

Multi-UAV Cooperative Power Inspection Base on Improved PSO Algorithm Obstacle Avoidance Path Planning

Changsheng Zhu, Minrui Zhao, Tianyu Li, Jingjie Li, Yafeng Zhao, Suchao Wang

Abstract—This study addresses the issues of poor obstacle avoidance and low efficiency in unmanned aerial vehicle (UAV) swarms during power inspections in complex areas. We propose a multi-UAV cooperative path planning method based on an improved Particle Swarm Optimization (PSO) algorithm. This method integrates spatial and temporal synergy constraints to ensure effective cooperative among UAVs in challenging terrains. To enhance path planning efficiency, we introduce a smoothing algorithm. This algorithm reduces fluctuations during flight. It decreases turning and pitch angles. As a result, the frequency and magnitude of attitude adjustments are minimized, improving flight stability and safety. To overcome the local optimal issue of the traditional PSO algorithm, we present a hybrid swarm intelligence algorithm. This combines PSO, Artificial Bee Colony (ABC), and Simulated Annealing (SA) algorithms. This approach enhances the UAV's obstacle avoidance capabilities and their ability to find globally optimal paths in complex environments. Experimental results show that the PSO-ABC-SA algorithm, compared to existing algorithms, reduces the optimal path length by 251 km in the same complex terrain. The obstacle avoidance rate reaches 94.2%, with a 5.7% improvement in average avoidance. The algorithm also outperforms others in optimal, worst, and average cost metrics. It effectively addresses the challenges of multi-UAV collaboration and obstacle avoidance path planning.

Index Terms—Particle swarm optimization, Obstacle avoidance algorithm, Complex terrain, Multi-UAV coordination, UAV, Power inspection, Mult-domain

I. INTRODUCTION

WITH the rapid development of power grid infrastructure, the total length of transmission lines continues to increase. This poses higher demands for inspection, maintenance, and upkeep of the power grid. The

Manuscript received January 7, 2025; revised May 21, 2025.

This paper was supported by Cooperation project between Lanzhou University of Technology and Longnan Power Supply Company of State Grid Gansu Electric Power Company(Grant Nos.HX2023C50800001) and Lanzhou University of Technology Higher Education Research Project(Grant Nos.202102001).

Changsheng Zhu is a professor at the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China.(corresponding author to provide e-mail: zhucs_2008@163.com)

Minrui Zhao is a graduate student at the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China.(e-mail: 2904104078@qq.com)

Tianyu Li is a doctoral student at the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China.(e-mail: li_ty1482916370@163.com)

Jingjie Li is a graduate student at the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China.(e-mail: 1251610185@qq.com)

Yafeng Zhao is a graduate student at the School of Computer and Communication, Lanzhou University of Technology, Lanzhou, 730050, China.(e-mail: 1323090227@qq.com)

Suchao Wang is an engineer from the Longnan Power Supply Company of State Grid Gansu Electric Power Company, Longnan, 746000, China.(e-mail: 421250345@qq.com)

traditional manual inspection methods are inefficient, high cost, and poor security. They struggle to meet the inspection needs of large-scale power grids. The development of UAV technology provides a new solution for power inspection. This method is efficient and low-cost. However, in areas with complex terrain, finding suitable obstacle avoidance paths for UAV swarms becomes very challenging. However, it becomes very difficult to find a set of obstacle avoidance paths suitable for UAV swarm inspection in areas with complex terrain [1].

Currently, most researchers use methods such as RRT, artificial potential fields, genetic algorithms, particle swarm optimization[2], and reinforcement learning [3-4] for obstacle avoidance path planning in UAVs [5]. Yang Fan et al. tackled the problems linked to the RRT algorithm, including excessive randomness, inefficient convergence, extended transmission durations, and non-linear flight trajectories. They proposed an improved RRT algorithm combined with Ant Colony Optimization (ACO) methods, which can effectively converge to the optimal solution. Although it can avoid falling into the local optima, the generated paths are not suitable for the flight conditions of UAVs [6-7]. Fusic et al. proposed an improved rapid random tree (IRRT) algorithm. This algorithm incorporates a triangular inequality rewiring technique. It is designed to find obstacle avoidance paths for UAVs in a 3D environment. In comparison with traditional RRT and improved RRT methods, the IRRT algorithm achieves lower planning time and cost-related distances. Additionally, it enhances applicability in formation path planning [8]. Enrique Aldao introduced a real-time algorithm aimed at preventing collisions during UAV autonomous navigation in structured environments. This algorithm can handle both fixed and moving obstacles. It utilizes simplified geometric models and 3D sensor data for obstacle detection and avoidance. This approach ensures the safety of UAVs while performing tasks [9]. In their evaluation of UAV path planning for collision avoidance, Zhao et al. examined the effectiveness of the fuzzy logic, artificial potential field and ant colony algorithms. They identified that the artificial potential field method tends to get trapped in local minima. To address this issue, they proposed an improvement by introducing the vertical guidance repulsion to help the UAV escape from the local minimum [10]. Wang et al. proposed a dual-mode control strategy to address the navigation problem of UAVs in environments with obstacles during formation flight. An improved Grossberg neural network algorithm was utilized to avoid obstacles and multi-UAV collision [11]. Yan et al. proposed a heuristic trajectory generation scheme for complex offshore environments that can generate optimal trajectories based on complicated

terrain conditions and avoid uncertain obstacles[12]. Tu enhanced path planning by leveraging the Q-learning algorithm and applied this reinforcement learning technique to obstacle avoidance within the AirSim simulation environment [13]. Lin introduced a real-time obstacle avoidance strategy based on a dual-game framework, which demonstrated the ability to guide UAV swarms through complex, narrow environments while also mitigating internal disorder, thereby reducing the risk of collisions [14].

Although previous studies have made progress in path planning and obstacle avoidance control, existing algorithms still face challenges[15-16]. These challenges include handling obstacles in complex environments, ensuring path smoothness, and facilitating multi-UAV collaboration. To address these issues, this paper proposes a novel obstacle avoidance path planning method. The method is based on the PSO-ABC-SA algorithm. Its aim is to enhance the obstacle avoidance capability and path planning efficiency of UAVs in complex scenarios, as depicted in Fig. 1. The main contributions of this work are summarized as follows:

(1) In this study, we have developed a collaborative inspection model for multi-UAV power systems, addressing the challenges of multi-UAV operations in complex terrains. By integrating spatial and temporal synergy constraint models, we ensure efficient collaborative operations of UAV swarms in intricate landscapes. Additionally, we introduce path smoothing techniques to minimize the turning angles and pitch variations during UAV flights, thereby maintaining the stability and safety of UAV operations.

(2) A novel obstacle avoidance path planning method based on the PSO-ABC-SA algorithm is proposed. This method integrates PSO, ABC, and SA algorithms to enhance the capability of UAVs in finding globally optimal paths while avoiding obstacles in complex environments. The algorithms PSO-SA, ABC-SA, QPSO, PSO, and AVOA are selected to be compared in five groups of test functions, and

among the nine sets of test functions, the optimization performance of the PSO-ABC-SA algorithm has significant advantages.

(3) Comparative experiments involving five optimization algorithms were performed to demonstrate the proposed algorithm's effectiveness. These experiments were performed in both simulated terrain, created with a two-dimensional grid method, and simulated mountainous terrain. The experimental results indicate that the PSO-ABC-SA algorithm demonstrates superior reliability in obstacle avoidance performance compared to the other algorithms. Secondly, this study extends the application scenario to the multi-domain environment of sea, land, and air. The effectiveness of the algorithm is similarly verified by three sets of comparative experimental results.

(4) Comparative experiments were conducted to evaluate the performance of the PSO-ABC-SA algorithm against five other algorithms. The algorithm exhibited strong performance across multiple evaluation metrics. In comparison with the PSO algorithm, the PSO-ABC-SA algorithm reduced the optimal path length by 251 km and the average path length by 130 km. Furthermore, it achieved an obstacle avoidance rate of 94.2%, representing a 5.7% improvement in average obstacle avoidance rate. The PSO-ABC-SA algorithm exhibited optimal performance in terms of best cost, worst cost, and average cost. These comprehensive results validate the reliability and effectiveness of the PSO-ABC-SA algorithm in path planning and obstacle avoidance.

II. OBSTACLE AVOIDANCE MODEL IN COMPLEX MOUNTAIN ENVIRONMENT

A. Multi-UAV Collision Constraint

Assume that there are N UAVs in the UAV swarm, and the position of each UAV is represented by a two-dimensional vector as $x_i = [x_{i1}, x_{i2}]$, where $i = \{1, \dots, i, \dots, N\}$. Then the objective function is:

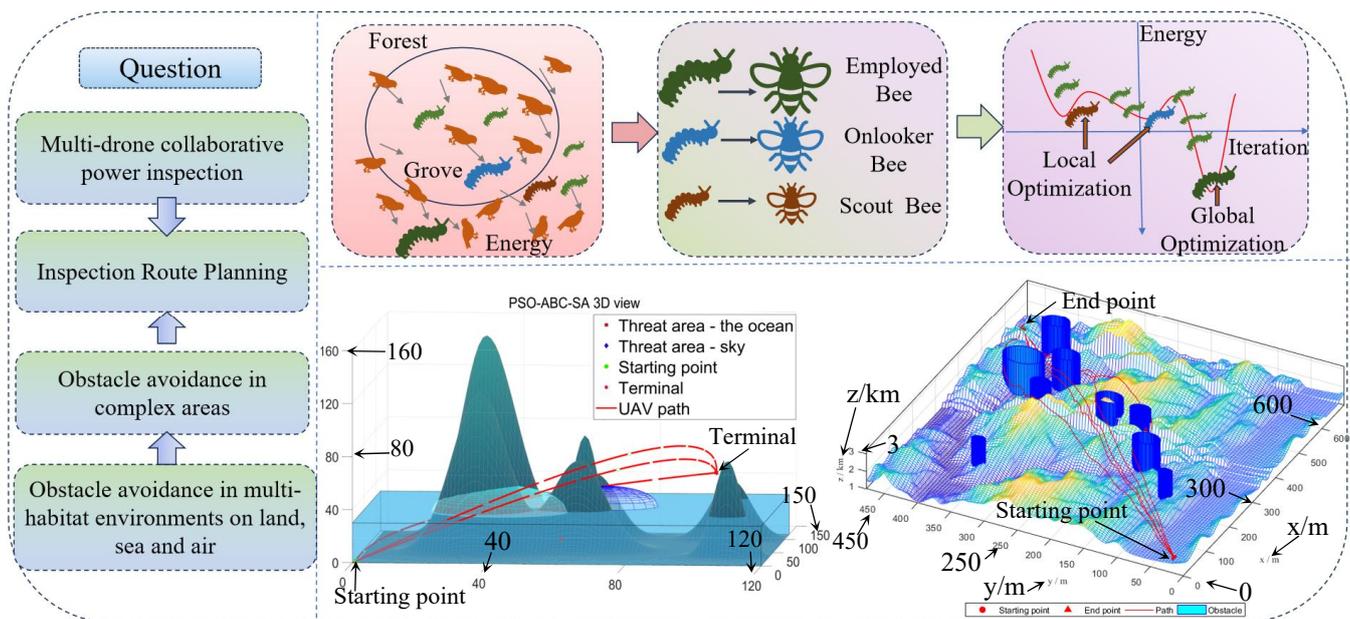


Fig. 1. Multi-UAV cooperative obstacle avoidance model

$$f_{obstacle} = \sum_{i=1}^N (\omega_1 d_{ic} + \omega_2 d_{io}) \quad (1)$$

Where d_{ic} is the distance between UAVs, d_{io} is the distance between UAVs and obstacles, ω_1 and ω_2 are weight coefficients used to balance the distances between UAVs and obstacles. Let D_{min} be the minimum distance. The constraint conditions are:

$$d_{icmin} \leq D_{min} \quad (2)$$

$$d_{iomn} \leq D_{min} \quad (3)$$

B. Flight Constraints in Complex Environments.

To maintain safe UAV inspection operations in the complex mountain environment, the flight altitude cannot be too high or too low. It typically lies between the given two extreme values, so it is necessary to set the maximum and minimum flight altitudes, denoted as H_{max} and H_{min} , as shown in equation (4). Meanwhile, the turning radius r_i during UAV flight should be greater than or equal to the minimum turning radius r_{min} , as shown in equation (5). The inspection area of the UAV is constrained according to the actual inspection task, as shown in equation (6), where S_{min} is the minimum inspection area. The pitch angle constraints generated by the UAV during flight due to climb or descent operations are shown in equation (7), where $Z_i \neq Z_{i+1}$. The UAV needs to constrain the turning angle during flight due to its own performance as well as inertia, as shown in equation (8), which satisfies the condition $Z_i = Z_{i+1}$. The above constraints are utilised to ensure that the UAV can avoid obstacles smoothly in complex terrain.

