A Differential Evolution for Task Allocation for Multi-UAVs with No-Fly Zone

1st Na Wang
College of Information and Communication, National University of Defense Technology
Wuhan 430035, China
syesun@hotmail.com

2nd Qingzheng Xu*
College of Information and Communication, National University of Defense Technology
Wuhan 430035, China
xuqingzheng@hotmail.com

3rd Feifan Liao
College of Information and Communication, National University of Defense Technology
Wuhan 430035, China
liaofeifan@126.com

4th Weihu Zhao
College of Information and Communication, National University of Defense Technology
Wuhan 430035, China
zhaoweihuandy@126.com

5th Zhiyuan Zhao
College of Information and Communication, National University of Defense Technology
Wuhan 430035, China
zhaozhiyuan1986@sina.com

6th Lei Wang
College of Information and Communication, National University of Defense Technology
Wuhan 430035, China
wanglei12a@nudt.edu.cn

2023 7th Asian Conference on Artificial Intelligence Technology (ACAIT) | 979-8-3503-5914-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ACAIT60137.2023.10528614

***Abstract*—Unmanned aerial vehicles (UAVs) have significant prospects in military and civilian fields. Multi-UAVs can cooperatively complete tasks more efficiently and economically than a single UAV. As a typical coordination pattern for multi-UAVs, task allocation is a combinatorial optimization problem by which they are allocated to accomplish many tasks. To date, several algorithms have been proposed for varied scenario. In this paper, the task allocation problem for multi-UAVs with no-fly zone is studied. First, the no-fly zone is defined as a circular in 2D surface, and then two accurate and fast approaches are proposed respectively, in order to calculate the flight distance with no-fly zone. The original differential evolution (DE) cannot be directly applied to this optimization problem due to its discrete feasible solution. Therefore, some key operations of DE are modified to suit the needs of this optimization problem as follows: solution coding, mutation operation, crossover operation, etc. To verify the proposed algorithm, some experiments are done on 10 UAVs and 10 tasks. For both simple and complex cases, the experimental results confirm that the mathematical model constructed in this paper is reasonable, and the proposed DE is effective, especially for task allocation problem for multi-UAVs with no-fly zones.**

Keywords—unmanned aerial vehicle, task allocation, differential evolution, no-fly zone, flight distance

I. INTRODUCTION

Unmanned aerial vehicles (UAVs) have some marvelous advantages, for instance, low cost, no casualties, flexible operation and reliable performance. In recent years, with the rapid development of UAVs hardware equipment and its post-processing software, it has attracted great attention all over the world [1], and has been widely applied in military and civilian fields, including target surveillance, remote sensing, multi-target tracking and so on. With the significant increase in the number and complexity of tasks, a single UAV cannot easily accomplish multiple tasks simultaneously. Therefore, the coordination among multi-UAVs is a key issue in complex application scenario [2].

The typical coordination patterns for multi-UAVs are collision avoidance, task allocation, path planning and formation reconstruction. In task allocation, multiple sub-tasks should be assigned to several UAVs in order to accomplish a complex task jointly under the task requirements and UAV performance [3]. To solve the task allocation problem for multi-UAVs, a number of approaches proposed can be divided into the following four categories: centralized algorithm, distributed algorithm, bio-inspired algorithm, and multi-fusion algorithm [4]. The differential evolution (DE) algorithm used in this paper belongs to the bio-inspired algorithm.

In information warfare, the effectiveness of UAVs is remarkably affected by complex electromagnetic environment and complex natural environment. As a new mode of air operation, setting up "no-fly zones" in local war, which is emerged after the 1990s, has the following features. First of all, its success is based on the absolute superiority of air offensive power. Secondly, air strategic deterrence and air tactical attack are well combined in this mode. Lastly, the scale may be large or small, the process may be fast or slow, and the degree may be tight or loose. Therefore, as a new way to military intervention by powerful countries, the establishment of "no-fly zones" has aroused great attention in the military field. When multi-UAVs execute the reconnaissance or strike task, they may inadvertently enter the no-fly zone, affecting the task execution inevitably. Therefore, it is possible that the task allocation result is not the best optimal solution.

The task allocation problem for multi-UAVs with no-fly zone is studied in this paper. The main contributions of this paper are as follows: (1) In order to calculate the flight distance with no-fly zone modeled as a circular in 2D surface, two accurate and fast approaches are proposed, respectively. (2) In order to suit the needs of the task allocation problem for multi-UAVs with/without no-fly zone, some key operations of the original DE are modified as follows: solution coding, mutation operation, crossover operation, etc.

The main structure of this paper is organized as follows: Section II describes the basic mathematical model of task allocation for multi-UAVs and the original DE algorithm. In Section III, the no-fly zone is first defined as a circular in 2D surface, and then two accurate and fast approaches are proposed respectively, in order to calculate the flight distance with no-fly zone. Some key operations of the original DE are introduced in detail to solve task allocation for multi-UAVs with/without no-fly zone in Section IV. The simulation configuration, experimental results, and discussion are given for both simple and complex cases in Section V. Finally, the conclusions and future research of this study are presented in Section VI.

*Corresponding author

II. BACKGROUND

A. Mathematical Model of Task Allocation for Multi-UAVs

A significant feature of task allocation problem for multi-UAVs is that there are many constraints, such as UAV capability constraints, tactical constraints, battlefield environment constraints, and task constraints. Therefore, as an optimization problem, the key of mathematical model construction is the reasonable definition of objective function and the corresponding constraints.

Before building a mathematical model, several assumptions are made as follows. (1) Once a certain task is assigned to a UAV, this task will definitely be accomplished. (2) The number of UAVs is exactly the same as the number of target tasks. (3) The battlefield area is empty, and the UAVs can fly directly to the task area assigned in advance.

