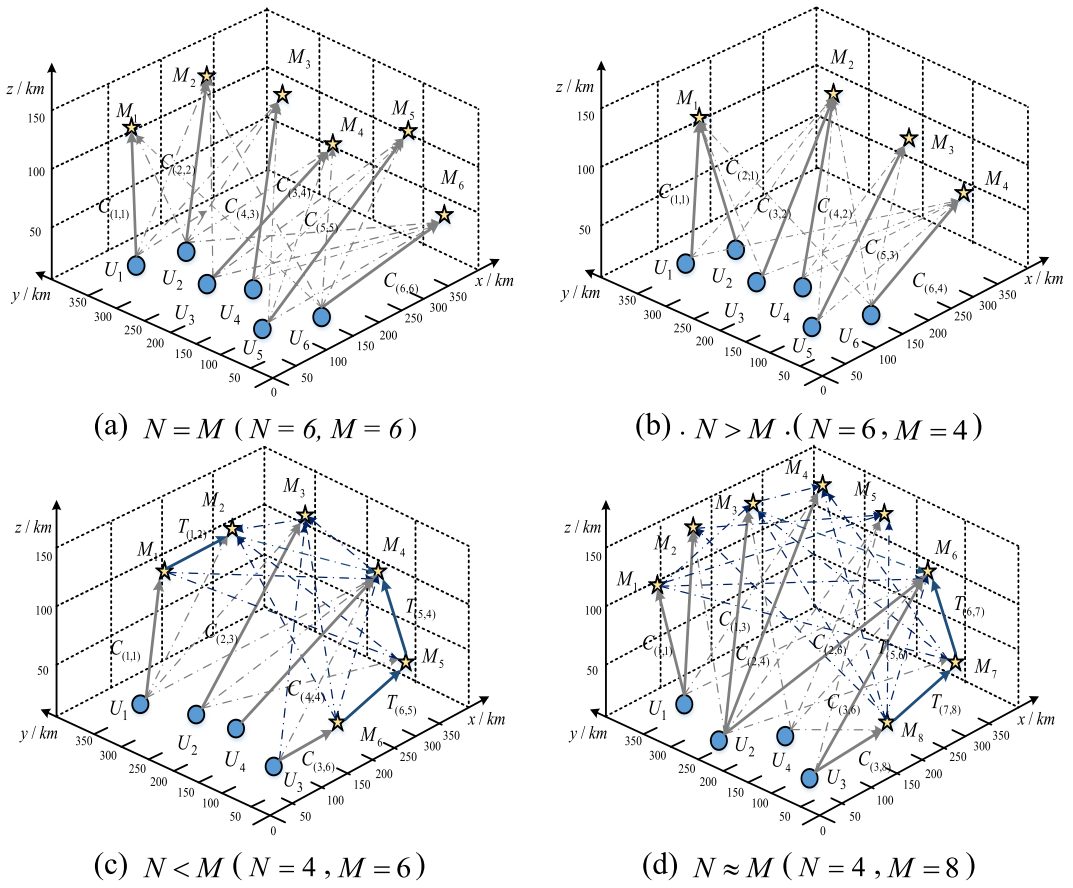
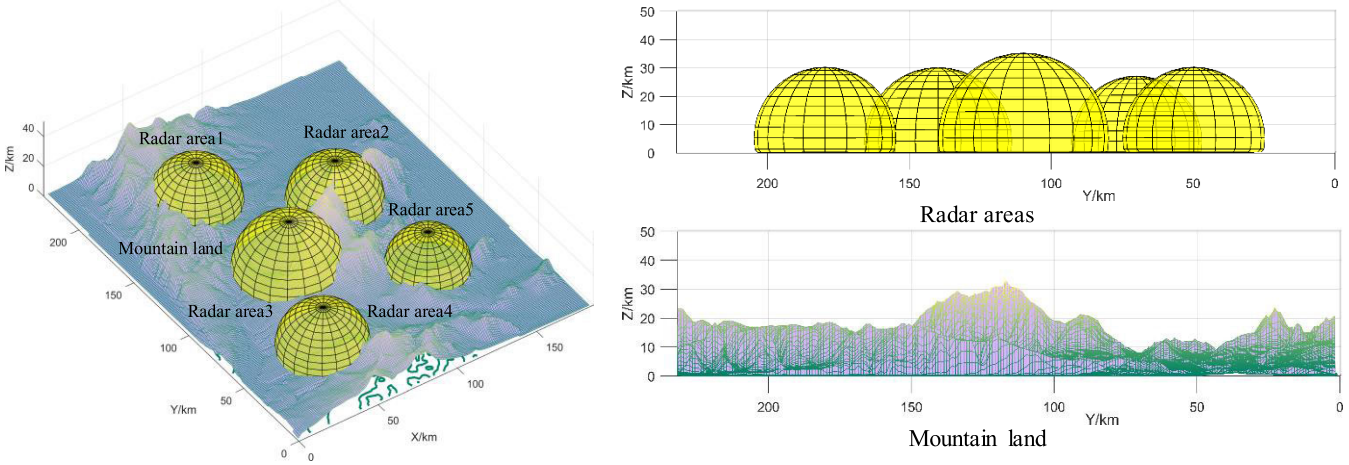


FIGURE 1. Schematic diagram of MUCMA under different assignment models.



(a) Unified simulation environment

(b) Decomposed simulation environment

FIGURE 2. Environmental modelling of MUCMA.

the format USGS 1:24000 digital elevation map file is read, which contains the real longitude, latitude and height. During the planning process, the planning system must effectively avoid UAVs crossing the mountainous terrain causing UAVs to be destroyed.

- Radar detection

Radar from the purpose of operation can be divided into fire control radar and early warning radar, it through the target's electromagnetic wave reflection for analysis to determine the location of the target. Radar detection performance

is susceptible to environmental and signal influences, and it is difficult to describe the detection characteristics of a radar threat considering various factor states, assuming that the radar threat is established on flat terrain with horizontal distance and detection height of the detection area boundary as in equation (1).

$$h_B = K_B \cdot L_B^2 \tag{1}$$

Formula (1), h_B is the radar detection height; L_B is the radar detection horizontal distance; K_B is the radar detection performance parameters. The above formula can be calculated from the maximum distance $d_{\max}$ of the radar detection area on the radar detection boundary.

$$d_{\max} = \sqrt{h_B^2 + L_B^2} \tag{2}$$

From equation (1) and equation (2), it can be seen that as the UAV is further away from the radar, the lower the probability of radar detection. The probability of radar detection of the target can be approximated by equation (3).

$$P_R(d_R) = \begin{cases} 0, & d_R \geq d_{R\max} \\ (1/d_r)^4, & d_{R\min} < d_R < d_{R\max} \\ 1, & d_R < d_{Rmin} \end{cases} \tag{3}$$

C. CONSTRUCTING THE FITNESS FUNCTION AND CONSTRAINTS FOR MUCMA

1) FITNESS FUNCTION FOR MUCMA

MUCMA is equivalent to the assignment problem in operations research. Assume that N UAVs are combatting M missions distributed in different locations in the simulation environment. MUCMA needs to be able to handle different assignment models in consideration of the shortest flight distance, shortest total flight time, and smallest constraint violation. The MUCMA fitness function is given as follows:

$$\begin{aligned} \min f(x) = \alpha \sum_{i=1}^{N}\sum_{j=1}^{M} w_j d_{(i,j)} D_{(i,j)} \\ + \ldots + \beta(\max(\sum_{i=1}^{N}\sum_{j=1}^{M} t_{(i,j)} D_{(i,j)})) \\ + \gamma \sum_{k=1}^{K} c_k, \alpha + \beta + \gamma = 1 \end{aligned} \tag{4}$$

where $d_{(i,j)}$ represents the approximate flight distance from UAV i to mission point j. w_j is the weight of mission j, and $0 < w_j \leq 1$. The larger the value of w_j is, the more important mission j is. $t_{(i,j)}$ represents the time from UAV i to mission point j. c_k denotes the penalty corresponding to the constraint. α, β and γ are scaling factors used to maintain the balance of the polynomials in the fitness function, which keeps the flight cost, flight time, and constraint violation in the same order of magnitude. $D_{(i,j)}$ is the decision variable that determines the mapping between the UAV and the mission point, and it has different expressions as follows:

$$D_{(i,j)} = \begin{cases} D_{(i,j)} & N = M \\ D_{([i_s,\ldots,i_l],j)} \forall s, l \in N \text{ and } s, l \in & NN > M \\ D_{(i,[j_p,\ldots,j_q])} \forall p, q \in M \text{ and } p, q \in & NN < M \end{cases} \tag{5}$$

In Equation (5), when $N = M$, $D_{(i,j)}$ denotes a one-to-one correspondence between UAVs and mission points. When $N > M$, $D_{([i_s,\ldots,i_l],j)}$ represents a many-to-one relationship between UAVs and mission points, which means that UAVs can attack the same mission point. When $N < M$, $D_{(i,[j_p,\ldots,j_q])}$ indicates a one-to-many relationship between UAVs and mission points, which means that a single UAV can attack a group of mission points.

2) CONSTRUCTING COOPERATIVE CONSTRAINTS

UAV and mission point decision variable constraints: When $N \geq M$, the number of UAVs is greater than or equal to the number of mission points. The mission decision is as follows: each UAV must execute one mission point, and each mission point must be executed by one UAV. When $N < M$, the number of UAVs is smaller than the number of mission points. Then, the mission decision is that one UAV must be assigned to each mission point, and a single UAV can cruise the mission points in the execution space.

$$\bigcap_{i=1}^{N} D_{(i,j)} = 1 \forall j = 1, \cdots, M \quad N \geq M \tag{6}$$

$$\bigcap_{j=1}^{M} D_{(i,j)} = 1 \forall i = 1, \cdots, N \quad N < M \tag{7}$$

Equations (6) and (7) represent the relationships corresponding to the two different decisions.

•Maximum flight constraint $MaxDis(km)$: Under different assignment models, the maximum flight cost refers to the sum of the approximate flight costs of all assignment relationships, and this constraint reflects each UAV's own performance, such as fuel consumption and effective communication.

$$\sum_{i=1}^{N}\sum_{j=1}^{M} d_{(i,j)} D_{(i,j)} \leq \sum_{k=1}^{M} D_k, \forall k = 1, \ldots, N \tag{8}$$

In Equation (8), $d_{(i,j)}$ denotes the approximate flight cost, and D_k denotes the limited maximum flight distance of each UAV.

•Minimum/maximum speed constraints for UAVs ($SpeFli(km/h)$):

$$V_{(i)} = [V_{(i)\min}, V_{(i)\max}] \tag{9}$$

In Equation (9), $V_{(i)\min}$ denotes the minimum speed of UAV i, and $V_{(i)\max}$ denotes the maximum speed of UAV i.

•Maximum flight time constraint ($MaxTime(h)$): It represents the maximum value of the time consumed by each UAV

to perform the mission in the cooperative assignment result.

$$\begin{cases} \max\left\{\sum_{i=1}^{n}\sum_{j=1}^{m} t_{(i,j)}D_{(i,j)}\right\} \le \sum_{i=1}^{k} T_k \forall k = 1,\ldots,n \\ T_{\max} = \max\{\dfrac{d_{(i,j)}D_{(i,j)}}{v_i}, \forall i,j \in N\} \end{cases} \tag{10}$$

The maximum flight time can be calculated based on the maximum flight constraint in Equation (8) and the maximum and minimum speed constraints in Equation (9).

•Maximum flight height constraint: In order to ensure the safety of the UAV's flight, it is required to fly at an appropriate height. On the one hand, if the UAV flies too high above the ground, it will increase the chances of being detected by radar. Full use should be made of the terrain in order to reduce the risk of detection. On the other hand, if the UAV flies too low, it is likely to collide with mountains or obstacles.

$$dh_j = \left| h_j - \frac{h_{\max} + h_{\min}}{2} \right|, h_{min} \le h_j \tag{11}$$

In equation (11), h_j represents the altitude value j of the UAV route at the h-th waypoint. h_{max} and h_{min} represent the maximum and minimum distances of the UAV from the mountains and the ground.

