IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 18, 2022, accepted May 1, 2022, date of publication May 4, 2022, date of current version May 12, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3172710

Chaotic Random Opposition-Based Learning and Cauchy Mutation Improved Moth-Flame Optimization Algorithm for Intelligent Route Planning of Multiple UAVs

MINGXI MA[1], JUN WU[1], YUE SHI[2], LONGFEI YUE[1], CHENG YANG[3], AND XUYI CHEN[3]

[1]Air Control and Navigation College, Air Force Engineering University, Xi'an 710051, China
[2]Equipment Management and UAV Engineering College, Air Force Engineering University, Xi'an 710051, China
[3]93787 Army, Beijing 102600, China

Corresponding author: Yue Shi (3159098298@qq.com)

ABSTRACT UAV route planning is the key issue for application of UAV in real-world scenarios. Compared with the traditional route planning methods, although the intelligent optimization algorithm has stronger applicability and optimization performance, it also has the problem of poor convergence accuracy and easy to fall into local optimization. Therefore, an intelligent route planning method for UAV based on chaotic random opposition-based learning and cauchy mutation improved Moth-flame optimization algorithm (OLTC-MFO) is proposed. First, the terrain environment is constructed by digital elevation map, and the threat model is established to realize the equivalent three-dimensional (3D) environment. Then, the opposite population is introduced to increase the diversity of solutions and improve the search speed of the algorithm. Then, the Logistic-Tent chaos map is introduced to realize random perturbation of flame position, which improves the global search capability of the algorithm. Finally, the probability operator and Cauchy mutation operator are introduced, which makes the algorithm not only accept the current solution with a certain probability, but also jump out of the current sub-optimal solution, thus enhancing the global search capability of the algorithm. The simulation results show that when the number of iterations is 1000, the length of route planning based on OLTC-MFO algorithm is 45.3716km shorter than MFO algorithm, and the convergence result of this method is stable and more accurate, which achieves the purpose of assisting UAV combat decision-making.

INDEX TERMS Route planning, moth-flame optimization algorithm, cauchy mutation, random opposition-based learning, chaotic mapping.

I. INTRODUCTION

With the rapid development of artificial intelligence technology, unmanned aerial vehicle (UAV) is more and more widely used, whether in aerial photography, plant protection, remote sensing surveying and mapping in the civil field, or in reconnaissance, electronic jamming and strike in the military field [1]–[19]. However, the primary link to complete the task is to plan the flight route of UAV. In short, the route planning problem refers to finding a route from the starting point to the end that meets the needs according to certain constraints in the established environmental model.

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Wang.

At present, two kinds of methods are widely used in UAV route planning. One is the traditional route planning method, and the other is the intelligent optimization algorithm.

Traditional route planning algorithms mainly include: visual graph method, Dijkstra algorithm, A* algorithm, RRT algorithm, etc. The visual graph method was proposed by Lozano Perez and Wesley in 1979 [10]. It regards the obstacle as a polygon and the robot as a point, and combine and connect all connecting lines of the starting point, each vertex of the obstacle and the end point. The key to building a view is to judge the visibility between each vertex of the obstacle. It is required that the connected line segment cannot pass through the interior of the polygon, and then select the optimal route through the weighted value of the line segment. Visual graph

method can find the shortest route, but the search time is long and inflexible. It is only suitable for polygonal obstacles, and it is invalid for circular obstacles. Dijkstra algorithm is a classical breadth first state space search algorithm [11], which obtains the global optimal route by calculating the shortest distance from the starting point to any point in the environment model. This calculation method is similar to the wave surface, which diffuses outward from the starting point. Due to its traversal calculation method, it greatly increases the amount of calculation, and will produce many useless calculations, which is inefficient. Based on it, A* algorithm is proposed, and a cost function is added on the basis of breadth first. It considers not only the distance cost between the current node and the starting point, but also the distance cost between the current node and the ending point. The accuracy and speed of the algorithm are adjusted by adjusting the size of H (n). However, due to the large amount of calculation, it is still not suitable for high-dimensional space. RRT algorithm is a simple and effective random algorithm. It generates a random extended tree by randomly spreading from the root node. When the leaf node contains the target point, it can obtain a route [12] Although RRT algorithm has the advantages of fast search efficiency and can be used in high-dimensional space, its random expansion makes the result may not be optimal, and RRT algorithm is not suitable for narrow and long search environment.

Due to the increase of constraints and the complexity of environmental model, the traditional route planning algorithm has poor adaptability and difficult to solve. In order to better solve the route planning problem, intelligent optimization algorithm came into being.

Intelligent optimization algorithms mainly include: genetic algorithm, ant colony algorithm, particle swarm optimization algorithm and so on. Genetic algorithm (GA) is a heuristic search algorithm based on Darwinian evolution theory and Mendelian genetic mechanism [13]. It is essentially a parallel and global search algorithm. 50. Wang *et al.* [14] proposed a double cycle genetic algorithm (DCGA). When the environmental model and motion constraints are considered in the experiment, the route planning result of DCGA algorithm is more realistic than that of traditional genetic algorithm

Ant colony algorithm (ACO) is a random search simulated evolutionary algorithm, which solves the optimization problem by simulating ant foraging. According to the pheromone concentration left by ants in the process of foraging, positive and negative feedback is generated to the algorithm, so as to find the shortest route with the highest pheromone concentration. Ant colony algorithm has good optimization ability in two-dimensional space, but it is still easy to fall into local optimization. Many scholars have extended the improved algorithm to three-dimensional space. For example, Miao C *et al.* [15] proposed an improved adaptive ant colony algorithm (IAACO), which improves the real-time and security of ACO algorithm by introducing angle guidance factor and obstacle removal factor, At the same time, the pheromone update rules of ant colony algorithm are improved, and the route planning problem is transformed into multi-objective planning problem. Finally, the simulation experiment is carried out in the two-dimensional grid environment. Through the comparative analysis with ACO algorithm and IACO algorithm, it is verified that IAACO algorithm has faster convergence speed and stronger global optimization ability while ensuring the quality of solution.

Particle swarm optimization (PSO) is a global random evolutionary search algorithm based on the migration and clustering behavior of birds in the process of foraging [16]. The particle position is changed by transmitting its own position information, and finally the food source is found. In view of its low search accuracy and easy to fall into local optimization, many scholars have improved it and then applied it. For example, B. Zhang *et al.* uses BSO algorithm, combines PSO algorithm with bas algorithm, uses longicorn foraging to replace particle optimization process, and applies it to three-dimensional route planning. Experiments show that BSO algorithm speeds up iterative convergence and reduces the probability of falling into local optimal solution [17].

Moth-flame algorithm (MFO) was first proposed by Seyedali Mirijalili et al in 2015 [18]. Compared with the above algorithms, it has the advantages of fast convergence speed, few adjustment parameters and simple implementation, and has been widely used in recent years. Li Zhiming *et al.* [19] proposed a Moth-flame optimization algorithm improved by Lévy flight trajectory. Which improved the convergence speed and optimization accuracy; Yue Longfei *et al.* [20] proposed Tent chaos and simulated annealing improved month-flame optimization algorithm, (TCSA-MFO), which increased the diversity of understanding and improved the ability of the algorithm to jump out of local optimization to a certain extent.

Although the problem of route planning has been solved by different methods in the above documents, there are still some deficiencies in these documents. Firstly, it is reflected in the establishment of environmental model. Although two-dimensional grid map can verify the performance of the algorithm, it lacks practical guiding significance; Secondly, in the search mechanism, the random search method slows down the convergence speed of the algorithm to a certain extent; Finally, the search mechanism of the algorithm is mostly linear search, which is not conducive to increasing the diversity of solution space.