Suppose there are n UAVs and n tasks to be accomplished, and exactly one UAV is required for each task. Thus, the task allocation problem for multi-UAVs is to determine which UAV to accomplish which task so that the total cost is the lowest or the total benefit is the highest after completing those tasks. In this paper, the total cost is directly described as the sum of the distances between each UAV and the corresponding task. Therefore, the objective function is defined as follows:

$$\min f = \sum_{i=1}^{n}\sum_{j=1}^{n} x_{ij} d_{ij} \tag{1}$$

where x_{ij} indicates whether the ith UAV accomplishes the jth task. If it does, $x_{ij} = 1$, otherwise $x_{ij} = 0$. And d_{ij} represents the Euclidean distance between the ith UAV and the jth task.

In this mathematical model, the constraints are defined as follows:

$$\sum_{i=1}^{n} x_{ij} = 1 \tag{2}$$

$$\sum_{j=1}^{n} x_{ij} = 1 \tag{3}$$

Equation (2) indicates that each UAV only needs to execute one task, and Equation (3) indicates that each task only needs one UAV to execute.

B. Differential Evolution

DE algorithm was proposed by Storn and Price in 1995 [5]. At first, it was designed to solve the Chebyshev polynomial problem, and then its advantages were gradually discovered, such as simple principle, few control parameters and convenient implementation [6]. The standard DE consists of the following steps:

(1) Population initialization. Suppose there are N_p individuals in the population, and each is a D-dimensional vector. Then, the initial population is randomly generated as follows:

$$x_{i,j} = x_{i\_min} + rand \times (x_{i\_max} - x_{i\_min}) \tag{4}$$

where $x_{i,j}$ represent the jth dimensional component of the ith individual, and x_{i_min} and x_{i_max} represent the lower and upper bounds of each dimensional component of each vector, respectively.

(2) Mutation operation. The mutation vector is calculated as follows:

$$v_i = x_{i3} + F(x_{i1} - x_{i2}) \tag{5}$$

where x_{i1}, x_{i2} and x_{i3} are three different individuals randomly selected from the current population, and F is the scale factor. As a results, v_i is the newly generated mutation vector.

(3) Crossover operation. Each vector selects its component from the mutation vector or the original vector according to a certain probability, as shown in Equation (6).

$$u_{i,j} = \begin{cases} v_{ij} & if(rand < CR \ \ or \ \ j = j_r) \\ x_{ij} & otherwise \end{cases} \tag{6}$$

where CR is the crossover probability and j_r is a random component, ensuring that, after crossover operation, at least one dimension of each individual is provided by the mutation vector.

(4) Out-of-bounds processing. When applying the mutation operation, it is possible that some dimensions of an individual are out of the bounds of feasible solutions. Therefore, it is generally necessary to reassign a feasible value to make it within the boundary range.

(5) Selection operation. Greedy selection operation is generally utilized. According to their fitness value, the better individual is selected from the original vector and the test vector into the next generation.

$$x_i^{G+1} = \begin{cases} u_i^G & if(f(u_i^G) < f(x_i^G)) \\ x_i^G & otherwise \end{cases} \tag{7}$$

C. Related Works

An on-line predictor-corrector reentry guidance algorithm was proposed to satisfy path and no-fly zone constraints for hypersonic vehicles [7]. L1-Penalized Sequential Convex Programming (LPSCP) method was proposed to solve the UAV trajectory optimization problem when there are several no-fly zones along the trajectory [8]. The discrete path decision and continuous trajectory optimization for no-fly zones avoidance was integrated as a mixed-integer optimal control problem (MIOCP) [9].

In [10], to cover all feasible avoidance paths by virtual waypoints, the mission space with no-fly zones was transformed into a directed-graph. Therefore, a no-fly zones avoidance path decision problem was modeled and solved as a path search problem. In [11], a novel Tabu search algorithm was designed to solve VRT-TDR (Vehicle Routing Problem with Truck and Drone Considering Regional Restriction). This algorithm stimulates human brain's short-term memory function to gradually seek the optimal solution.

A novel collision avoidance algorithm was presented to ensure minimum separation between the vehicles considering no-fly zones [12]. The proposed algorithm has been addressed in two steps: conflict detection and resolution. A hybrid particle swarm algorithm was also designed that combined obstacle avoidance and path planning [13]. Specifically, an improved A* algorithm was added to solve the obstacle avoidance.

Recently, a rigorous mathematical model was conducted for the UAV-assisted item delivery scenario in the presence of a number of no-fly zones [14]. The original problem was converted into an equivalent problem with a difference-of-convex structure and then solved by the penalty convex-concave procedure method.

III. Mathematical Model of No-fly Zone and Calculation of Flight Distance

A. Mathematical Model of No-fly Zone

In [15], the no-fly zone of obstacles or threats can be approximated by irregular convex polygons at the same horizontal height. In this case, UAVs need to avoid the space field above it, and there is enough distance between some no-fly zones for UAVs to pass. In [7, 9], no-fly zones are modeled as a cylinder with its center at the specified longitude and latitude and a given radius. In [16], the no-fly zone was defined as a circular zone of infinite height, and determined the no-fly zone by giving the central coordinates and radius of the no-fly zone. Recently, dynamic no-fly zone was designed based on a sphere centered on the current flight with a radius [17].

In this paper, the no-fly zone is defined as a circular in 2D surface. As shown in Fig. 1, the center and radius of this circular are O and *R*, respectively. The UAV is located at U, the target point to execute the reconnaissance or strike task is located at T_1 or T_2.