$$H_{min} < H_i < H_{max} \quad (4)$$

$$r_i \geq r_{min} \quad (5)$$

$$S \geq S_{min} \quad (6)$$

$$\frac{|Z_i - Z_{i-1}|}{\sqrt{(X_i - X_{i-1})^2 + (Y_i - Y_{i-1})^2}} \leq \tan \gamma \quad (i = 2, \dots, n) \quad (7)$$

$$0 \leq \frac{1}{\cos} \left[\frac{X_{i+1}X_iX_{i-1} + Y_{i+1}Y_iY_{i-1}}{\sqrt{X_{i+1}^2 + X_i^2 + X_{i-1}^2} \sqrt{Y_{i+1}^2 + Y_i^2 + Y_{i-1}^2}} \right] \leq \theta_{max} \quad (8)$$

C. Multi-UAV Spatial and Temporal Synergy

In order to avoid the collision of UAVs during the flight process, the distance between UAVs is constrained in the spatial. d_{ij} denotes the distance between the i -th UAV and the j -th UAV, and d_{min} denotes the minimum distance between UAVs as shown in equation (9). In time, the speed of the UAV is v can be set between $[V_{min}, V_{max}]$, the length of the route is L , and the arrival time at the target position is the time used as shown in equation (10). If $T_{max} = t_1 \cap t_2 \cap \dots \cap t_n$ is a non-empty set, then t in any set can be used as a synergy time.

$$d_{ij} \geq d_{min} \quad (9)$$

$$t_i = \left[\frac{L}{V_{max}}, \frac{L}{V_{min}} \right] \quad (10)$$

D. Path Smoothness Processing

The optimization of the obstacle avoidance path smoothing algorithm is a key technical link to achieve the smoothness of UAV movement and the improvement of operational efficiency in the UAV power inspection task. The path planning model in this study uses the shortest Euclidean distance between discrete waypoints as the initial optimization objective. Global paths are generated by connecting the waypoints. However, such paths are prone to triggering frequent UAV attitude adjustments due to curvature discontinuities at turning points. This in turn leads

to problems such as increased energy loss. In order to solve the above defects, this paper adopts the multiconstrained path smoothing algorithm based on cubic spline interpolation, and equation (12) presents the relevant mathematical expression. The algorithm effectively suppresses the phenomenon of abrupt change of heading angle by ensuring the continuous property of the second-order derivative of the trajectory. It significantly improves the stability of UAV flight. In the optimization process, multi-objective constraints such as cost of obstacle avoidance, cost of voyage, and cost of inspection are added in this study, as shown in equations (1), (14), and (15), respectively. By introducing an adaptive weight adjustment mechanism, the competition between the subgoals is dynamically balanced. The continuous flyable path that meets the UAV dynamics constraints is finally generated. The specific smoothing process is shown in equation (11), where f_{smooth} represents the smoothing function, S denotes the target path, and (a_i, b_i, c_i) denotes the coordinate points in the 3D space.

$$S = \begin{pmatrix} a_1 & a_2 \dots a_i \\ b_1 & b_2 \dots b_i \\ c_1 & c_2 \dots c_i \end{pmatrix} \quad (11)$$

$$f_{smooth} = \sum_{i < j} \sqrt{\|S(:, i+1) - S(:, i)\|^2} \quad (12)$$

E. Revenue Function

In this paper, the revenue function consists of a range cost, an obstacle avoidance cost, an inspection cost, and a smoothness processing cost. Different weight coefficients are set for each cost, $\omega_1, \omega_2, \omega_3, \omega_4$, with different costs indicating different degrees of influence. The revenue function is:

$$f = \omega_1 f_{voyage} + \omega_2 f_{obstacle} + \omega_3 f_{inspection} + \omega_4 f_{smooth} \quad (13)$$

$$f_{voyage} = d_{ij} = \sum_{j=0}^N \sum_{j=1}^{N+1} \sqrt{dx^2 + dy^2 + dz^2} \quad (14)$$

where dx , dy , dz are the components of the Euclidean distance, respectively, and N is the number of UAVs.

Maximising the sum of weights of high-voltage electric towers inspected by UAV swarms as an objective function.

$$f_{inspection} = \text{Max} \sum_{i=1}^N \sum_{j=1}^{N+1} w_i x_{ij} \quad (15)$$

Assuming that the set of the high-voltage electric towers is $A = \{0, 1, \dots, N, N+1\}$, and let the high-voltage electric towers that have not been inspected be i . Then w_i ($i \in T$) is the time interval since the last inspection in which the i -th high-voltage electric tower has not been inspected. If the value of w_i is larger, it means that the priority of the high-voltage electric towers that are not inspected is also higher. $P = \{0, i, \dots, j, N+1\}$, i denotes the UAV and x denotes the distance.

III. OBSTACLE AVOIDANCE MODEL IN MULTI-DOMAIN ENVIRONMENT

A. Peak Collision Avoidance Constraint

In a multi-domain environment, drones need to avoid obstacles such as mountain peaks during flight to ensure safe flight. To effectively represent the characteristics of complex terrain, this study defines the geometry of the peak as a cone, C denotes the center coordinates of the peak, R denotes the horizontal radius of the peak, and H denotes the height of the peak. Define $f_{collision}$ as the collision function,

which is used to detect whether the path of the UAV collides with the mountain peak.

$$f_{collision} = \begin{cases} 0 & z_{path} < z_c \text{ or } \sqrt{(x-x_c)^2 + (y-y_c)^2} \geq R \\ (z_{path} - H)^2 & \sqrt{(x-x_c)^2 + (y-y_c)^2} < R \text{ and } z_{path} \geq z_c \\ \infty & z_{path} \geq H \end{cases} \quad (16)$$

Z_{path} denotes the flight altitude of the UAV, (x_c, y_c, z_c) denotes the coordinates of the center point of the horizontal cut of the peak, R denotes the radius.

B. Radar Detection Constraint

For radar detection facilities distributed in the environment, to maintain UAV flight stability, the radar detection zone must be avoided. The definition function f_{radar} represents the distance relationship between the UAV and the radar, and r_{radar} represents the detection radius of the radar. When the UAV's distance to the radar falls below its detection threshold, a penalty is triggered.

$$f_{radar} = \begin{cases} 0 & d_i \geq r_{radar} \\ (r_{radar} - d_i)^2 & d_i < r_{radar} \end{cases} \quad (17)$$

C. Atmospheric Impact Constraint

In the atmosphere, taking into account the differences in aerodynamic characteristics of the UAV when flying at different altitudes. Define the function $f_{atmosphere}$, which represents the effect of flight altitude on energy consumption. k_a is a constant related to atmospheric drag.

$$f_{atmosphere} = k_a z_{path}^2 \quad (18)$$

D. Marine Environmental Constraint

There may be unidentified obstacles in the ocean, so UAV need to keep a safe distance during flight to avoid collisions. Define the function f_{ocean} , which represents the restriction of the marine environment on the path of the UAV power inspection process.

$$f_{ocean} = \begin{cases} 0 & \sqrt{(x_i - x_{oc})^2 + (y_i - y_{oc})^2 + (z_i - z_{oc})^2} \geq R_{oc} \\ \infty & \sqrt{(x_i - x_{oc})^2 + (y_i - y_{oc})^2 + (z_i - z_{oc})^2} < R_{oc} \end{cases} \quad (19)$$

(x_i, y_i, z_i) represents the position of the i -th UAV, (x_{oc}, y_{oc}, z_{oc}) represents the position of the obstacle in the ocean, and R_{oc} represents a safe range between the UAV and the obstacle.

E. Objective Optimization Function

When conducting power inspection tasks in multi-domain environments encompassing land, sea, and air, it is essential to comprehensively consider the multi-dimensional constraints imposed by the distinct characteristics of each environment. In this study, the influencing factors present in the mountain peaks, radar, atmosphere, and ocean are used as constraints. Meanwhile, ensuring that the UAV follows a smooth flight path requires careful planning, the same cubic spline interpolation method is used as the path smoothing algorithm to ensure that the planned path meets the conditions of stable UAV flight, as in equation (12). Let the objective optimization function of multi-UAVs for power inspection in multi-domain environments be F_{min} . Set the weight coefficients for each constraint to indicate different degrees of influence.

$$F_{min} = \omega_1 f_{collision} + \omega_2 f_{radar} + \omega_3 f_{atmosphere} + \omega_4 f_{ocean} + \omega_5 f_{smooth} \quad (20)$$

IV. DESCRIPTION OF THE BASE ALGORITHM

A. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a population-based optimization method inspired by swarm intelligence. In this algorithm, each individual — referred to as a "particle" — navigates the search space, representing a potential optimal solution. These particles navigate through a multi-dimensional space, guided by two key "extrema": the personal best, which is the optimal solution discovered by the individual particle, and the global best, which represents the best solution found by the whole swarm of particles. The particles update their positions and velocities, taking into account not only their own experience but also the wisdom of the group, especially the experience of other particles. In this way, the PSO algorithm can effectively search through the solution space and converge to a region close to the global optimal solution [17].

$$v_{ij}(t+1) = w * v_{ij}(t) + c_1 * r_1 * (Pbest_{ij} - x_{ij}(t)) + c_2 * r_2 * (Gbest_{ij} - x_{ij}(t)) \quad (21)$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (22)$$

where $v_{ij}^{(t+1)}$ is the velocity of the i -th particle in the j -th dimension under generation $t+1$. $v_{ij}^{(t)}$ denotes the velocity of the i -th particle in the j -th dimension under generation t . The parameters c_1 and c_2 are the learning factors, w is the inertia weight, r_1 and r_2 are random numbers between [0,1] used to increase the randomness of the algorithm. Additionally, $Gbest_{ij}$ is the global optimal position of the entire particle swarm in the j -th dimension, and $Pbest_{ij}$ is the localized position of the i -th particle in the j -th dimension optimal position

B. Artificial Bee Colony

Inspired by the foraging activities of real bees, the Artificial Bee Colony (ABC) algorithm serves as a nature-based optimization technique. The algorithm identifies optimal solutions by leveraging the cooperative behavior of three bee roles: employed, onlooker, and scout bees. The updating of the solution can be expressed as [18-19]:

$$X_{ij} = X_{minj} + rand[0,1](X_{maxj} - X_{minj}) \quad (23)$$

$$V_{ij} = X_{ij} + \phi_{ij}(X_{ij} - X_{kj}) \quad (24)$$

Where X_{ij} denotes the position of the i -th bee in the j -th dimension, and $rand[0,1]$ is a random number in the range [-1,1] denoting a random number. The search phase, where the scout bees searches for a new nectar source is denoted as equation (19).