To sum up, this paper selects MFO algorithm and proposes an improved Moth-flame optimization algorithm based on chaotic random opposite learning and Cauchy mutation (OLTC-MFO), which improves it from the following three aspects:

(1) Firstly, the opposite population is generated by random opposite learning, which increases the diversity of the initial population;

(2) The flame position is disturbed by logistic tent chaotic map, which enhances the global optimization ability of the algorithm;

TABLE 1. Algorithm comparison.

Algorithm Name	Advantage	Shortcoming
GA	Strong global search ability	Weak local optimization ability
ACO	Distributed computing, easy to combine with other optimization algorithms	Complex parameter setting
PSO	The ability of global exploration for multi-modal problems is strong and easy to implement	Straight line exploration, easy premature convergence
MFO	Strong development ability and few adjustment parameters	The convergence speed is relatively slow

(3) Adding probability operator allows the algorithm to accept the current solution with a certain probability or disturb the optimal position through Cauchy mutation operator to produce a new solution, which improves the local optimization ability of the algorithm;

(4) Based on the improved algorithm, the simulation experiment is carried out on three-dimensional terrain, and compared with MFO and TCSA-MFO algorithm is compared to verify the effectiveness and feasibility of the algorithm.

The rest of the paper is organized as follows: the second section introduces the UAV route planning model. The third section introduces the improvement process of the algorithm. In the fourth section, the performance of the algorithm is compared by using the benchmark function. In the fifth section, the comparative simulation experiment analysis of UAV route planning is carried out. Finally, the sixth section gives the conclusion.

II. UAV ROUTE PLANNING MODELING

A. TERRAIN ENVIRONMENT MODELING

In UAV intelligent route planning, the construction of environment model is the basis of the problem, which affects the complexity of UAV route planning to a certain extent. In order to better simulate the real environment, this paper uses digital elevation map (DEM) to establish three-dimensional terrain, which is described as follows:

$$\psi = \{(x, y, z) \mid 0 \le x \le x_{\max}, 0 \le y \le y_{\max}, 0 \le z \le z_{\max}\} \quad (1)$$

where x, y and z represent the horizontal and vertical coordinates and heights of any point in the three-dimensional environment, and $x_{\max}, y_{\max}, z_{\max}$ represent the boundary values of the environment model.

B. THREAT MODEL CONSTRAINTS

Radar position, surface to air missile position and no-fly zone are the main threats that need to be avoided in UAV route planning. This paper mainly considers UAV intelligent route planning in static environment, which is to plan UAV route under the condition of known threat array.

1) RADAR MODELING

Radar mainly emits electromagnetic waves and reflects echoes when electromagnetic waves touch objects. Radar receives echo information and judges the relevant information of objects, so as to report the information to the surface to air missile position and attack. Therefore, in UAV route planning, the primary task is to avoid detection of radar positions. The classical radar equation describing detection characteristics of radar is as follows [14]:

$$R = \left(\frac{P_t G^2 \sigma \lambda^2}{(4\pi)^3 (S/N) F_n k T_o B_n L} \right)^{1/4} \quad (2)$$

R represents the detection range of radar, which is the key information to determine whether radar can detect the target. In this paper, two radar positions are set up in three-dimensional terrain, and their coordinates are: $(196.1km, 355.3km, 389.9m)$ and $(86.2km, 135.4km, 286.2m)$. The detection radius is $R = 30km$ and $R = 45km$. In order to ensure the safety of UAV route, this paper requires that UAV route planning should completely avoid radar positions.

2) GROUND-TO-AIR MISSILE

Surface to air missile position is mainly responsible for the main firepower for attacking the aircraft with threats after receiving the information of the radar position. In this paper, the ground missile position is simulated as a highly concealed unit hidden in the terrain, and its firepower coverage is simplified as a sphere. Two ground missile positions are arranged in a three-dimensional terrain environment, and their position coordinates are: $(393.9km, 414.1km, 349.2m)$ and $(358.9km, 146.8km, 266.5m)$. The detection radius is $R = 40km$ and $R = 50km$.

3) NO-FLY ZONE

The no-fly zone is also called prohibited flight area. Due to the existence of some controlled areas in the real environment and their special nature, they are subject to military control and aircraft are forbidden to cross. In this paper, the no-fly zone model is simplified and represented by a cylinder with coordinates of (234.2, 200), and 355.6 km in height and 35km in radius.

To sum up, the three-dimensional environment and threat model of path planning are shown in Figure 1.The spherical area represents the radar and ground-to-air missile position, and the cylindrical area represents the no-fly zone.

C. UAV MOTION CONSTRAINTS

In the process of UAV route planning, the flight path of UAV is not only affected by environmental factors, but also due to its own performance limitations that have been decided in the design and manufacturing process. In this paper, the constraint conditions of UAV are mainly constructed from four aspects: maximum range, minimum flight height, pitch Angle and turning Angle.

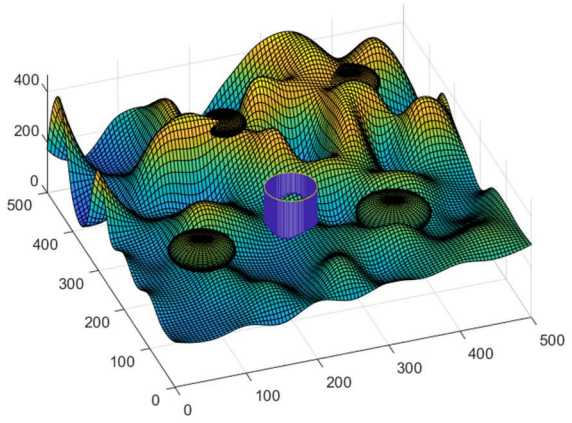


FIGURE 1. Schematic diagram of terrain environment and threat model.

1) MAXIMUM RANGE CONSTRAINT

Due to the limitation of UAV fuel tank volume, as a result, the amount of fuel consumed by UAV in one flight is limited. In order to ensure that UAV can reach the mission area and return smoothly, it is required that the amount of fuel consumed should be less than the maximum fuel load. In the route planning problem, the constraint is converted into a distance constraint, and its mathematical expression is as follows:

$$2\sum_{i=1}^{n} S_i \leq S_{\max} \tag{3}$$

The cost function is:

$$loss_{hc} = 2\sum_{i=1}^{n} S_i \tag{4}$$

Among them, $S_{\max}$ is the maximum range of UAV, and S_i is the route length between two waypoints.

2) MINIMUM FLIGHT ALTITUDE

Due to the flight characteristics and flight safety requirements of UAV, it is required that UAV should keep a certain safe distance from the ground when flying, that is, the minimum flight height in any route is greater than the minimum flight height required by UAV, which is mathematically described as: $\forall H_{i,\min} \geq H_{\min}$

The cost function is:

$$loss_{gd} = 2\sum_{i=1}^{n} H_i \tag{5}$$

Among them, $H_{\min}$ is the minimum flight altitude required by UAV and $H_{i,\min}$ is the minimum flight altitude in Section I route.

3) MAXIMUM PITCH ANGLE AND TURN ANGLE

During the flight, if the UAV dives, climbs or turns at too high an Angle, it will stall and fail to complete the mission. Assuming that the current position coordinate of the UAV is (x_i, y_i, z_i) and the position coordinate at the next moment is $(x_{i+1}, y_{i+1}, z_{i+1})$, the maximum pitch angle constraint to be satisfied by the UAV is:

$$\begin{aligned} &\forall \alpha_i \leq \alpha_{\max} \\ &\alpha_i = \arcsin \frac{\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2}}{\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2+(z_{i+1}-z_i)^2}} \end{aligned} \tag{6}$$

Among them, $\alpha_{\max}$ is the maximum pitch angle of UAV, and α_i is the dive or climb angle of UAV in Section I route.