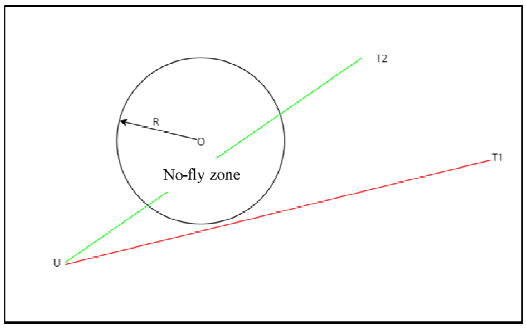

Fig. 1 Schematic figure of no-fly zone

Obviously, if there is no no-fly zone in the battlefield, the flight path of this UAV is a line segment between the UAV (located at U) and the target point (located at T_1 or T_2).

If the distance *d* from the center O to the flight trajectory of the UAV is greater than the radius *R* (as shown by the red line U-T_1), the flight of UAV is considered as safety. Conversely, if the distance *d* is less than the radius *R* (as shown by green line U-T_2), the flight of UAV is considered as unsafety and then the UAV needs to fly in another way to avoid the no-fly zone. At this time, the flight trajectory of this UAV needs to be replanned, and the flight distance also needs to be recalculated.

B. Accurate Calculation Method of Flight Distance

In this section, the accurate method is put forward to calculate the flight distance for two cases: a no-fly zone and two no-fly zones.

(1) A no-fly zone. Suppose there is only one no-fly zone. In Fig. 2, U is a UAV, T is a target task, and circle O is a no-fly zone in the battlefield.

As shown in Fig. 2, the shortest path between the UAV and the target task is line segment U-A + arc A-B + line segment B-T. Among them, U-A and B-T are two tangent lines of the circle O. Obviously, the flight distance between U and T is the sum of the lengths of line segment U-A, line segment B-T and arc A-B. Therefore, the specific calculation process can be described as follows:

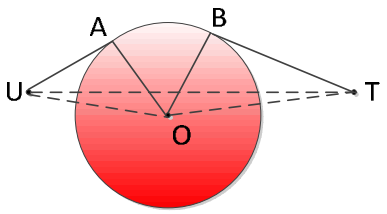

Fig. 2 Case 1: A no-fly zone

Step 1: In the right triangle OAU, given the radius O-A and length of line segment O-U, the length of line segment U-A can be calculated easily based on the Pythagorean theorem, and the angle AOU can also be calculated easily based on the cosine theorem.

Step 2: Similarly, it is easy to obtain the length of line segment B-T and the angle BOT.

Step 3: In the triangle UOT, the length of the three sides (U-O, O-T, and T-U) is known, and then the angle UOT can further be obtained based on the cosine theorem.

Step 4: Calculate angle AOB = angle UOT - angle AOU - angle BOT.

Step 5: In the circle O, it is easy to obtain the length of arc A-B = the radian of angle AOB × the radius *R*.

Step 6: The flight distance between U and T = length of line segment U-A + length of line segment B-T + length of arc A-B.

(2) Two no-fly zones. Suppose there are two no-fly zones. As shown in Fig. 3, the UAV is located at U, and the target task is located at T. Two no-fly zones are circle O_1 and circle O_2, respectively. And, their center O_1, O_2 and radius *R* (the same radius for tow circles) are known in advance. In addition, lines A_1-A_2 and B_1-B_2 are two common tangents to circles O_1 and O_2.

It is obvious that, the UAV will pass through two no-fly zones at the same time only if U and T are located on line segments A1-B1 and A2-B2, respectively. In other cases, UAV can only cross one of two no-fly zones.

If the UAV only passes through one of two no-fly zones, the calculation process of the flight distance is described in case 1. Thus, we will not repeat it here.

Next, we will present the accurate method in detail where the UAV must pass through two no-fly zones simultaneously. According to the elementary geometry, the shortest path between the UAV and the target task is line segment U-Q_1 + arc Q_1-P_1 + line segment P_1-P_2 - arc P_2-Q_2 + line segment Q_2-T. Thus, the flight distance between U and T is the sum of the lengths of line segment U-Q_1, arc Q_1-P_1, line segment P_1-P_2, arc P_2-Q_2 and line segment Q_2-T. Therefore, the specific calculation process can be described as follows:

Step 1: Obviously, the length of line segment P_1-P_2 is the length between center O_1 and center O_2.

Step 2: The calculation process of line segments U-Q_1 and Q_2-T is similar to the first case (a no-fly zone).

Step 3: According to the coordinates of U and O_1, the coordinates of tangent point Q_1 can be obtained.

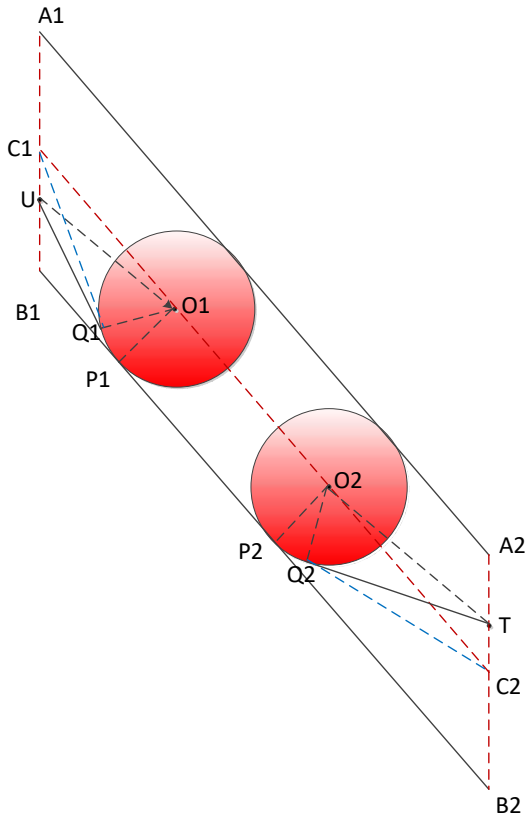

Fig. 3 Case 2: Two no-fly zones

Step 4: In the triangle $C_1O_1Q_1$, given three sides (C_1-O_1, O_1-Q_1, and Q_1-C_1), the angle $C_1O_1Q_1$ can be obtained based on the cosine theorem.