C. Simulated Annealing

Simulated Annealing (SA) is a stochastic optimization technique that mimics the thermal annealing process in physics to locate a function's global minimum. The fundamental idea of the algorithm is that, in the initial phase, the system is allowed to conduct a broad search at a high temperature. As the "temperature" gradually decreases, the search range narrows, ultimately converging to the global optimum or an approximate global optimum. The iterative process is as follows [20-21]:

(1) Randomly select a new solution from the neighborhood of the current solution as a candidate solution.

(2) Calculate the energy difference between the candidate

solution and the current solution. If the candidate solution is better than the current solution, accept the candidate solution as the new current solution.

(3)When the new solution performs worse than the current one, it may still be accepted according to a probability defined by the Metropolis criterion: $p = \exp(-\frac{\Delta E}{KT})$, where ΔE is the energy difference, K is a constant, and T is the current temperature.

V. ALGORITHM IMPROVEMENT STRATEGY

A. Algorithm Description

This paper explores a method that combines the PSO and ABC algorithms, utilizing the update mechanism of the ABC algorithm to update the iterations of the particle swarm, thereby avoiding getting trapped in local optima cycles. The aim is to leverage the strengths of both algorithms, enhance global search capability, and avoid getting stuck in local optimal solutions.

In the initial stage, an initial swarm of particles is randomly generated, with each particle representing a potential solution. Based on the fitness of the particles and the grouping strategy of the ABC algorithm, the particles are divided into n groups, with each group consisting of scout bees, employed bees, and onlooker bees. During the algorithm iteration process, scout bees are responsible for searching and updating the solution space to prevent the algorithm from getting trapped in local optimal solutions. Employed bees focus on exploiting known areas of the solution space, searching locally in the domain of the current solution for a better solution, and sharing this information with onlooker bees. Onlooker bees conduct optimal searches in local regions based on the information brought by employed bees, and the ranking of particles within the group is adjusted based on changes in fitness. As the algorithm continues to iterate, the weight of particle updates gradually decreases. To address this, the SA algorithm is introduced to enhance local search capabilities and help the algorithm escape local optimal solutions.

The update operation for these optimal particles is as follows:

$$v_{ij}(t+1) = w * v_{ij}(t) + c_1 * r_1 * (Pbest_{ij} - x_{ij}(t)) + c_2 * r_2 * (Gbest_{ij} - x_{ij}(t)) \quad (25)$$

$$X_{ij}(t+1) = X_{min_{ij}}(t+1) + rand[0,1](X_{max_{ij}}(t+1) - X_{min_{ij}}(t+1)) \quad (26)$$

$$fit_{ij} \begin{cases} 1/(1+f_{i,j}), & f_{i,j} \geq 0 \\ 1+abs(f_{i,j}), & otherwise \end{cases} \quad (27)$$

$$C_1 = \exp\{1/(1+f_{i,j})\} \quad (28)$$

$$C_2 = \exp\{1+abs(f_{i,j} - f_{k,j})\} \quad (29)$$

X_{ij} denotes the i -th particle of the j -th group, C_1 and C_2 are the improved learning factors. The learning factor can be made to decrease with the number of iterations, thus enhancing the convergence of the algorithm. f_{ij} is the fitness of the current particle. $Pbest_{ij}$ is the optimal solution of X_{ij} in the j -th group, and $Gbest_{ij}$ is the optimal solution in the j -th group except $Pbest_{ij}$. $f_{k,j}$ is the fitness of any particle in the j -th group. Each particle within the

group is updated according to the following criterion:

$$v_i(t+1) = w * v_i(t) + c_1 * r_1 * (Pbest_i - x_i(t)) + c_2 * r_2 * (Gbest_i - x_i(t)) \quad (30)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (31)$$

Then, the simulated annealing algorithm is modified to update the local search capability based on the Metropolis criterion in the algorithm:

$$x_i(t+1) = \begin{cases} x_i(t+1) & \Delta f < 0 \\ x_i(t) & otherwise \end{cases} \quad (32)$$

$$\exp(-\Delta f/T(t)) > rand \quad (33)$$

When $\Delta f < 0$, it means that the particle is poorly adapted, and it can be adjusted by equation (33). The adjustment can avoid falling into the local optimal solution problem again, and the other particles do not change the state. The pseudocode of the algorithm is shown in Algorithm 1:

Algorithm 1: PSO-ABC-SA algorithm

```

1: Input: PopSize(number of particles in the swarm), n(population size), T (the number of iterations), d (the problem dimension), T0( initial temperature), Tend(final temperature), w (weight), C1 and C2 (learning factors)
2: Output: Global optimal solution
3: for i=1 to PopSize do
4:   | Initialise the particle swarm
5: end for
6: Calculate the particle fitness, divide the particles into n groups according to the fitness, and determine the scout bees, employed bees, and onlooker bees in each group
7: While t<T do
8:   | Initialise the optimal particle Pbest for the group and Gbest for the global optimal particle
9:   for i=1 to n do
10:    | for j=1 to d do
11:      | Calculate the fitness of the particles
12:      | if fitness[i][j] > fitness[Pbest]
13:        |   | Pbest=fitness[i][j]
14:      | if fitness[i][j] > fitness[Gbest]
15:        |   | Gbest=fitness[i][j]
16:      | The velocity and position of the optimal particles in the group are updated by equations. (25)(26)(27)(28)
17:      | Updating the velocity and position of the particles in the group by equations. (30)(31)
18:    | end
19:   end
20: Calculate the change in fitness of the particle  $\Delta f$ 
21:   while (T0<Tend) do
22:     | if ( $\Delta f \geq 0$ )
23:       | Adjustment of poorly adapted particles.
24:     end
25:   Recalculating comparative fitness and updating scout, employed, onlooker bees
26:   t=t+1
end

```

B. Algorithm Performance Evaluation in Benchmarking Functions

In order to evaluate the optimization ability of the PSO-ABC-SA algorithm, 10 sets of test functions were selected for experiments, as shown in Table 1. The algorithms PSO-SA [22], ABC-SA, QPSO [23], PSO, and AVOA [24] are selected to compare with the PSO-ABC-SA

algorithm. The search space range and fitness evolution curve of the test function are shown in Fig. 2. The three-dimensional search space can intuitively reflect the solution value region within the range and the curve graph clearly compares the performance of the algorithms. Table 2 shows the comparison of the test results of the ten test functions. Comparing Fig. 2 and Table 2, it can be found that among the nine sets of test functions, the PSO-ABC-SA algorithm reached the optimal. In summary, The PSO-ABC-SA algorithm demonstrates significant strengths in both optimization performance and convergence behavior. And a better solution can be found in a random experiment.

C. Computational Complexity Analysis

The PSO-ABC-SA algorithm integrates the characteristics of three optimization techniques: PSO, ABC, and SA algorithms. So its particle search process exhibits significant nonlinearity and complexity. Consequently, analyzing the algorithm's complexity qualitatively and directly from a theoretical standpoint proves difficult. Therefore, in order to quantitatively analyze the computational complexity of an algorithm, researchers usually evaluate the actual running time of the algorithm as a measure. Suganthan et al. proposed an evaluation method that enables quantitative analysis and comparison of the computational complexity of different AI algorithms.

$$CB = \frac{\widehat{T}_2 - T_1}{T_0} \tag{34}$$

Where CB denotes the computational complexity of the algorithm and \widehat{T}_2, T_1, T_0 denote the computational time required for each mathematical operation, respectively.

According to the literature [25], T_0 means that 1000000 iterations were performed for the input parameter $x = 5.55$.

In each iteration, a series of calculations are performed, and the total time consumed is found, including self-increment of x , division of x by 2, squaring of x , logarithm of x , exponentiation of x and division of x by x . T_1 denotes the total computational time consumed by the algorithm to solve the test function F_4 in the table once when the number of iterations is 200000. \widehat{T}_2 denotes the average computational time consumed by the algorithm to solve the function f_4 five times for the same number of iterations. As shown in Table 3, findings suggest that the PSO-ABC-SA algorithm introduced in this study can substantially decrease computational complexity, thus improving the reliability of the search and the average convergence speed.

VI. EXPERIMENTAL SIMULATION AND RESULTS ANALYSIS

Aiming at the problem of obstacle avoidance path planning in UAV swarms, this study designs a series of experiments based on intelligent optimization algorithms to evaluate its navigation capability in a complex mountain environment and multi-habitat. To evaluate the algorithm's obstacle avoidance capability, we examine several key aspects. These include the smoothness of the planned path, the degree of turning, and the spatial and temporal synergy of multi-UAV collaborative operations. The experiment assesses the algorithm's effectiveness from multiple perspectives. The specific experimental parameters are as follows: Particle group size: 3, Number of iterations: 200, Population size: 50, Weight: 0.9, Degradation rate: 0.99, Learning factors C_1 : 1.5 and C_2 : 1.5, Internal iterations for the local annealing algorithm: 100, Initial temperature: 10,000, and Temperature decay coefficient: 0.99.

TABLE I
TEST FUNCTION

Function	Dim	Range	Minnum
$f_1(x) = \max_i\{ x_i , 1 \leq i \leq n\}$	30	[-100,100]	0
$f_2(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[-5,5]	0.1484
$f_3(x) = 0.5 + \frac{\sin^2(x_1^2 - x_2^2) - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	2	[-100,100]	0
$f_4(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10 \left(1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	[-5,5]	0.3
$f_5(x) = - \sum_{i=1}^4 c_i \exp \left[- \sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2 \right]$	6	[0,1]	-3
$f_6(x) = - (x_2 + 47) \sin \left(\sqrt{\left x_2 + \frac{x_1}{2} + 47 \right } \right) - x_1 \sin \left(\sqrt{ x_1 - (x_2 + 47) } \right)$	2	[-512,512]	-959.6407
$f_7(x) = \sum_{i=1}^d \frac{x_i^2}{4000} - \prod_{i=1}^d \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1$	d	[-600,600]	0
$f_8(x) = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2 [1 + \sin^2(2\pi x_2)]$	2	[-10,10]	0
$f_9(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90(x_3^2 - x_4)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$	4	[-10,10]	0