After the route segment is projected onto the horizontal plane, the maximum turning Angle constraint to be satisfied by the UAV is (7), as shown at the bottom of the page. Among them, $\beta_{\max}$ is the maximum turning angle of UAV, β_i is the turning angle of UAV in Section I route.

D. THE OBJECTIVE FUNCTION OF UAV ROUTE PLANNING

1) OBJECTIVE FUNCTION OF ROUTE PLANNING FOR SINGLE UAV

UAV route planning is essentially an optimization problem, so the objective function can be obtained as follows:

$$\begin{aligned} \min Q &= \mu_1 \cdot loss_{hc} + \mu_2 \cdot loss_{gd} + \gamma \cdot constraint \\ constraint &= \sum_{i}^{n} \zeta_i \\ \xi_i &= \begin{cases} 0 & \text{Satisfy constraint} \\ 1 & \text{Dissatisfyconstraint} \end{cases} \end{aligned} \tag{8}$$

Among them, the loss_{hc} is the cost function of the voyage, loss_{gd} is the cost function of flight height, constraint is the penalty function of constraint condition, and μ_1 is the voyage cost and μ_2 is height cost weight coefficient. γ is the penalty function factor.

2) OBJECTIVE FUNCTION OF COOPERATIVE ROUTE PLANNING FOR MULTI-UAVs

In order to adapt to the future war style and realize multi-UAVs cooperative operations, this paper establishes a multi-UAVs intelligent route planning model. The model is divided

$$\begin{aligned} &\forall \beta_i \leq \beta_{\max} \\ &\beta_i = \arccos \frac{\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2}+\sqrt{(x_{i+2}-x_{i+1})^2+(y_{i+2}-y_{i+1})^2}-\sqrt{(x_{i+2}-x_i)^2+(y_{i+2}-y_i)^2}}{2\sqrt{(x_{i+2}-x_{i+1})^2+(y_{i+2}-y_{i+1})^2}\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2}} \end{aligned} \tag{7}$$

into two parts, one is to plan the route of each UAV at the single UAV planning level; second, consider the cooperative relationship of multi-UAVs at the multi-UAVs planning level. The multi-UAVs route planning framework is shown in Figure 2:

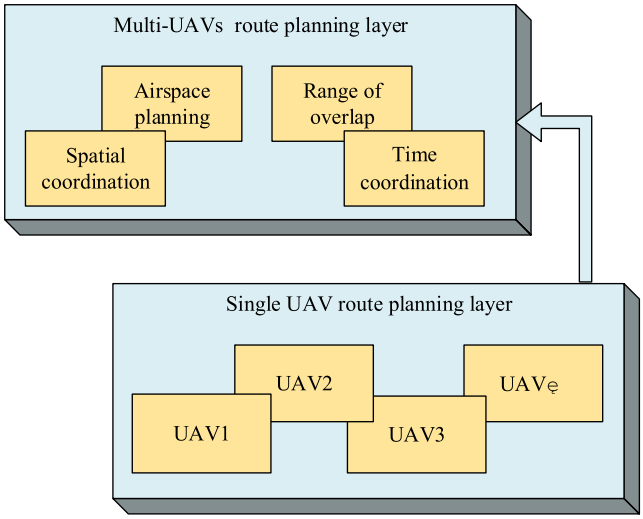


FIGURE 2. Route planning framework of multi-UAVs.

Multi-UAVs cooperative relationship is usually divided into two types, one is spatial cooperation; the other is Time cooperation. Spatial cooperation can divide the airspace by the air traffic control department, and adjust the flying heights of different UAVs in the airspace, so as to avoid the collision of UAVs and successfully complete the task, as shown in Figure 3. Time cooperation can adjust the flight speed of UAVs, calculate the fastest and slowest time for each UAV to arrive at the mission place, and get the time interval for each UAV to execute the mission. Through interval intersection, the flight speed of each UAV can be determined or its departure time can be adjusted, so as to successfully complete the mission, as shown in Figure 4.

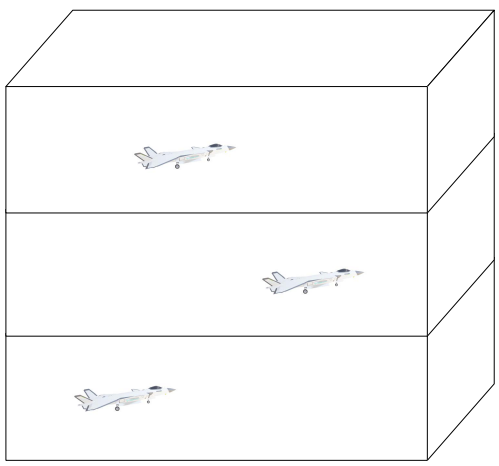


FIGURE 3. Schematic diagram of spatial cooperation planning.

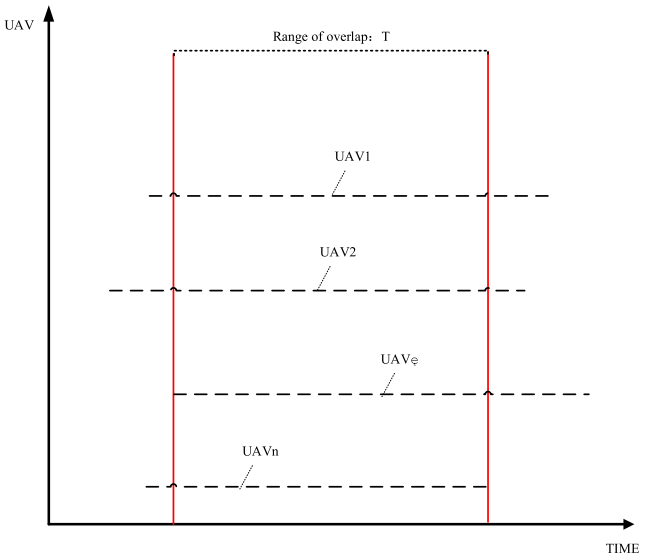


FIGURE 4. Schematic diagram of time cooperation planning.

The expression of UAV interval intersection is as follows:

$$T_{time,i} = \left(\frac{S_i}{V_{\max}}, \frac{S_i}{V_{\min}} \right) \tag{9}$$

S_i represents the route length of the i-th UAV, and $V_{\max}$ represents the maximum speed and $V_{\min}$ is the minimum speed of the UAV.

Because the flight speed of UAV is not restricted in this paper. Based on the route planning of single UAV, the time cooperation of multi-UAVs is emphatically considered, and the objective function of time cooperation route planning of many UAVs is as follows:

$$\min Q_{xt} = \sum_{i=1}^{number} Q_i + T_{cost} + \gamma \cdot constraint \tag{10}$$

$\sum_{i=1}^{number} Q_i$ is the cost of selected route, and T_{cost} is the cost of time cooperation of multi-UAVs. In order to enable multi-UAVs to complete tasks quickly, the intersection of time interval is taken as the surrogate value, that is

$$T_{cost} = \left(\frac{S_i}{V_{\max}}, \frac{S_i}{V_{\min}} \right)$$

III. OLTC-MFO ALGORITHM

A. MOTHING-FLAME OPTIMIZATION ALGORITHM

Mothing-flame Optimization Algorithm (MFO) is a new intelligent optimization algorithm proposed by Seyedali Mirijalili and others in 2015. Its inspiration comes from the lateral positioning of moths flying at night [18], that is, moths always keep a certain included angle with the moon when flying at night, so as to keep their straight flight path.