Step 5: The angle $Q_1O_1P_1$ = angle $C_1O_1P_1$ (90 degrees) - angle $C_1O_1Q_1$. Then the length of arc P_1-Q_1 = the radian of angle $P_1O_1Q_1$ × radius R.

Step 6: Similarly, the length of arc P_2-Q_2 can be obtained.

Step 7: The flight distance between U and T = length of line segment U-Q_1 + length of arc Q_1-P_1 + length of line segment P_1-P_2 + length of arc P_2-Q_2 + length of line segment Q_2-T.

Note that the accurate calculation process of flight distance described in this section is relatively complex and cannot automatically be applied to more complex situations. Therefore, this method is only suitable for some simple situation where there are 1 or 2 no-fly zones, and not for the complex situation where there are more than 2 no-fly zones. In addition, in the actual battlefield, whether the accurate flight distance has practical military significance is also debatable in most cases.

C. Fast Calculation Method of Flight Distance

From a historical viewpoint, the UAVs can encounter a very limited number of no-fly zones over the whole flight course. Therefore, in order to simplify the calculation process of flight distance, in this paper the maximal detour distance is utilized no matter how the UAV flies realistically.

As shown in Fig. 1, the maximal detour distance *MaxD* for a no-fly zone is calculated as follows:

$$MaxD = \pi R - 2R = (\pi - 2)R \tag{8}$$

Therefore, the flight distance D_{UT} between U and T can be calculated as follows:

$$D_{UT} = Dis_{UT} + m \times MaxD \tag{9}$$

where Dis_{UT} represents the Euclidean distance between U and T, and m is the number of no-fly zones encountered during the whole flight course. This calculation method can quickly calculate the flight distance between U and T, even if there are multiple no-fly zones. As a result, it can meet the realistic needs for complex battlefield in most cases.

IV. Differential Evolution for Task Allocation Problem

A. Task Allocation Problem for Multi-UAVs

The goal of task allocation problem for multi-UAVs is that the total flight distance of all UAVs is minimized when each target task can be executed. However, the original DE described in Section II.B cannot be directly applied to this optimization problem due to its discrete feasible solution. Therefore, it is necessary to make some changes to the traditional DE based on the features of task allocation problem. The key operations of DE are described as follows.

(1) Solution coding and population initialization. The individuals in DE adopt the integer coding mechanism in this paper. The code length is the number of tasks to be executed, and each gene of the individual is an integer value, representing the task sequence number.

Suppose that there are 5 UAVs and 5 target tasks to be executed. As shown in Fig. 4, this vector indicates that the 1st UAV executes the 3rd task, the 2nd UAV executes the 5th task, the 3rd UAV executes the 1st task, the 4th UAV executes the 4th task, and the 5th UAV executes the 2nd task.

3	5	1	4	2

Fig. 4 Example of coding of an individual

According to the above coding method, N_p individuals can be randomly generated and then constitute the initial population.

(2) Mutation operation. Since the scale factor F is usually a decimal, the mutation vector v_i is no longer a feasible candidate solution. In this paper, a repair strategy of mutation operation is proposed in order to obtain a feasible solution quickly.

First, a temporary mutation vector v'_i is calculated according to the DE/rand/1 mutation strategy. Then, each dimension of v'_i is sorted in ascending order. Finally, the order number is assigned to the corresponding dimension, and then a feasible mutation vector v_i is obtained in this way.

For example, three vectors x_{i1} = [3 5 1 4 2], x_{i2} = [1 5 4 2 3 3], and x_{i3} = [4 1 2 3 5] are randomly selected, and the scale factor F = 0.5. According to the repair strategy proposed above, v'_i = [5 1 0.5 4 4.5] is firstly calculated by Equation (5). Then, all dimensions are sorted in ascending order and a feasible mutation vector v_i = [5 2 1 3 4] is obtained.

(3) Crossover operation. Generally, DE adopts the binomial crossover strategy as shown in Equation (6). Note that the crossover operation in this paper is no longer applied to each dimension of the mutation vector, but to each mutation vector as a whole as follows. Obviously, this crossover

method can effectively exclude the infeasible solution, leading to improve the algorithm efficiency.

$$u_i = \begin{cases} v_i & if(rand < CR) \\ x_i & otherwise \end{cases} \quad (8)$$

(4) Out-of-bounds processing. After the mutation operation and crossover operation, the newly generated test vector is still a feasible solution. It is an important distinction compared with the traditional DE. Therefore, no additional out-of-bounds processing is required in this paper.

(5) Selection operation. DE in this paper still selects vectors into the next generation based on the greedy selection principle, as shown in Equation (7), without any modification.

(6) Fitness calculation. Because the integer coding mechanism is utilized in this paper, the fitness of each individual cannot be directly calculated. Therefore, the fitness can be calculated by mapping different attribute values of individuals into the distance matrix. Note that, for task allocation problem for multi-UAVs without no-fly zones, the values in distance matrix D are the Euclidean distances between all UAVs and all tasks.

