IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 11 July 2023, accepted 8 August 2023, date of publication 14 August 2023, date of current version 23 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3304708

RESEARCH ARTICLE

UAV Flight Path Planning Based on Multi-Strategy Improved White Sharks Optimization

RAN ZHANG, XINGDA LI, HONGHONG REN, YUANMING DING, YIFEI MENG, AND QINGYU XIA

School of Information Engineering, Dalian University, Dalian 116622, China
Communication and Network Laboratory, Dalian University, Dalian 116622, China

Corresponding author: Ran Zhang (nancy444@163.com)

ABSTRACT Due to the development of its own technology, the Unmanned Aerial Vehicle (UAV) play an increasingly important role in today's social production practice. The complex and changeable environment requires the development of innovative UAV path planning algorithms. In order to meet the requirements of the increasingly complex UAV flight environment, a new UAV flight path planning algorithm based on a version of the White Sharks Optimization (WSO) is proposed in this research. Firstly, the terrain matrix is used to establish the three-dimensional terrain environment and constraint function, and then WSO is improved for handling the path planning. In the process of path planning, multi-trajectory search, nonlinear convergence factor and the model of fish movement behavior are adopted to enrich the population diversity, excavate the search space, speed up the convergence and reduce the likelihood of falling into local optima. Based on the simulation results, it can be observed that the proposed algorithm outperforms in terms of optimization accuracy, convergence speed, and robustness, leading to improved outcomes in UAV flight path planning.

INDEX TERMS Fight path planning, unmanned aerial vehicle, white sharks optimization, multi-trajectory search, nonlinear convergence factor.

I. INTRODUCTION

The Unmanned Aerial Vehicle (UAV) can perform complex and dangerous tasks, such as intelligence collection, reconnaissance, and surveillance, through remote controls without pilots in some extreme cases [1]. It plays a crucial role in modern military wars [2] in particular, and its application field is still expanding rapidly [3]. Nevertheless, as non-contact combat continues to evolve, UAVs frequently encounter a multitude of threats including ground radar, jamming equipment, anti-UAV missiles, and enemy reconnaissance aircraft [4]. It is still difficult to fully realize autonomous driving. Therefore, it is urgent to develop an UAV flight path planning algorithm that can adapt to more combat scenarios [4].

UAV path planning involves global and local components. Global planning uses all available information to determine an optimal path [5], while local planning adjusts the flight direction based on current constraints and dynamic obstacles, enhancing obstacle avoidance capabilities [6]. The research in this paper is global static path planning for UAV. Commonly employed algorithms encompass the graph search algorithm [7] and mathematical methods [8], intelligent optimization algorithm [9], and reinforcement learning [10].

The swarm intelligence algorithm, as a multi-agent and gradient-free optimization technique [11], has demonstrated good performance in many complex and coupled engineering problems. At present, many scholars have been studying the use of the swarm intelligence algorithms to improve the path planning performance of UAVs. The traditional intelligent optimization algorithm includes the Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Fish Swarm Optimization (FSO), Bacterial Foraging Optimization Algorithm (BFOA) and other improved strategies based on these algorithms [12], [13], [14], [15], [16]. Although these traditional algorithms are effective for simple problems,

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

as the complexity of real-world problems increases, there is a need for more powerful algorithms to handle them.

Maintaining a balance between global exploration and local exploitation is crucial for the performance of swarm intelligence algorithms. In recent years, researchers in the field have been exploring various approaches to enhance the algorithms. Wu and Tan [17] to enhance the exploration and exploitation capabilities while maintaining a satisfactory convergence rate, combines the hierarchical approach of Grey Wolf Optimization (GWO) with the greedy strategy of the Differential Evolution Algorithm (DE) within the framework of the Whale Optimization Algorithm (WOA). Jiaqi et al. [18] take the whale algorithm as a reference, introduced spiral update, modified parameters and added an adaptive leadership mechanism, which enhanced the overall performance of the algorithm and solved the problems of slow convergence of path planning and insufficient flight path. Zhou et al. [19] integrated the improved BA algorithm into the ABC algorithm based on the features of the standard bat algorithm (BA) and artificial bee colony algorithm (ABC). IBA uses ABC to modify BA, which overcomes the problem of poor local search ability of BA. Zhang et al. [20] proposed an improved Harris Hawke optimization (HHO) algorithm. The algorithm combines the Cauchy mutation strategy and adaptive weighting, and uses the oscillation characteristics of the sine-cosine algorithm (SCA) to gradually converge to the global optimum. This approach can combine the characteristics and advantages of different algorithms, but it is difficult to overcome the inherent limitations of the algorithms themselves.

The use of mathematical models or mathematical theories to improve or modify algorithms provides a new and solid and effective means of improvement for the improvement of algorithms. Ji et al. [21] proposed a new learning particle swarm optimization (DDBLPSO) algorithm based on dual dynamic biogeography, which can avoid premature convergence and maximize the chance of finding the global optimal. Wan et al. [22] proposed an Elite-Guided Orthogonal Krill Swarm (EODKH) algorithm that combines the benefits of elite-guided evolution with traditional krill swarming methods to solve the path planning problem of UAVs in complex terrain three-dimensional space. Zhang et al. [23] proposed the QFOA algorithm based on the quantum theory. QFOA introduces a search mechanism based on the quantum behavior in FOA. By leveraging the principles of probability and uncertainty from quantum theory, the proposed algorithm effectively addresses the issues of premature convergence and susceptibility to local optima that are commonly encountered in the Firefly Optimization Algorithm (FOA). Tang et al. [24] proposed an improved ant colony optimization (ACO) considering path security. Firstly, Tyson polygons are formulated according to the position of the high-altitude mountain range to obtain a feasible UAV path under the flight conditions. Secondly, dense peak areas are avoided by framing path safety constraints. Thirdly, the ACO algorithm is used to search for the shortest path. Finally, the path is smoothened to produce the best safe path that can be used for actual flight. Mathematical methods can indeed be effective in improving algorithms, but they can sometimes encounter theoretical bottlenecks.

In recent years, some scholars have tried to combine swarm intelligence algorithms and neural networks to solve the problem of UAV path planning in dynamic environments. Liu et al. [25] introduced a UAV path planning method that combines the Sparrow Search Algorithm (SSA) with a Bionic Neural Network (BINN). This integration of SSA and BINN enables efficient and adaptive UAV path planning, considering both static and dynamic environmental factors. Although various swarm intelligence algorithms are constantly emerging, no single algorithm could adapt to all the situations. The above algorithms have different types of improvements in efficiency, speed, etc. But they cannot be well balanced, and cannot adapt to a wide range of environments having complex and changeable obstacles. The MSWSO algorithm proposed in this paper adopts the method of disorderly exploration and then orderly development, taking into account its efficiency and speed, so as to adapt to a more complex and large-scale environment. However, the universality and robustness of this approach still need to be validated.

Based on the advantages and limitations of the aforementioned algorithms, a method has emerged that utilizes multiple strategies to address different problems. Chai et al. [26] proposed a multi-strategy fusion differential evolution (MSFDE) algorithm. The algorithm combines multiple population strategies, adaptive strategies, and interactive mutation strategies to strike a balance between development and exploration capabilities. Hu et al. [27] proposed an improved version of the QPIO algorithm. Firstly, the initial population is logically mapped, and then the sub-parameters are adaptively up-dated in each iteration. In the milestone operations, an updated strategy of gradually reducing the number of pigeons was introduced to prevent premature convergence and local optima. Qian and Lei [28] proposed a jointly improved adaptive cuttlefish algorithm that combines chaotic perturbation and mutation learning into an adaptive weighting mechanism. In addition, an automatic filtering mechanism based on individual fitness is used to adjust population diversity and eliminate local optimum, promoting robustness optimization performance. Using a variety of strategies, certain improvements are made at different stages of the algorithm, and the algorithm is reconstructed to a certain extent, which can improve the performance of the algorithm to a high extent.

WSO has an advantage in handling global optimization problems as it has the flexibility to deal with many different types of problems. The mathematical model is suitable for solving various engineering optimization problems, especially high dimensional optimization problems. Its simplicity and robustness enable it to find global optimum solutions to difficult optimization problems quickly and accurately, and it has a high convergence rate also. Although the White Sharks Optimization (WSO) algorithm has shown promise in terms of its low development cost and its ability to find effective solutions for real-world optimization problems, it still faces certain challenges in its basic form. Some of the common issues include relying on a single search method, limited exploration of the search space, and a tendency to converge on local optima during optimization. These shortcomings highlight the need for further improvements and enhancements to enhance the algorithm's performance and overcome these limitations.

To further enhance the quality and efficiency of UAV flight path planning, a flight path planning algorithm (MSWSO) based on the multi-strategy improvement of WSO [29] is proposed. Building upon the traditional WSO algorithm, the MSWSO algorithm incorporates a multi-trajectory search strategy to explore the planning space effectively. Additionally, a nonlinear convergence factor is introduced to guide the entire algorithm development process. The inclusion of a fish movement behavior model enhances population diversity during later stages of the optimization. Furthermore, to ensure the algorithm's versatility across different terrains, two sets of experiments are conducted, involving eight diverse complex terrains along with their associated threat conditions.