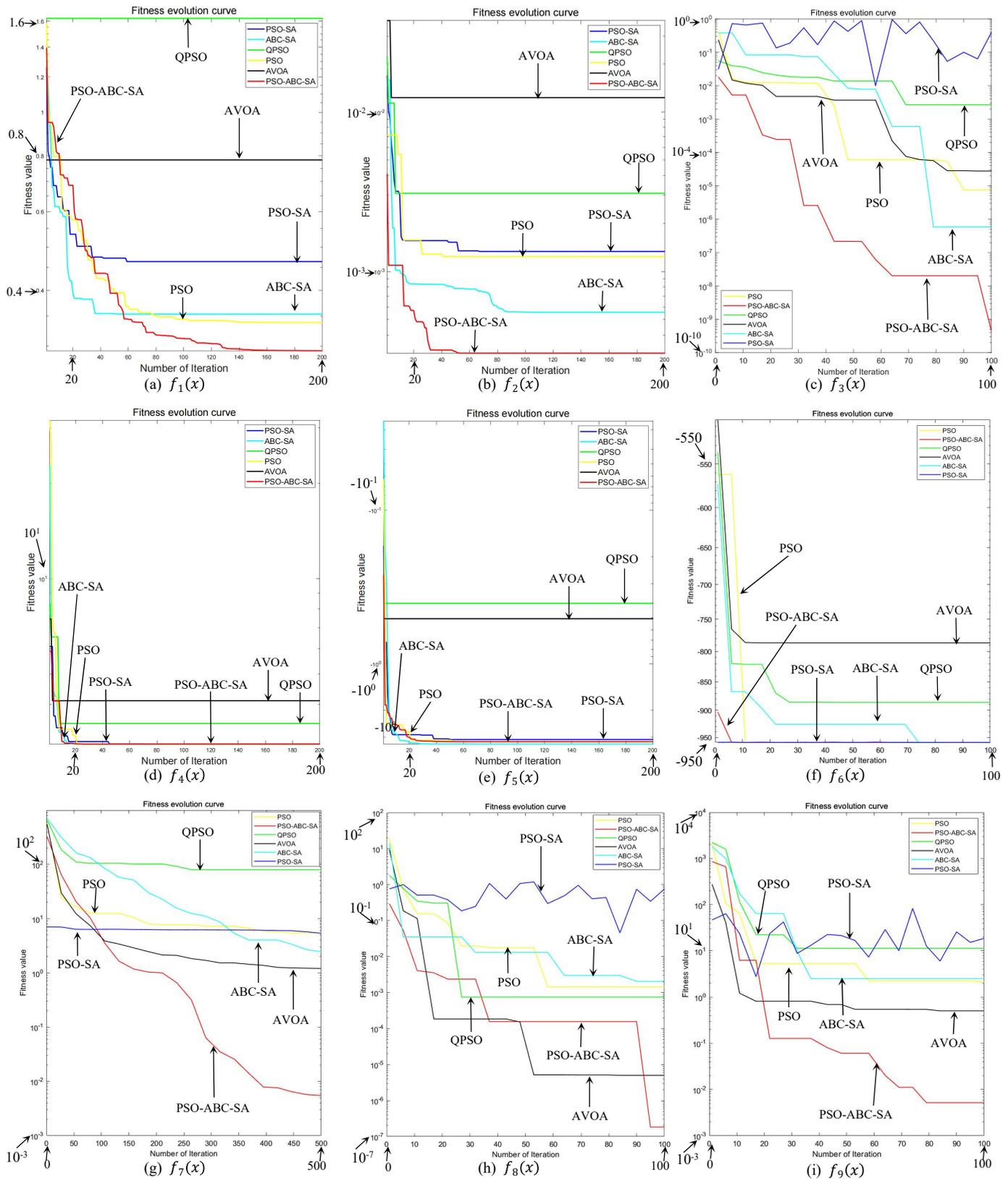


Fig. 2. Average function curves comparison

A. Obstacle Avoidance Performance Validation in Single-domain Environment

1) Two-dimensional Scene Simulation

i) Experiments in Sparse Obstacle Environments

In this study, in order to comprehensively verify the effectiveness and feasibility of the PSO-ABC-SA algorithm, a 2D terrain simulation environment based on a two-dimensional grid method[26] is constructed. This method can provide an intuitive view of path planning. As

shown in Fig. 3, by designing comparative experiments, the path planning performance of the Particle Swarm Optimization (PSO), the Genetic Algorithm (GA)[27], Grey Wolf Optimization (GWO)[28], Sparrow Search Algorithm (SSA)[29], and the PSO-ABC-SA hybrid algorithm proposed in this study is evaluated under a unified terrain environment. The results show that the PSO-ABC-SA algorithm not only has excellent obstacle avoidance capability but also demonstrates significant advantages in

path optimization efficiency.

ii) Experiments in Intensive Obstacle Environments

To further assess how well the PSO-ABC-SA algorithm performs in handling obstacle avoidance within more challenging environments. The simulated terrain was enhanced with an increased number of obstacles and irregular distribution patterns, thereby elevating the difficulty of the test scenarios. As shown in Fig. 4, even with a higher obstacle density and more complex terrain conditions, the PSO-ABC-SA algorithm consistently identifies the global optimal path. Its fitness value reaches the lowest point, demonstrating excellent path planning

capabilities. The outcome illustrates that the algorithm remains both robust and efficient, even in highly complex environments.

2) Three-Dimensional Mountain Environment Simulation

i) Map Construction

To assess the feasibility of applying the PSO-ABC-SA algorithm in real-world scenarios[30]. This study conducts experimental simulations of obstacle avoidance paths for UAV swarms. The simulation environment was modeled using two-dimensional cubic convolution interpolation, with the size set to 400 m× 800 m× 3000 m, as shown in Fig. 5.

TABLE II
OPTIMIZATION RESULTS FOR DIFFERENT FUNCTIONS

Function	Optimization Result Value						Minnum
	PSO-SA	ABC-SA	QPSO	PSO	AVOA	PSO-ABC-SA	
$f_1(x)$	0.4637	0.3536	1.623	0.3387	0.7828	0.2928	0
$f_2(x)$	0.001336	0.0005548	0.003095	0.00124	0.01231	0.0003075	0.1484
$f_3(x)$	0.39286	5.8706e-07	0.0026845	7.4979e-06	2.7764e-05	4.4917e-10	0
$f_4(x)$	3.004	3	3.487	3	4.114	3	0.3
$f_5(x)$	-3.102	-0.4034	-0.4034	-3.203	-0.5078	-3.32	-3
$f_6(x)$	-959.6403	-959.5563	-885.7882	-959.6407	-786.5241	-959.6407	-959.6407
$f_7(x)$	5.3721	2.4601	79.7525	5.4311	1.2067	0.0054575	0
$f_8(x)$	0.75412	0.0020573	0.00075313	0.0014425	5.0308e-06	1.8316e-07	0
$f_9(x)$	18.6828	2.51	11.3647	2.118	0.50424	0.0051871	0

TABLE III
COMPUTATIONAL TIME (IN SECONDS) CONSUMED BY THE MULTI-ALGORITHM TO SOLVE THE MULTIMODAL BENCHMARK FUNCTION f_4

Formula mode	Algorithm						Minnum
	PSO-GA	PSO-SA	GRO	SWO	COA	PSO-ABC-SA	
T_0	1.8103E-01	1.8103E-01	1.8103E-01	1.8103E-01	1.8103E-01	1.8103E-01	1.8103E-01
T_1	1.2006E+02	1.2530E+02	4.6861E+03	4.4315E+02	4.4413E+04	8.5129E+02	1.2006E+02
\widehat{T}_2	1.4763E+05	1.7291E+04	4.9739E+04	4.6720E+02	4.8064E+03	8.7443E+02	1.4763E+05
CB	1.4898E+01	2.6299E+01	1.5897E+01	1.3285E+01	2.0167E+01	1.2782 E+01	1.4898E+01

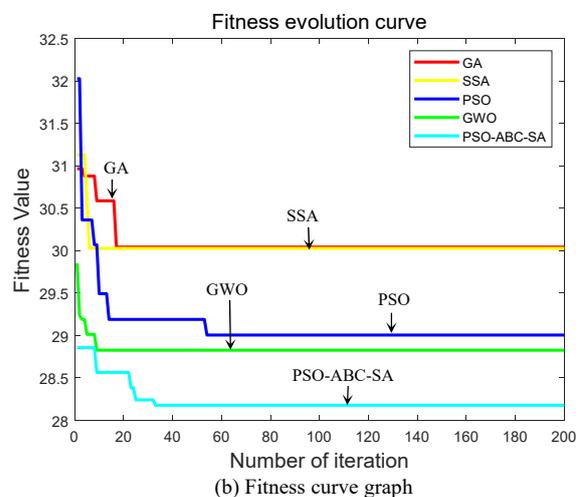
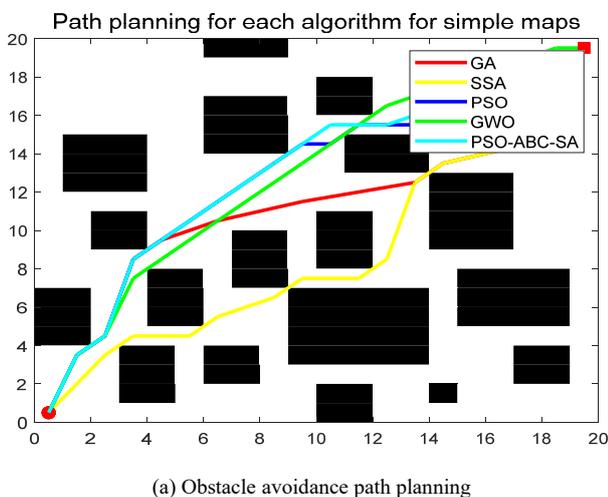


Fig. 3. Simple 2D Obstacle Avoidance

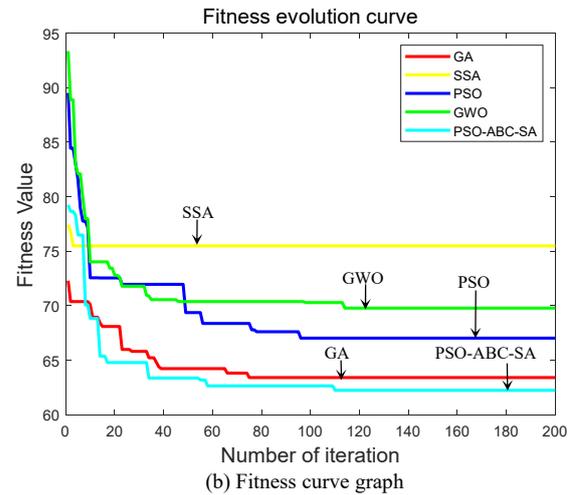
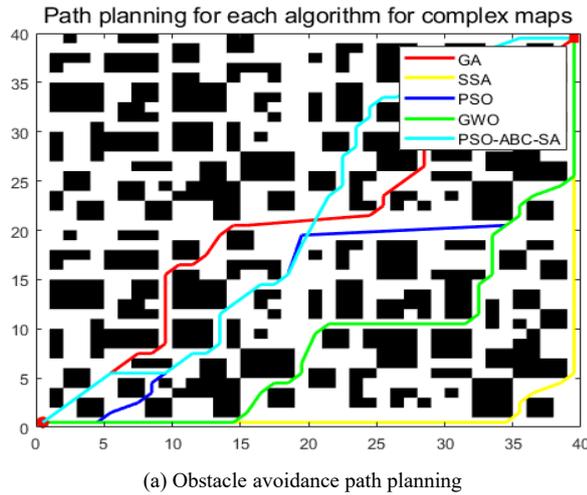