For example, there are 5 UAVs and 5 tasks. The distance matrix D is first constructed, where D_{ij} represents the distance between the ith UAV and the jth target task. Suppose that the distance matrix $D = \begin{bmatrix} 9 & 5 & 3 & 2 & 3 \\ 3 & 5 & 6 & 7 & 2 \\ 3 & 2 & 8 & 9 & 6 \\ 4 & 1 & 3 & 7 & 4 \\ 3 & 7 & 8 & 3 & 1 \end{bmatrix}$, where the elements in the first row indicates that the distance from the first UAV to 5 tasks, respectively. For example, the distance to the 1st task is 9, the distance to the 2nd task is 5, the distance to the 3rd task is 3, etc. Therefore, the fitness of individual x = [3 5 1 4 2] is the sum of the value with underline in the distance matrix $D = \begin{bmatrix} 9 & 5 & \underline{3} & 2 & 3 \\ 3 & 5 & 6 & 7 & \underline{2} \\ \underline{3} & 2 & 8 & 9 & 6 \\ 4 & 1 & 3 & \underline{7} & 4 \\ 3 & \underline{7} & 8 & 3 & 1 \end{bmatrix}$. Finally, we can obtain the fitness $f(x) = 3 + 2 + 3 + 7 + 7 = 22$.

The pseudo-code of DE for task allocation problem for multi-UAVs is given in Algorithm 1 as follows.

Algorithm 1: Differential evolution for task allocation problem for multi-UAVs

Input: population size N_p, scale factor F, crossover probability CR, maximum function evaluation times MaxNFE

1. Initialization parameter: generation $t = 0$; number of function evaluation NFE = 0
2. Construct the distance matrix D according to the coordinates of the UAVs and the target tasks
3. Randomly generate the initial population P_0
4. **while** the termination condition is not meet **do**
5. Calculate the fitness of each individual
6. Perform DE/rand/1 mutation operation with the repair policy described in this section to generate a mutation vector
7. Perform crossover operation to generate test vector
8. Calculate the fitness of the test vector
9. Perform the selection operation to generate the population P_{t+1} in the next generation
10. Update parameters: $t = t + 1$; NFE = NFE + N_p
11. **end**

Output: Optimal individual x_{opt} and the corresponding fitness in population P_t

B. Task Allocation Problem for Multi-UAVs with No-fly Zones

When solving the task allocation problem for multi-UAVs with no-fly zones, the key modification of DE is the calculation method of the flight distance. In order to construct the distance matrix D, DE needs to quickly calculate the flight distances between all UAVs and all target tasks according to the Equation (9). This is the key difference of DE algorithm for two kinds of task allocation problem.

The rest part of DE is the same as Algorithm 1, described in Section IV.A. Thus, the details of DE are not repeated here.

V. Simulation Experiments

In this section, we will examine the reasonableness of the mathematical model constructed in Section III and the effectiveness of DE for task allocation problem for multi-UAVs described by Algorithm 1 in Section IV. In the proposed DE algorithm, some key operations of the original DE is modified to suit the needs of task allocation problem for multi-UAVs with/without no-fly zones. Note that, the goal of this paper is not to propose an efficient new algorithm for task allocation problem. Therefore, we will not compare it with the state-of-the-art bio-inspired algorithms in this section.

A. Set Up

Suppose that our army has a total of 10 UAVs in the battlefield. And there are 10 different tasks to be executed by them. For a simple description, these UAVs and target tasks are both numbered 1 to 10, respectively. Furthermore, our army can determine the geographical coordinates of these target tasks with the help of some technical means. The problem we faced is how to assign these tasks to different UAVs. Two supposed cases are considered in this paper. For each case, those coordinates for UAVs and tasks are known in advance and chosen randomly in this section.

(1) Simple case. Tables 1 and 2 show the coordinates of 10 UAVs and 10 target tasks, respectively. In this battlefield, there are two no-fly zones with the centers are (-50, 50) and (-60, 60), and the radius R is 5.

Table 1 Coordinates of the UAVs for simple case

UAV number	Coordinates	UAV number	Coordinates
1	(100, 10)	2	(100, 20)
3	(100, 30)	4	(100, 40)
5	(100, 50)	6	(100, 60)
7	(100, 70)	8	(100, 80)
9	(100, 90)	10	(100, 100)

Table 2 Coordinates of the tasks for simple case

Task number	Coordinates	Task number	Coordinates
1	(-100, 5)	2	(-100, -15)

3	(-100, -25)	4	(-100, -35)
5	(-100, -45)	6	(-100, -55)
7	(-100, -65)	8	(-100, -75)
9	(-100, -85)	10	(-100, -95)

(2) Complex case. Tables 3 and 4 list the coordinates of 10 UAVs and 10 target tasks. In this battlefield, there are two no-fly zones with the centers are (0,0) and (1000,1000), and the radius R is 200.

Table 3 Coordinates of the UAVs for complex case

UAV number	Coordinates	UAV number	Coordinates
1	(2000, 1920)	2	(1900, 1720)
3	(1990, 1430)	4	(1720, 2020)
5	(1840, 1630)	6	(1660, 1850)
7	(1850, 1740)	8	(1970, 1540)
9	(2090, 1510)	10	(2320, 1920)

Table 4 Coordinates of the tasks for complex case

Task number	Coordinates	Task number	Coordinates
1	(-2000, -1920)	2	(-1320, -1630)
3	(-2360, -1580)	4	(-1770, -1690)
5	(-1880, -1450)	6	(-1480, -1750)
7	(-1950, -1660)	8	(-1920, -1620)
9	(-1380, -1530)	10	(-1860, -1430)

For population-based intelligent algorithm, parameter selection can greatly affect its performance. However, the purpose of this paper is not to find the optimal parameters for DE algorithm. Therefore, some parameters of DE algorithm are borrowed directly from [18], and used to solve the task allocation problem for multi-UAVs in both cases as follows: population number $N_p = 100$, dimension of variables $D = 10$, maximum generation $G = 200$, scale factor $F = 0.5$, crossover probability $CR = 0.9$. Note that, the theoretical meaning of these main parameters can be found in Section II.A. In order to eliminate the random of population-based DE, the algorithm run independently 30 times for each case.