The article is structured as follows. Section II provides an introduction to the terrain environment model and the constraints based on UAV flight conditions for path planning. Section III presents an overview of the basic WSO algorithm, its enhancements, and the application of the improved WSO algorithm in UAV path planning. Section IV discusses the testing and experimental results of the proposed algorithm. Finally, Section V concludes the article by summarizing its key findings and contributions.

II. MODELING AND CONSTRAINTS

A. MODELING

This paper focuses on the study of UAV trajectory planning in a three-dimensional environment. The mountain altitude matrix is used for modeling a terrain, where the two-dimensional matrix is visualized in three-dimension with the Z-axis to control the bump of the terrain. The real mountain surface is steep and often presents nonlinear changes. Therefore, the matrix is resampled by using nonlinear interpolation to smooth the terrain model. Some threat areas and obstacles are also presented through a matrix simulation.

Figure 1 shows the simulated terrain environment, where the randomly generated complex terrain is on the surface, and the red hemispherical and cylindrical models rep-resent the threat area of the radar.

In general, a UAV has a certain volume. If simply the center point of a drone is used to calculate its body distance, the edge of the drone may hit the mountain. Therefore, it is necessary to check whether the distance between the fuselage of the drone and the mountain will cause a collision. Therefore, the drone model is reduced to a sphere, as shown in Fig. 2. The

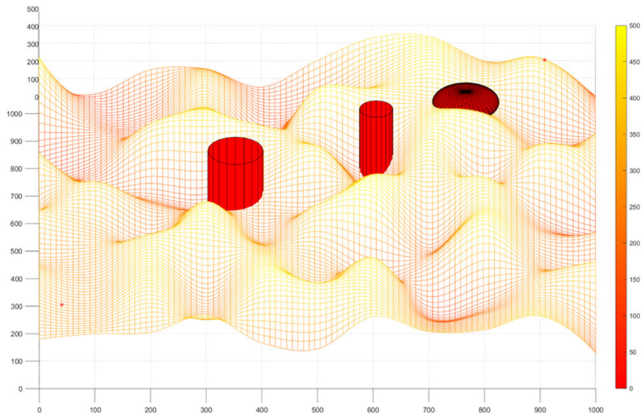


FIGURE 1. UAV flight terrain.

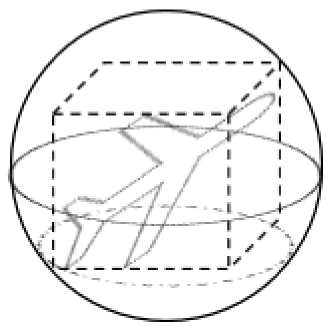


FIGURE 2. UAV flight model.

requirement for the safe transportation of a drone is that all the obstacles can be avoided during flight. In order to ensure the flight safety, the conditions in Eq. (1) should be met.

Where, $Z(x_{m,}y_m)$ is the height of a single mountain, $U(x, y, z)$ is a point on UAV, and D(x,y,z) is a point on the threat area.

$$\begin{cases} U(x, y, z) > Z(x_{m,}y_m) \\ U(x, y, z) > D(x, y, z) \end{cases} \tag{1}$$

B. CONSTRAINTS

To evaluate the quality of UAV flight paths, it is essential to construct evaluation metrics that specifically assess factors such as height, length, and smoothness of the flight path. These metrics will provide a comprehensive assessment of the flight path's quality and help in analyzing and comparing different trajectory planning algorithms. Equation (2) gives the distance between UAV and the planned two points.

$$l_i = ||obj_{ij}obj_{i,j+1}|| \tag{2}$$

where obj_{ij} and $obj_{i,j+1}$ are the two planned nodes, and l_i is the planned shortest path.

In addition, the flight height of UAV should also be considered to be out of the range of the wireless control signals. The easily exposed problems caused by the flight height of UAV exceeding a limiting value, as well as the possible collision of UAV with ground objects such as surface plants and stones,

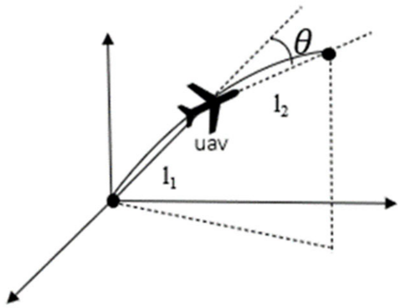


FIGURE 3. UAV flight angle.

are too low. Therefore, the flight height of UAV should also meet the condition in Eq. (3).

$$h_{\min} \leq h_i \leq h_{\max} \tag{3}$$

where, $h_{\min}$ is the minimum flight height of UAV, $h_{\max}$ is its maximum flying height.

In order to ensure the smooth flight of the UAV and reduce the energy consumption, UAV should keep smooth flight as far as possible. The flight angle is shown in Fig. 3, and the smoothness of the path is expressed by Eq (4).

l_1 and l_2 are two adjacent displacements of UAV, θ is the flight pitch angle or turning angle obtained through Eq. (4), ϕ is the set of these angles throughout the entire flight process, and the magnitude of the ϕ reflects the smoothness of the path throughout the entire flight process.

$$\phi_i = \frac{l_{\mathrm{i}} \cdot l_{i+1}}{||l_i||||l_{i+1}||} \tag{4}$$

where ϕ_i is the smoothness of the path between the planned nodes.

By combining Eqs. (1) to (4), the objective function for UAV flight path planning can be expressed as Eq. (5), taking into consideration the specific requirements of the task.

$$F = \sum_{i=1}^{\mathrm{N}} (\omega_1 l_i + \omega_2 h_i + \omega_3 \phi_i) \tag{5}$$

where ω_1, ω_2 and ω_3 are the weight parameters obtained experimentally.

III. UAV FLIGHT PATH PLANNING BASED ON MSWSO

A. BASIC WHITE SHARKS OPTIMIZATION

WSO is inspired by the auditory and olfactory hunting behavior of white sharks in the ocean, specifically targeting prey on the sea floor. In WSO, the search agents, resembling the white sharks, update their positions randomly based on the best solution found so far, aiming to reach the optimal solution. The algorithm consists of three distinct phases: global exploration, local exploitation, and fish behavior. These phases ensure a balance between exploration and exploitation, taking into account the intensity of both auditory and olfactory cues in the search process.

1) GLOBAL EXPLORATION

The movements of prey create waves. The great whites use their organs to sense these changes and locate their prey. In this case, the white shark uses its associated sense of hearing and smell, and undulate wave motion to navigate to prey, as expressed by Eq (6).

$$w_{k+1}^i = \begin{cases} w_k^i \cdot \neg \oplus w_o + u \cdot a + l \cdot b; rand < mv \\ w_k^i + v_k^i / f; rand \geq mv \end{cases} \tag{6}$$

where, $\neg$ is a negative number, a and b, are one-dimensional binary vectors-, l and u represent the limits of the solution space, wo represents a logical vector, f indicates the wave movement frequency of the white shark, $rand$ is a random number created in the range from 0 to 1, and v_k^i is the speed at which the individual white shark moves with the wave.

2) LOCAL EXPLOITATION

The great white shark updates its position according to the best white shark. So, it can occupy the best position to stay close to its prey. This behavior is expressed by Eq (7).

$$w_{k+1}^{\prime i} = w_{gbest_k} + r_1 \vec{D}_w sgn(r_2 - 0.5) r_3 < S_s \tag{7}$$

where w_k^i is the updated position of the white shark relative to its prey, w_{gbest_k} is the global optimal individual of the kth iteration, sgn(r2-0.5) gives 1/-1 to change the search direction, is the distance between the prey and the white shark r_1, and, r_2; and r_3 are random numbers in the range of [1, 0].

3) FISH BEHAVIOR

The behavior of the population of white sharks is mathematically modeled in such a way that the two best solutions are preserved, while the positions of the remaining white sharks are updated based on these two best solutions. Equation (8) is used to define the behavior of the fish school of white sharks.

$$w_{k+1}^i = \frac{w_k^i + w_{k+1}^{\prime i}}{2 \times rand} \tag{8}$$

where w_k^i and $w_{k+1}^{\prime i}$ are the first two optimal solutions for retaining, and rand represents a random number evenly distributed within the interval of [1, 0].

Equation (8) describes how a white shark updates its position by referencing the best position in proximity to its prey. Consequently, the final location of the white shark will closely align with the best prey found in the search space. This behavior, coupled with the movement of white sharks towards the best individual, exemplifies the collective behavior observed in WSO. Such behavior expands the exploration and exploitation capabilities of the algorithm.