However, in real life, moths are always deceived by artificial light sources, and keeping a similar angle makes moths show a spiral flight path to the light source. Mirijalili

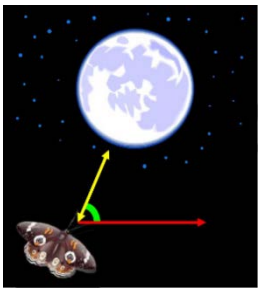


FIGURE 5. Transverse positioning of moths.

has taken advantage of this phenomenon to create the ''Mothing-flame Optimization Algorithm''.

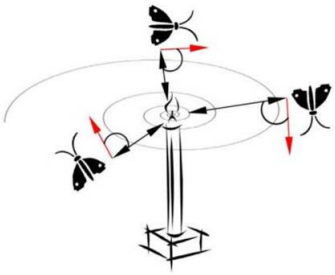


FIGURE 6. Moth to flame.

1) POPULATION INITIALIZATION OF MOTHS

In MFO algorithm, moth location is the candidate solution of the problem to be solved, and the variable to be solved is the position of moth in space, which is essentially a population intelligent optimization algorithm. Therefore, the initial population of moth is defined as matrix M, which is expressed as follows:

$$M = \begin{bmatrix} m_{11} & m_{12} & \cdots & m_{1d} \\ m_{21} & m_{22} & \cdots & m_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ m_{n1} & m_{n2} & \cdots & m_{nd} \end{bmatrix}_{n\times d} \tag{11}$$

n represents the number of moths, and d represents the characteristic dimension of moths, and d represents the number of set path points in solving the path planning problem in this paper. At the same time, M_i is defined as the i-th moth in moth population M, and its fitness matrix is M_F, which is expressed as follows:

$$M_F = \begin{bmatrix} M_{f1} \\ M_{f2} \\ \vdots \\ M_{fn} \end{bmatrix}_{1\times n} \tag{12}$$

M_{fi} is the fitness of the i-th moth, which can be obtained by solving the corresponding cost function. The initial flame set has the same size as the moth population set, but gradually decreases with subsequent iterations. The flame set is the matrix F, which is expressed as follows:

$$F = \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1d} \\ f_{21} & f_{22} & \cdots & f_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ f_{n1} & f_{n2} & \cdots & f_{nd} \end{bmatrix}_{n\times d} \tag{13}$$

The i-th flame is represented by F_i, and the corresponding flame fitness is the same as the moth fitness, which is the matrix F_F, as shown below:

$$F_F = \begin{bmatrix} F_{f1} \\ F_{f2} \\ \vdots \\ F_{fn} \end{bmatrix}_{1\times n} \tag{14}$$

F_{fi} is the fitness of the i-th flame, which is obtained by ordering the fitness of moth, that is, the matrix F_F is the result of matrix M_F ascending ordering, which also indicates that flame F is the optimal solution of moth M in the current iterative search.

2) ITERATIVE UPDATE

The updating mechanism of each iteration of MFO algorithm is mainly divided into two parts, one is the updating mechanism of moth position, the other is the updating mechanism of flame position.

(1) Updating mechanism of moth position: MFO algorithm adopts the flight principle of moth under artificial light source. According to the phototaxis of moths, the flight mode of moths is fitted with logarithmic spiral curve, and the moth M_i moves in logarithmic spiral around the corresponding flame F_i. The updated mode of the position of moths is defined as follows:

$$S\left(M_i, F_j\right) = D_i \cdot e^{bt} \cdot \cos\left(2\pi t\right) + F_j \tag{15}$$

$S\left(M_i, F_j\right)$ is the updated position of moth M_i, and is a constant parameter, and its value determines the shape of logarithmic isogonal spiral, usually taking 1; Parameter T is a random number within the range, and its value controls the distance between the moth and the flame. The smaller the value of t is, the closer the moth is to the flame. By changing the value of parameter T, the position where moth finally arrives around flame is adjusted, which reflects the local optimization ability of the algorithm; D_i is the euclidean distance between the j-th flame and the i-th moth: $D_i = \left|M_i - F_j\right|$, and the trajectory of the moth is shown in Figure 7.

(2) Updating mechanism of flame position: In MFO algorithm, flame represents the optimal solution obtained by moths in the current iteration, and each moth is required to update its own position only by using a unique flame. However, if n moths update their positions according to N different positions in the search space each time, the local optimization capability of the algorithm will be reduced. Therefore, an adaptive updating mechanism of flame is proposed in the MFO algorithm, that is, the number of flame

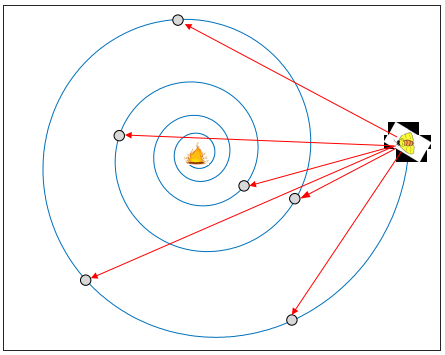


FIGURE 7. Schematic diagram of moth movement track.

decreases with the increase of iterations, so as to balance global search and local optimization. The adaptive updating mechanism of flame is as follows:

$$flameno = round\left(N - iteration \times \frac{N-1}{\max\_iteration}\right) \quad (16)$$

where N represents the maximum number of flames, iteration represents the current iteration number, and max_iteration represents the maximum iteration number of the algorithm. The number of flames changes with the number of iterations of the algorithm, as shown in Figure 8:

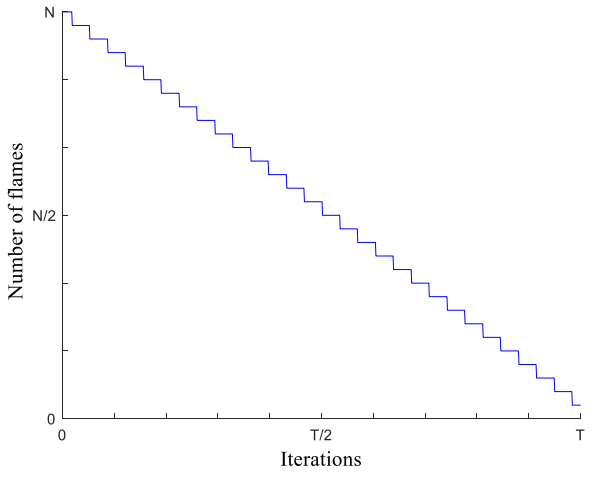


FIGURE 8. Adaptive updating of flame number.

B. 2.2 UAV ROUTE PLANNING BASED ON IMPROVED MFO ALGORITHM

1) ROUTE ENCODING

In practical application, each route is a line formed by connecting several points obtained by searching. In the moth population M, any moth will form a flight path: $M_i = [m_{i1}, m_{i2}, \cdots, m_{id}]$, and each point in the flight path has three-dimensional attributes: (x, y, z). Therefore, when MFO algorithm is used to solve the UAV intelligent route planning problem, the number of waypoints D should be determined first, and the coordinates of waypoints in the equidistant X direction are fixed and unchanged, and the optimal path should be determined by searching the Y and Z coordinates.

2) RANDOM OPPOSITION-BASED LEARNING

In 2005, Hamid R. Izhoosh first introduced opposition-based learning (OBL) as a new intelligent computing scheme [21] As is known to all, generating a random population in an optimization algorithm will affect the optimality of search results, and often result in the algorithm cannot converge to the desired solution. An OBL strategy is to consider a solution to a problem while thinking about its opposite solution. Experiments show that the probability of the opposite candidate solution reaching the global optimum is higher than that of the random solution. Therefore, in the MFO algorithm, in order to enhance the population diversity and improve the global search ability of the algorithm, after the random generation of moth population M, the opposite population was introduced, denoted as matrix OM. Its mathematical description is as follows:

$$\tilde{x} = lb + ub - x \quad (17)$$

In which, $x \in [lb, ub]$, and $\tilde{x}$ is the opposite solution of x. At the same time, considering that the upper and lower limits of some functions are negative to each other in the performance test of the algorithm, the distance between the opposite moth and the moth is fixed, which cannot fully enhance the diversity of the population. So, the random opposition-based learning strategy is adopted, and its mathematical description is as follows:

$$\tilde{x} = lb + ub - rand \cdot x \quad (18)$$

And rand is the random number between (0, 1).