•Timing constraint between mission points (*OrdMis*): It reflects the order of execution between mission points. Important mission points must be executed first, and the other mission points are executed afterward in accordance with their constraints.

$$\begin{cases} T_j \max < T_{(j+\tau)} \min \\ \forall \tau \in N \text{ and } \tau < m - j \end{cases} \tag{12}$$

Equation (12) indicates that mission point j must be executed later than mission point $j + \tau$, where τ is a positive integer.

•Multi-time-window constraint: Missions are grouped by different time-windows and the mission points within the group cannot be executed beyond the time-window range $[T_{k-1}, T_k]$ of the group.

$$\begin{cases} [T_{k-1}, T_k] \ne \Phi \\ \max\{t_i, \ldots., t_j\} \le T_k \qquad i,j \in 1,\ldots., m \\ \min\{t_i, \ldots, t_j\} \ge T_{k-1} \qquad i,j \in 1,\ldots, m \end{cases} \tag{13}$$

Simultaneous arrival constraint (*SimArr*): It reflects the ability of UAVs in Model $N > M$ to arrive at a mission at the same time and the ability of UAVs to work together to perform a mission, which means that UAVs must arrive at a mission at the same time.

$$\bigcap_{i=1}^{N} [t(i)_{\min}, t(i)_{\max}] \ne \phi \quad N \in 1,\ldots,n \tag{14}$$

In Equation (14), $t(i)$ denotes the execution time of the i-th UAV, and N denotes the number of UAVs participating in the cooperative operation.

•Waiting time constraint (T_{wait}): To ensure that the UAVs in Models $N = M$ and $N > M$ arrive at the specified mission at the same time, some of the UAVs are allowed to wait for some time before departing.

$$T_{wait}(i) \le T_{\max}(i) \tag{15}$$

In Equation (15), $T_{wait}(i)$ denotes the i-th UAV waiting time, and $T_{\max}(i)$ denotes the i-th UAV maximum waiting time.

III. SOLUTION OF APPROXIMATE FLIGHT COST BASED ON VERTICAL TANGENT METHOD

At present, the study of MUCMA is mostly about UAVs striking mission points in the 2D plane or an equal number of UAVs and mission points in the 3D space with a one-to-one assignment relationship [19], [20], [21], [22]. The reasons for this are: 1) the small amount of information in the 2D plane does not require consideration of the impact of 3D data information on the conflicting assignment of UAVs. 2) the use of an equal number of UAVs and mission points only requires consideration of a one-to-one approximate flight cost base construction method, and the transition simplifies the assignment mechanism. Therefore, this section proposes a strategy for solving the approximate flight cost and spatial mapping using the vertical cut method.

A. APPROXIMATE FLIGHT COST METHOD

When the simulation environment is determined, UAVs and mission points can be initially assigned relationships by approximating the flight cost. Currently, some of the literature often uses the straight-line distance between the UAV and the mission point or the distance of the trajectory segment on the Voronoi diagram to approximate the flight cost of the UAVs. These methods are less complex to compute, but do not take into account the probability of destruction of UAVs during the system planning process and are not applicable in a 3D complex environment. Therefore, the method of approximating the flight cost representation greatly determines the reliability and reasonableness of the assignment results. In this section, a vertical slice is used to estimate the flight cost, which makes the approximate flight cost more reasonable in a three-dimensional complex environment.

The main idea of this method is: firstly, the DEM (Digital Elevation Model) is used to describe the geomorphology of the landscape, and at the same time combined with the radar parameters to obtain a unified simulation environment. Then, a vertical horizontal section is made through the line between the UAV and the mission point, and the elevation value of the section intersecting the simulation environment is used as a flight point for the approximate flight path, and the intersecting discrete values are mapped to the new coordinates through a coordinate transformation. Finally, only the distance of each discrete value needs to be calculated. Although this method is not the shortest actual distance of the UAV, it makes full use of the 3D terrain information and is closer to the optimal distance than the method of calculating the

straight-line distance between two points, and is suitable for solving multi-UAV cooperative mission assignment problems in complex 3D environments.

Figure 3 shows the construction method of the approximate flight cost. The black circle and pentagram in Figure 3(a) indicate the location of the UAV and the location of the mission point respectively, the semi-circular sphere indicates the detection area of the radar, and the grey triangle indicates the mountain The grey dashed line indicates the elevation value of the vertical tangent between the UAV and the mission point that intersects the simulated environment. Figure 3(b) shows the approximate flight cost generated in the real simulation environment. Figure 3(c) shows the approximate flight cost after hiding the mountain and radar.

It is worth noting that the calculation of the approximate flight cost consists of two main distance formulas: the Euclidean distance between mountains $L_1 = (U, A, B, \ldots, G, H)$ and the radar detection zone arc length formula $L_2 = (H, I)$. The higher weighting of the approximate flight cost θ_1 for the intersection of the vertical tangent with the mountain is due to the need to avoid damage to the UAVs caused by the UAVs impacting the mountain. The equation for the approximate flight cost is shown in (16).

$$\begin{aligned} Cost_{Approximate} &= \theta_1 \cdot \sum_{j=0}^{E} \left\| K_{ij} K_{i,j+1} \right\| + \theta_2 \cdot \sum_{j=0}^{A} \left\| L_j \right\| \\ &= \theta_1 \cdot \sum_{j=0}^{E} \sqrt{(x_{ij+1} - x_{i,j})^2 + (y_{ij+1} - y_{i,j})^2} \\ &\quad + \ldots + \theta_2 \cdot \sum_{j=0}^{A} n_j \pi r_j / 180^\circ, \theta_1 \gg \theta_2 \end{aligned} \tag{16}$$

In equation (16), K_{ij} and $K_{i,j+1}$ in equation (15) denote the j-th elevation value and the $j+1$-th elevation value respectively. When $j = 0$, K_0 denotes the starting point of the UAV, K_{N+1} denotes the mission point, (x_j, y_j) and (x_{j+1}, y_{j+1}) denote the coordinates of the elevation values K_j and K_{j+1}, n_j and r_j denote the j-th radial angle and radius of the j-th radar, θ_1 and θ_2 denote the weights of the Euclidean distance and radar arc length, respectively, and the value of θ_1 is much larger than the value of θ_2.

B. SPATIAL MAPPING STRATEGIES

MUCMA has different assignment models, and different models have different assignment mechanisms. Mapping the approximate flight cost of discrete space UAVs to a physically meaningful continuous space necessitates the design of an effective mapping strategy. In this section, mapping strategies for different models are designed using the following principles.

1) AUXILIARY FLIGHT COST MATRIX METHOD

In accordance with the assignment mechanisms of different models, the auxiliary flight cost matrix is constructed as

Algorithm 1 Approximate Flight Cost Method

Input: Location of UAVs: U; Location of mission points: M; Mountain parameters: Latitude: lat; Altitude: Zz; Radar parameters: $Radar$.

Output: Approximate flight cost: $Cost_{approximate}$.

1 Experimental simulations with radar and mountain parameters and integration of radar detection area and mountain type data.
2 Set the position of the UAVs and the mission points.
3 ***if*** $(x_i \le x_j \&\& y_i \le y_j \,\|\, x_i \ge x_j \&\& y_i \ge y_j)$ /* Calculate the elevation of all points at a unit distance */
4 ***While*** $(abs(x - x_j) > 1 \,\|\, abs(y - y_j) > 1)$
5 $x = x_i + k/(\sqrt{(1+k^2)}); y = y_i + 1/(\sqrt{(1+k^2)}); M(i,1) = x; M(i,2) = y;$
6 $i{=}i+1;$
7 *end*
8 *else*
9 ***While*** $(abs(x - x_j) > 1 \,\|\, abs(y - y_j) > 1)$
10 $x = x_i - k/(\sqrt{(1+k^2)}); y = y_i - 1/(\sqrt{(1+k^2)}); M(i,1) = x; M(i,2) = y;$
11 $i{=}i+1;$
12 ***end***
13 ***end***
14 $M = [fix(M(:,1)), fix(M(:,2))]$.
15 ***for*** i=1: size(M,1)
16 zheight(i, 1) = Zzx($fix(M(i,2)), fix(M(i,1))$). /* *Approximate flight cost elevation values are obtained by the vertical cut method*/
17 ***end***
18 Use equation (11) to set the flight altitude of the UAVs.
19 Calculate the approximate flight cost of UAVs to mission points using equation (16).

AuM_1, AuM_2, and AuM_3. The auxiliary flight cost matrix stores the flight generation values of each UAV and mission point, and WAFC-RRAS calls the flight generation values of each UAV and mission point during the evolution process. This process reduces the repeated calculation of flight costs and effectively improves the planning efficiency of the algorithm. It is worth noting that AuM_3 stores not only the flight cost $C(U_i, M_j)$ of each UAV executing the mission point, but also the flight cost $T(M_i, M_j)$ between two mission points.

2) MAIN ORDER MATCHING PRINCIPLE METHOD

The first column U_1 of reference cost matrix $Cost$ is compared with the flight cost of the first column U_1 of auxiliary flight cost AuM_1. The smaller flight cost is selected as the mapping medium for $U1$, the corresponding mission point M_5 is outputted, and the corresponding rows and columns are deleted. Then, the execution assignment schemes for the other UAVs are selected in turn. This method effectively avoids repeated mapping of assigned execution relation and

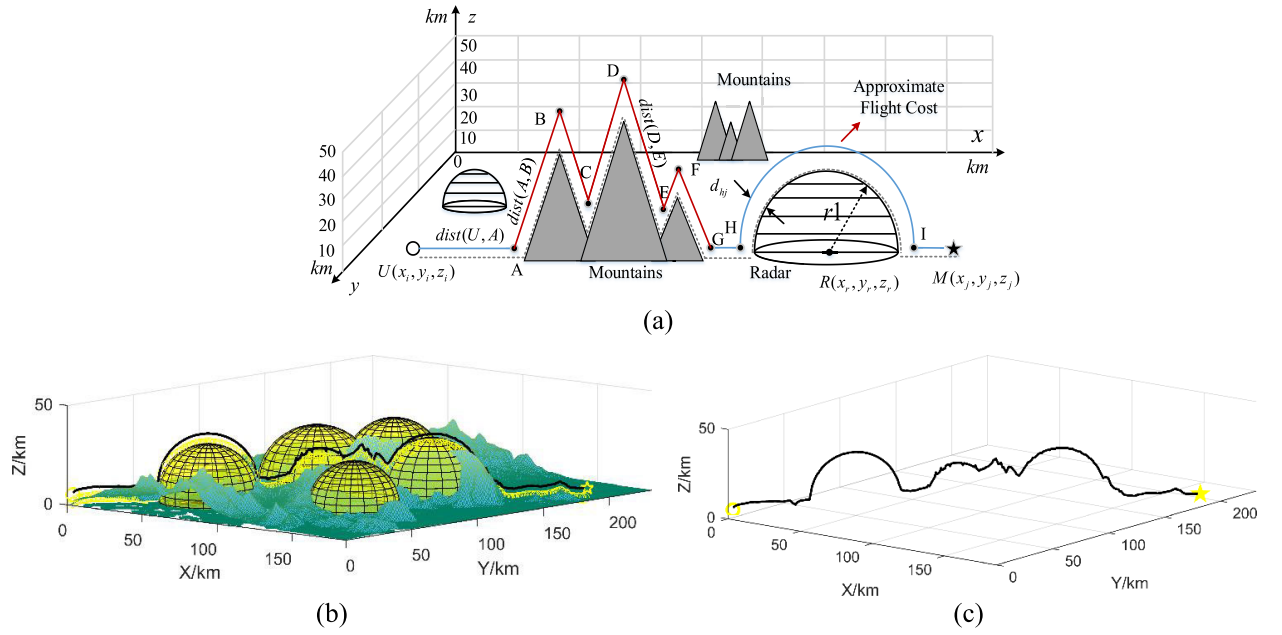


FIGURE 3. Construction method of approximate flight cost.