B. MULTI-STRATEGY IMPROVED WSO

The original WSO algorithm employs random search and wave-trajectory based search. However, it suffers from limitations such as a single search method, insufficient search space, susceptibility to local optima, and low convergence accuracy. To address these issues, this study introduces a

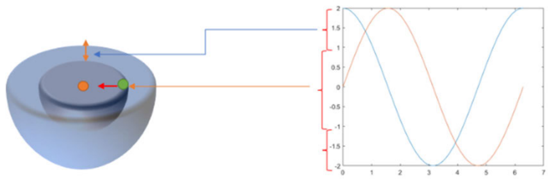


FIGURE 4. Sine and cosine model.

multi-trajectory search strategy by combining the sine and cosine search algorithms with a variable spiral search strategy. This strategy effectively expands the search space and enhances search efficiency. Furthermore, a nonlinear convergence factor is introduced to govern the entire iterative process of the algorithm. In addition, fish movement behavior rules are incorporated to increase population diversity.

1) MULITY-TRAJECTORY SEARCH

SCA adopts the oscillation and periodicity of the sine and cosine functions, and alters its position with the trajectory of the wavy curve. The spiral search updates the position in the form of a thread in the space, through which the solution space can be developed quickly and effectively.

a: SINE AND COSINE ALGORITHM

The Sine Cosine Algorithm (SCA) utilizes the mathematical properties of sine and cosine functions to dynamically adjust their amplitudes. By striking a balance between global exploration and local exploitation, the algorithm effectively adapts its search capabilities during the optimization process. After exploring around one solution, the agent can run around another solution on the next cycle, making full use of the space defined between the two functions. SCA requires its search agents to change dramatically during the initial phase of optimization and slightly at the end of optimization. This search method enables the agent to search the solution space adequately and develop reliably near the location with better fitness. SCA uses Eq. (9) to update the location as shown in Fig. 4.

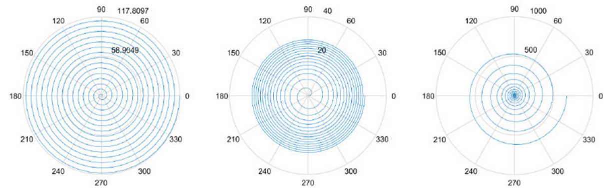
$$x_{t+1}^i = \begin{cases} x_t^i + r_1 \cdot \sin(r_2) \cdot |r_3 p_t - x_t|, & r_4 < 0.5 \\ x_t^i + r_1 \cdot \cos(r_2) \cdot |r_3 p_t - x_t|, & r_4 \geq 0.5 \end{cases} \tag{9}$$

where, x_t^i is the position of the individual at the t_{th} iteration; represents the position of the individual with the highest fitness at the t_{th} iteration; r_2, r_3, and r_4 are random numbers, and r_1 is a function of the number of iterations.

b: SPIRAL SEARCH MODEL

There are three commonly used spiral searches: Archimedes spiral, Fermat spiral, and logarithmic spiral as shown in Fig.5.

Among the three spiral search models mentioned above, the logarithmic helix exhibits a wide exploration range in the initial stage and gradually transitions to a localized search in the later stage. This behavior closely resembles the search process observed in swarm intelligence algorithms. Hence,



(a)Archimedes spiral (b)Fermat spiral (c)logarithm spiral

FIGURE 5. Spiral search model.

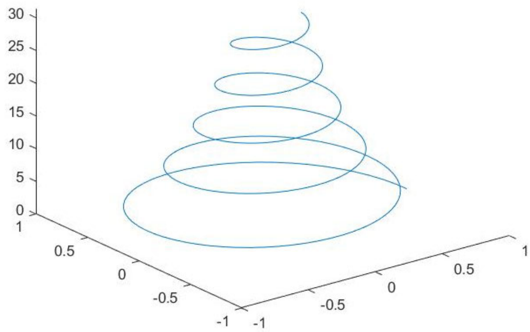


FIGURE 6. Logarithmic spiral search model.

the logarithmic spiral is adopted in this research. Equation (10) gives the spiral search formula. Figure 6 shows the simulation of the logarithmic spiral search in a three-dimensional space.

$$\mathrm{P} = e^{bl} * \cos(2\pi l) \tag{10}$$

where P is the exploration factor, b is the logarithmic spiral shape constant, and l is the exploration step size.

The span between the successive turns of the logarithmic helix increases geometrically. The longer the distance from the center, the larger is the span. As shown in Figure 6, the individuals in each generation progressively converge towards a spiral pattern during the position update process. This behavior enables the algorithm to explore the neighboring search space, preserve population diversity, and enhance its overall exploration capability. However, there might be insufficient search space in the later stages of the search. So, it is necessary to combine other methods. In this research, nonlinear convergence factor is adopted.

2) NONLINEAR CONVERGENCE FACTOR

Nonlinearity, as opposed to linearity, is more closely aligned with the intrinsic nature of objective phenomena. It represents an important category for quantitative research and the comprehension of complex knowledge. By introducing the nonlinear excitation function tahn in the neural network, along with its derivative and some modifications, the nonlinear con-vergence factor used in this paper is obtained. The improved nonlinear convergence factor is expressed by Eq. (11). Figure 7 shows how the nonlinear convergence factor α

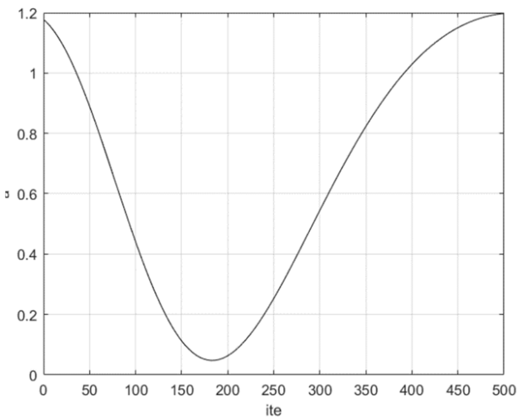


FIGURE 7. Nonlinear convergent silver changes with iteration.

changes throughout the iteration.

$$\alpha = \omega - \frac{\sigma}{\frac{t}{T_{\max}} + e^{(-\lambda \cdot \frac{t}{T_{\max}})^2}} - \varphi \tag{11}$$

where, ω, σ, λ; and φ are three control parameters of the convergence factor, T_{max} is the maximum number of iterations, t is the iteration counter, and is an experimental parameter. The early value of the algorithm is large, which makes it to pay more attention to the search for the global optimum. However, in the iterative algorithm, the value gradually decreases, which is convenient for fine search in the middle and late stages, thus improving the convergence accuracy of the algorithm. In the later iterations of the algorithm, this value gradually increases to jump out of the local optimum.

Overall, in this research, the cyclic and oscillatory properties of WSO are combined with the ability to develop a three-dimensional space of the logarithmic spiral strategy and the nonlinear factor that controls the search strategy in the iterative process. This al-lows WSO to fully explore the entire search space during the global search phase and reliably develop and leverage near the best individuals. The convergence speed of the algorithm is also improved, which can effectively help it to jump out of any local optimum. The improved search is mathematically expressed by Eqs. (12) and (13).

$$w_{k+1}^{i} = \alpha \cdot w_{k}^{i} \cdot \neg \oplus w_{o} + u \cdot a + l \cdot b;\ rand < mv \tag{12}$$

$$w_{k+1}^{i} = \begin{cases} \alpha \cdot e^{bl} \cdot \sin(2\pi l) \cdot w_{k}^{i};\ rand{<}0.5 \\ \alpha \cdot e^{bl} \cdot \cos(2\pi l) \cdot w_{k}^{i};\ rand \geq 0.5 \end{cases} \quad rand \geq mv \tag{13}$$

3) RULES OF MOVEMENT BEHAVIOR OF FISH

WSO believes that the optimal solution found by the population of individuals is not necessarily the location of prey, but may also be the smell left by prey. Therefore, in this research the movement behavior of fish is further used to change the location of the optimal solution, i.e., the location where from the prey may escape to avoid white sharks. The position after the initialization of the population is considered to be the position of the shark that has sensed the prey. At each time, the updated position is the position of the prey that has been sensed earlier by the shark. The movement of the fish is simulated ac-cording to these two positions, i.e., the optimal solution is updated. The symbolic description of the rules of movement behavior of fish is presented in Table 1.

TABLE 1. Symbolic description of the rules of fish movement behavior.

Symbol	Description of Symbol
A_i	An individual fish in a school
P_{t+1}	The direction of movement of a single fish during the next time step cycle
P_{nt}	The direction in which an individual fish can escape in the fastest way to avoid a threat

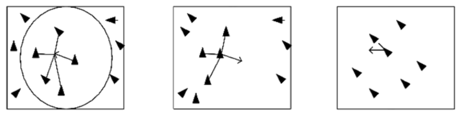


FIGURE 8. Motion behavior of a school of fish.

The rules of sports are described below:

(1) Inertia rule: When a fish gets a signal requiring to change its motion, the swimming direction of the fish cannot be changed immediately; due to the effect of its inertia.

(2) Proximity rule: For not leaving the fish, it is necessary to get as close to the center of the neighbors as possible, as shown in Fig. 8(a).

(3) Alignment rule: In order to maintain the coherence of the fish's movement, each fish should try to move in the same direction as its neighbors, as shown in Fig. 7(b).