3) LOGISTIC-TENT CHAOS MAPPING

In the field of optimization, chaotic mapping is widely used in intelligent optimization algorithms because of its ergodicity, randomness and nonlinearity [22]. Logistic chaos map and Tent chaos map are two kinds of chaos map commonly used at present [23]. However, Logistic chaos map has strong global ergodicity and poor performance in local search, while Tent chaos has strong disturbance ability, which can make up for this deficiency [24], [25] Therefore, Logistic-Tent chaos mapping is proposed to improve MFO algorithm. When updating the flame position after an iteration, if the position with the smallest fitness value is selected as the optimal flame every time, it is not conducive to global search. At this time, when updating the flame position, a random disturbance is introduced. When the disturbance value is less than 0.2, Logistic-Tent chaos mapping is used to map the optimal flame position in chaos, so as to avoid premature algorithm and fall into local optimum. The Logistic-Tent chaos mapping is as follows:

$$x_{n+1} = \begin{cases} \left[rx_n(1-x_n) + \dfrac{(4-r)}{2}x_n\right] \bmod 1 & if\ x_n < 0.5 \\ \left[rx_n(1-x_n) + \dfrac{(4-r)(1-x_n)}{2}\right] \bmod 1 & if\ x_n \geq 0.5 \end{cases} \quad (19)$$

Logistic-Tent chaotic mapping distribution is shown in Figure. 9:

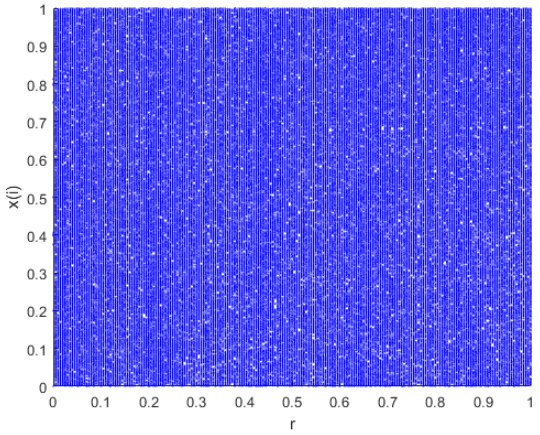


FIGURE 9. Logistic-tent chaos map distribution.

The pseudo-code of the optimal flame position chaotic mapping is as follows:

TABLE 2. Chaotic mapping pseudocode.

Algorithm 1. The pseudocode of the optimal flame position chaotic mapping
1. Generate a random disturbance
2. Assign the flame position and score
3. Cycle to compare moth positions
4. ***If****<constant*
5. Update the optimal position of flame according to Equation (20)
6. ***else***
7. The location updates normally
8. ***end***

4) CAUCHY MUTATION OPERATOR

In the MFO algorithm, moths change their positions by logarithmic isogonal spiral flight. Due to its curve characteristics, although MFO algorithm has a fast convergence speed, it is easy to fall into local optimal. When dealing with the route planning in complex environment, the problem is more serious, which makes the route appear large deviation. Therefore, Cauchy operator is introduced to perturb the waypoints. At the same time, the selection probability operator is introduced to allow the algorithm to accept the current bad solution with a certain probability, so that the algorithm can jump out of the local optimal solution and find the global optimal solution.

Suppose that one of the flames: $F = [f_1, f_2, \cdots, f_d]$, each dimension judges whether there is variation according to the selection probability. Suppose that the i-th dimension produces Cauchy variation, and the expression is as follows:

$$X'_{best} = X_{best} + X_{best} \times cauchy\,(0, 1) \tag{20}$$

The generating function of cauchy distribution random variable is:

$$cauchy = \tan\left[\left(\xi - \frac{1}{2}\right)\pi\right] \tag{21}$$

The expression of selection probability operator is as follows:

$$\begin{aligned} Q &= fobj\left(X'_{best}\right) \\ p &= Q - Best\_flane\_score \end{aligned} \tag{22}$$

$fobj\,(x)$ is the fitness function calculation formula. When P value is less than 0, cauchy variation is used to make the algorithm jump out of the local optimal and obtain the ability to continue searching for the global optimal solution.

C. OLTC-MFO ALGORITHM FLOW

TABLE 3. OLTC-MFO algorithm flow.

Algorithm 2 The pseudocode of OLTC-MFO
1. Initializing terrain and threat model
2. Initializing of parameters
3. Initializing the moth population and flame matrix.
4. Initializing the opposite population according to Equation (19)
5. ***while Iteration<Max_iteration+1***
6. Calculate the fitness of moth population and flame matrix
7. Sort the fitness matrix of moths
8. Assign to the flame fitness matrix
9. Updating the flame adaptively according to Equation (17)
10. Generate a random disturbance
11. Assign the flame position and score
12. Cycle to compare moth positions
13. ***If<constant***
14. Update the optimal position of flame according to Equation (20)
15. ***else***
16. The location updates normally
17. ***end if***
18. Generate probability operator according to Equation (22)
19. ***if*** *probability operator<0*
20. Generate cauchy mutation operator Equation (21)
21. ***else***
22. Accept the current solution
23. ***end if***
24. Updating moths location according to Equation (16)
25. ***end while***
26. Return the optimal path and minimum generation value

IV. COMPARISON EXPERIMENT OF OLTC-MFO ALGORITHM

A. TEST FUNCTIONS

UAV route planning is ultimately an optimization problem. Due to the complex environmental model, changeable terrain and many constraints, UAV may fall into local optimum or fall into terrain environment and cannot find a way out when searching for optimization. Therefore, it is particularly important to test the performance of UAV route planning algorithm to ensure that it can effectively cope with the complex environment and find the global optimal solution.

TABLE 4. Benchmark functions.

Function	Expression	Global minimum	Features
Schwefel 2.21	$f(X)=\max\{\lvert x_i\rvert, 1\le i\le 30\}, x_i\in[-100,100]$	$x_i=0, f(X)=0$	Unimodal
Rosenbrock	$f(X)=\sum_{i=1}^{D-1}(100(x_{i+1}-x_i^2)^2+(x_i-1)^2), x_i\in[-30,30]$	$x_i=1, f(X)=0$	Unimodal
Rastrgin	$f(X)=\sum_{i=1}^{D}(100(x_i^2-10\cos(2\pi x_i)+10)), x_i\in[-5.12,5.12]$	$x_i=0, f(X)=0$	Multimodal
Griewank	$f(X)=\frac{1}{1000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\frac{x_i}{\sqrt{i}}+1, x_i\in[-600,600]$	$x_i=0, f(X)=0$	Multimodal
Penalized_2	$f(X)=0.1\left\{\begin{array}{l}\sin^2(3\pi x_1)+\sum_{i=1}^{29}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+\\ (x_n-1)^2[1+\sin^2(2\pi x_{30})]\end{array}\right\}+\sum_{i=1}^{30}\mu(x_i,5,100,4),\ x_i\in[-50,50]$	$x_i=1, f(X)=0$	Multimodal

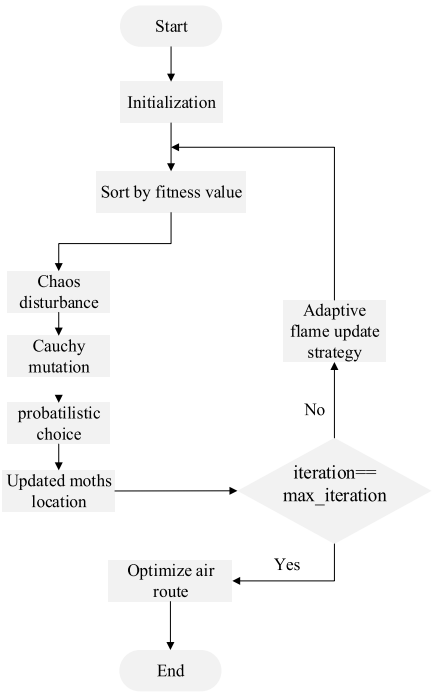


FIGURE 10. Flow chart of OLTC-MFO algorithm.