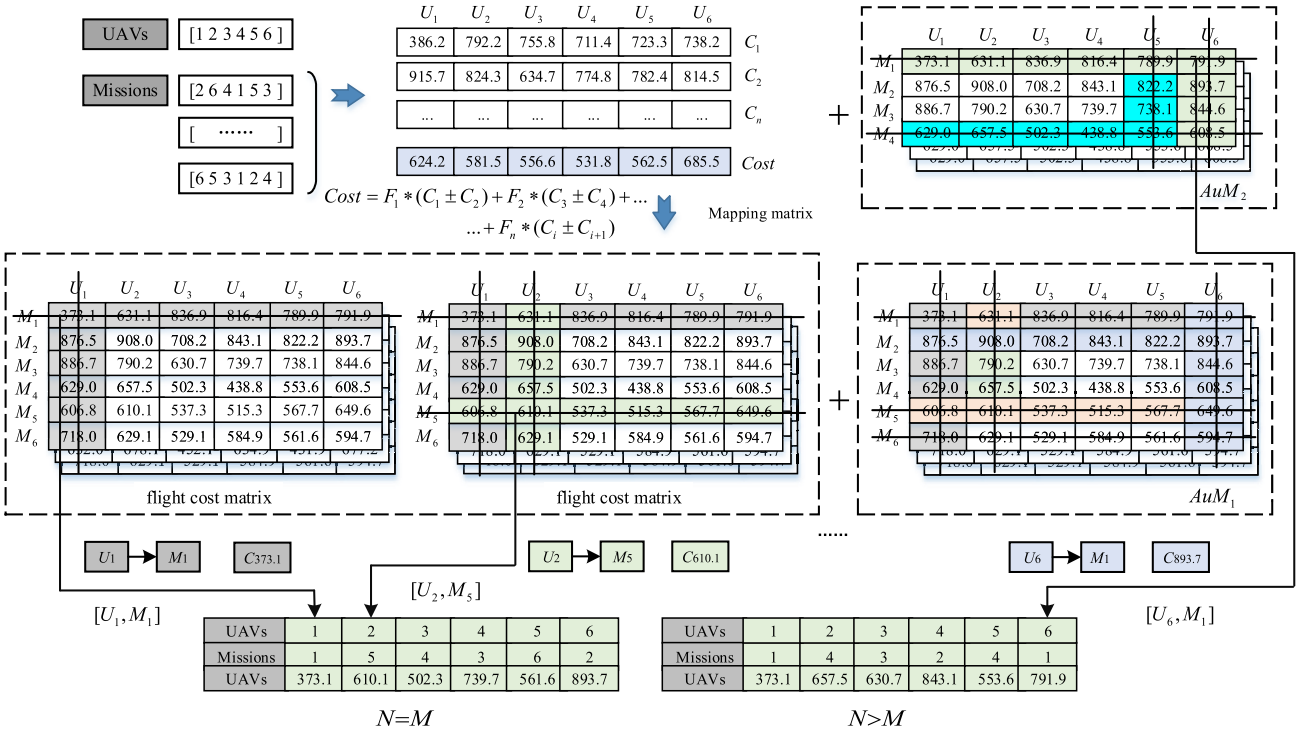


FIGURE 4. Schematic of the mapping of $N = M$ and $N > M$.

avoids UAVs assignment conflict from the perspective of mathematics.

Figure 5 shows the mapping strategy for Model $N = M$ and $N > M$. In accordance with the distribution mechanism of Model $N = M$, six UAVs $[U_1, U_2, U_3, U_4, U_5, U_6]$ are assumed to strike each of the six mission points $[M_1, M_2, M_3, M_4, M_5, M_6]$ distributed in different locations in space. First, we calculate the approximate flight cost matrix $[C_1, C_2, \ldots, C_n]$ for UAVs based on the approximate flight cost method. Then, the reference flight cost *Cost*, the reference flight cost matrix *Cost* and the auxiliary flight cost matrix AuM_1 are calculated for each column of the UAVs based on the approximate flight cost, after the mutation strategy of the difference evolution algorithm and the crossover

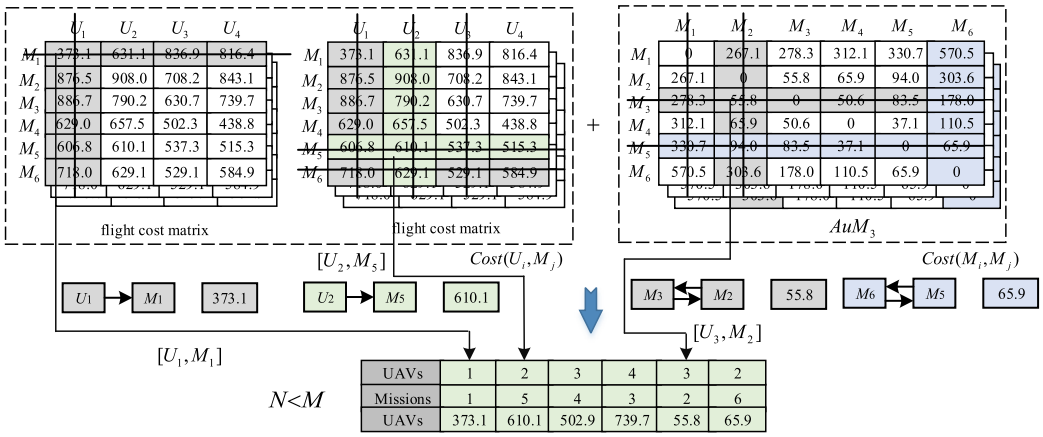


FIGURE 5. Schematic of the mapping of $N < M$.

strategy. Similarly, the sequence of assignments under model $N > M$ can be obtained.

Figure 6 shows a schematic of the mapping for model $N < M$, in which it is assumed that four UAVs $[U_1, U_2, U_3, U_4]$ strike six mission points $[M_1, M_2, M_3, M_4, M_5, M_6]$ distributed at different locations in space. We need to add auxiliary flight cost matrix AuM_3, where AuM_3 denotes the flight cost between mission points. First, the sequence of UAVs striking the mission point is obtained from the reference flight cost matrix *Cost* and the auxiliary range cost matrix AuM_1. Then, the sequence of UAVs being executed at other mission points is calculated from the auxiliary flight cost matrix AuM_3. Algorithm 2 is the principle of the mapping strategy under different models.

IV. RRAS ALGORITHM

This section first describes the basic process of the differential evolution algorithm. Then it describes the differential evolution algorithm for reducing redundant data.

A. DIFFERENTIAL EVOLUTION ALGORITHM

Population intelligence algorithms have received increasing attention in solving various optimisation problems and have been widely used in various applications. In the past decades, many population intelligence algorithms have been developed, including Ant Colony Optimisation (ACO), Particle Swarm Optimisation (PSO), Differential Evolution (DE), Bacterial Foraging Optimisation (BFO) and Artificial Bee Colony (ABC) [23]. Differential evolution algorithm (DE), similar to many classical evolutionary algorithms, is a random heuristic search algorithm, which is simple to use, has strong robustness and global optimization ability, and is widely used in multi-UAV cooperative mission planning at present [24], [25], [26]. In the process of evolution, each individual in the population corresponds to a solution vector, and new individuals are generated through different mutation strategies. Crossover operators mix individuals and target individuals to generate experimental individuals. The selection operator determines the next generation individual according to the fitness value of the test individual compared with the fitness value of the target individual. Mutation operators, crossover operators and selection operators constitute the main loop of DE.

1) INITIALIZE THE POPULATION

NP individuals are randomly and uniformly generated in the solution space.

$$\left\{X(0)|x_{j,i}^{L} \le x_{j,i}(0) \le x_{j,i}^{U};\ i = 1, 2, \ldots, NP;\ j = 1, 2, \ldots, D\right\} \times x_{j,i}(0) = x_{j,i}^{L} + rand(0, 1) \cdot (x_{j,i}^{U} - x_{j,i}^{L}) \quad (17)$$

In equation (17), *NP* represents the size of the population, D represents the dimension of the solution space, $x_{j,i}^{L}$ and $x_{j,i}^{U}$ respectively represent the upper and lower bounds of the j-th "gene" of the i-th "individual" in the population, and $x_{j,i}(0)$ represents the j-th "gene" of the i-th "individual" in the 0-th generation.

2) MUTATION OPERATOR

After initialization of the population, the DE algorithm randomly selects two individuals from the population for the difference by the mutation operator, and the resulting difference vector is weighted and summed with a third individual to produce the mutated individuals. The construction of the mutation strategy not only affects the diversity and convergence of the population, but also determines the effectiveness of the algorithm. Therefore, a reasonable mutation strategy construction method can effectively improve the efficiency of the algorithm search. Five commonly used mutation strategies are listed below.

- DE/rand/1

$$v_{j,i}(g+1) = x_{j,r1}(g) + F \times (x_{j,r2}(g) - x_{j,r3}(g)) \quad (18)$$

- DE/best/1

$$v_{j,i}(g+1) = x_{j,best}(g) + F \times (x_{j,r1}(g) - x_{j,r2}(g)) \quad (19)$$

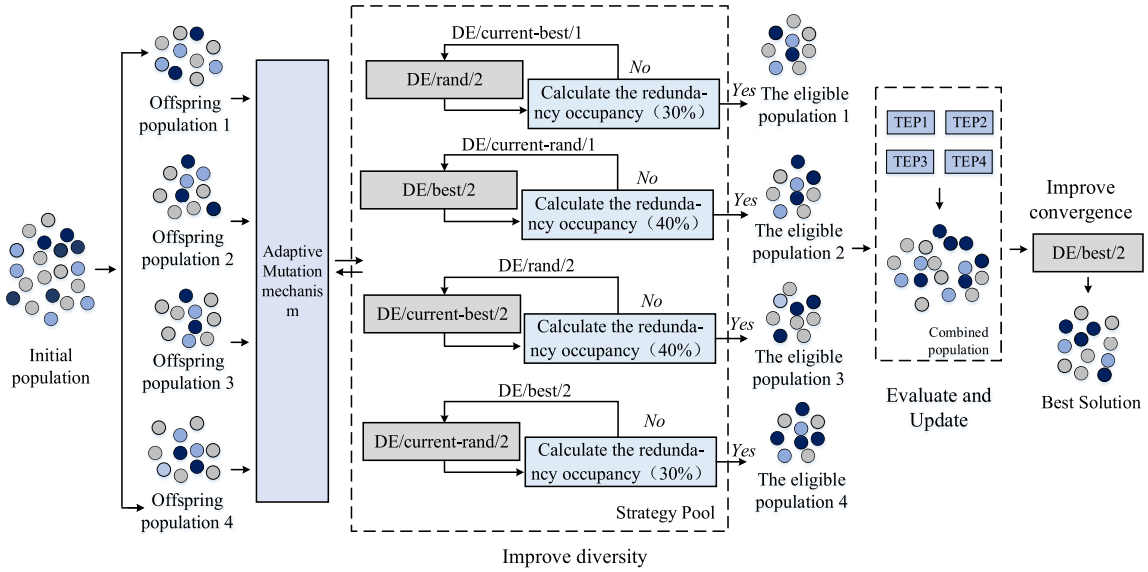


FIGURE 6. Schematic of the process of eliminating redundant flight cost data.

•DE/rand/2

$$v_{j,i}(g+1) = x_{j,r1}(g) + F \times (x_{j,r2}(g) - \ldots - x_{j,r3}(g) + x_{j,r4}(g) - x_{j,r5}(g)) \quad (20)$$

•DE/best/2

$$v_{j,i}(g+1) = x_{j,best}(g) + F \times (x_{j,r1}(g) - \ldots - x_{j,r2}(g) + x_{j,r3}(g) - x_{j,r4}(g)) \quad (21)$$

•DE/current-to-best/1

$$v_{j,i}(g+1) = x_{j,i}(g) + F \times (x_{j,best}(g) - \ldots - x_{j,i}(g) + x_{j,r1}(g) - x_{j,r2}(g)) \quad (22)$$

In equations (18) to (22), $r1, r2, r3, r4, r5$ is the five integers randomly selected in $[1, N]$, and $i \neq r1 \neq r2 \neq r3 \neq r4 \neq r5$. $x_{j,i}(g)$ represent the i-th individual in the g generation population, $x_{j,best}(g)$ represents the best individual in the g generation population, and $v_{j,i}(g+1)$ represents the mutant individual. Parameter F is the scaling factor, and the value of F directly affects the global optimization ability of the algorithm.

3) CROSSOVER OPERATOR

After the mutation, DE usually performs a binomial crossover operation where the crossover is determined by the crossover rate CR and a partial exchange is made from $x_{j,i}(g)$ and $v_{j,i}(g+1)$ to form a new trial vector $u_{j,i}(g+1)$.

$$u_{j,i}(g+1) = \begin{cases} v_{j,i}(g+1), & if\ rand \leq CR \\ x_{j,i}(g), & otherwise \end{cases} \quad (23)$$

In Formula (23), CR represents crossover rate, which determines the degree of exchange of individual genes between offspring and parents. The larger the value of CR, the more genes exchanged between individuals, and the increase of population diversity, otherwise, the decrease of population diversity, which is not conducive to global optimization.