B. Experimental Results and Discussion

(1) Simple case. Table 5 records the results of 30 independent experiments. It can be seen that the global optimal solution x_{opt} = [1 2 4 5 3 9 6 7 8 10], and the corresponding fitness (the sum of all flight distances) is 2005.1105. Compared to the task allocation problem without no-fly zone, the optimal fitness is increased by 0.22%. Furthermore, we can find from Table 5 that DE has obtained the global optimal solution 13 times, and the success rate is 43.33%. From another point of view, the worst fitness obtained is 2009.8237, which is 0.26% larger than the optimal fitness. In most cases, this tiny error can be neglected in the battlefield. Fig. 5 illustrates the convergence curve of the objective function during the whole evolution, and Fig. 6 illustrates the optimal allocation scheme finally obtained by the proposed DE. These experimental results fully indicate that the proposed DE can effectively solve the task allocation problem for multi-UAVs with no-fly zones.

Table 5 Results of 30 independent experiments

Experiment	Fitness	Optimal solution
1	2006.1052	[1 2 4 5 3 9 6 7 10 8]
2	2006.1052	[1 2 4 5 3 9 6 7 10 8]
3	2007.8268	[1 2 4 5 3 6 7 8 9 10]
4	2005.6060	[1 2 4 5 3 9 6 8 7 10]
5	2005.1105	[1 2 4 5 3 9 6 7 8 10]
6	2005.1105	[1 2 4 5 3 9 6 7 8 10]
7	2005.1105	[1 2 4 5 3 9 6 7 8 10]
8	2009.8237	[1 3 2 4 5 9 6 7 8 10]
9	2005.1105	[1 2 4 5 3 9 6 7 8 10]
10	2006.1089	[1 4 2 5 3 9 6 7 8 10]
11	2006.1052	[1 2 4 5 3 9 6 7 10 8]
12	2006.1052	[1 2 4 5 3 9 6 7 10 8]
13	2006.1052	[1 2 4 5 3 9 6 7 10 8]
14	2005.1105	[1 2 4 5 3 9 6 7 8 10]
15	2005.1105	[1 2 4 5 3 9 6 7 8 10]
16	2007.8268	[1 2 4 5 3 6 7 8 9 10]
17	2005.1105	[1 2 4 5 3 9 6 7 8 10]
18	2007.8268	[1 2 4 5 3 6 7 8 9 10]
19	2007.1036	[1 4 2 5 3 9 6 7 10 8]
20	2006.1052	[1 2 4 5 3 9 6 7 10 8]
21	2005.1105	[1 2 4 5 3 9 6 7 8 10]
22	2007.8268	[1 2 4 5 3 6 7 8 9 10]
23	2005.1105	[1 2 4 5 3 9 6 7 8 10]
24	2005.1105	[1 2 4 5 3 9 6 7 8 10]
25	2006.1089	[1 4 2 5 3 9 6 7 8 10]
26	2005.1105	[1 2 4 5 3 9 6 7 8 10]
27	2009.8237	[2 1 3 4 5 9 6 7 8 10]
28	2005.1105	[1 2 4 5 3 9 6 7 8 10]
29	2008.5939	[1 5 4 2 3 9 7 6 8 10]
30	2005.1105	[1 2 4 5 3 9 6 7 8 10]

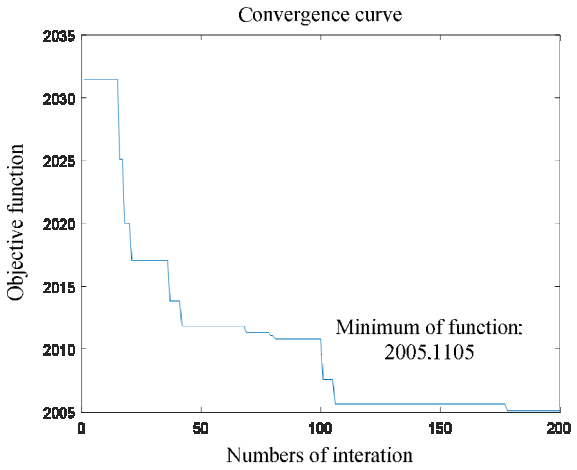

Fig. 5 The convergence curve of the objective function

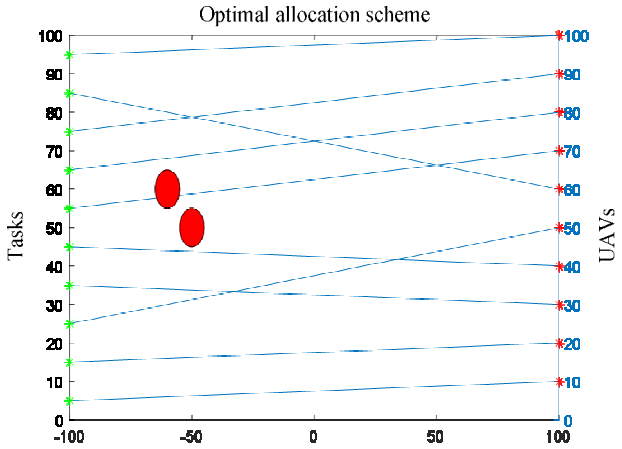

Fig. 6 The optimal allocation scheme obtained

Next, the influence of the radius of the no-fly zone on the experimental results is studied. As can be seen from Table 6, when the radius of the no-fly zone increases from 5 to 20, the optimal solution varies, and the corresponding fitness increases slowly. We think that there are two reasons of the trend. Firstly, when the radius of the no-fly zone increases, the additional flight distance caused by no-fly zone will increase inevitably. Secondly, in order to avoid crossing the no-fly zone, the UAV will choose a relatively farther target task. As a result, the corresponding fitness of the optimal solution is larger and larger.