In this paper, five classical benchmark functions in CEC2010 were selected to test the performance of the algorithm [29], among which Schwefel 2.21 and Rosenbrock functions were single-peak functions, which were mainly used to test the convergence speed and accuracy of the algorithm. Rastrgin, Griewank and Penalized_2 functions are complex multi-peak functions with multiple local optima, which are mainly used to test whether the algorithm can effectively escape from local optima and then search for global optimal values. The expression and basic information of the algorithm are shown in Table 3:

B. EXPERIMENTAL VERIFICATION

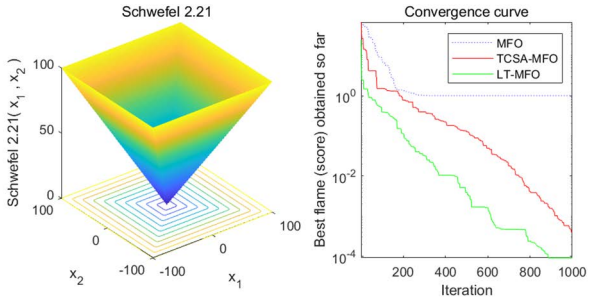
The above five classical benchmark functions were selected to conduct comparative experiments on MFO algorithm, TCSA-MFO algorithm [16] and OLTC-MFO algorithm to test the performance of the algorithm in this paper. Figure 11 shows the graphs of each test function and the fitness convergence curves of each algorithm on different test functions. Table 5 lists the optimal value and convergence algebra of each algorithm as well as the average value in 20 tests. The initial parameter Settings of the algorithm are shown in Table 4:

TABLE 5. Algorithm parameter settings.

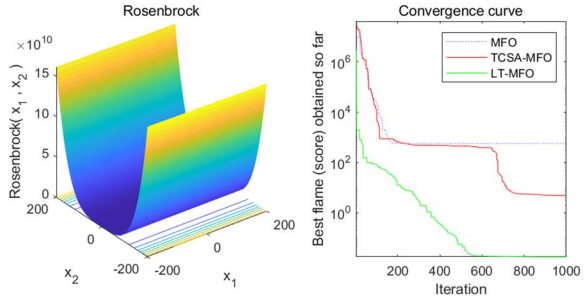
The parameter name	Numerical value
Population	30
iterations	1000
Logarithmic spiral parameters	1

The five classical test function diagrams and convergence curves are as follows in Fig. 11.

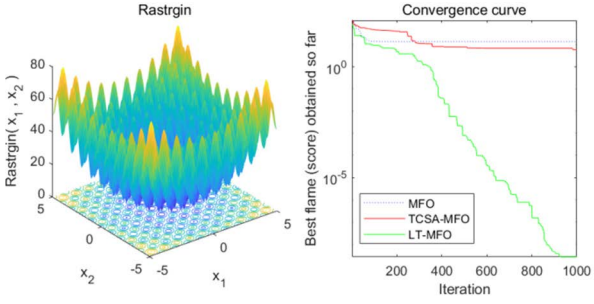
It can be seen from Figure. 11 that OLTC-MFO algorithm is superior to MFO algorithm and TCSA-MFO algorithm in both convergence speed and accuracy in the benchmark function in 5. Especially in the Rastrgin test function experiment, it can be seen that the performance of OLTC-MFO algorithm is significantly better than the other two algorithms, which proves that the OLTC-MFO algorithm proposed in this paper has strong performance in convergence speed and accuracy, and has the ability to jump out of the local



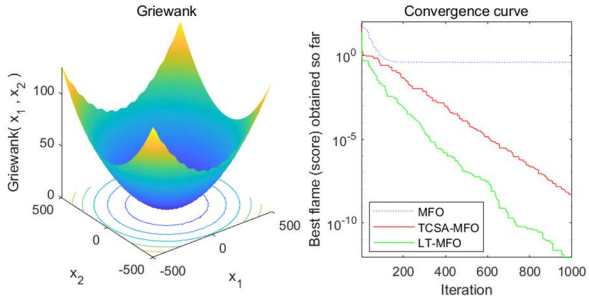
(a) Schwefel 2.21 function



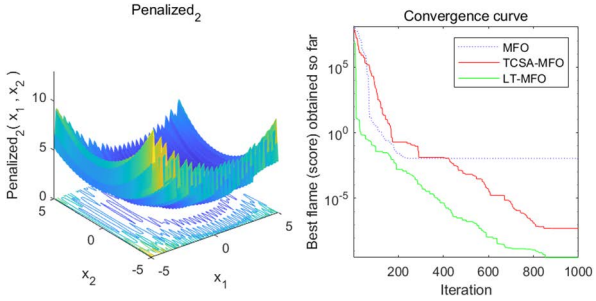
(b) Rosenbrock function



(c) Rastrgin function



(d) Griewank function



(e) Penalized_2 function

FIGURE 11. **Fitness convergence curves of each algorithm on different test functions.**

TABLE 6. **Comparison of the results of the three algorithms on the test functions.**

Test functions	Algorithm	The optimal value	Iterations	Mean value
Schwefel 2.21	MFO	1.0045	329	1.6656
	TCSA-MFO	6.4990×10^{-4}	962	7.5983×10^{-5}
	OLTC-MFO	9.7070×10^{-5}	886	9.8956×10^{-5}
Rosenbrock	MFO	555.1400	207	5.5364
	TCSA-MFO	4.7114	890	1.2653
	OLTC-MFO	0.171	588	0.1532
Rastrgin	MFO	13.9294	128	5.6934
	TCSA-MFO	6.1313	514	1.4253
	OLTC-MFO	2.7401×10^{-9}	923	3.6956×10^{-9}
Griewank	MFO	0.3983	186	0.4726
	TCSA-MFO	6.489×10^{-9}		3.600×10^{-10}
	OLTC-MFO	1.132×10^{-12}	983	3.600×10^{-10}
Penalized_2	MFO	0.0110	293	0.0253
	TCSA-MFO	5.985×10^{-8}	877	3.600×10^{-10}
	OLTC-MFO	3.600×10^{-10}	883	3.600×10^{-10}

TABLE 7. Algorithm parameter settings.