(4) Avoiding rule. In order to maintain the consistency of the fish movement, collisions of individual fish should be avoided as much as possible, as shown in Fig. 8 (c).

The problem is simplified according to the path planning model, where the prey will change its position only when there is a threat (will not actively forage and move), and there will be no neighbor (i.e., no individual collision, and no population center), as shown in Fig. 9.

After simplifying the problem, let. The movement direction of individual fish at the next moment is expressed by Eq. (14).

$$p_{5t} = \arctan \frac{x_0 - x_s}{y_0 - y_s} \tag{14}$$

where (x_s, y_s) is the current position of the shark, i.e., the optimal position during the population initialization, and (x_0, y_0) is the current position of individual fish.

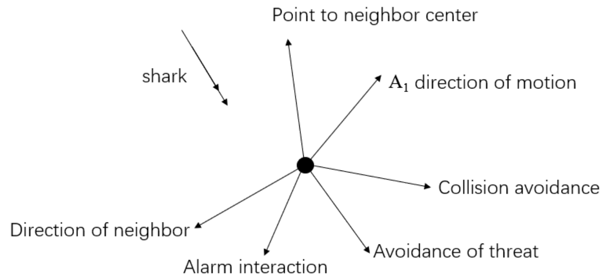


FIGURE 9. **Motion model of fish.**

FIGURE 10. **Simple fish swarm prediction model.**

Algorithm 1 MSWSO

Inputs: Population size N and iterations, T, and algorithm parameters
Outputs: The position and fitness value of the solved individual
Initialize the population X_i (i=1; 2;...; N)
While(t<T) // Loop T times
Calculate the initial fitness value for each individual //Loop T times
Calculate and update parameters *mv, Ss, alpha* //Loop T times
for (each White sharks (X_i)) **do** //Loop N times
  if(rand< mv) **then**
  Update the individual position of the population using Equation (12)
  else
  Update the individual position of the population using Equation (13)
  end if
 end for
 for (each White sharks (X_i)) **do** //Loop N times
  if (rand< *Ss*) **then**
   if(t==1) **then**
   The white shark population closing to the optimal white shark, according to Equation (7)
   else
   Update the white shark population according to Equation (16)
   end if
  end if
end for
Do
Calculate the new fitness value for each individual
Update and save the best individuals
end while
Return the best optimal solution;
Record the mean, best optimal solution and standard deviation.
T(n)=T(T+T+N+N) = C+CN=N

The position coordinates of individual fish at the next moment are determined by using Eq. (18).

$$w_j^{t+1} = \begin{cases} w_j^t + v \cdot \cos p_t \theta < 0.5 \\ w_j^t + v \cdot \sin p_t \theta \geq 0.5 \end{cases} \tag{15}$$

where p_t is the determined direction angle.

In this paper, the solution after updating the current positions is set as the position of the current shark, which is also the position of individual fish in the last iteration. In this paper, the two solutions are brought into the fish movement model to predict the next position of the fish. Figure 10 shows the simple prediction model.

In this case, the optimal position of the shark is not the position of the prey, but the position where the prey once existed, i.e., the smell of the prey attracts the shark. Equation (16) represents the modified fish behavior.

$$W_{i+1}^j = W_i^j + \sin p * V \tag{16}$$

where is the position of the shark when capturing its prey, and is the speed at which the shark approaches its prey.

The pseudocode of the proposed MSWSO is as follows.

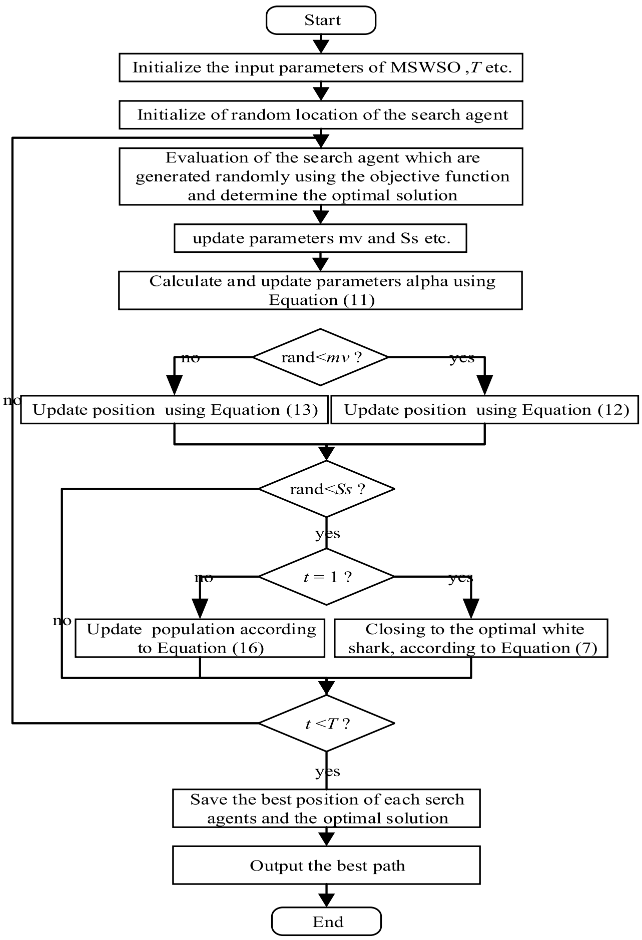


FIGURE 11. **Flowchart of the proposed path planning algorithm.**

C. UAV FLIGHT PATH PLANNING BASED ON MSWSO

The flow chart of the UAV path planning based on MSWSO is shown in Fig. 11, and its steps are summarized below.

Step1: Establish a 3D model based on topographic data

Step2: Initialize the population and algorithmic parameters mv, Ss and α, and then calculate the initial fitness value of each individual

TABLE 2. Eight benchmark functions.

Function	Dimension	Interval
$F_1(x)=\sum_{i=1}^{n} x_i^2$	30	[-100,100]
$F_2(x)=\sum \lvert x_i\rvert + \Pi_{i=1}^{n} \lvert x_i\rvert$	30	[-10,10]
$F_3(x)=\sum_{i=1}(\sum_{j-1}^{i} x_j)^2$	30	[-100,100]
$F_4(x)=\max_i\{\lvert x_i\rvert, 1\le i\le n\}$	30	[-100,100]
$F_5(x)=\sum_{i=1}^{n} ix_i^4 + random[0,1)$	30	[-1.28,1.28]
$F_6(x)=\sum_{i=1}^{n}[x_i^2-10\cos(2\pi x_i+10)]$	30	[-5.12,5.12]
$F_7(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	30	[-32,32]
$F_8(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\Pi_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	30	[-600,600]

Step3: When the individual white shark is far away from the prey, a small kinetic force is obtained and Eq. (12) is used for global search. When approaching the prey, a larger kinetic force is obtained, and Eq. (13) is used for local exploitation

Step4: When individual renewal is completed, the white shark population approaches the optimal individual using Eq. (7), and then the population is updated using Eq. (16).

Step5: If the algorithm does not reach the completion condition, continue the optimization process, otherwise report the optimal path.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this research, eight benchmark test functions are used to evaluate the improved MSWSO, and several common path planning algorithms are used to verify the performance of the proposed path planning algorithm. The simulation test environment was as follows: operating system win10, 64-bit operating system, memory 8 GB, CPU Intel i7 6500u, main frequency 2.60GHz, power supply 150W, and simulation software matlabR2018b.

A. EXPERIMENT ON BENCHMARK

In order to test the performance of the intelligent algorithm in finding the global optimum, this article uses the standard eight test functions presented in Table 2, which include both unimodal and multimodal test functions. The performance of the proposed algorithm is compared with those of SSA, SOA, GWO, WOA, WSO, and MSWSO. The same test environment and same test parameters are used in all the experiments in order to ensure their fairness.

The experiments are divided into two groups. The first group is to conduct 500 iterations on each test function and obtain the solution at each iteration. The second group is to conduct 100 iterations for 30 times on each test function to obtain the optimal value, mean value and standard deviation. Then, the convergence speed, convergence accuracy and robustness of the algorithm are analyzed in both groups of experiments. The results of the two experiments are shown in Fig. 12 and Table 3, respectively.

Fig. 12 shows the convergence plots of all test algorithms under different test functions. It can be found that MSWSO has a fast convergence speed in the test function and is significantly ahead of all comparison algorithms. MSWSO has strong exploration capabilities in the initial optimization process, and then quickly switches to the development stage in the middle, and can jump out to the global exploration in the later stage. MSWSO can usually converge to the global optimal solution based on this search characteristic. In addition, it can be seen that the iterative process of MSWSO is smooth and has a longer duration. The results show that MSWSO has high robustness and adaptability in solving different optimization problems because it can achieve a balance between exploration and development.

As shown in Table 3, MSWSO is able to obtain the optimal values of these eight test functions and approach the theoretical optimal values of each function. MSWSO has higher search accuracy than the improved algorithm using a single strategy, indicating that under the joint influence of different strategies, the optimization ability and stability of the algorithm have been maximized. The STD of the data can reflect the degree of dispersion. Based on the test results in Table 3, MSWSO has a minimal STD for each test function, indicating that it is more robust and stable when dealing with real-world problems.