4) SELECTION OPERATOR

DE uses a greedy algorithm to select individuals to enter the next generation of the population, which selects the better individual from the trial vector $u_{j,i}(g+1)$ and the parent vector $x_{j,i}(g)$ to enter the next generation.

$$x_{j,i}(g+1) = \begin{cases} u_{j,i}(g+1), \\ \qquad if\ f(u_{j,i}(g+1)) \geq f(x_{j,i}(g)) \\ x_{j,i}(g), \\ \qquad otherwise \end{cases} \quad (24)$$

In equation (24), $f(x)$ represents the fitness function constructed according to specific problems. The selection operator determines whether an individual can enter the next generation according to its fitness value. Algorithm 3 is the framework of DE.

B. METHOD FOR REDUCING REDUNDANT DATA

From the DE evolution process in Part IV, Section A, it is known that individuals in the population calculate the fitness value based on the mutation operator and the crossover operator, and then use the greedy algorithm to obtain the optimal solution based on the selection operator [27], [28], [29], [30]. In the MUCMA planning process, we usually think of a UAV as a prime, and a prime is an individual in a population. The population represents the candidate assignment schemes of UAVs. As the number of populations and the number of iterations increase, a large number of duplicate candidate assignment schemes will be generated, i.e. the same assignment scheme appears repeatedly in the same set of

Algorithm 2 Schematic of the Mapping Strategies for Different Models

Input: Number of UAVs: N; Number of mission points: M; Auxiliary flight cost matrix: AuM_1, AuM_2, and AuM_3.
Output: Assignment sequence $[U_i, M_j]$ and flight cost $C_{(i,j)}$ corresponding to the different model UAVs and mission points.
First model $N = M$.
Construct the auxiliary flight cost matrix AuM_1.
1 ***for*** $i = 1 : N$
2 $difference$ = min(min(abs($AuM_1(i)$ − $Cost(i)$))). /∗ *Compare the size of the reference flight cost with the auxiliary flight cost*∗/
3 [row,column] = find(abs($AuM_1(i)$ − $Cost(i)$) == *difference*).
4 new$U(i)$ = row(1); new$M(i)$ = column(1). /∗ *Output modelN = M sequence of UAVs and mission point assignments* ∗/
5 new$C$(i) = $obj.AuM_1$(new$U(i)$,new$M(i)$). /∗ *Output model N = M UAV with the corresponding flight cost of the assignment sequence of mission points* ∗/
6 AuM_1(row(1),:) = [ ]. AuM_1(:,column(1)) = []. /∗ *Delete exported assignment sequence* ∗/
7 ***end***
Second model $N > M$.
Construct separate auxiliary flight cost matrices AuM_1 and . AuM_2..
8 ***for*** $i = 1 : M$
9 Repeat steps 1-7. /∗ *Output equivalent sequence of UAVs and mission points assignments*∗/
10 ***end***
11 ***for*** $j = (M + 1) : N$ /∗ *Calculate the assignment sequence for other UAVs.* ∗/
12 $difference$ 1 = min(min(abs($AuM_2(i)$ − $Cost(j)$))).
13 [row,column] = find(abs($AuM_2(i)$ − $Cost(j)$) == *difference*1).
14 new$U(j)$ = row(1); new$T(j)$ = column(1).
15 new$C(j)$ = $obj.AuM_2$(new$U(j)$,new$T(j)$). /∗ *Output model N > M UAV with the corresponding flight cost of the assignment sequence of mission points* ∗/
16 AuM_2(row(1),:) = 0.
17 ***end***
Third model $N < M$.
Construct the auxiliary flight cost matrix AuM_1 andAuM_3.
18 ***for*** $i = 1 : N$
19 Repeat steps 1-7.
20 ***end***
21 ***for*** i= $(N + 1)$: M
22 $difference$ 2 = min(min(abs($AuM_3(i)$ − $Cost(j)$))).
23 [row,column] = find(abs($AuM_3(i)$ − $Cost(j)$) == *difference*2).
24 newU(i) = row (1); new$T(i)$ = column(1).
25 new$C(i)$ = $obj.AuM_3$(row(1), new$T(i)$). /∗ *Output model N < M UAV with the corresponding flight cost of the assignment sequence of mission points*∗/
26 ***end***

Algorithm 3 Framework of the DE Algorithm

Input: Population: NP; Dimebison: D; Genetation: Gen.
Output: The best vector(solution).
1 $t = 1$
2 Initialize the population $X(0)$ by equation (17).
3 ***while*** $t \leq Gen$
4 ***for****i* =1 to NP
5 ***for*** $j = 1$ to D
6 $v_{j,i}(g+1) = Mutation(x_{j,i}(g))$. /∗ Mutant individuals are generated by equations (18)-(22) or other mutation strategies. ∗/
7 $u_{j,i}(g+1) = Crossover(v_{j,i}(g+1), x_{j,i}(g))$. /∗ The trial individuals were generated by equation (23). ∗/
8 ***endfor***
9 ***if*** $f(u_{j,i}(g+1)) \geq f(x_{j,i}(g))$ *then* /∗ The more adapted individuals were selected by equation (24). ∗/
10 $x_{j,i}(g+1) = u_{j,i}(g+1)$.
11 ***else***
12 $x_{j,i}(g+1) = x_{j,i}(g)$.
13 ***endif***
14 ***endfor***
15 $t = t + 1$.
16 ***endwhile***

assignment sequences, and the redundancy of the assignment schemes will lead to anomalies in the assignment sequences of UAVs, so the redundant assignment schemes must be dealt with.

In recent decades, researchers have designed various methods to eliminate redundant data or extract data features for large-scale data optimization. These methods include the relevant-redundant weight-based feature criterion [31], joint multi-objective optimization method for feature selection and classifier design [32], and binary differential evolution with self-learning [33]. These methods have demonstrated good performance in handling irrelevant and redundant data. However, MUCMA requires solving the optimal assignment solution in a short time, and it would be impractical to optimize both the diversity of the populations and the redundancy of the data with the main research objective of improving the accuracy of the assignment solution. The reason is that a part of the population will be added to the next generation cycle as a feasible solution during the iterative process. Therefore, a method for reducing redundant data based on fitness values is proposed in this section, and the schematic diagram of the algorithm is shown in Figure 7.

In WAFC-RRAS, the entire population is randomly divided into four populations with the same number of individuals, and an archive of mutation strategies (AMS) is constructed to store four mutation strategies with different characteristics. Next, in accordance with the data redundancy reduction rule, the voyage cost of each individual is calculated, and whether the redundancy rate of individuals in the subpopulation exceeds the specified value (30% or 40%) is determined through the flight cost. When the redundancy

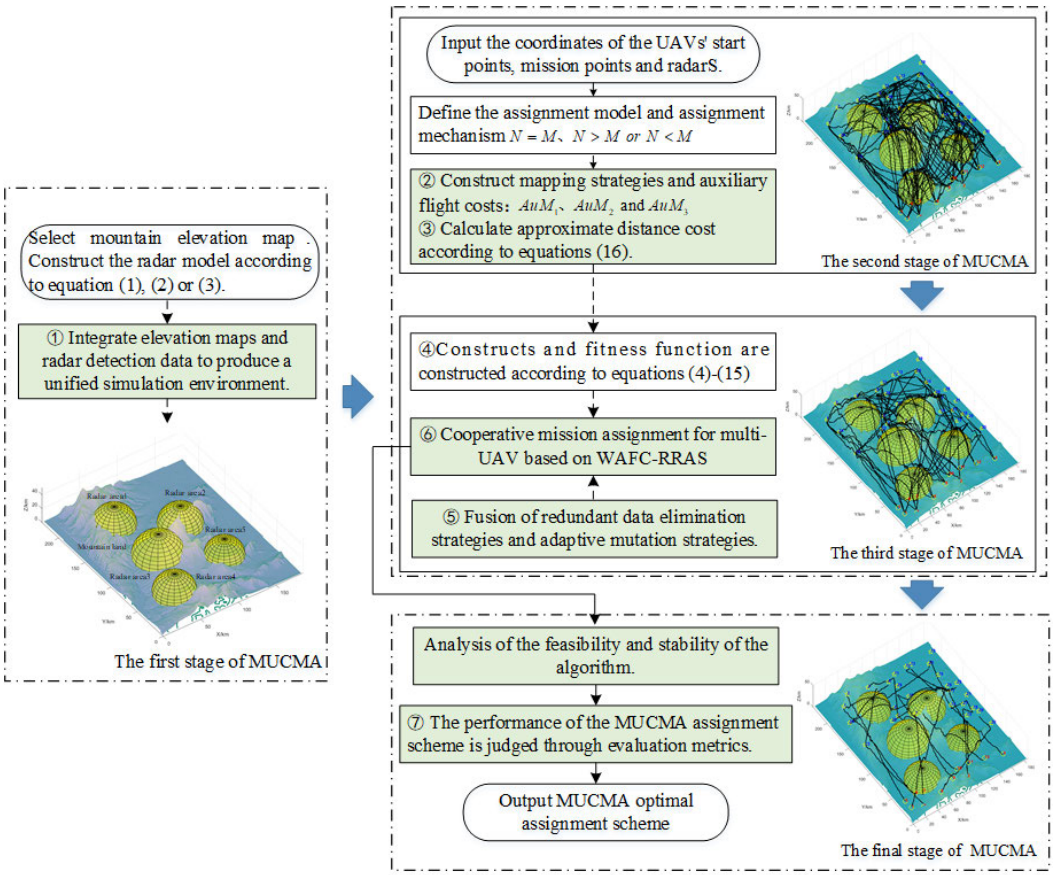


FIGURE 7. Schematic of the process of eliminating redundant flight cost data.

TABLE 1. Parameter selection in the mutation strategies.

Type	Mutation strategy	Scaling factor F	Crossover rate CR
I	DE/current-rand/1	$F = 0.6$	$CR = 0.5$
II	DE/rand/2	$F = 0.8$	$CR = 0.7$
III	DE/best/2	$F = 0.2$	$CR = 0.1$
IV	DE/current-best/1	$F = 0.3$	$CR = 0.5$

rate exceeds its regular value, the corresponding mutation strategy is changed, and the update is continued iteratively; otherwise, a new population set of qualified individuals is produced.

At the same time, we use an intergenerational variation mechanism instead of random coverage to improve population diversity. The reason is that the fitness value of the optimal solution is low during the evolutionary process and likely to be overwritten as the number of iterations increases, leading to a waste of computational resources. Therefore, in this study, individuals that are better than the others are retained. This approach effectively changes the evolutionary direction of the population and increases population diversity.

The parameter settings ($F$and CR) of the mutation strategies are shown in Table 1, and the redundancy rate of flight cost reduction is shown in Algorithm 4.

C. EXECUTION PROCESS OF MULTI-UAV COOPERATIVE MISSION ASSIGNMENT BASED ON WAFC-RRAS ALGORITHM

Figure 8 depicts the specific flow of multi-UAV cooperative mission assignment based on the WAFC-RRAS algorithm.

V. EXPERIMENTAL RESULTS AND ANALYSIS

To verify the effectiveness and stability of MUCMA based on WAFC-RRAS, this study conducts three sets of simulation experiments in MatlabR2016b. In Experiment 1, small numbers of UAVs and mission points are selected to verify the effectiveness of WAFC-RRAS. In Experiment 2, large numbers of UAVs and mission points are selected to verify the stability of WAFC-RRAS. Meanwhile, in Experiment 3, the performance of WAFC-RRAS is compared with that of other algorithms.