The parameter name	number
Population	30
Iterations	1000
Logarithmic spiral parameters	1
Start point	(15.15,30.3,180.9)
Threat 1 coordinates	(196.1,355.3,389.9)
Threat 2 coordinates	(393.9,414.1,349.2)
Threat 3 coordinates	(86.2,135.4,286.2)
Threat 4 coordinates	(358.9,146.8,266.5)
Threat 5 coordinates	(234.2,200)
End point coordinates	(449.5,459.6,482.0)
Route points	40

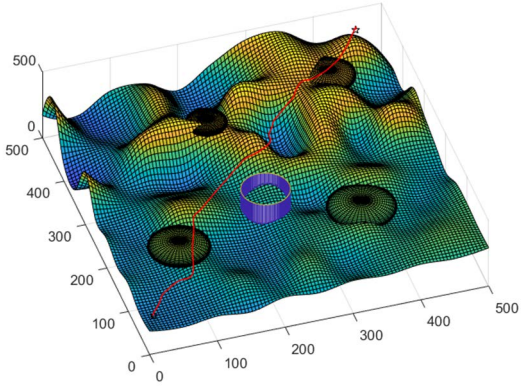
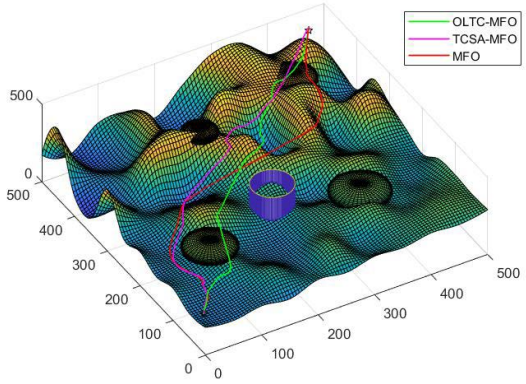
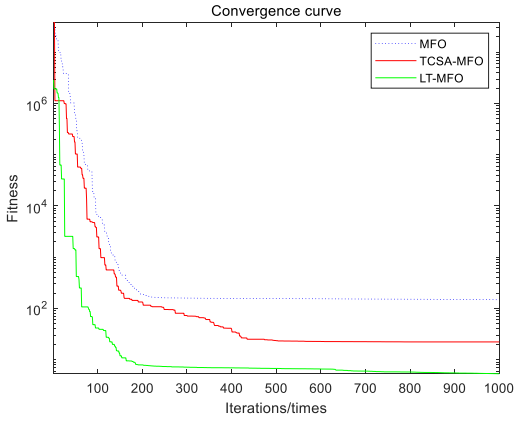


FIGURE 12. Single UAV route planning route.



(a) Route planning path diagram of each algorithm



(b) Fitness convergence curves of each algorithm

FIGURE 13. Route planning results and route cost convergence curve.

optimal value, and can better search for the global optimal value.

V. SIMULATION ANALYSIS OF UAV AUTONOMOUS ROUTE PLANNING

The CPU of the simulation computer is Intel Core I7-9750H CPU 2.60ghz, the operating system is Windows 10, and the simulation software is MATLAB R2020a. The terrain area is set as $500km \times 500km$, and the height limit is 500km. The algorithm parameters and model initialization parameters are set as shown in Table 6:

A. ROUTE PLANNING OF SINGLE UAV

Firstly, OLTC-MFO algorithm is used to carry out route planning for a single UAV on the simulated terrain. The results are shown in Figure 12:

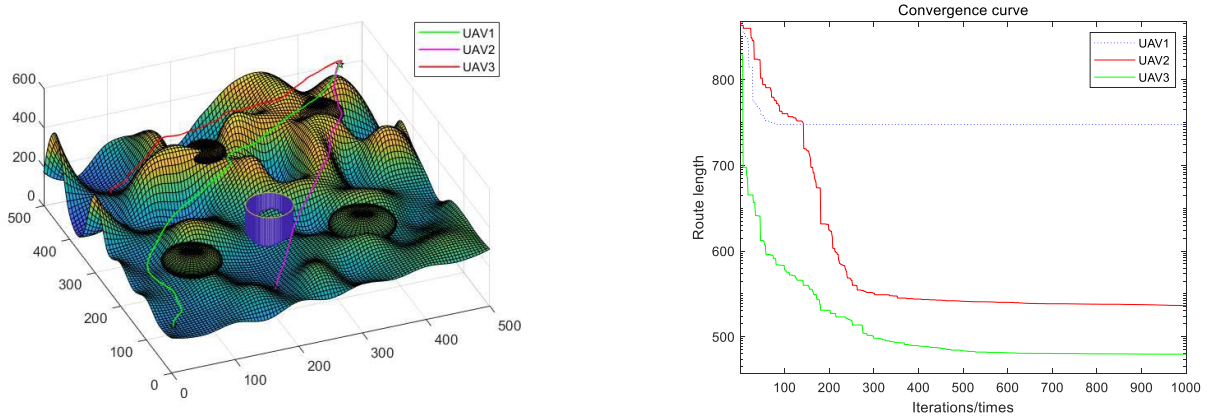
Then, MFO algorithm and TCSA-MFO algorithm are used to plan the UAV route on the simulated terrain, Compared with the experimental results of OLTC-MFO algorithm, the experimental results show that OLTC-MFO algorithm is superior to the other two algorithms in convergence speed and optimization ability, and the route range found is shorter and the route cost is obviously lower than the other two algorithms as a whole, which can ensure the UAV to complete the task safely and quickly with less cost. The route planning results and route cost convergence curve are shown in Fig 13.

TABLE 8. Single UAV route planning results of each algorithm.

Algorithm name	Length /km	Optimal fitness	Convergence times
MFO	783.6566	155.5	522
TCSA-MFO	760.1211	22.32	747
OLTC-MFO	738.2885	5.463	910

B. ROUTE PLANNING OF MULTI-UAVs

In order to meet the actual combat requirements, give full play to the advantages of multi-UAV cooperative combat, and prove the feasibility of the algorithm in route planning of multi-UAV, the simulation verification of route planning of multi-UAV is realized to some extent with three UAVs as the representative. UAV1 and UAV2 represent the reconnaissance and combat integrated UAV, and the goal is to penetrate and destroy the enemy's high-value targets at the mission site; UAV3 stands for anti-radiation UAV, which is mainly used to attack enemy radar and air defense positions at target sites after penetration. The starting points of the three UAVs are: (15.15, 30.3, 180.9), (175.5, 30.3, 260.9), and (49.48, 358.9,272.4). The route planning route of multi-UAVs is shown in Figure 14.



(a) Route planning for multi-UAV routes

(b) Route range convergence curve

FIGURE 14. Route planning results and convergence curves of multi-UAVs.

TABLE 9. Results of time coordinated route planning for multi-UAVs.

Number	Speed range /km/h	Route length /km	Time interval /T	Range of overlap
UAV1	(400,860).	760.1211	(0.884,1.900)	
UAV2	(400,740).	564.7756	(0.763,1.412)	(1.062, 1.412)
UAV3	(280,460).	488.7411	(1.062,1.746)	

The results of time cooperation planning of multi-UAVs are shown in Table 8.

VI. CONCLUSION

In this paper, a route scale model based on OLTC-MFO algorithm is proposed for multi-UAV route intelligent route planning. Firstly, a three-dimensional terrain model and a threat model are established, and the performance of UAV is constrained and controlled to determine the cost function of UAV route planning. At the same time, the cooperative route planning model of multiple UAVs was established to provide technical support for future practical application. Secondly, MFO algorithm is improved to solve the problem of slow convergence and easy to fall into local optimum in the search process:

(1) According to the principle of reverse learning, the reverse population is generated, which makes the initial search of the algorithm more purposeful and accelerates the convergence speed of the algorithm;

(2) The Logistic Tent chaos principle is used to conduct chaos disturbance to the flame position, which enhances the global search ability of the algorithm;

(3) Cauchy mutation operator and probability operator are introduced to perturb the optimal flame position, so that the algorithm can not only accept the current solution with a certain probability, but also jump out with a certain probability, which enhances the local search ability of the algorithm and avoids the algorithm falling into local optimal.

Finally, experiments prove that OLTC-MFO algorithm has better convergence speed and accuracy than MFO algorithm and TCSA-MFO algorithm in algorithm performance and route planning application process, and can also better complete tasks in route planning of multiple UAVs. In the next research the proposed algorithm will be integrated into the UAV mission planning software, so as to achieve engineering application.