Fig. 4. Complex 2D Obstacle Avoidance

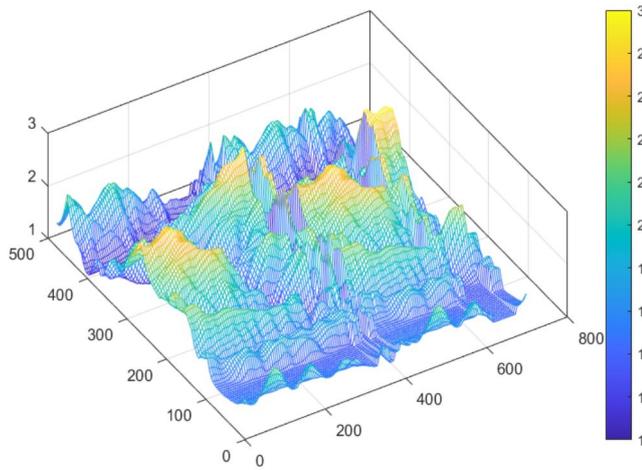


Fig. 5. Simulation map

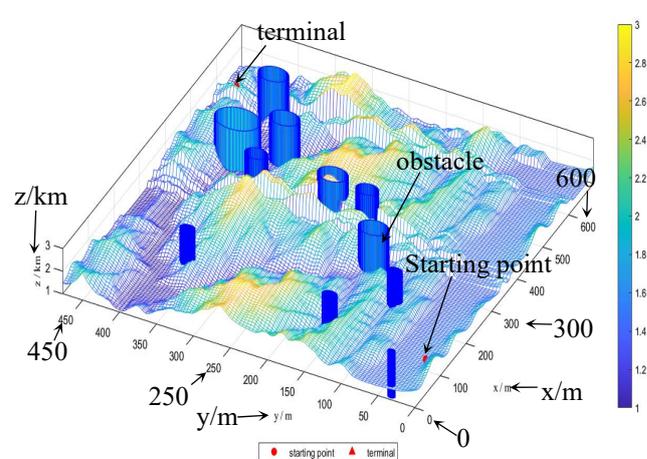


Fig. 6. Obstacle distribution map

To additionally confirm the algorithm’s ability to avoid obstacles effectively, this study established 11 obstacles of varying sizes on the designated map. These obstacles are represented by blue bars in Fig. 6. The two-dimensional coordinates and radii of the obstacles are as follows: (10, 35, 5), (100, 150, 10), (150, 360, 10), (210, 100, 10), (300, 160, 20), (390, 200, 15), (390, 350, 15), (460, 400, 30), (500, 350, 20), (600, 400, 20), and (400, 250, 20). Additionally, the starting and ending points of the UAV inspection path were defined at coordinates (100, 20, 1500) and (600, 450, 2000), respectively. The UAV swarm is required to depart from the starting point and navigate to the endpoint following the obstacle avoidance paths planned by the respective algorithms. To evaluate the algorithm's obstacle avoidance capabilities and the spatial and temporal coordination of multi-UAV operations, five obstacle avoidance paths were designed for testing. The objective is to assess the safety and reliability of the planned paths in actual UAV cluster flight missions. Six algorithms—PSO-ABC-SA, PSO-SA, ABC-SA, QPSO, PSO, and AVOA—were employed to optimize the obstacle avoidance paths for the UAV swarm within the simulation environment.

ii) Multi-Algorithm Obstacle Avoidance Comparison

In the 3D view, this study compares and analyzes the reliability of the PSO-ABC-SA algorithm with the other five algorithms in terms of obstacle avoidance capability. The

results show that all five paths planned based on the PSO-ABC-SA algorithm successfully avoid obstacles and reach the endpoint smoothly, as shown in Fig. 7(a). These five paths comprehensively consider the synergy of time and space, effectively avoiding the collision problem between multiple UAVs. The superiority of the algorithm in path planning is demonstrated. However, two of the paths planned using the PSO-SA algorithm failed to achieve effective obstacle avoidance and directly passed through the blue obstacle, as shown in Fig. 7(b). Similarly, one path was found to have failed to achieve successful obstacle avoidance in Fig. 7(c). Among the five paths planned based on the PSO algorithm, there are some overlapping paths, which do not meet the requirements of cooperative UAV flight, as shown in Fig. 7(e). Fig. 7(d) and Fig. 7(f), on the other hand, show that in the paths planned by the QPSO and AVOA algorithms, there are the same cases of failing to avoid obstacles, and there are crossings between the paths. In addition, the distribution of the five paths in the start and end positions is too concentrated, which may lead to collision accidents during actual UAV cluster flights. In summary, the PSO-ABC-SA algorithm shows high reliability in obstacle avoidance and can provide safe and efficient flight paths for UAV swarms, while the other algorithms have some limitations in obstacle avoidance. This result provides important theoretical support for UAV

swarms for power inspections in complex environments.

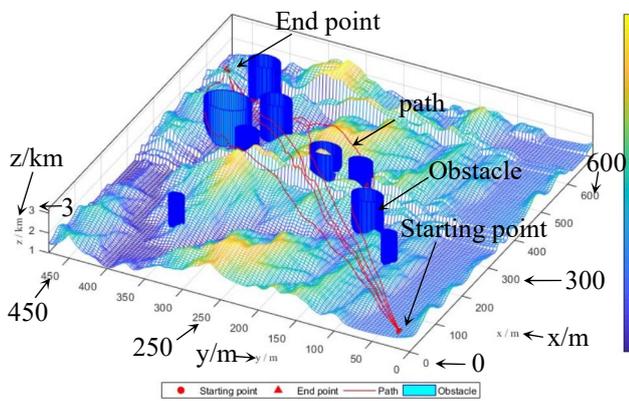
B. Collaborative Obstacle Avoidance Validation in Multi-domain Environment

1) Multi-domain experimental scenario construction

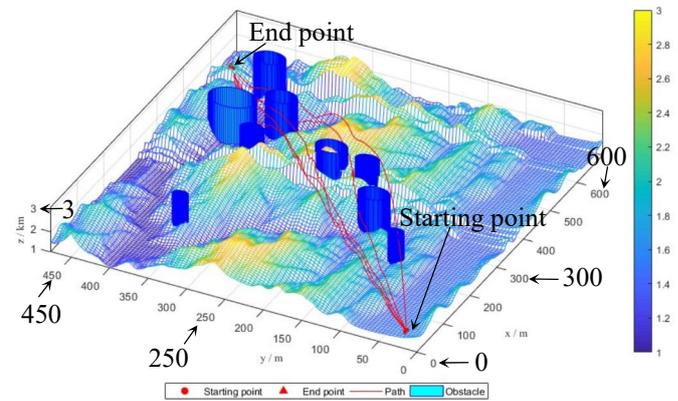
In this study, parametric modeling and spatial overlay are mainly used to randomly generate core features of mountain peaks, including location, height, and extent. And ensure that these features are uniformly distributed in a predetermined area. Subsequently, a Gaussian function model was used to calculate the height value for each ground point, taking into account the mutual influence of multiple peaks, thereby forming a height dataset. Finally, the interpolation technique is utilized to construct a continuous peak surface for realistic peak simulation. In this 3D terrain model, we also set up the marine environment and radar distribution area to increase the diversity of the simulation experiment environment. This design provides a high-confidence experimental basis for validating the cross-domain adaptability and real-time decision-making capability of multi-UAV cooperative obstacle avoidance algorithms. As shown in Fig. 8, the multi-domain simulation experimental environment contains multiple mountain peaks representing land terrain. The bottom of the terrain model represents the ocean area, shown in a blue plane. Meanwhile, red squares and blue diamonds indicate threat areas in the ocean and sky, respectively, and dark and light blue hemispheres indicate two radar areas. With the above diverse simulation setups, this study aims to provide an effective testing platform for the multi-UAV cooperative obstacle avoidance algorithm to evaluate its performance in complex environments.

2) Experimental Results in a Multi-domain Environment

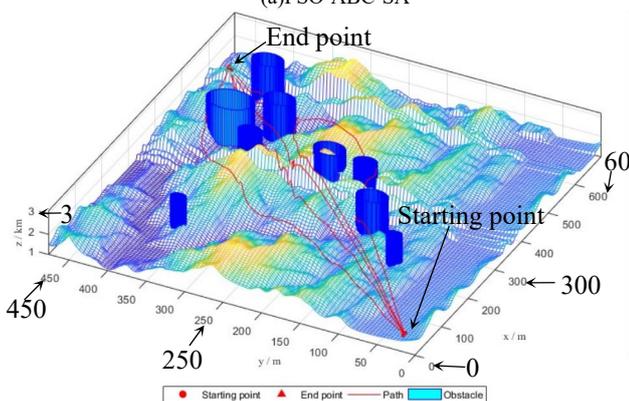
To evaluate the proposed algorithm's capability in obstacle avoidance. The study will extend the application scenario to the multi-domain environment of land, sea, and air. The start and end points were set in the experiment, and the red dotted line indicated the flight path of the UAV. As shown in Fig. 8, the UAV successfully avoided all threat areas and obstacles during its flight, skillfully bypassing mountain peaks and radar areas. In the end, it selected a safe and smooth flight path. In addition, combined with the application of path smoothing algorithms, the planned paths are not only safe but also suitable for the flight characteristics of UAVs. As shown in Figs. 8(a) and 8(b), the 3D view and top-down view clearly show the ability of the PSO-ABC-SA algorithm to effectively plan obstacle avoidance paths for UAVs in a multi-domain environment. Comparing Fig. 9 and Fig. 10, the routes planned based on the PSO algorithm and QPSO algorithm have routes that do not successfully avoid mountain peaks or radar areas. This does not fulfill the condition of multi-UAV cooperative work. In addition, the fitness curve shown in Fig. 11(a) further demonstrates the optimization ability of the PSO-ABC-SA algorithm during the iterative process. This rapid improvement in fitness values implies that the algorithm can identify optimal directions early in the search process. The subsequent stabilization of fitness values shows the algorithm's good convergence. It can find the optimal solution within a limited number of iterations. Additionally, comparing Figs. 11(b) and 11(c), it has the smallest fitness value. In summary, results from the experiments provide strong evidence that the PSO-ABC-SA algorithm has good application prospects in multi-domain environments and provides an effective solution for UAV obstacle avoidance path planning.