A. EXPERIMENT 1: SELECTING SMALL NUMBERS OF UAVs AND MISSION POINTS TO VERIFY THE EFFECTIVENESS OF THE ALGORITHM

Table 2 shows the parameter information for each UAV and mission point under the three assignment models. *Start*, *Mission*, and *Radar* denote the coordinates of the UAVs, the mission points, and the radar scanning area in the 3D simulation environment, respectively. *MaxDis* refers to

Algorithm 4 Reduce the Redundancy Rate

Input: Population: P_n; Mutation strategies: I, II, III, and IV.
Output: Output the redundant data occupancy *Ocrd* and the better set of individuals of the population.
1 Determine the size of the population and divide it randomly into 4 populations: $offspr_1$,, $offspr_3$ and $offspr_4$.
2 Calculate the fitness values for the various populations and determine the occupancy of redundant data.
3 ***fori*** =1: Np/4
4

$$v_{j,i}(g+1) = x_{j,r1}(g) + F \times (x_{j,r2}(g) - x_{j,r3}(g)) + \ldots + F \times (x_{j,r4}(g) - x_{j,r5}(g)).$$

5 $fitness(i) = FliDCost(i) + FliTCost(i) + VioBCost(i)$. /∗ *Calculate the fitness values for individuals in the population separately* ∗/
6 ***end***
7 *ReC*= tabulate(*fitness*). /∗ *Output repeated fitness values and output the proportion of that number repeated* ∗/
8 ***if*** *ReC* >30%
9

$$v_{j,i}(g+1) = x_{j,i}(g) + F \times (x_{j,best}(g) - x_{j,i}(g)) + \ldots + F \times (x_{j,r1}(g) - x_{j,r2}(g)).$$

10 ***else***
11 Export eligible individuals to *TEP*1.
12 ***for*** i = Np/4+1: Np/4∗2
13

$$v_{j,i}(g+1) = x_{j,best}(g) + F \times (x_{j,r1}(g) - x_{j,r2}(g)) + \ldots + F \times (x_{j,r3}(g) - x_{j,r4}(g)).$$

14 $fitness\ (i) = FliDCost(i) + FliTCost(i) + VioBCost(i)$.
15 ***end***
16 *ReC* = tabulate(*fitness*).
17 ***if*** *ReC* >30%
18

$$v_{j,i}(g+1) = x_{j,i}(g) + F \times (x_{j,r1}(g) - x_{j,i}(g)) + \ldots + F \times (x_{j,r2}(g) - x_{j,r3}(g)).$$

19 ***else***
20 Export eligible individuals to *TEP*2.
21 And so on, output *TEP3* and *TEP4*.
22 *TEP* = [*TEP*1; *TEP*2; *TEP*3; *TEP*4]. /∗ *Combine individuals from eligible populations*∗/
23 Improve the convergence performance of the algorithm.
24

$$v_{j,i}(g+1) = x_{j,best}(g) + F \times (x_{j,r1}(g) - x_{j,r2}(g)) + \ldots + F \times (x_{j,r3}(g) - x_{j,r4}(g)).$$

25 Output the set of optimal solutions.
end

the maximum flight of each UAV. *SpeFli* is the minimum and maximum flight speeds of each UAV. *WeiMis* denotes the weight of each mission point. *OrdMis* indicates the time sequence in which each mission point is executed, and *WaitS* is the waiting time of each UAV.

The algorithm parameters for the three assignment models in Experiment 1 were set as follows: *NP*=500, *NP* is the size of the population. *Gen*=1000, *Gen* is the number of evolutionary iterations. As the differential evolution algorithm is somewhat stochastic, in order to better show the stability of the algorithm in this paper, the experiment will be repeated continuously for the same model, and *Num* sets the number of repetitions, *Num*=20.

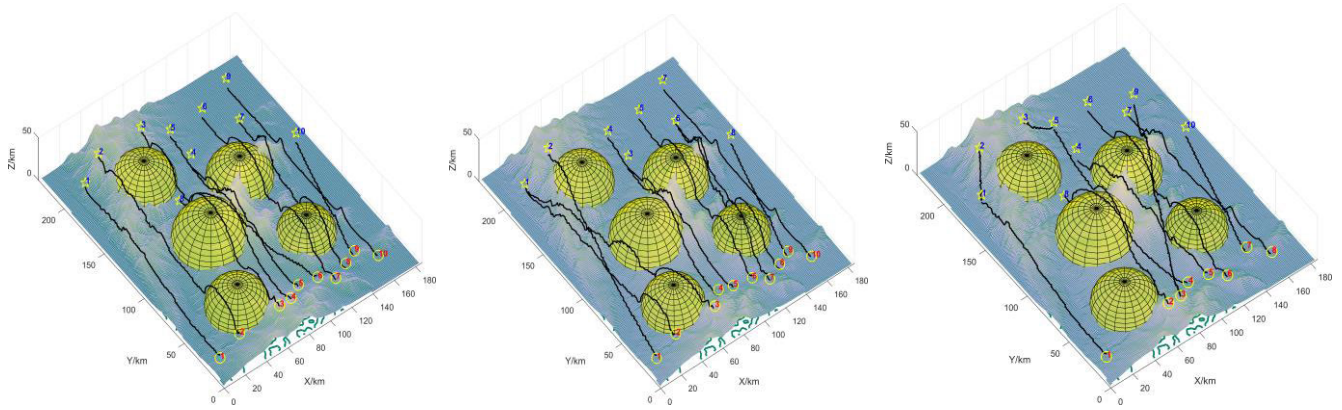
Figure 9 shows the results of the assignment of UAVs to missions in Experiment 1. The yellow circles and yellow pentagrams represent the positions of UAVs and mission points in 3D space, respectively. The yellow hemispheres indicate the radar scanning range, and the black solid lines represent the relationship between UAVs and the mission point assignment. As can be seen in Figure 9, the planning system was able to follow the UAVs assignment mechanism without assignment conflicts for the different assignment models, while the approximate flight cost method was able to efficiently calculate the distance between the UAVs and each mission and the UAVs did not cross the obstacles.

Table 3 shows the results of the assignment of MUCMA under different models for Experiment 1. UAV and Mission represent the UAV sequence and mission assignment sequence, respectively. *Tcost* represents the total flight cost of the UAV to execute the mission, and *Time* represents the algorithm's running time. As can be seen from the assignment results in Table 3, accurate approximate flight costs can be solved for both UAVs and mission points, while MUCMA can solve the assignment scheme in a short time.

Figure 10 shows a comparison of the single convergence and average convergence curves of the MUCMA fitness values for Experiment 1. The solid blue and red lines in Figure 10 indicate the single and average convergence curves, respectively, while the red circles and blue asterisks indicate the optimal solution solved by the WAFC-RRAS algorithm and the optimal solution searched for by the intergenerational variation during the iterative process, respectively. The single convergence curve shows that each assignment model finds as many assignment solutions as possible in the beginning of the iteration. As the number of iterations increases, the different assignment models converge quickly to the optimal solution. The average convergence curve indicates that the convergence trend is relatively smooth and similar to that of the single convergence curve. As a result, the WAFC-RRAS algorithm has good diversity and convergence, and is highly effective in solving MUCMA for a small number of UAVs and missions.

B. EXPERIMENT 2: SELECTING LARGE NUMBERS OF UAVs AND MISSION POINTS TO VERIFY THE STABILITY OF WAFC-RRAS

In Experiment 2, large numbers of UAVs and mission points are selected to verify the stability of WAFC-RRAS, and the simulation environment and algorithm parameters are the



$N = M$ ($N = M = 10$) $N > M$ ($N = 10, M = 8$) $N < M$ ($N = 8, M = 10$)

FIGURE 8. Simulation diagram of UAVs and mission assignment in Experiment 1.

TABLE 2. Experimental initial data for UAVs and mission points.

Mode	Data type	1	2	3	4	5	6	7	8	9	10
$N = M$	*Start*	[11 20 3]	[34 26 7]	[67 20 20]	[78 20 20]	[93 31 12]	[112 32 5]	[120 20 10]	[134 26 10]	[145 30 12]	[160 20 5]
	Mission	[20 200 12]	[40 210 20]	[80 210 20]	[97 170 16]	[100 200 11]	[130 200 12]	[150 180 5]	[70 149 5]	[160 210 13]	[173 140 12]
	Radar	[130 70 4,23]	[120 140 4 26]	[70 110 5 30]	[60 180 5 25]	[50 50 5 25]	-	-	-	-	-
	MaxDis	800	600	500	700	600	500	800	600	500	600
	SpeFli	[65 80]	[70 100]	[65 120]	[75 125]	[60 80]	[75 110]	[90 120]	[100 120]	[70 90]	[80 110]
	WeiMis	0.6	0.7	1	0.8	0.6	0.4	0.8	0.3	0.5	0.2
	WaitS	15	20	20	25	15	20	15	20	20	15

Mode	Data type	1	2	3	4	5	6	7	8	9	10
$N > M$	*Start*	[11 20 3]	[34 26 7]	[67 20 20]	[78 20 20]	[93 31 12]	[112 32 5]	[120 20 10]	[134 26 10]	[145 30 12]	[160 20 5]
	Mission	[20 200 12]	[50 210 20]	[97 170 16]	[100 200 11]	[130 200 12]	[150 180 5]	[160 210 13]	[173 140 12]	-	-
	Radar	[130 70 4,23]	[120 140 4 26]	[70 110 5 30]	[60 180 5 25]	[50 50 5 25]	-	-	-	-	-
	MaxDis	800	600	500	700	600	500	800	600	500	600
	SpeFli	[65 80]	[70 100]	[65 120]	[75 125]	[60 80]	[75 110]	[90 120]	[100 120]	[70 90]	[80 110]
	WeiMis	0.8	0.7	0.2	0.8	0.5	0.3	0.6	0.5		
	WaitS	15	20	20	25	15	20	15	20	20	15

Mode	Data type	1	2	3	4	5	6	7	8	9	10
$N < M$	*Start*	[11 20 3]	[67 20 20]	[78 20 20]	[93 31 12]	[112 32 5]	[120 20 10]	[145 30 12]	[160 20 5]	-	-
	Mission	[20 200 12]	[40 210 20]	[80 210 20]	[97 170 16]	[100 200 11]	[130 200 12]	[150 180 5]	[70 149 5]	[160 210 13]	[173 140 12]
	Radar	[130 70 4,23]	[120 140 4 26]	[70 110 5 30]	[60 180 5 25]	[50 50 5 25]	-	-	-	-	-
	MaxDis	800	600	500	700	600	500	800	600	-	-
	SpeFli	[65 80]	[70 100]	[65 120]	[75 125]	[60 80]	[75 110]	[90 120]	[100 120]	[70 90]	[80 110]
	WeiMis	0.6	0.7	1	0.8	0.6	0.4	0.8	0.3	0.5	0.2
	OrdMis	[3 4]	[5 2]	[6 1]	[7 4]	-	-	-	-	-	-

same as in Experiment 1. The performance metrics include average time $T_{average}$, average cost value $C_{average}$, optimum cost value $C_{optimum}$, constraint violation $C_{violation}$, optimum solution rate $R_{optimum}$ and redundancy rate $R_{redundancy}$.

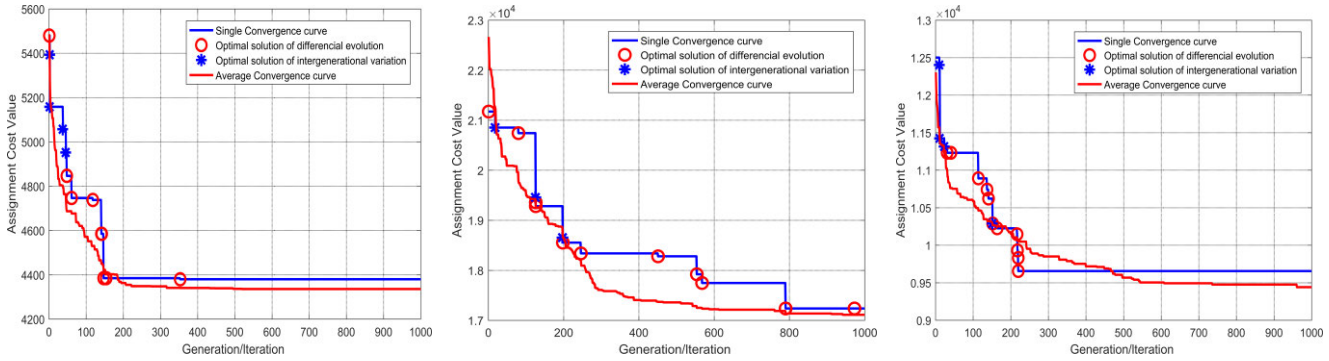


FIGURE 9. Single and average convergence curves for each model in Experiment 1.