Based on the analysis of Fig. 12 and Table 3, which combine the results from single experiments and multiple experiments, it is evident that the MSWSO algorithm demonstrates significant advantages over other algorithms in terms of optimal values, average optimal values, and algorithmic stability. The overall performance of MSWSO outperforms other algorithms in achieving optimal solutions. Despite the inherent characteristics of specific functions affecting the performance, MSWSO consistently converges to the optimal solution with fewer iterations compared to other algorithms. The convergence speed of MSWSO is noticeably superior. Furthermore, the obtained standard deviation of MSWSO indicates excellent performance in terms of algorithm robustness. These findings validate the effectiveness and reliability of the MSWSO algorithm.

The experimental results demonstrate that the enhanced multi-trajectory strategy achieves a balance between randomness and order in different stages of the algorithm. This approach maintains sufficient exploration in the early stage and introduces more targeted search in the later stage, effectively expanding the search space and accelerating convergence. The nonlinear convergence factor plays a crucial role in guiding the algorithm to escape local optima during the later stages of exploration. Additionally, the incorporation of fish movement rules enhances population diversity, further aiding in the escape from local optima.

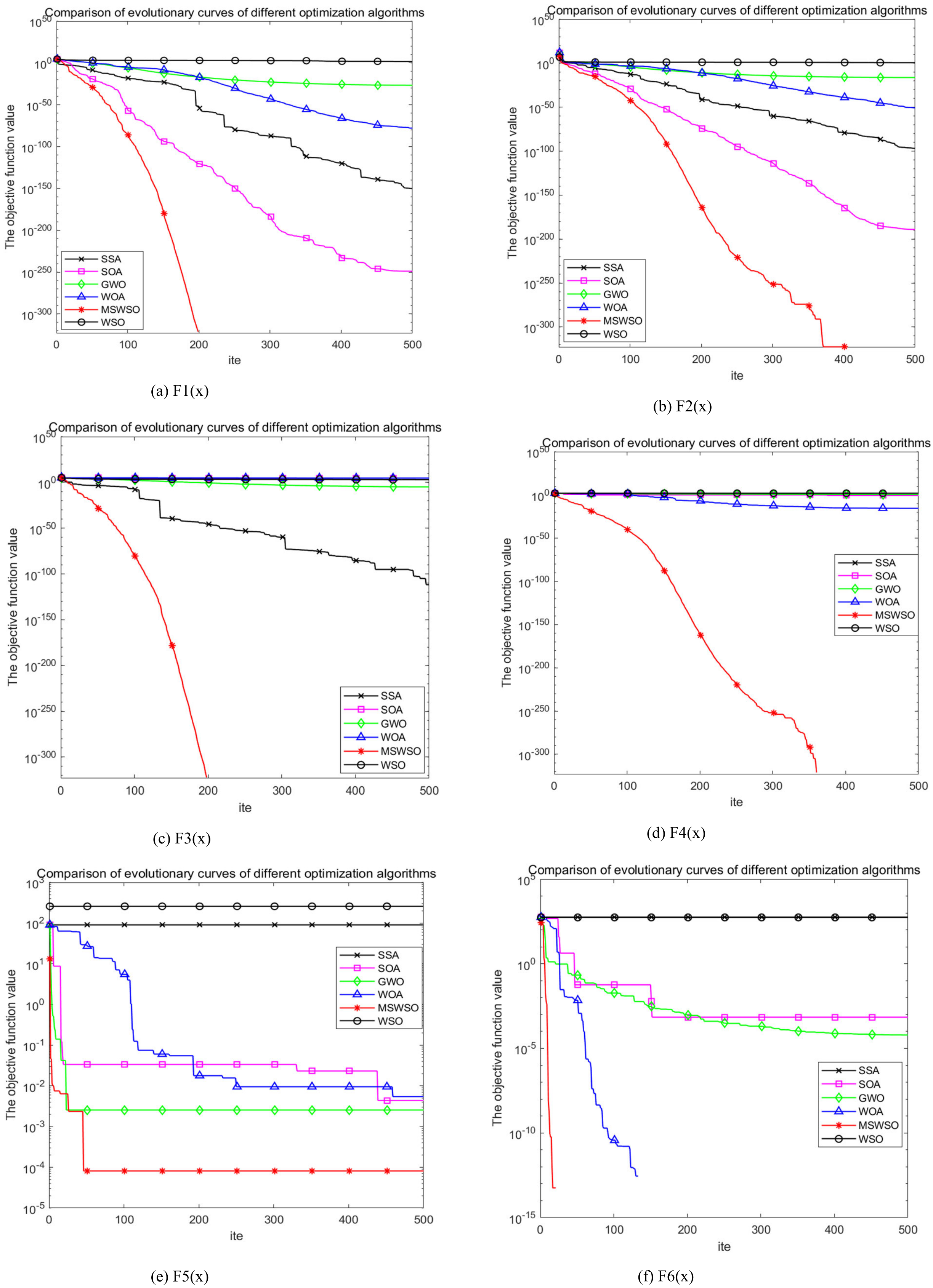
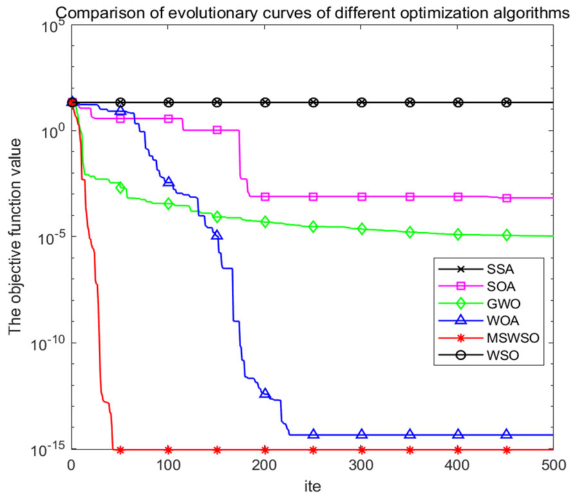
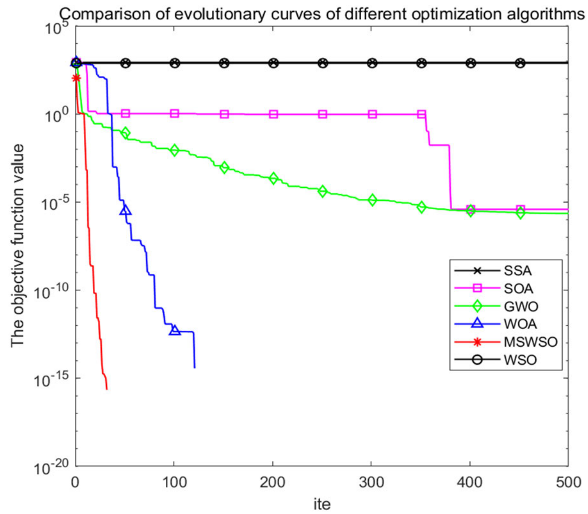


FIGURE 12. Comparison of evolutionary curves of different optimization algorithms.



(g) F7(x)



(h) F8(x)