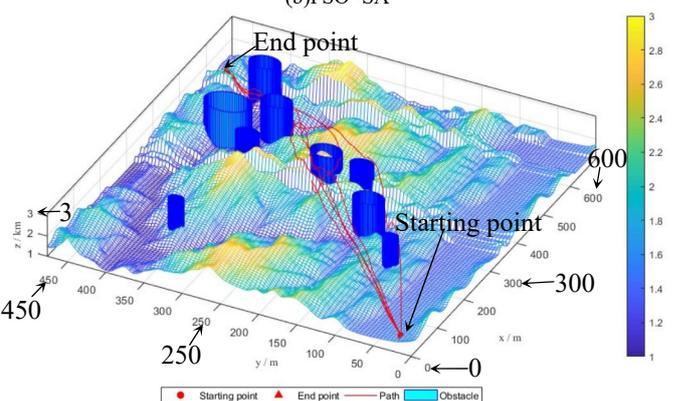
(a) PSO-ABC-SA



(b) PSO-SA



(c) ABC-SA



(d) QPSO

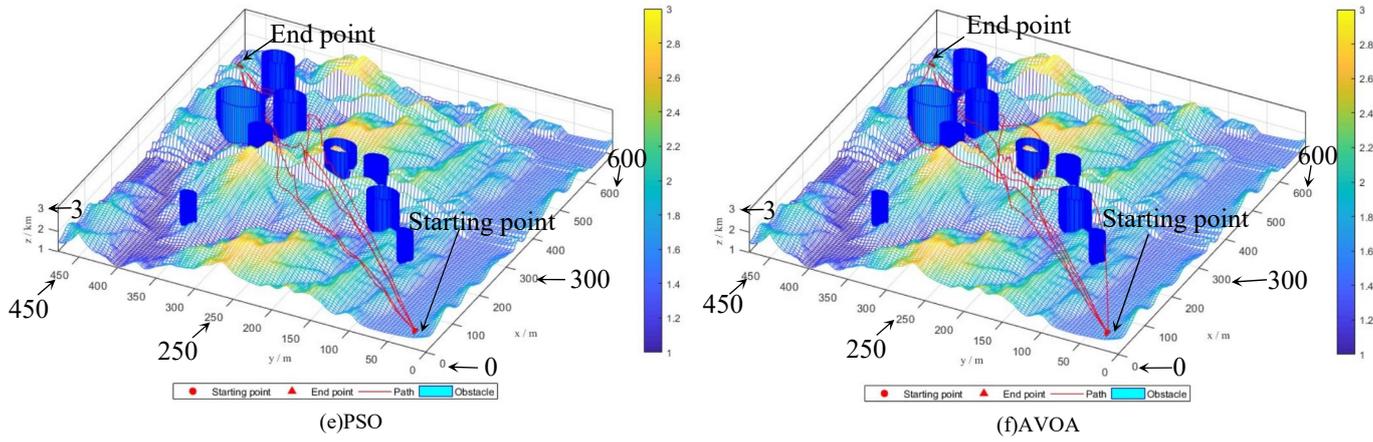


Fig. 7. Obstacle avoidance algorithms comparison

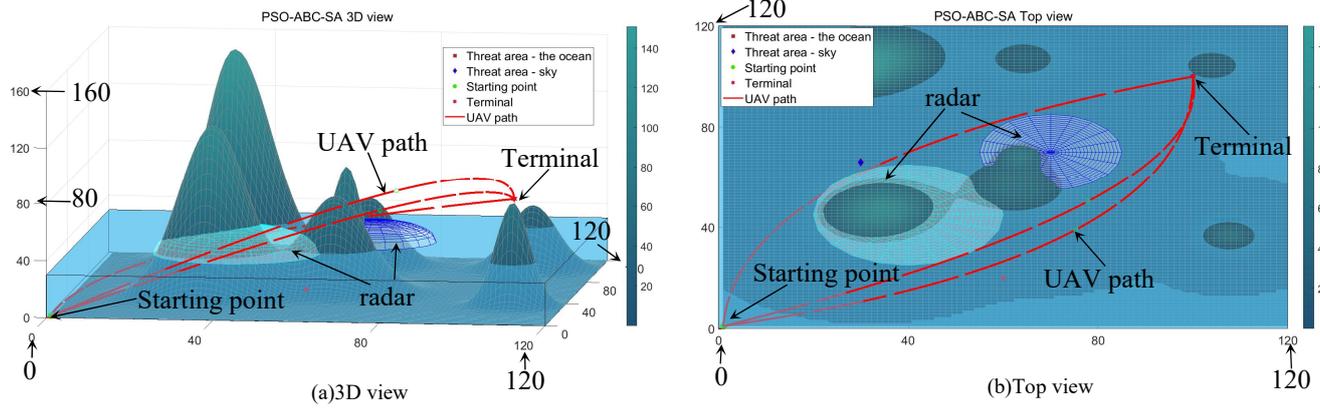


Fig. 8. PSO-ABC-SA

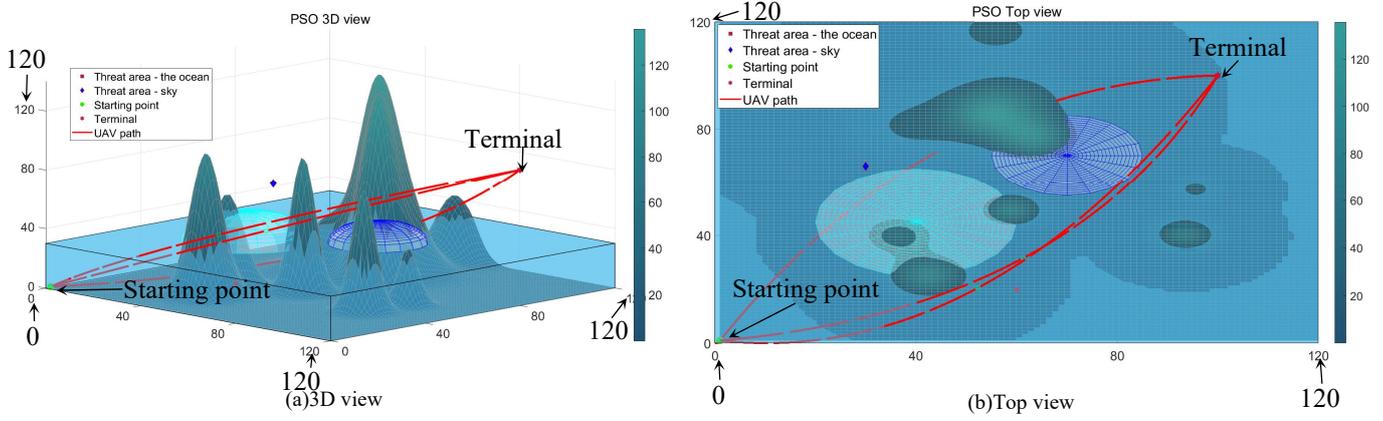


Fig. 9. PSO

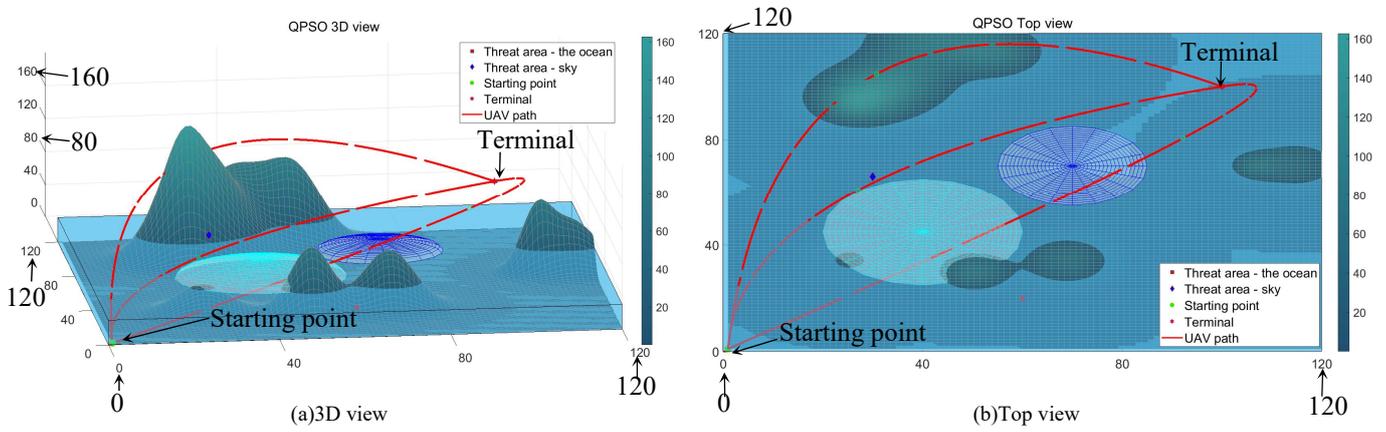


Fig. 10. QPSO

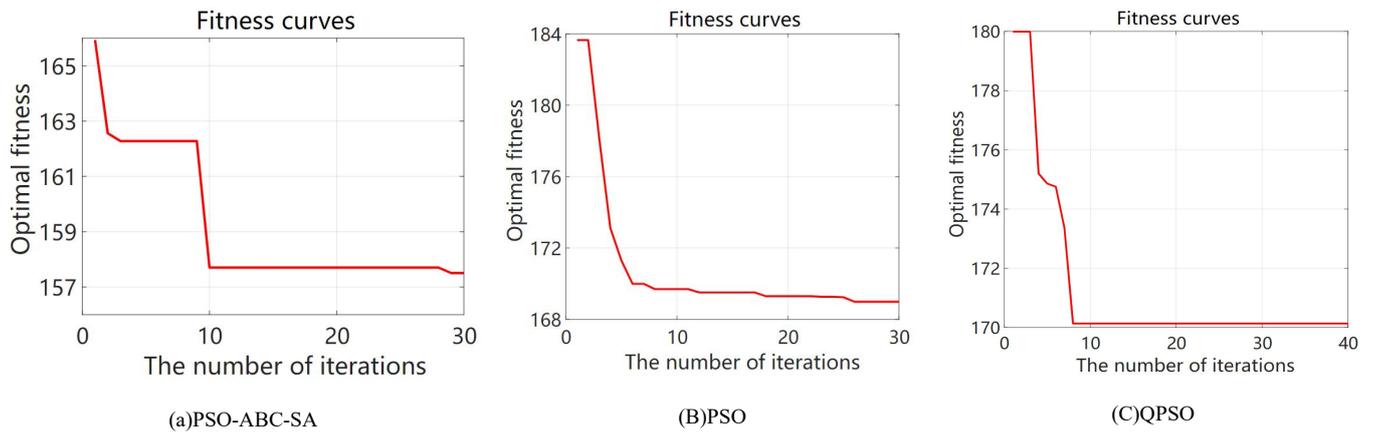


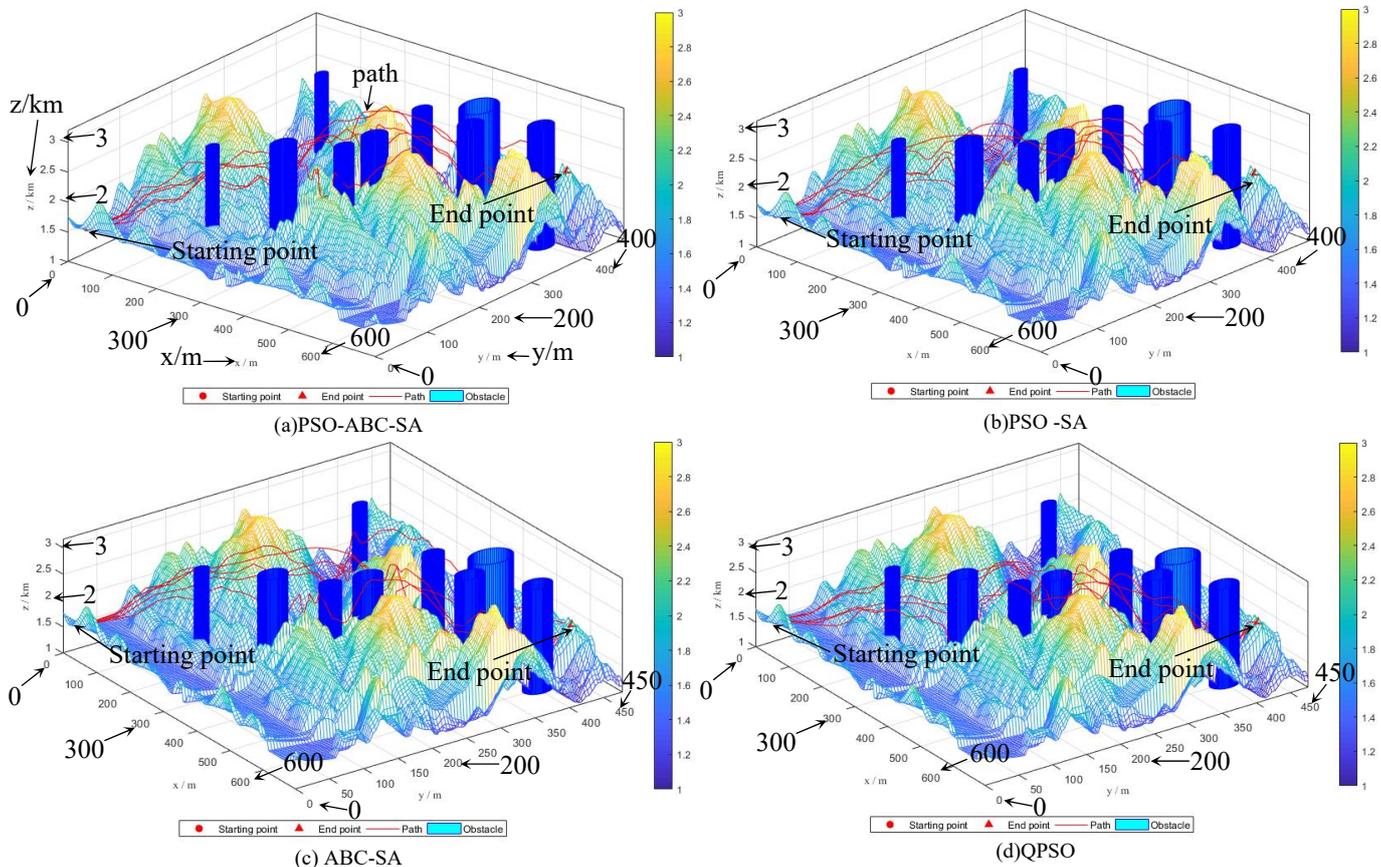
Fig. 11. Comparison of fitness curves

C. Safety Assessment of Obstacle Avoidance Paths

1) Comparison of Pitch Angles

In studying path planning with obstacle avoidance for UAV swarms, the perspective can be adjusted by varying the pitch angle to study how to avoid simulated mountains. In this analysis, the PSO-ABC-SA approach is compared with five existing algorithms for obstacle avoidance path planning in UAV applications. By controlling the pitch angle during the UAV flight using equations (7) and (8), the five obstacle avoidance paths planned by the PSO-ABC-SA algorithm show relatively small fluctuations, as shown in Fig. 12. This indicates that the algorithm has a low requirement for attitude adjustments during flight, which helps maintain the balance of the UAV's body. However, the paths planned using the PSO and PSO-SA algorithms have

issues with being too high relative to the ground, which is not beneficial for actual UAV flight, as shown in Figs. 12(b) and 12(e). In contrast, the paths planned using the AVOA algorithm are smoother and have smaller fluctuations. The UAV does not need to make frequent attitude adjustments during flight, as shown in Fig. 12(f). However, it is noteworthy that not all five paths in this algorithm successfully avoided obstacles, failing to achieve the expected results. Furthermore, the paths planned using the ABC-SA and QPSO algorithms generally have too large a fluctuation angle, as shown in Figs. 12(c) and 12(d). This may require the UAV to make more frequent and larger attitude adjustments during flight. Such significant pitch angle changes increase the complexity of UAV control and may affect flight stability and safety.



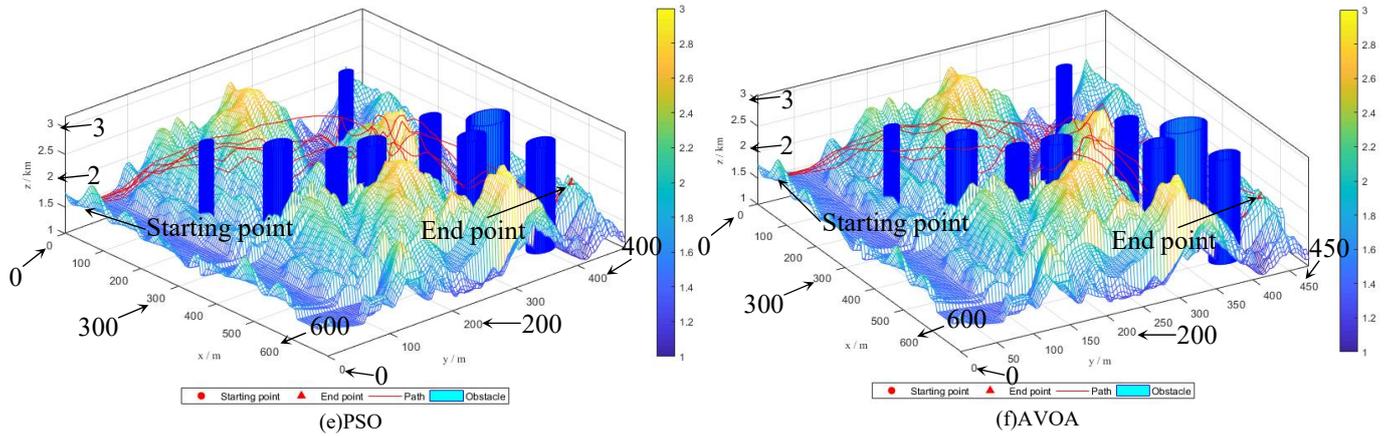


Fig. 12. Multi-algorithm pitch angle comparison

2) Comparison of Algorithmic Path Smoothness

The superiority of the PSO-ABC-SA algorithm in UAV obstacle avoidance path planning is evaluated from the perspective of path smoothness, as shown in Fig. 13. The results indicate that the five paths planned by the PSO-ABC-SA algorithm successfully avoid all obstacles (represented by blue circles) and exhibit smoothness. This characteristic significantly reduces the turning angles of the UAV during flight, minimizing its impact and effectively maintaining flight stability. Moreover, it reduces energy consumption and structural load due to large-angle turns. In contrast, path planning based on the PSO-SA and ABC-SA algorithms (shown in Figs. 13(b) and 13(c)) strives for both the shortest path and smoothness. However, some routes in these algorithms fail to avoid all obstacles, resulting in suboptimal experimental outcomes. Furthermore, as shown in Fig. 13(d), although this algorithm successfully avoids obstacles, it features intersecting paths, which may lead to potential collisions among UAVs. This makes it unsuitable for collective flight tasks, and the excessive turning angles do not meet the practical flight conditions. The algorithms presented in Figs. 13(e) and 13(f) also fail to achieve effective collective obstacle avoidance for UAV swarms. Additionally, they exhibit issues such as large turning angles, insufficient path smoothness, and intersecting routes. These shortcomings further highlight the superior performance of the PSO-ABC-SA algorithm to UAV path planning with obstacle avoidance.