TABLE 3. Results of MUCMA assignment and running times in experiment 1.

Model		Assignment results										$Tcost(km)$	$Time(s)$
$N = M$	UAV	1	2	3	4	5	6	7	8	9	10	4373.8	11.2
	Mission	1	2	3	5	4	8	6	7	10	9		
	Cost	373.1	565.2	908.0	630.7	396.4	385.0	515.3	561.6	178.1	331.1		
$N > M$	UAV	1	2	3	4	5	6	7	8	9	10	16780.1	13.1
	Mission	2	1	1	4	3	5	6	6	8	7		
	Cost	449.2	489.7	822.2	433.0	396.4	438.8	606.8	561.6	178.1	331.1		
$N < M$	UAV	1	1	2	2	3	4	5	6	7	8	9394.2	13.2
	Mission	1	2	5	3	9	8	4	6	10	7		
	Cost	334.8	134.5	843.1	232.2	629.0	396.4	385.0	592.4	178.1	342.8		

Figure 11(a), Figure 11(b) and Figure 11(c) show the UAVs and mission assignment results for Experiment 2, respectively. Figure 11(d), Figure 11(e) and Figure 11(d) hide the mountainous model and it can be seen more clearly that the approximate flight cost takes full account of the influence of obstacles on the planning system during the planning process, and it can also be seen that although the number of UAVs and missions is increased, the UAVs do not suffer from assignment conflicts and strictly follow the assignment mechanism of each assignment model.

Figure 12 shows the single and average MUCMA convergence curves for Experiment 2. The graph shows that although the numbers of UAVs and mission points are increased, the WAFC-RRAS algorithm is still able to search a wider solution space at the beginning of the iteration and converges quickly to the optimal assignment solution later in the iteration.

Meanwhile, the convergence curve is analyzed using a box line plot, as shown in Figure 13. A boxplot is a statistical plot that uses five statistics (minimum, upper quartile, mean, lower quartile, and maximum) to describe the data.

Figure 13 randomly selects five convergence curves in the different assignment models as the object of analysis. With Model $N = M$ convergence curve A as an example, the black diamond points in the figure indicate the data outliers (20686.8, 20134.8). The outliers show a large difference in a set of data relative to other convergence values. In the MUCMA assignment system, the outliers indicate that at the beginning of the iteration, WAFC-RRAS can search a broad population space. The rectangles denote concentrated convergence data, and the squares denote the mean of the convergence curve. As can be seen from the boxplot of convergence curve A, convergence curve A can search for a wider solution space in the early iteration, search for more optimal solutions near the optimal solution in the middle iteration, and converge to the optimal solution quickly in the late iteration.

Table 4 presents the data on the cooperative assignment of UAVs for Experiments 1 and 2, and six evaluation metrics are selected to analyze the performance of WAFC-RRAS.

- Average time: $T_{average} = \sum_{i=1}^{n} T_i/Num$, where T_i denotes the time of the i-th algorithm run and Num denotes the number of times the system is run.
- Average cost: $C_{average} = \sum_{i=1}^{n} T_{C_i}/Num$, where TC_i denotes the total flight cost of the i-th UAVs. This metric measures the stability of the algorithm.
- Optimum cost: $C_{optimum} = min[T_{C1}, T_{C2}, \ldots, T_{Cn}]$, where T_{Cn} represents the minimum generation value in the number of iterations. It measures the ability of the algorithm to find the optimal solution.

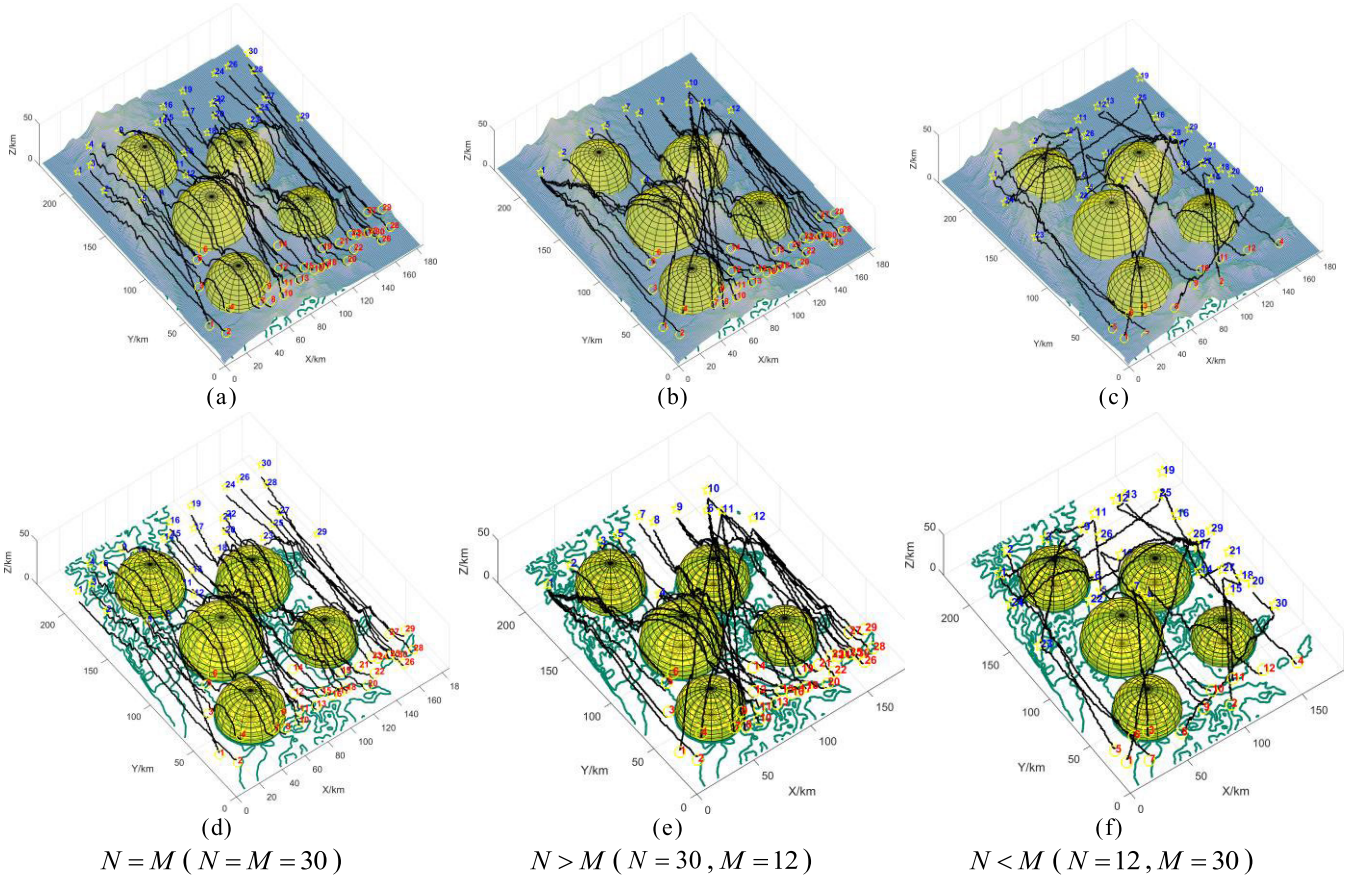


FIGURE 10. UAVs and mission assignment results for Experiment 2.

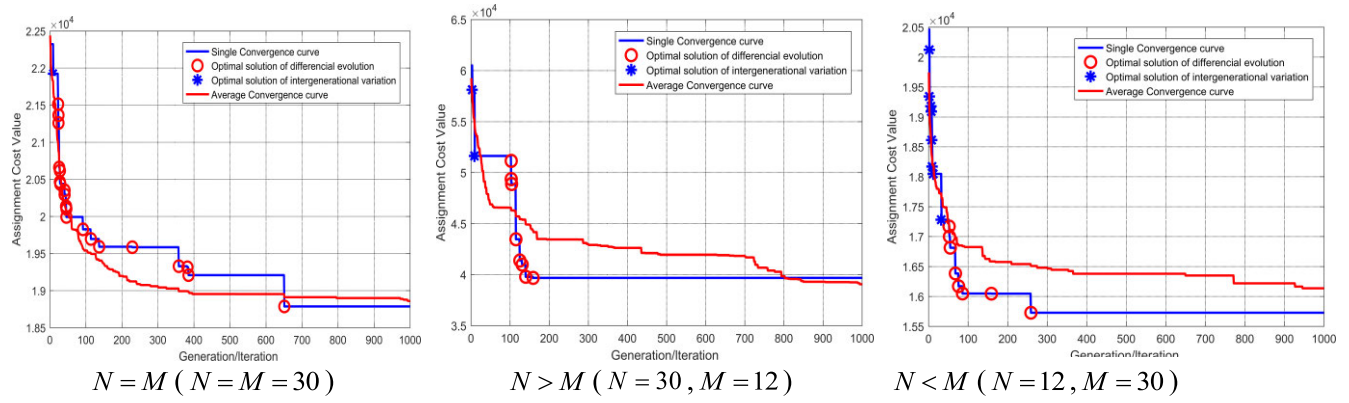


FIGURE 11. Single and average convergence curves for each model in Experiment 2.

- Constraint violation: $C_{violation} = max[Conv(i)/T\cos t(i)]$, where $Conv(i)$ denotes the amount of violation in generation i and $T\cos t(i)$ is the total flight cost in generation i. This metric measures the cooperative performance of WAFC-RRAS.
- Optimum solution $R_{optimum}$: It indicates the number of times of falling into a local optimum and the number of times that are currently better than the average cost as a percentage of the total number of experiments.
- Redundancy rate $R_{redundancy}$: It indicates the repetition rate of the UAV assignment scheme at the beginning of the algorithm iteration, which is usually considered to be the first 30% of iterations at the beginning of the iteration.

The following conclusions are drawn from the data in Table 4.

Conclusion 1: In Experiment 1, the average time of the cooperative mission assignment of a small number of UAVs

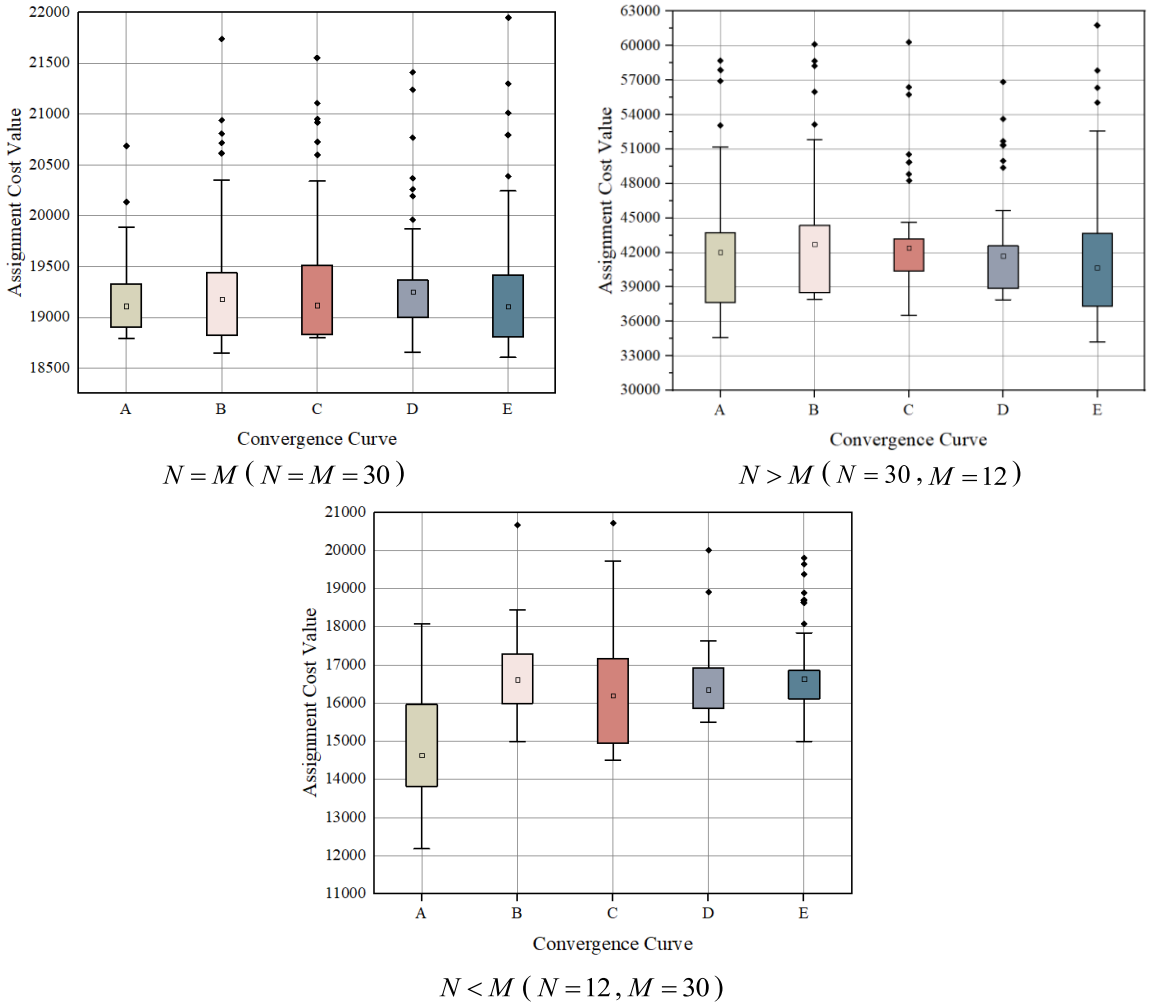


FIGURE 12. Data analysis of convergence curves under different assignment models.