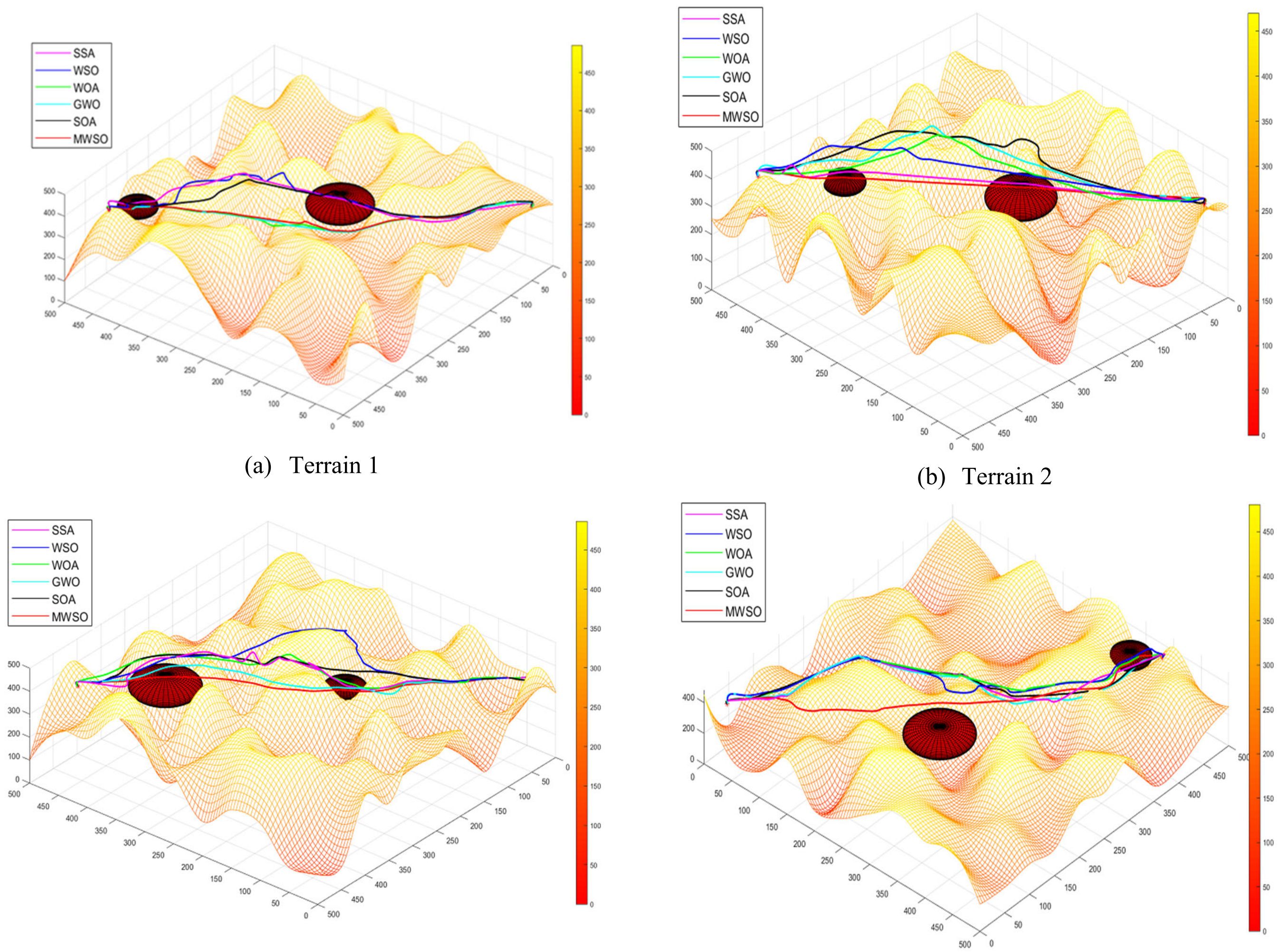
FIGURE 12. ***(Continued.)*** **Comparison of evolutionary curves of different optimization algorithms.**

TABLE 3. Comparative analysis of performance of 6 swarm intelligence algorithms.

Function	Measure	SSA	SOA	GWO	WOA	MSWSO	WSO
F1	BEST	0	1.9821e-67	0.0032245	2.7035e-15	0	791.8631
	WORST	1.5385e-38	1.7334e-28	0.039607	1.4256e-10	9.8302e-111	2975.327
	MEAN	5.1282e-40	5.778e-30	0.016602	1.2414e-11	3.2767e-112	1720.0212
	STD	2.8088e-39	3.1647e-29	0.011113	3.4751e-11	1.7947e-111	594.0909
F2	BEST	0	4.7095e-49	0.014527	1.6606e-11	0	12.1266
	WORST	1.8102e-31	1.0005e-28	0.04928	6.8432e-08	3.1558e-67	22.7407
	MEAN	6.0377e-33	3.9409e-30	0.029037	7.1362e-09	1.0519e-68	18.0373
	STD	3.3048e-32	1.8281e-29	0.0079269	1.2974e-08	5.7617e-68	2.5269
F3	BEST	0	140270.6671	51.6897	122379.6409	0	903.4483
	WORST	1.6635e-37	241883.7137	1088.0096	184512.6548	3.5712e-154	5278.317
	MEAN	5.5449e-39	195916.4771	345.4433	161400.4009	1.2634e-155	2652.3027
	STD	3.0371e-38	26861.1517	238.2551	15566.5716	6.5186e-155	1055.3458
F4	BEST	0	1.3823e-12	0.61189	13.4532	0	10.7524
	WORST	1.563e-18	82.2891	2.0209	86.9138	1.6545e-91	22.1858
	MEAN	5.2099e-20	34.7105	1.2741	66.1047	5.5434e-93	15.8093
	STD	2.8536e-19	35.9576	0.39237	22.7814	3.0201e-92	2.9186
F5	BEST	6.1882e-05	3.5154e-05	0.0043817	0.00083589	2.4228e-05	0.046812
	WORST	0.0046577	0.0029553	0.059186	0.0647	0.0015502	0.49588
	MEAN	0.001254	0.0010239	0.020192	0.020537	0.00030212	0.21376
	STD	0.00099377	0.00077124	0.011441	0.018185	0.0003818	0.11848
F6	BEST	0	0	12.4201	0	0	154.3723
	WORST	0	8.5912e-05	43.0877	4.4342e-05	0	218.0278
	MEAN	0	2.8637e-06	27.2243	1.6989e-06	0	192.7501
	STD	0	1.5685e-05	7.9163	8.1404e-06	0	14.1588
F7	BEST	8.8818e-16	8.8818e-16	0.014059	4.9886e-09	8.8818e-16	7.8302
	WORST	8.8818e-16	9.2415e-12	0.055205	3.2114e-06	8.8818e-16	11.0914
	MEAN	8.8818e-16	3.0891e-13	0.030304	4.5916e-07	8.8818e-16	9.5336
	STD	0	1.6871e-12	0.01177	7.1527e-07	0	0.91664
F8	BEST	0	0	0.0098306	1.8874e-14	0	8.7866
	WORST	0	0.97952	0.23141	0.6639	0	32.573
	MEAN	0	0.032651	0.080165	0.02213	0	17.6522
	STD	0	0.17883	0.072583	0.12121	0	5.4008

B. EXPERIMENT ON UAV FLIGHT PATH PLANNING

To evaluate the performance of MSWSO in tackling the UAV flight path planning problem, comprehensive experiments are conducted in two different scenarios. The first scenario involves normal task execution, where the map terrain, threat area locations, and task start and end points are randomly initialized to ensure unbiased and objective assessments. This scenario aims to assess the algorithm's ability to handle typical flight planning situations. In the second scenario, the complexity is increased by introducing a higher number of obstacles. Additionally, the map terrain is randomized to further challenge the algorithm's exploration and optimization



(a) Terrain 1 (b) Terrain 2

(c) Terrain 3 (d) Terrain 4

FIGURE 13. Planned path under ordinary terrain.

TABLE 4. Topographic data of the first group of experiments.

Terrain	Threat location	Threat radius
Terrain 1	[207.1 232.3 389.9]	30
	[449.9 393.4 349.2]	50
Terrain 2	[368.7 373.7 400.6]	30
	[200.3 200.6 350.7]	50
Terrain 3	[200.3 200.6 350.7]	30
	[368.7 373.7 400.6]	50
Terrain 4	[353.3 464.6 350.7]	30
	[267.7 176.8 350.5]	50

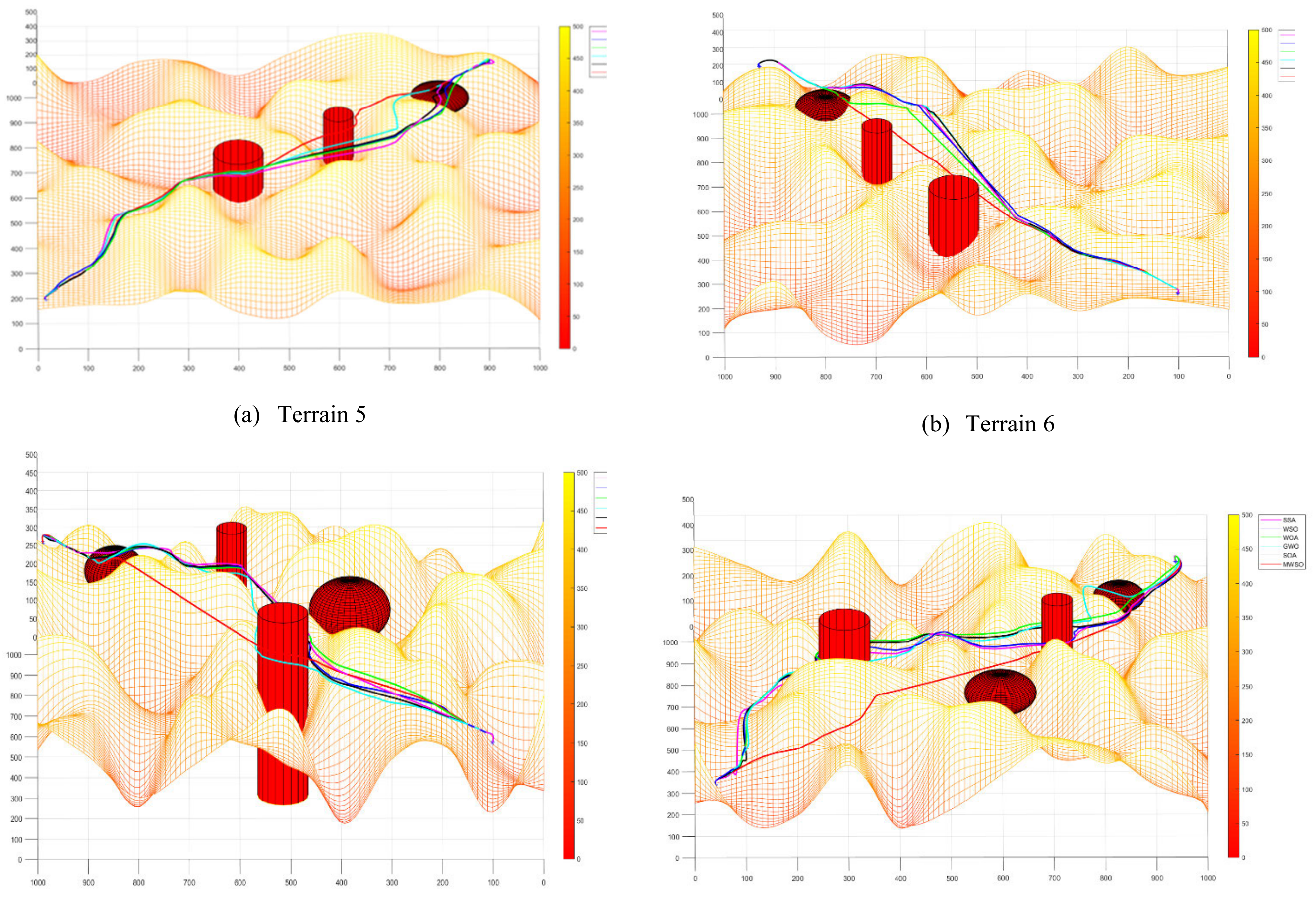
capabilities. This scenario aims to test the algorithm's robustness and effectiveness in handling more challenging and unpredictable environments.