D. Analysis of the Performance Advantages of the Algorithm

Under the same experimental conditions, 51 simulations were conducted for the PSO-SA, ABC-SA, QPSO, PSO, AVOA, and PSO-ABC-SA algorithms, as shown in Fig. 14. The results indicate that the PSO-ABC-SA algorithm achieved the optimal fitness value of 182. This is better than the PSO-SA algorithm, which had the worst performance with an optimal solution of 221.2. Statistical results in six areas (optimal cost, worst cost, average cost, average computation time, total flight distance, and flight time) are presented in Table 5. In terms of optimal cost, worst cost, and average cost, the PSO-ABC-SA algorithm outperformed the other five algorithms and exhibited higher convergence. However, in terms of average computation time, the PSO-ABC-SA algorithm was not optimal. This is due to

repeated calculations during position updates and fitness adjustments, leading to longer average times. Additionally, in optimal path length, the PSO-ABC-SA algorithm shortened the distance by 251 kilometers compared to the PSO algorithm, which had the worst performance. It also reduced the distance by 29 kilometers compared to the QPSO algorithm, which had the second worst performance, and shortened the average path by 130 kilometers. Regarding obstacle avoidance rate, the PSO-ABC-SA algorithm achieved 94.2%. This is an increase of 0.9% compared to the QPSO algorithm and an increase of 11.1% compared to the ABC-SA algorithm. The average obstacle avoidance rate improved by 5.7%. These results indicate that the PSO-ABC-SA algorithm has significant advantages in UAV intelligent obstacle avoidance and lowering power consumption.

VII. CONCLUSION

To address the issues of poor obstacle avoidance and low work efficiency in UAV power inspection in complex areas, this study proposes a new hybrid algorithm that integrates PSO, ABC, and SA algorithms. Based on the PSO-ABC-SA algorithm, a multi-UAV collaborative obstacle avoidance model with temporal and spatial coordination is introduced. The algorithm also considers constraints such as UAV flight height, turning range, and pitch angle. Experiments were conducted in simulated terrain to validate the superior performance of the PSO-ABC-SA algorithm in obstacle avoidance and inspection efficiency in complex mountainous environments and multi-domain environments. Additionally, the algorithm demonstrated effective path smoothing and spatial-temporal coordination in multi-UAV collaborative operations.

(1) By integrating the PSO, ABC, and SA algorithms, significant advantages have been demonstrated in multi-UAV collaborative intelligent obstacle avoidance, obstacle path smoothness processing, and improved inspection efficiency. This provides an efficient and reliable path planning strategy for power inspection tasks in complex terrains. The combination of the three algorithms effectively avoids local optimal solutions and enhances global search capabilities.