TABLE 4. Statistics on the assignment results for Experiments 1 and 2.

Experiment	Model	Average time (s)	Average cost (km)	Optimum cost (km)	Violation of constraints (%)	Optimum solution rate (%)	Redundancy rate (%)
Experiment 1	$N = M$	9.5	4583.9	3778.3	0.12%	88.7%	0.2%
	$N > M$	13.7	16867.4	16775.5	0.17%	80.9%	0.9%
	$N < M$	13.0	9782.7	9151.8	0.13%	86.5%	0.2%
Experiment 2	$N = M$	25.3	18591.5	18470.1	0.65%	85.2%	11.4%
	$N > M$	28.4	39455.9	35213.9	0.76%	81.3%	6.8%
	$N < M$	27.6	16399.2	16260.2	0.60%	83.5%	7.4%

under the different models is short, the difference between the average and optimal cost values is small, the violation constraint is small, the optimal solution rate is high, and the redundancy rate is low. These results indicate that the proposed WAFC-RRAS can effectively deal with the few cooperative mission assignment problem of UAVs under different assignment models.

Conclusion 2: In Experiment 2, although the number of UAVs and mission points is increased, resulting in a long average time, the difference between the average and optimal cost values is still small, indicating that the algorithm has strong stability. The low violation constraint and high optimization rate indicate that the algorithm has strong cooperative performance, and the redundancy rate is increased but

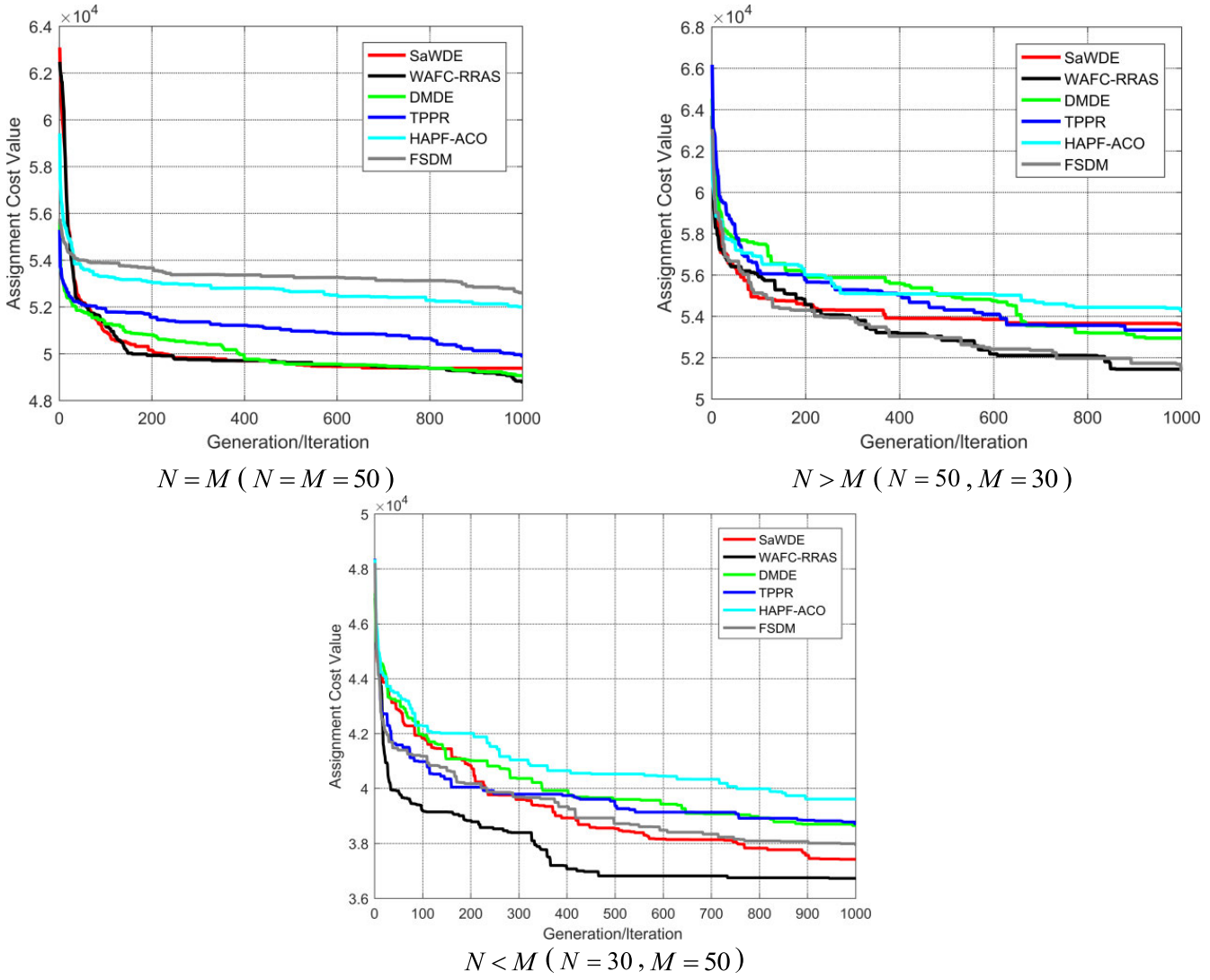


FIGURE 13. Convergence curves of average fitness values for different algorithms.

still does not exceed the limited requirements. This result shows that the proposed WAFC-RRAS can effectively handle the cooperative mission assignment problem.

C. EXPERIMENT 3: COMPARISON OF THE PERFORMANCE OF WAFC-RRAS AND OTHER ALGORITHMS

This section compares WAFC-RRAS's performance with that of the self-adaptive weighted DE approach (SaWDE), discrete mapping DE (DMDE), two-level parameter cooperation-based population regeneration (TPPR), hybrid artificial potential field and ant colony(HAPF-ACO), and feature selection-based decision model(FSDM).

The SaWDE algorithm adopts six mutation strategies and 10 selection mechanisms to improve the diversity of data features and reduce the redundancy rate of data, and it is suitable for large-scale data feature extraction and application [34]. The DMDE algorithm uses a unified genetic coding approach and an evolutionary algorithm framework to handle different assignment models, and it employs dynamic cross-rate and hybrid differencing strategies during evolution to improve the search efficiency of discrete differencing algorithms [21]. The TPPR algorithm adopts a two-layer parametric framework to cooperate continuous space optimization problems, thereby reducing population stagnation and early convergence problems [35]. The HAPF-ACO algorithm fully considers multiple UAV constraint types, and constructs target attraction and threat repulsion zones for environment cognition through HAPF, focusing on multi-UAV execution efficiency and obstacle avoidance performance [8]. The FSDM algorithm proposes a new feature selection-based decision-making model to improve multi-UAV recognition of the simulated environment, while the superior search capability of the A∗ algorithm is used to improve the efficiency of multi-UAV cooperative mission assignment [36]. The different model parameters and algorithm performance are shown in Table 5.

TABLE 5. Comparison of the performance metric data of different algorithms.

Model	Algorithm	$N \& M$	P_n	Gen	Num	Optimum cost (km)	Average cost (km)	Average time (s)
$N=M$	SaWDE	$N=M=50$	500	1000	20	50323.2	53230.8	46.5
	WAFC-RRAS	$N=M=50$	500	1000	20	47936.1	49411.5	37.3
	DMDE	$N=M=50$	500	1000	20	49337.8	51370.4	48.9
	TPPR	$N=M=50$	500	1000	20	50635.3	51860.3	45.7
	HAPF-ACO	$N=M=50$	500	1000	20	52413.5	58558.7	60.8
	FSDM	$N=M=50$	500	1000	20	53249.0	56801.6	59.5
$N>M$	SaWDE	$N=50, M=30$	500	1000	20	55464.2	57263.2	68.4
	WAFC-RRAS	$N=50, M=30$	500	1000	20	47388.4	54635.8	48.7
	DMDE	$N=50, M=30$	500	1000	20	50927.0	55101.9	60.9
	TPPR	$N=50, M=30$	500	1000	20	51265.8	55816.7	61.6
	HAPF-ACO	$N=50, M=30$	500	1000	20	54615.3	55505.3	79.3
	FSDM	$N=50, M=30$	500	1000	20	53801.6	53801.6	68.5
$N<M$	SaWDE	$N=30, M=50$	500	1000	20	38284.3	41515.7	78.9
	WAFC-RRAS	$N=30, M=50$	500	1000	20	33430.5	37438.8	50.4
	DMDE	$N=30, M=50$	500	1000	20	40489.9	42175.3	75.6
	TPPR	$N=30, M=50$	500	1000	20	39730.3	41521.2	73.8
	HAPF-ACO	$N=30, M=50$	500	1000	20	41081.7	41345.7	87.4
	FSDM	$N=30, M=50$	500	1000	20	37426.0	39193.0	75.1

Table 5 summarizes the comparative data on the performance metrics of the algorithms under different assignment models. It can be seen from the data that the WAFC-RRAS algorithm is significantly better than the other three algorithms in the two metrics of optimal flight cost, and average flight cost for the same model. The average time is slightly higher than the SaWDE algorithm, but the difference is smaller.

The solid lines with different colors in Figure 4 represent the convergence trend of the average fitness values of the six algorithms. As can be seen from model $N = M$, WAFC-RRAS and SaWDE can search a wide solution space at the beginning of the iteration, but WAFC-RRAS can find more superior solutions at the end of the iteration. The diversity and convergence of WAFC-RRAS are superior to those of the five other algorithms. As can be seen from model $N > M$, all algorithms search a wide solution space early in the iteration, FSDM and WAFC-RRAS convergence trends are consistent, but WAFC-RRAS can converge quickly to the optimal assignment scheme in the middle of convergence. In model $N < M$, all six algorithms converge to the optimal solution, but WAFC-RRAS has a much better optimal solution than the three other algorithms. In summary, WAFC-RRAS outperforms the other algorithms in terms of diversity and convergence under the different models, and it can search for a good assignment solution within a short time.

VI. CONCLUSION

In this study, a Multi-UAV cooperative mission assignment optimization method based on an auxiliary flight cost matrix is proposed. The method maps the flight cost of discrete-space UAVs to a physically meaningful continuous space and designs main order matching and auxiliary cost matrix methods under different assignment models to prevent UAVs assignment conflicts. Then, an RRAS is proposed to simplify the complexity of reducing redundant data by fusing different mutation strategies. The method ensures the diversity of the population and speeds up the convergence to the improved assignment scheme. Simulation experiments show that the method can effectively and quickly solve the problem of cooperative mission assignment of UAVs under different models.

Our next research will focus on the impact of communication security and communication delay on MUCMA. As the number of UAVs in the cooperative operation system increases, the communication signal strength weakens. A common method to improve the communication performance of UAVs is to increase the signal strength and extend the communication range. However, this procedure causes new communication security and communication delays. Communication security and avoidance of communication delays are prerequisites to ensure stable and reliable information interaction in UAVs. Therefore, investigating MUCMA

in environments involving communication security and communication delay environment is necessary.