Table 6 Experimental results of different radius of no-fly zone

Radius of the no-fly zone	5	10	15	20
Fitness	2005.1105	2012.5257	2046.8170	2120.7490
Optimal solution	[1 2 4 5 3 9 6 7 8 10]	[2 3 5 4 1 10 8 6 7 9]	[1 4 5 2 3 9 10 7 6 8]	[1 5 3 4 2 6 8 9 7 10]

(2) Complex case. We conduct three groups of independent experiments with different number of no-fly zone: without no-fly zone, one no-fly zone (see Fig. 7), and two no-fly zones (see Fig. 8). DE algorithm runs independently 30 times for each group of experiments runs, and the experimental results are concluded in Table 7.

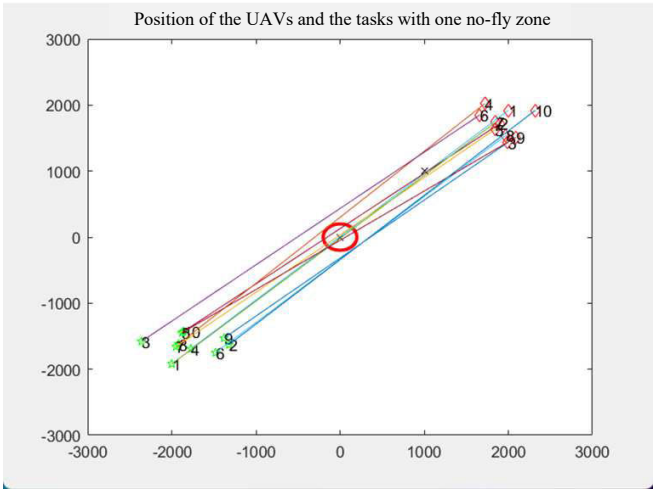

Fig. 7 Position of the UAVs and the tasks with one no-fly zone

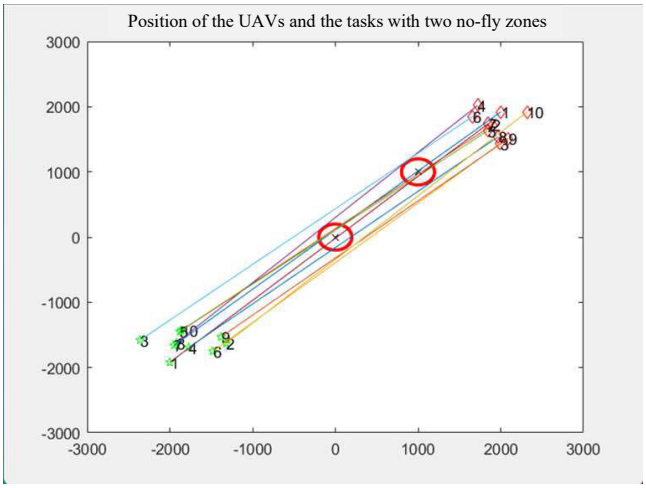

Fig. 8 Position of the UAVs and the tasks with two no-fly zones

Table 7 Results of 30 independent experiments for three groups of independent experiments

Experiment	Fitness for the first group of experiments	Fitness for the second group of experiments	Fitness for the third group of experiments
1	50137.9711	51184.8429	52359.7537
2	50146.1900	51159.2725	52407.5131
3	50142.5623	51144.7466	52367.6354
4	50140.1714	51182.5715	52452.6450
5	50144.4083	51213.9848	52347.8558
6	50140.9634	51185.2934	52328.9658
7	50146.4763	51168.3267	52408.7941
8	50143.4208	51152.3816	52362.3244
9	50145.0041	51153.2295	52371.0100
10	50144.2857	51149.2809	52424.7669
11	50144.6146	51153.2295	52489.7243
12	50143.8966	51155.5730	52421.0454
13	50137.4074	51148.8370	52349.3961
14	50146.0060	51207.0960	52346.0215
15	50142.0659	51211.3227	52422.6191
16	50141.0807	51161.0218	52419.3724
17	50143.7502	51182.0808	52604.7398
18	50139.5943	51164.2310	52349.3961
19	50140.6821	51165.3466	52462.8439
20	50144.5816	51306.7107	52332.1069
21	50144.3752	51145.3425	52350.1896
22	50143.9231	51182.4730	52515.0735
23	50143.4946	51250.6854	52473.5691
24	50141.0807	51179.0099	52384.1915
25	50137.2444	51249.5388	52489.3112
26	50141.9231	51202.8200	52403.8105
27	50144.8039	51301.1561	52421.1017
28	50143.2145	51183.4718	52423.0185
29	50141.3555	51171.4958	52380.4611
30	50139.0378	51156.4576	52496.6669
Maximum	50146.4763	51306.7107	52604.7398
Minimum	50137.2444	51144.7466	52328.9658
Mean	50142.4783	51188.2277	52415.6134
Median	50143.2145	51179.0099	52408.1536
Variance	6.6128	1800.7718	4128.1977

As can be seen from Table 7, when solving the task allocation problem for multi-UAVs with no-fly zones, the variance of fitness for different runs is very small. This results further indicates that the proposed DE in this paper runs stably and it can effectively solve the task allocation problem for multi-UAVs.

In addition, when the number of no-fly zones increases, the fitness also happens a lot of changes. It is consistent with common sense. From this result we inferred that the mathematical model constructed in this paper is reasonable and has a significant role in future wars.