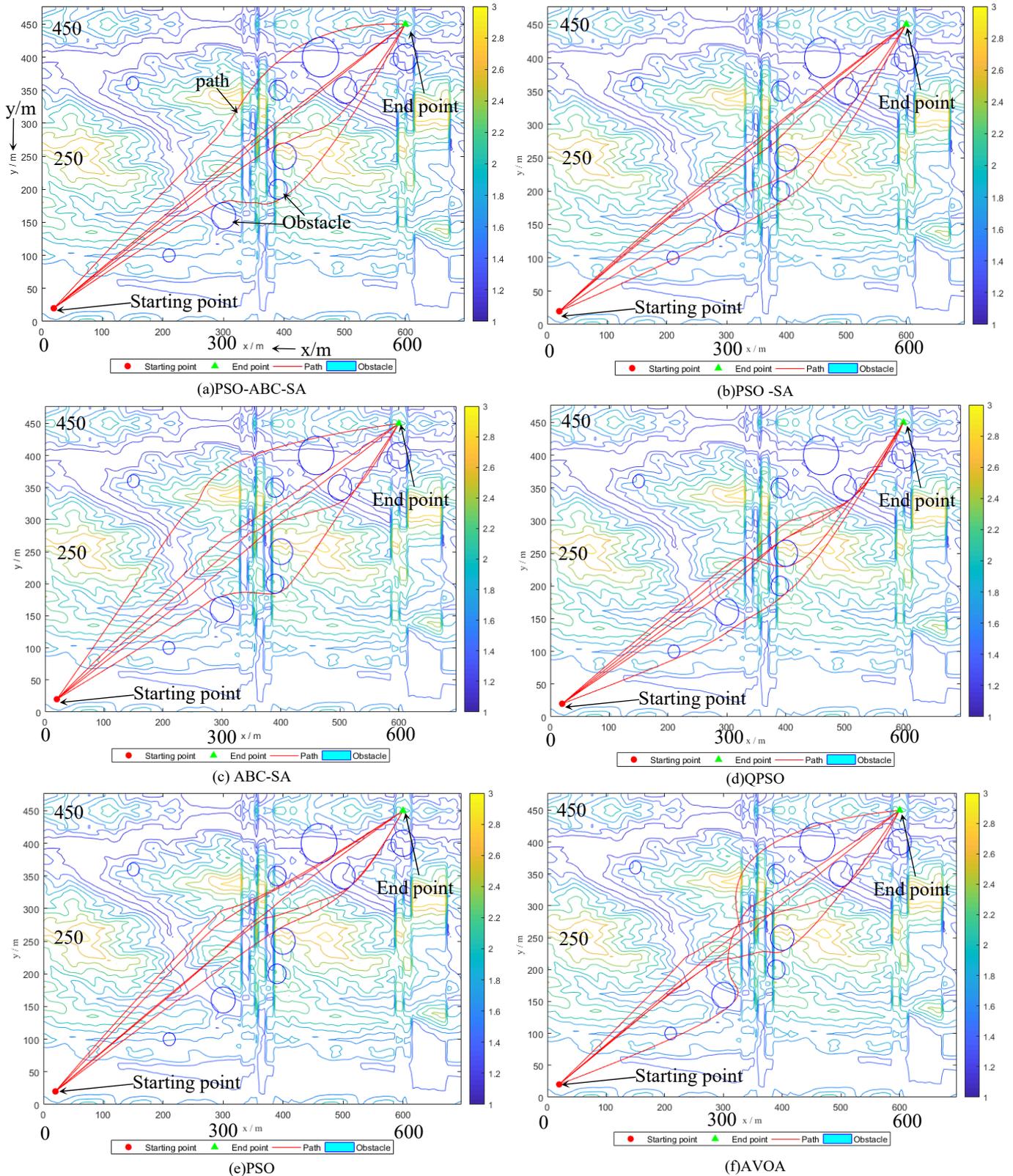


Fig. 13. Multi-algorithm path smoothness comparison

(2) This study used a two-dimensional grid method to conduct simulation experiments on simple and complex terrains. In both experiments, the PSO-ABC-SA algorithm outperformed other comparison algorithms in terms of obstacle avoidance performance, shortest path, and fitness metrics. In addition, comparative experiments conducted in a 3D simulation environment further validate the superior performance of the proposed algorithm in obstacle avoidance and smoothness processing. Extending the

application scenario to a multi-domain environment, three sets of comparative experiments further proved its excellent performance.

(3) This study evaluated the algorithm's performance across six aspects. Results showed that the PSO-ABC-SA algorithm shortened the average path by 130 km and achieved a 94.2% obstacle avoidance rate, improving by 5.7%. It also outperformed in optimal, worst, and average costs, confirming its reliability.

TABLE IV
ANALYSIS OF EXPERIMENTAL RESULTS

Algorithm	Optimal cost	Worst cost	Average cost	Average operation time	Optimal path length	Avoidance rate
PSO-SA	182.1	221.2	182.9	46.36s	1816km	85.9%
ABC-SA	189.2	201.4	190.9	41.31s	1703km	83.1%
QPSO	183.3	191.7	185.4	36.23s	1653km	93.3%
PSO	182.7	193.9	185.1	32.18s	1875km	89.7%
AVOA	188.5	209.8	189.1	32.87s	1723km	90.5%
PSO-ABC-SA	181.1	190.6	182	40.32s	1624km	94.2%

(4) Future work will aim to enhance the computational efficiency of the algorithm and exploring its adaptability to dynamic environmental changes. Additionally, we will integrate reinforcement learning algorithms to enhance obstacle recognition and detection capabilities. The goal is to achieve more efficient and intelligent multi-UAV collaborative obstacle avoidance path planning.

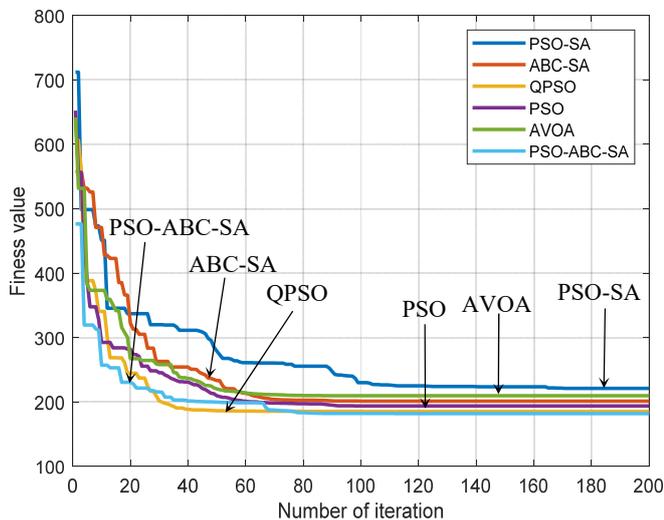


Fig. 14. Algorithm adaptation comparison

REFERENCES

- [1] D. Debnath, F. Vanegas, J. Sandino, A. F. Hawary, and F. Gonzalez, "A Review of UAV Path-Planning Algorithms and Obstacle Avoidance Methods for Remote Sensing Applications," *Remote Sensing*, vol. 16, no. 21, p. 4019, 2024.
- [2] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Real-Time UAV Path Planning," *IEEE Trans. Ind. Inf.*, vol. 9, no. 1, pp. 132–141, 2013.
- [3] R. A. Saeed, M. Omri, S. Abdel-Khalek, E. S. Ali, and M. F. Alotaibi, "Optimal path planning for drones based on swarm intelligence algorithm," *Neural Comput. & Applic.*, vol. 34, no. 12, pp. 10133–10155, 2022.
- [4] A. F. U. Din, I. Mir, F. Gul, M. R. Al Nasar, and L. Abualigah, "Reinforced Learning-Based Robust Control Design for Unmanned Aerial Vehicle," *Arab J Sci Eng*, vol. 48, no. 2, pp. 1221–1236, 2023.
- [5] G. Ahmed, T. Sheltami, and A. Mahmoud, "Energy-Efficient Multi-UAV Multi-Region Coverage Path Planning Approach," *Arab J Sci Eng*, vol. 49, no. 9, pp. 13185–13202, 2024.
- [6] F. Yang et al., "Obstacle Avoidance Path Planning for UAV Based on Improved RRT Algorithm," *Discrete Dynamics in Nature and Society*, vol. 2022, no. 1, p. 4544499, 2022.
- [7] X. Meng, X. Zhu, and J. Zhao, "Obstacle Avoidance Path Planning Using the Elite Ant Colony Algorithm for Parameter Optimization of Unmanned Aerial Vehicles," *Arab J Sci Eng*, vol. 48, no. 2, pp. 2261–2275, 2023.
- [8] S. J. Fusic and R. Sitharthan, "Improved RRT* Algorithm-Based Path Planning for Unmanned Aerial Vehicle in a 3D Metropolitan Environment," *Un. Sys.*, vol. 12, no. 05, pp. 859–875, 2024.
- [9] E. Aldao, L. González-deSantos, H. Michinel, and H. González-Jorge, "UAV Obstacle Avoidance Algorithm to Navigate in Dynamic Building Environments," *Drones*, vol. 6, no. 1, p. 16, 2022.
- [10] Z. Xu, J. Hu, Y. Ma, M. Wang, and C. Zhao, "A Study on Path Planning Algorithms of UAV Collision Avoidance," *JNWPU*, vol. 37, no. 1, pp. 100–106, 2019.
- [11] X. Wang, V. Yadav, and S. N. Balakrishnan, "Cooperative UAV Formation Flying With Obstacle/Collision Avoidance," *IEEE Trans. Contr. Syst. Technol.*, vol. 15, no. 4, pp. 672–679, 2007.
- [12] M. Yan, C. Aun Chan, A. F. Gyax, C. Li, A. Nirmalathas, and I. Chih-Lin, "Efficient Generation of Optimal UAV Trajectories With Uncertain Obstacle Avoidance in MEC Networks," *IEEE Internet Things J.*, vol. 11, no. 23, pp. 38380–38392, 2024.
- [13] G.-T. Tu and J.-G. Juang, "UAV Path Planning and Obstacle Avoidance Based on Reinforcement Learning in 3D Environments," *Actuators*, vol. 12, no. 2, p. 57, 2023.
- [14] Y. Lin, Z. Na, Z. Feng, B. Lin, and Y. Lin, "Dual-game based UAV swarm obstacle avoidance algorithm in multi-narrow type obstacle scenarios," *EURASIP J. Adv. Signal Process.*, vol. 2023, no. 1, p. 118, 2023.
- [15] Z-F Hu, X-Y Wang, and Junhao Zhang, "Research on Dynamic Obstacle Avoidance Method of Mobile Robot based on Improved A* Algorithm and Dynamic Window Method," *Engineering Letters*, vol. 32, no. 3, pp.520-530, 2024
- [16] Q. Meng, K. Chen, and Q. Qu, "PPSwarm: Multi-UAV Path Planning Based on Hybrid PSO in Complex Scenarios," *Drones*, vol. 8, no. 5, p. 192, 2024.
- [17] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometrics and Intelligent Laboratory Systems*, vol. 149, pp. 153–165, 2015.
- [18] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," *J Glob Optim*, vol. 39, no. 3, pp. 459–471, 2007.
- [19] M. Abid, S. El Kafhali, A. Amzil, and M. Hanini, "Optimization of UAV Flight Paths in Multi-UAV Networks for Efficient Data Collection," *Arab J Sci Eng*, 2024.
- [20] D. Delahaye, S. Chaimatanan, and M. Mongeau, "Simulated annealing: From basics to applications," *Handbook of Metaheuristics*, pp. 1–35, 2019.
- [21] D. Bertsimas and J. Tsitsiklis, "Simulated annealing," *Stat. Sci.*, vol. 8, no. 1, pp. 10–15, 1993.
- [22] C. Huang, J. Fei, and W. Deng, "A Novel Route Planning Method of Fixed-Wing Unmanned Aerial Vehicle Based on Improved QPSO," *IEEE Access*, vol. 8, pp. 65071–65084, 2020.
- [23] M. Haris, D. M. S. Bhatti, and H. Nam, "A Fast-Convergent Hyperbolic Tangent PSO Algorithm for UAVs Path Planning," *IEEE Open Journal of Vehicular Technology*, vol. 5, pp. 681–694, 2024.
- [24] Xin Yao, Yong Liu, and Guangming Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, 1999.
- [25] P. N. Suganthan, N. Hansen, J. J. Liang, and K. Deb, "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization". *KanGAL Report*, 2005.
- [26] C. Masafu, R. Williams, X. Shi, Q. Yuan, and M. Trigg, "Unpiloted Aerial Vehicle (UAV) image velocimetry for validation of two-dimensional hydraulic model simulations," *Journal of Hydrology*, vol. 612, p. 128217, 2022.
- [27] H. Garg, "A hybrid PSO-GA algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, pp.

292–305, 2016.

- [28] A. Gupta, A. Trivedi, and B. Prasad, "B-GWO based multi-UAV deployment and power allocation in NOMA assisted wireless networks," *Wireless Netw.*, vol. 28, no. 7, pp. 3199–3211, 2022.
- [29] M. A. Awadallah, M. A. Al-Betar, I. A. Doush, S. N. Makhadmeh, and G. Al-Naymat, "Recent Versions and Applications of Sparrow Search Algorithm," *Arch Computat Methods Eng*, vol. 30, no. 5, pp. 2831–2858, 2023.
- [30] Y-B Wang, J-S Wang, and X-F Sui, "Improved Particle Swarm Optimization Algorithm with Logistic Function and Trigonometric Function for Three-dimensional Path Planning Problems," *Engineering Letters*, vol. 33, no. 2, pp.442-459, 2025