REFERENCES

[1] Y. Wu, J. Gou, H. Ji, and J. Deng, "Hierarchical mission replanning for multiple UAV formations performing tasks in dynamic situation," *Comput. Commun.*, vol. 200, pp. 132–148, Feb. 2023, doi: 10.1016/j.comcom.2023.01.011.

[2] J. Li, X. Xiong, Y. Yan, and Y. Yang, "A survey of indoor UAV obstacle avoidance research," *IEEE Access*, vol. 11, pp. 51861–51891, 2023, doi: 10.1109/ACCESS.2023.3262668.

[3] M. M. Alam, M. Y. Arafat, S. Moh, and J. Shen, "Topology control algorithms in multi-unmanned aerial vehicle networks: An extensive survey," *J. Netw. Comput. Appl.*, vol. 207, no. 1, Nov. 2022, Art. no. 103495, doi: 10.1016/j.jnca.2022.103495.

[4] X. Zhang, S. Xia, T. Zhang, and X. Li, "Hybrid FWPS cooperation algorithm based unmanned aerial vehicle constrained path planning," *Aerosp. Sci. Technol.*, vol. 118, Nov. 2021, Art. no. 107004, doi: 10.1016/j.ast.2021.107004.

[5] R. I. Mukhamediev, K. Yakunin, M. Aubakirov, I. Assanov, Y. Kuchin, A. Symagulov, V. Levashenko, E. Zaitseva, D. Sokolov, and Y. Amirgaliyev, "Coverage path planning optimization of heterogeneous UAVs group for precision agriculture," *IEEE Access*, vol. 11, pp. 5789–5803, 2023, doi: 10.1109/ACCESS.2023.3235207.

[6] W. Liu, T. Zhang, S. Huang, and K. Li, "A hybrid optimization framework for UAV reconnaissance mission planning," *Comput. Ind. Eng.*, vol. 173, Nov. 2022, Art. no. 108653, doi: 10.1016/j.cie.2022.108653.

[7] J. Tang, X. Chen, X. Zhu, and F. Zhu, "Dynamic reallocation model of multiple unmanned aerial vehicle tasks in emergent adjustment scenarios," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 2, pp. 1139–1155, Apr. 2023, doi: 10.1109/TAES.2022.3195478.

[8] Z. Zhen, Y. Chen, L. Wen, and B. Han, "An intelligent cooperative mission planning scheme of UAV swarm in uncertain dynamic environment," *Aerosp. Sci. Technol.*, vol. 100, May 2020, Art. no. 105826, doi: 10.1016/j.ast.2020.105826.

[9] Y. Wang, K. Li, Y. Han, and X. Yan, "Distributed multi-UAV cooperation for dynamic target tracking optimized by an SAQPSO algorithm," *ISA Trans.*, vol. 129, pp. 230–242, Oct. 2022, doi: 10.1016/j.isatra.2021.12.014.

[10] T. Liu, D. Han, Y. Lin, and K. Liu, "Distributed multi-UAV trajectory optimization over directed networks," *J. Franklin Inst.*, vol. 358, no. 10, pp. 5470–5487, Jul. 2021, doi: 10.1016/j.jfranklin.2021.04.044.

[11] Z. Liu, Y. Li, and Y. Wu, "Multiple UAV formations delivery task planning based on a distributed adaptive algorithm," *J. Franklin Inst.*, vol. 360, no. 4, pp. 3047–3076, Mar. 2023, doi: 10.1016/j.jfranklin.2023.01.008.

[12] B. Shirani, M. Najafi, and I. Izadi, "Cooperative load transportation using multiple UAVs," *Aerosp. Sci. Technol.*, vol. 84, pp. 158–169, Jan. 2019, doi: 10.1016/j.ast.2018.10.027.

[13] H. Qiu and H. Duan, "A multi-objective pigeon-inspired optimization approach to UAV distributed flocking among obstacles," *Inf. Sci.*, vol. 509, pp. 515–529, Jan. 2020, doi: 10.1016/j.ins.2018.06.061.

[14] M. R. Ramzan, M. Naeem, O. Chughtai, W. Ejaz, and M. Altaf, "Radio resource management in energy harvesting cooperative cognitive UAV assisted IoT networks: A multi-objective approach," *Digit. Commun. Netw.*, vol. 509, no. 4, pp. 515–529, Jan. 2023, doi: 10.1016/j.dcan.2023.01.006.

[15] S. Chai, Z. Yang, J. Huang, X. Li, Y. Zhao, and D. Zhou, "Cooperative UAV search strategy based on DMPC-AACO algorithm in restricted communication scenarios," *Defence Technol.*, vol. 509, no. 4, pp. 515–529, Jan. 2023, doi: 10.1016/j.dt.2022.12.012.

[16] J. Guo, G. Huang, Q. Li, N. N. Xiong, S. Zhang, and T. Wang, "STMTO: A smart and trust multi-UAV task offloading system," *Inf. Sci.*, vol. 573, pp. 519–540, Sep. 2021, doi: 10.1016/j.ins.2021.05.020.

[17] Y. Li, T. Han, H. Zhou, S. Tang, and H. Zhao, "A novel adaptive L-SHADE algorithm and its application in UAV swarm resource configuration problem," *Inf. Sci.*, vol. 606, pp. 350–367, Aug. 2022, doi: 10.1016/j.ins.2022.05.058.

[18] L. Wen, Z. Zhen, T. Wan, Z. Hu, and C. Yan, "Distributed cooperative fencing scheme for UAV swarm based on self-organized behaviors," *Aerosp. Sci. Technol.*, vol. 138, Jul. 2023, Art. no. 108327, doi: 10.1016/j.ast.2023.108327.

[19] L. Xu, X. Cao, W. Du, and Y. Li, "Cooperative path planning optimization for multiple UAVs with communication constraints," *Knowl.-Based Syst.*, vol. 260, Jan. 2023, Art. no. 110164, doi: 10.1016/j.knosys.2022.110164.

[20] C. Ramirez-Atencia, V. Rodriguez-Fernandez, and D. Camacho, "A revision on multi-criteria decision making methods for multi-UAV mission planning support," *Exp. Syst. Appl.*, vol. 160, Dec. 2020, Art. no. 113708, doi: 10.1016/j.eswa.2020.113708.

[21] Z. Ming, Z. Lingling, S. Xiaohong, M. Peijun, and Z. Yanhang, "Improved discrete mapping differential evolution for multi-unmanned aerial vehicles cooperative multi-targets assignment under unified model," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 3, pp. 765–780, Jun. 2017, doi: 10.1007/s13042-015-0364-3.

[22] J. G. Falcón-Cardona, R. H. Gómez, C. A. C. Coello, and M. G. C. Tapia, "Parallel multi-objective evolutionary algorithms: A comprehensive survey," *Swarm Evol. Comput.*, vol. 67, no. 3, Dec. 2021, Art. no. 100960, doi: 10.1016/j.swevo.2021.100960.0.

[23] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 10, pp. 1627–1643, Oct. 2021, doi: 10.1109/JAS.2021.1004129.

[24] X. Chai, Z. Zheng, J. Xiao, L. Yan, B. Qu, P. Wen, H. Wang, Y. Zhou, and H. Sun, "Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment," *Aerosp. Sci. Technol.*, vol. 121, Feb. 2022, Art. no. 107287, doi: 10.1016/j.ast.2021.107287.

[25] C. Huang, X. Zhou, X. Ran, J. Wang, H. Chen, and W. Deng, "Adaptive cylinder vector particle swarm optimization with differential evolution for UAV path planning," *Eng. Appl. Artif. Intell.*, vol. 121, May 2023, Art. no. 105942, doi: 10.1016/j.engappai.2023.105942.

[26] Vikas, D. R. Parhi, and A. K. Kashyap, "Humanoid robot path planning using memory-based gravity search algorithm and enhanced differential evolution approach in a complex environment," *Exp. Syst. Appl.*, vol. 215, Apr. 2023, Art. no. 119423, doi: 10.1016/j.eswa.2022.119423.

[27] I. P. Souza, M. C. S. Boeres, and R. E. N. Moraes, "A robust algorithm based on differential evolution with local search for the capacitated vehicle routing problem," *Swarm Evol. Comput.*, vol. 77, Mar. 2023, Art. no. 101245, doi: 10.1016/j.swevo.2023.101245.

[28] S. Aggarwal and K. K. Mishra, "X-MODE: Extended multi-operator differential evolution algorithm," *Math. Comput. Simul.*, vol. 211, pp. 85–108, Sep. 2023, doi: 10.1016/j.matcom.2023.01.018.

[29] J. Cortez-González, A. Hernández-Aguirre, R. Murrieta-Dueñas, R. Gutiérrez-Guerra, S. Hernández, and J. G. Segovia-Hernández, "Process optimization using a dynamic self-adaptive constraint handling technique coupled to a differential evolution algorithm," *Chem. Eng. Res. Design*, vol. 189, pp. 98–116, Jan. 2023, doi: 10.1016/j.cherd.2022.11.006.

[30] Z. Liao, X. Mi, Q. Pang, and Y. Sun, "History archive assisted niching differential evolution with variable neighborhood for multimodal optimization," *Swarm Evol. Comput.*, vol. 76, Feb. 2023, Art. no. 101206, doi: 10.1016/j.swevo.2022.101206.

[31] S. Zhao, M. Wang, S. Ma, and Q. Cui, "A feature selection method via relevant-redundant weight," *Exp. Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 117923, doi: 10.1016/j.eswa.2022.117923.

[32] L. Bai, H. Li, W. Gao, J. Xie, and H. Wang, "A joint multiobjective optimization of feature selection and classifier design for high-dimensional data classification," *Inf. Sci.*, vol. 626, pp. 457–473, May 2023, doi: 10.1016/j.ins.2023.01.069.

[33] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020, doi: 10.1016/j.ins.2019.08.040.

[34] X. Wang, Y. Wang, K.-C. Wong, and X. Li, "A self-adaptive weighted differential evolution approach for large-scale feature selection," *Knowl.-Based Syst.*, vol. 235, Jan. 2022, Art. no. 107633, doi: 10.1016/j.knosys.2021.107633.

[35] G. Sun, G. Yang, and G. Zhang, "Two-level parameter cooperation-based population regeneration framework for differential evolution," *Swarm Evol. Comput.*, vol. 75, Dec. 2022, Art. no. 101122, doi: 10.1016/j.swevo.2022.101122.

[36] H. Ali, G. Xiong, M. H. Haider, T. S. Tamir, X. Dong, and Z. Shen, "Feature selection-based decision model for UAV path planning on rough terrains," *Exp. Syst. Appl.*, vol. 232, Dec. 2023, Art. no. 120713, doi: 10.1016/j.eswa.2023.120713.

GANG HUANG was born in Jiangsu, China, in 1991. He received the bachelor's degree from the Jincheng College, Nanjing University of Aeronautics and Astronautics, in 2015, and the master's degree from Nanchang Hangkong University, in 2020. He is currently pursuing the Ph.D. degree with the School of Aerospace Science and Technology, Space Engineering University. His research interests include evolutionary computing and intelligent collaboration.

MIN HU was born in 1983. He received the B.S. and M.S. degrees from the Equipment College, China, in 2006 and 2008, respectively, and the Ph.D. degree from Space Engineering University, Beijing, China, in 2012. He is currently an Associate Professor with Space Engineering University. His research interests include spacecraft orbit dynamics, distributed spacecraft dynamics and control, the design of constellation, and space mission analysis. He has published more than 60 articles and authorized 25 national invention patents.

XUEYING YANG was born in 1998. She received the B.S. degrees from Nanchang Hangkong University, Nanchang, China, in 2016 and 2020, and the M.S. degrees from the Belarusian State University of Informatics and Radio-Electronics, Belarus, in 2020 and 2022. She is currently pursuing the Ph.D. degree in aeronautical and astronautical science and technology with Space Engineering University, Beijing, China. Her research interests include multi-satellite mission planning and scheduling.

FEIYAO HUANG was born in Henan, China, in 1998. He received the B.S. degree in computer science and technology from Jimei University, in 2021. He is currently pursuing the M.S. degree in aerospace science and technology with Space Engineering University. His research interests include visual recognition and machine learning, most recently focusing on the application of deep learning to remote sensing image object recognition.