VI. Conclusion

Task allocation is a combinatorial optimization problem by which multi-UAVs is allocated to accomplish many tasks. Several algorithms have been designed for UAV networks. In

this paper, the no-fly zone is first defined as a circular in 2D surface, and then the accurate method and the fast method are proposed in order to calculate the flight distance. According to the feature of task allocation problem for multi-UAVs, some key operations of DE are modified as follows: solution coding, mutation operation, crossover operation, etc. Simulation experiments on 10 UAVs and 10 tasks are carried out for both simple and complex cases. The experimental results confirm that the mathematical model constructed in this paper is reasonable, and the proposed DE is effective, especially for task allocation problem for multi-UAVs with no-fly zones well. Although the number of UAVs and tasks in two cases studied in this paper is limited to 10, we have reason to believe that the proposed method can also be applied to a larger scale.

In the future, one of our main research directions will be to further improve the efficiency of the proposed DE algorithm and compare it with other swarm intelligence algorithms [19]. It would also be interesting to explore ways to expand the applicability of our approach to a real-world optimization scenario.

ACKNOWLEDGMENT

This work was supported by the National Natural Science Foundation of China (Grant no. 62272384).

REFERENCES

[1] Department of Defense, "Unmanned Systems Integrated Roadmap FY2013-2038: 14-S-0553," Washington, USA, Department of Defense, 2013.

[2] H. Zhang, B. Xin, L. H. Dou, J. Chen, and K. Hirota, "A review of cooperative path planning of an unmanned aerial vehicle group," Frontiers of Information Technology & Electronic Engineering, vol. 21, no. 12, pp. 1671-1694, 2020.

[3] C. Q. Wang, D. Mei, Y. Wang, X. W. Yu, W. Sun, D. Wang, and J. Q. Chen, "Task allocation for Multi-AUV system: A review," Ocean Engineering, vol. 266, no. 1, Article ID: 112911, 2022.

[4] S. Poudel, and S. Moh. "Task assignment algorithms for unmanned aerial vehicle networks: A comprehensive survey," Vehicular Communications, vol. 35, no. 1, Article ID: 100469, 2022.

[5] R. Storn, and K. Price, "Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces," Journal of Global Optimization, vol. 11, no. 4, pp. 341-359, 1997.

[6] S. Das, and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," IEEE Transactions on Evolutionary Computation, vol. 15, no. 1, pp. 4-31, 2011.

[7] D. Zhang, L. Liu, and Y. J. Wang, "On-line reentry guidance algorithm with both path and no-fly zone constraints," Acta Astronautica, vol. 117, no. 1, pp. 243-253, 2015.

[8] Y. J. Oh, H. Roh, and M. J. Tahk, "Fast trajectory optimization using sequential convex programming with no-fly zone constraints," IFAC PapersOnLine, vol. 52, no. 12, pp. 298-303, 2019.

[9] Y. Zhang, R. Zhang, and H. F. Li, "Mixed-integer trajectory optimization with no-fly zone constraints for a hypersonic vehicle," Acta Astronautica, vol. 207, no. 1, pp. 331-339, 2023.

[10] Y. Zhang, R. Zhang, and H. F. Li, "Graph-based path decision modeling for hypersonic vehicles with no-fly zone constraints," Aerospace Science and Technology, vol. 116, no. 1, pp. Article ID: 106857, 2021.

[11] R. Yan, L. S. Chen, X. N. Zhu, H. T. Tian, Y. Wen, and Q. Zhang, "Research on vehicle routing problem with truck and drone considering regional restriction," Chinese Journal of Management Science, vol. 30, no. 5, pp. 144-155, 2022. (in Chinese)

[12] H. I. Lee, H. S. Shin, and A. Tsourdos, "UAV collision avoidance considering no-fly-zones," IFAC PapersOnLine, vol. 53, no. 2, pp. 14748-14753, 2020.

[13] M. Miao, Y. Q. Niu, X. H. Li, Y. Zhao, Y. Dong, and P. Wang, "Path planning of unmanned logistic aerial vehicle group under the constraint of a no-fly zone," Journal of Lanzhou University (Natural Science), vol. 59, no. 1, pp. 98-105, 2023.

[14] W. L. Wen, K. Luo, L. H. Liu, Y. Zhang, and Y. J. Jia, "Joint trajectory and pick-up design for UAV-assisted item delivery under no-fly zone constraints," IEEE Transactions on Vehicular Technology, vol. 72, no. 2, pp. 2587-2592, 2023.

[15] E. Koyuncu,·N. K. Ure, and G. Inalhan, "Integration of path/maneuver planning in complex environments for agile maneuvering UCAVs," Journal of Intelligent & Robotic Systems, vol. 57, no. 1/4, pp. 143-170, 2010.

[16] Y. Zhang, R. Zhang, and H. F. Li, "Dual-level path-trajectory generation with complex no-fly zone constraints for hypersonic vehicle," Journal of Astronautics, vol. 43, no. 5, pp. 615-627, 2022. (in Chinese)

[17] G. Y. Tian, C. H. Xiao, and W. C. Liu, "Dynamic no-fly zone for drones," IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, August 19-23, 2019, Leicester, UK, pp. 761-766.

[18] A. K. Qin, and X. D. Li, "Differential evolution on the CEC-2013 single-objective continuous optimization testbed," IEEE Congress on Evolutionary Computation, June 20-23, 2013, Cancún, México, pp. 1099-1106.

[19] J. Tang, H. B. Duan, and S. Y. Lao, "Swarm intelligence algorithms for multiple unmanned aerial vehicles collaboration: A comprehensive review," Artificial Intelligence Review, vol. 56, no. 5, pp. 4295-4327, 2023.