The topographic data for these experiments are presented in Tables 4 and 5, respectively. These tables provide important information about the characteristics of the terrains, including elevation, obstacle locations, and other relevant details. By conducting experiments in these different scenarios, the algorithm's performance can be thoroughly evaluated under various conditions, providing valuable insights into its effectiveness and suitability for real-world UAV flight path planning applications.

The SSA, SOA, GWO, WOA, WSO and proposed MSWSO algorithms are used to plan the flight path of UAVs. All the algorithms adopt their original rules. The algorithmic population is formed in the same way, whose size is 60. Each algorithm is executed for 100 iterations.

Figure 13 shows the UAV planned path under the first group of ordinary terrain, and Fig. 14 shows that under the second group of large range and multi-factor terrain. The UAV first chooses the route with the shortest distance, and then it gives priority to side-way bypassing rather than climbing, when it needs to bypass obstacles. The average height of the flight path is also taken into account throughout the flight.

The results shown in Figs. 13 and 14 demonstrate that the improved algorithm, when applied to UAV path planning, is capable of generating high-quality paths in both ordinary and diverse environments. This indicates that the improved



(a) Terrain 5

(b) Terrain 6

(c) Terrain 7

(d) Terrain 8

FIGURE 14. Planned path under large range and multi-factor terrain.

algorithm has practical applicability and can adapt effectively to various scenarios.

The length and safety of the flight path of UAV directly determine the success or failure of the mission, no matter whether executed in ordinary or complex terrains. An excellent path can not only improve the efficiency of the executed task, but also ensures the safety of the UAV itself. In addition, minimization of the climbing process during the flight can reduce the energy consumption, so as to ensure the normal execution of subsequent tasks. In the experiment, all UAVs are set to drive at a constant speed, and the flight path and altitude of the drones are used to reflect their flight costs. Tables 6 and VII shows the results of 30 times of execution of all the compared algorithms in each terrain environment under eight terrain conditions in the two groups of experiments.

As can be seen from Tables 6, the average path length and height of the UAV flight have been improved significantly. Although the obtained results do not have a large difference under the action of the planning model and constraints, the improved algorithm still has an excellent performance in these two aspects. Compared with other algorithms, the path length and height are approximately 12% to 15% and 2% to 5%, respectively. This is due to the introduction of the multi-trajectory strategy, which enables a UAV to fully and quickly explore the available space. Moreover, due to the existence of nonlinear convergence factors, the UAV can timely get rid of any local optimum and not easily lose itself. Moreover, the application of the motion behavior rules of fish swarm

TABLE 5. Topographic data of the second group of experiments.

Terrain	Threat location	Threat radius
Terrain 5	[600 650 500] [400 500 500] [800 800 350]	30 50 60
Terrain 6	[596 596 500] [404 707 500] [737 899 350]	30 50 60
Terrain 7	[596 596 500] [282 616 500] [717 787 350] [404 323 350]	30 50 60 60
Terrain 8	[707 606 500] [292 515 500] [596 353 350] [828 798 350]	30 50 70 50

TABLE 6. The results of 30 times of execution of all the compared algorithms.

Terrain	COST	SSA	SOA	GWO	WOA	MSWSO	WSO
Terrain1	Path	742.9	731.1	788.1	746.3	670.6	685.6
	Altitude	394.1	400.2	388.9	391.1	385.3	396.8
	Time	2.56	2.57	2.59	2.56	2.49	2.50
Terrain2	Path	635.1	657.4	633.4	643.9	622.2	706.4
	Altitude	413.5	416.5	414.3	415.3	413.2	416.9
	Time	2.48	2.49	2.47	2.48	2.46	2.49
Terrain3	Path	704.2	800.8	688.4	714.2	679.7	742.5
	Altitude	427.4	438.2	429.4	427.4	414.9	417.3
	Time	2.48	2.60	2.50	2.52	2.48	2.55
Terrain4	Path	671.4	684.4	695.7	689.2	665.9	669.6
	Altitude	398.2	391.6	389.3	391.1	388.6	398.1
	Time	2.48	2.49	2.49	2.49	2.48	2.48
Terrain5	Path	1387.5	1506.9	1403.7	1427.3	1298.3	1324.5
	Altitude	422.4	413.3	406.5	406.1	399.7	407.7
	Time	7.15	7.20	7.18	7.18	6.95	7.09
Terrain6	Path	1270.2	1316.2	1299.3	1318.2	1261.5	1355.6
	Altitude	418.5	424.5	437.2	422.9	415.2	424.6
	Time	6.90	7.09	6.99	7.09	6.84	7.10
Terrain7	Path	1307.9	1302.5	1302.3	1374.7	1300.6	1364.7
	Altitude	420.6	428.15	418.9	419.5	415.9	419.2
	Time	7.09	7.09	7.09	7.12	7.09	7.12
Terrain8	Path	1578.3	1519.8	1593.8	1457.3	1387.7	1566.4
	Altitude	364.5	355.1	364.8	375.8	347.3	356.5
	Time	7.33	7.21	7.34	7.19	7.12	7.31

enables it to further get rid of any local optimum in the planning process and improve the convergence speed.

The application of the improved algorithm in UAV flight path planning is evident in Figs. 13, 14, and Tables 6. The obtained paths exhibit higher stability and efficiency compared to other algorithms, showcasing the algorithm's capability to assist UAVs in performing tasks effectively across diverse environments.

V. CONCLUSION

This research is dedicated to applying the enhanced algorithm MSWSO to UAV flight path planning. The algorithm incorporates a combination of random exploration and systematic development to efficiently explore and exploit the solution space. Furthermore, the integration of a nonlinear control factor enables the algorithm to effectively avoid getting trapped in local optima during later iterations. Moreover, the inclusion of fish movement behavior rules promotes population diversity, thereby improving the overall performance of the algorithm. To assess the efficacy of the enhanced algorithm in path planning, two sets of experiments were conducted, yielding positive outcomes that underscore the algorithm's effectiveness and its significance in the field of UAV flight path planning. And in the future research work, the research on the dynamic path planning of UAVs under dynamic constraints will also be carried out.

REFERENCES

[1] L. Salt, D. Howard, G. Indiveri, and Y. Sandamirskaya, "Parameter optimization and learning in a spiking neural network for UAV obstacle avoidance targeting neuromorphic processors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3305–3318, Sep. 2020, doi: 10.1109/tnnls.2019.2941506.

[2] C. Zhao, Y. Liu, L. Yu, and W. Li, "Stochastic heuristic algorithms for multi-UAV cooperative path planning," in *Proc. 40th Chin. Control Conf. (CCC)*, Jul. 2021, pp. 7677–7682, doi: 10.23919/ccc52363.2021.9549984.

[3] S. Zeng and K. Liu, "Research status and development trend of UAV path planning algorithms," *J. Phys., Conf. Ser.*, vol. 2283, no. 1, Jun. 2022, Art. no. 012004, doi: 10.1088/1742-6596/2283/1/012004.

[4] C. Xu, M. Xu, and C. Yin, "Optimized multi-UAV cooperative path planning under the complex confrontation environment," *Comput. Commun.*, vol. 162, pp. 196–203, Oct. 2020, doi: 10.1016/j.comcom.2020.04.050.

[5] Z. Jian, S. Zhang, S. Chen, Z. Nan, and N. Zheng, "A global-local coupling two-stage path planning method for mobile robots," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5349–5356, Jul. 2021, doi: 10.1109/LRA.2021.3074878.