REFERENCES

[1] R. Stevens and F. Sadjadi, "Small unmanned aerial vehicle real-time intelligence, surveillance and reconnaissance (ISR) using onboard pre-processing," *Proc. SPIE*, vol. 6967, pp. 400–407, May 2008.
[2] Y. Liu, Z. Luo, Z. Liu, J. Shi, and G. Cheng, "Cooperative routing problem for ground vehicle and unmanned aerial vehicle: The application on intelligence, surveillance, and reconnaissance missions," *IEEE Access*, vol. 7, pp. 63504–63518, 2019.
[3] Z. Zhao, Y. Niu, and L. Shen, "Adaptive level of autonomy for human-UAVs collaborative surveillance using situated fuzzy cognitive maps," *Chin. J. Aeronaut.*, vol. 33, no. 11, pp. 2835–2850, Nov. 2020.
[4] V. Shaferman and T. Shima, "Unmanned aerial vehicles cooperative tracking of moving ground target in urban environments," *J. Guid., Control, Dyn.*, vol. 31, no. 5, pp. 1360–1371, Sep. 2008.
[5] M. Suresh and D. Ghose, "UAV grouping and coordination tactics for ground attack missions," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 1, pp. 673–692, Jan. 2012.
[6] R. Larson, M. Pachter, and M. Mears, "Path planning by unmanned air vehicles for engaging an integrated radar network," in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, Aug. 2005, p. 6191.
[7] Y. Duan, X. Ji, M. Li, and Y. Li, "Route planning method design for UAV under radar ECM scenario," in *Proc. 12th Int. Conf. Signal Process. (ICSP)*, Oct. 2014, pp. 108–114.
[8] M. Darrah, W. Niland, B. Stolarik, and L. Walp, "UAV cooperative task assignments for a SEAD mission using genetic algorithms," in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, Aug. 2006, p. 6456.
[9] M. Haque, M. Egerstedt, and A. Rahmani, "Multilevel coalition formation strategy for suppression of enemy air defenses missions," *J. Aerosp. Inf. Syst.*, vol. 10, no. 6, pp. 287–296, Jun. 2013.
[10] T. Lozano-Pérez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Commun. ACM*, vol. 22, no. 10, pp. 560–570, 1979.
[11] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Math.*, vol. 1, no. 1, pp. 269–271, 1959.
[12] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report*, 1999.

[13] J. H. Holland, "Adaptation in natural and artificial systems," *Ann Arbor*, 1975, vol. 6, no 2, pp. 126–137.

[14] L. Wang, Z. Zhang, Q. Zhu, and S. Ma, "Ship route planning based on double-cycling genetic algorithm considering ship maneuverability constraint," *IEEE Access*, vol. 8, pp. 190746–190759, 2020, doi: 10.1109/ACCESS.2020.3031739.

[15] C. Miao, G. Chen, C. Yan, and Y. Wu, "Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm," *Comput. Ind. Eng.*, vol. 156, Jun. 2021, Art. no. 107230.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, 1995, pp. 16–2.

[17] B. Zhang, Y. Duan, Y. Zhang, and Y. Wang, "Particle swarm optimization algorithm based on beetle antennae search algorithm to solve path planning problem," *Proc. IEEE 4th Inf. Technol., Netw., Electron. Automat. Control Conf. (ITNEC)*, vol. 1, Jul. 2020, pp. 1586–1589, doi: 10.1109/ITNEC48623.2020.9085035.

[18] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *J. Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[19] L. Zhiming and M. Yuanbin, "Optimization algorithm based on Levy flight puhuo to J," *Comput. Eng. Des.*, vol. 38, no. 3, 2017, Art. no. 807813.

[20] Y. Longfei, Y. Rennong, Z. Yijie, Y. Yang, and Z. Zhenxing, "Tent chaos and simulated annealing improved moth fire suppression optimization algorithm," *J. Harbin Inst. Technol.*, vol. 51, no. 5, p. 49, 2019.

[21] H. R. Tizhoosh, "Opposition-based learning: A new scheme for machine intelligence," in *Proc. Int. Conf. Comput. Intell. Modelling, Control Automat., Int. Conf. Intell. Agents, Web Technol. Internet Commerce*, vol. 1, 2005, pp. 695–701.

[22] E. V. Altay and B. Alatas, "Bird swarm algorithms with chaotic mapping," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 1373–1414, 2019.

[23] H. Bingol and B. Alatas, "Chaotic league championship algorithms," *Arabian J. Sci. Eng.*, vol. 41, no. 12, pp. 5123–5147, Dec. 2016.

[24] B. Alatas, *Chaotic Bee Colony Algorithms for Global Numerical Optimization*. New York, NY, USA: Pergamon, 2010.

[25] B. Alatas, "Chaotic harmony search algorithms," *Appl. Math. Comput.*, vol. 216, no. 9, pp. 2687–2699, 2010.

[26] A. Mandalika, O. Salzman, and S. Srinivasa, "Lazy receding horizon A* for efficient path planning in graphs with expensive-to-evaluate edges," in *Proc. 28th Int. Conf. Automated Planning Scheduling*, vol. 28, 2018, pp. 476–484.

[27] G. Ying, C. Xu, S. Zhou, and S. Guo, "Planning Based on Imorived Ant Colony Algorithm Multiple Batches Collaborative Three-Dimensional Track," *J. Northwestern Polytechnical Univ.*, vol. 34, no. 1, pp. 41–46, 2016.

[28] Z. Mingyou and W. Xuegang, *Radar System*. Beijing, China: Publishing House of Electronics Industry, 2011.

[29] T. Ke, X. Yáo, P. N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen, and Z. Yang, "Benchmark functions for the CEC'2010 special session and competition on large-scale global optimization," *Nature Inspired Comput. Appl. Lab.*, 2009.

[30] H. Wang, Z. Wu, S. Rahnamayan, Y. Liu, and M. Ventresca, "Enhancing particle swarm optimization using generalized opposition-based learning," *Inf. Sci.*, vol. 181, no. 20, pp. 4699–4714, 2011, doi: 10.1016/j.ins.2011.03.016.

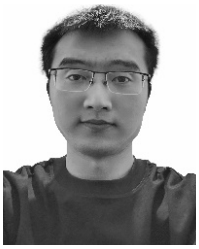
[31] Y. Duan, X. Ji, M. Li, and Y. Li, "Route planning method design for UAV under radar ECM scenario," in *Proc. 12th Int. Conf. Signal Process. (ICSP)*, Oct. 2014, pp. 108–114.

[32] M. Darrah, W. Niland, B. Stolarik, and L. Walp, "UAV cooperative task assignments for a SEAD mission using genetic algorithms," in *Proc. AIAA Guid., Navigat., Control Conf. Exhib.*, Aug. 2006, p. 6456.

MINGXI MA was born in Baotou, Inner Mongolia in 1998. He is currently pursuing the master's degree in management science and engineering with Air Force Engineering University. His research interests include operation planning and mission planning, and the application of intelligent algorithms.

JUN WU was born in Hunan, China. He is currently a Professor and a Master's Tutor with Air Force Engineering University. His current research interests include dynamic planning, expert systems, and autonomous planning.

YUE SHI was born in Changchun, Jilin, in 1968. She is currently a Professor and a Master's Tutor with Air Force Engineering University. Her major research interest includes management science and engineering.



LONGFEI YUE was born in Baoji, Shaanxi, in 1995. He received the M.S. degree from Air Force Engineering University. He is currently pursuing the Ph.D. degree in research of management science and engineering. His research interests include deep reinforcement learning, intelligent air combat, and maneuver decision.

CHENG YANG was born in Mianyang, Sichuan, in 1982. He graduated in aviation communication and navigation from Air Force Aviation University, in 2006.

XUYI CHEN was born in Beijing, in 1989. She received the master's degree in military command from Air Force Engineering University. Her current research interests include battle control and navigation planning.