[6] M. Kobayashi and N. Motoi, "Local path planning: Dynamic window approach with virtual manipulators considering dynamic obstacles," *IEEE Access*, vol. 10, pp. 17018–17029, 2022, doi: 10.1109/ACCESS.2022.3150036.

[7] H. Wang and W. Pan, "Research on UAV path planning algorithms," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 693, no. 1, 2021, Art. no. 012120, doi: 10.1088/1755-1315/693/1/012120.

[8] J. J. Ruz, O. Arevalo, J. M. de la Cruz, and G. Pajares, "Using MILP for UAVs trajectory optimization under radar detection risk," in *Proc. IEEE Conf. Emerg. technol. Factory Autom.*, Sep. 2006, pp. 957–960, doi: 10.1109/ETFA.2006.355225.

[9] J. Xin, J. Zhong, F. Yang, Y. Cui, and J. Sheng, "An improved genetic algorithm for path-planning of unmanned surface vehicle," *Sensors*, vol. 19, no. 11, p. 2640, Jun. 2019, doi: 10.3390/s19112640.

[10] J. Lian, W. Yu, K. Xiao, and W. Liu, "Cubic spline interpolation-based robot path planning using a chaotic adaptive particle swarm optimization algorithm," *Math. Problems Eng.*, vol. 2020, pp. 1–20, Feb. 2020, doi: 10.1155/2020/1849240.

[11] A. A. Maw, M. Tyan, T. A. Nguyen, and J.-W. Lee, "iADA*-RL : Anytime graph-based path planning with deep reinforcement learning for an autonomous UAV," *Appl. Sci.*, vol. 11, no. 9, p. 3948, Apr. 2021, doi: 10.3390/app11093948.

[12] P. Li and H. Zhu, "Parameter selection for ant colony algorithm based on bacterial foraging algorithm," *Math. Problems Eng.*, vol. 2016, pp. 1–12, Dec. 2016, doi: 10.1155/2016/6469721.

[13] Y. Zhang, G. Guan, and X. Pu, "The robot path planning based on improved artificial fish swarm algorithm," *Math. Problems Eng.*, vol. 2016, pp. 1–11, Aug. 2016, doi: 10.1155/2016/3297585.

[14] M. K. Muni, D. R. Parhi, and P. B. Kumar, "Improved motion planning of humanoid robots using bacterial foraging optimization," *Robotica*, vol. 39, no. 1, pp. 123–136, Jan. 2021, doi: 10.1017/s0263574720000235.

[15] X.-D. Liang, L.-Y. Li, J.-G. Wu, and H.-N. Chen, "Mobile robot path planning based on adaptive bacterial foraging algorithm," *J. Central South Univ.*, vol. 20, no. 12, pp. 3391–3400, Dec. 2013, doi: 10.1007/s11771-013-1864-5.

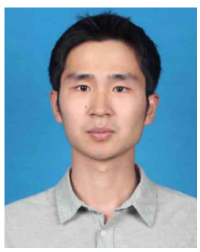
[16] B. Pang, Y. Song, C. Zhang, H. Wang, and R. Yang, "Bacterial foraging optimization based on improved chemotaxis process and novel swarming strategy," *Appl. Intell.*, vol. 49, no. 4, pp. 1283–1305, Apr. 2019, doi: 10.1007/s10489-018-1317-9.
[17] K. Wu and S. Tan, "Path planning of UAVs based on improved whale optimization algorithm," *Hangkong Xuebao/Acta Aeronautica Et Astronautica Sinica*, vol. 41, no. S2, 2020, Art. no. 724286, doi: 10.7527/S1000-6893.2020.24286.
[18] S. Jiaqi, T. Li, Z. Hongtao, L. Xiaofeng, and X. Tianying, "Adaptive multi-UAV path planning method based on improved gray wolf algorithm," *Comput. Electr. Eng.*, vol. 104, Dec. 2022, Art. no. 108377, doi: 10.1016/j.compeleceng.2022.108377.
[19] X. Zhou, F. Gao, X. Fang, and Z. Lan, "Improved bat algorithm for UAV path planning in three-dimensional space," *IEEE Access*, vol. 9, pp. 20100–20116, 2021, doi: 10.1109/access.2021.3054179.
[20] R. Zhang, S. Li, Y. Ding, X. Qin, and Q. Xia, "UAV path planning algorithm based on improved Harris hawks optimization," *Sensors*, vol. 22, no. 14, p. 5232, Jul. 2022, doi: 10.3390/s22145232.
[21] Y. Ji, X. Zhao, and J. Hao, "A novel UAV path planning algorithm based on double-dynamic biogeography-based learning particle swarm optimization," *Mobile Inf. Syst.*, vol. 2022, pp. 1–23, Jan. 2022, doi: 10.1155/2022/8519708.
[22] R. Wan, F. Zhang, and T. Zhou, "UAV path planning based on an elite-guided orthogonal diagonalised Krill Herd algorithm," *Int. J. Wireless Mobile Comput.*, vol. 23, no. 1, pp. 1–8, 2022, doi: 10.1504/ijwmc.2022.125527.
[23] X. Zhang, S. Xia, and X. Li, "Quantum behavior-based enhanced fruit fly optimization algorithm with application to UAV path planning," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, p. 1315, 2020, doi: 10.2991/ijcis.d.200825.001.
[24] L. Tang, P. Hao, and X.-J. Zhang, "An UAV path planning method in mountainous area based on an improved ant colony algorithm," *Jiaotong Yunshu Xitong Gongcheng Yu Xinxi/J. Transp. Syst. Eng. Inf. Technol.*, vol. 19, no. 1, pp. 158–164, 2019, doi: 10.16097/j.cnki.1009-6744.2019.01.024.
[25] Q. Liu, Y. Zhang, M. Li, Z. Zhang, N. Cao, and J. Shang, "Multi-UAV path planning based on fusion of sparrow search algorithm and improved Bioinspired Neural Network," *IEEE Access*, vol. 9, pp. 124670–124681, 2021, doi: 10.1109/access.2021.3109879.
[26] X. Chai, Z. Zheng, J. Xiao, L. Yan, B. Qu, P. Wen, H. Wang, Y. Zhou, and H. Sun, "Multi-strategy fusion differential evolution algorithm for UAV path planning in complex environment," *Aerosp. Sci. Technol.*, vol. 121, Feb. 2022, Art. no. 107287, doi: 10.1016/j.ast.2021.107287.
[27] C. Hu, Y. Xia, and J. Zhang, "Adaptive operator quantum-behaved pigeon-inspired optimization algorithm with application to UAV path planning," *algorithms*, vol. 12, no. 1, p. 3, Dec. 2018, doi: 10.3390/a12010003.
[28] Z. Qian and M. Lei, "Adaptive cuttlefish algorithm for UAV path planning," *Harbin Gongye Daxue Xuebao/J. Harbin Inst. Technol.*, vol. 51, no. 10, pp. 37–46, 2019, doi: 10.11918/j.issn.0367-6234.201805004.
[29] M. Braik, A. Hammouri, J. Atwan, M. A. Al-Betar, and M. A. Awadallah, "White shark optimizer: A novel bio-inspired meta-heuristic algorithm for global optimization problems," *Knowleadge.-Based Syst.*, vol. 243, May 2022, Art. no. 108457, doi: 10.1016/j.knosys.2022.108457.

RAN ZHANG received the Dr.Eng. degree from Waseda University, Japan, in 2016. Since 2016, she has been a Lecturer with Dalian University, China. Her current research interests include optimization algorithms for unmanned aerial vehicle networks and PCB routing.

XINGDA LI received the B.Eng. degree in software from Qingdao University, Qingdao, China, in 2021. He is currently pursuing the master's degree with the Communication and Network Laboratory, Dalian University. His current research interests include path planning, dynamic obstacle avoidance, and intelligent optimization.

HONGHONG REN received the bachelor's degree in automation from Dalian University, Dalian, China, in 2020, where she is currently pursuing the master's degree with the Communication and Network Laboratory. Her current research interests include dynamic and static task assignment and the intelligent optimization of UAV cluster.

YUANMING DING received the Ph.D. degree from Keio University, Japan, in 2004. From November 2004 to November 2016, he was a Postdoctoral Fellow with JSPS. Since 2009, he has been a Professor with the School of Information Engineering, Dalian University, China. His research interests include communication signal processing, underwater optical communication, and network technologies.



YIFEI MENG received the B.Eng. degree in information management from Shandong Technology and Business University, Yantai, China, in 2021. He is currently pursuing the master's degree with the Communication and Network Laboratory, Dalian University. His current research interest includes EEG signal emotion recognition.

QINGYU XIA received the B.Eng. degree in electronic information engineering from the Anhui University of Finance and Economics, Bengbu, China, in 2020. He is currently pursuing the master's degree with the Communication and Network Laboratory, Dalian University. His current research interests include swarm intelligence, blind source separation, and intelligent optimization.