Hindawi
Scientific Programming
Volume 2021, Article ID 5158304, 15 pages
https://doi.org/10.1155/2021/5158304

Research Article

A Novel Sparrow Particle Swarm Algorithm (SPSA) for Unmanned Aerial Vehicle Path Planning

Wangwang. Yu, Jun. Liu, and Jie. Zhou

School of Electrical Engineering, Shanghai Dianji University, Shanghai 201306, China

Correspondence should be addressed to Jun. Liu; liujun@sdju.edu.cn

Received 14 September 2021; Accepted 23 November 2021; Published 9 December 2021

Academic Editor: Pengwei Wang

Unmanned aerial vehicle (UAV) has been widely used in various fields, and meeting practical high-quality flight paths is one of the crucial functions of UAV. Many algorithms have the problem of too fast convergence and premature in UAV path planning. This study proposed a sparrow particle swarm algorithm for UAV path planning, the SPSA. The algorithm selects a suitable model for path initialization, changes the discoverer position update, and reinforces the influence of start-end line on path search, which can significantly reduce blind search. The number of target points reached is increased by adaptive variable speed escapes in areas of deadlock. In this case, the planned trajectory will fluctuate, and adaptive oscillation optimization can effectively reduce the fluctuation of the path. Finally, the optimal path is simplified, and the path nodes are interpolated with cubic splines to improve the smoothness of the path, which improves the smoothness of the UAV flight trajectory and makes it more suitable for use as the UAV real flight trajectory. By comparison, it can be concluded that the improved SPSA has good convergence speed and better search results and can avoid local optimality.

1. Introduction

Unmanned aerial vehicle (UAV) has been widely used in various fields, such as transportation, rescue, and military, with the continuous progress of technology. Compared with ground robot path planning, the path planning of UAV in 3D environment increases the complexity of trajectory calculation as the spatial dimension rises. UAVs are affected by buildings, weather, and their energy consumption during flight, so how to plan a safer, more efficient, and faster path is an ongoing hot research problem. Traditional algorithms such as the A* algorithm [1], artificial potential field method [2, 3], and RRT [4] have been used for path planning, but most of them still suffer from high computational complexity and local convergence.

A heuristic algorithm is a new solution algorithm obtained by imitating various natural phenomena and summarizing and refining that natural phenomenon. The heuristic algorithm can solve some problems that ordinary methods cannot solve. The UAV path planning problem is a multivariate problem in the real environment to find the optimal solution. Many scholars have studied this problem with heuristic algorithms [5–11] to find suitable solutions. Duan [12] et al. used the maximum-minimum adaptive ant colony optimization algorithm to replan the UAV collaborative path in dynamic and uncertain environments, determine the time for the UAV to reach the conflict point, and then determine the flight trajectory and speed of the UAV to avoid static threats and popup threats. Shiri et al. [13] proposed a neural network-assisted online control algorithm for remote UAV paths that enable UAV to make control decisions locally in harsh communication environments, thereby reducing travel time. Qu [14] et al. proposed a reinforcement learning algorithm based on the grey wolf algorithm that adaptively switches selection by controlling the performance accumulated by individual four operations of exploration, exploitation, geometric adjustment, and optimal adjustment, which has better results than other improved grey wolf algorithms. Duan [15] et al. proposed a UAV path planning based on an improved water droplet algorithm, which considers the effects of ice accumulation, Richardson number, meteorological changes, and different flight altitudes on UAV path planning. They used the virtual potential field method to adjust the flight direction of the UAV to perform static path planning and dynamic path planning for the UAV. Liu [16] et al. proposed an evolutionary algorithm based on the

evolutionary algorithm for UAV path planning, which changes the t distribution algorithm to effectively solve the high computational complexity and low efficiency in UAV dynamic path planning.

The sparrow search algorithm [17] is a new heuristic algorithm proposed by Jiankai Xue in 2020, and it is a group intelligence algorithm that imitates sparrow predation. The algorithm has the characteristics of fast convergence and merit-seeking solid ability. The sparrow search algorithm has been widely used in the field of control engineering, for example, tracking the maximum power point [18], network configuration [19], and path planning field [20]. Liu [21] proposed a modified sparrow search algorithm (CASSA) to balance the convergence speed and searchability. The algorithm improved the efficiency of path planning by the Corsi–Gaussian variation strategy and adaptive inertia weights. Ouyang [22] proposed the learning sparrow search algorithm (LSSA) to improve stability and security. This algorithm solves the shortcomings of strong randomness and easy to fall into local optimum. Abdulhammed [23] applied the sparrow search algorithm to the load balancing problem of cloud computing by separately provisioning different tasks to shorten the server response time and reduce power consumption. The algorithm shows that the advantages of fast convergence and good searchability of the sparrow search algorithm are highly applicable in solving engineering problems.

Particle swarm algorithm [24] is an algorithm proposed by James Kennedy and Russell Eberhart, which has the characteristics of fast convergence and simple structure. Many scholars have applied it to various optimization scenarios [25–30] and achieved good results. Li [31] proposed an SLPSO algorithm for solving collision hazards, insufficiently smooth paths, and long planning paths of mobile robots in path planning, and the experimental results showed that the algorithm is effective and feasible. Phung [32] proposed an improved discrete particle swarm algorithm (DPSO) to be used in UAV path planning to improve the algorithm's performance through qualitative initialization, random variation, and edge swapping by taking advantage of parallel computing. Foo [33] proposed a particle swarm algorithm-based 3D UAV path planning that considers the fuel consumption of the UAV, flies over specified obstacles, and leaps over specified reconnaissance targets. Thabit [34] proposed a multi-robot particle swarm optimization algorithm for multi-UAV path planning in unknown situations. The robot decides the direction of movement based on the information collected by sensors and combines probabilistic windows to obtain current information and previous robot experience to select the path with better fitness.

In the field of human-machine path planning, the current UAV still suffers from the problem of locally optimal solutions, making it difficult to search for globally optimal paths. A novel SPSA is proposed for this problem, which guides the subsequent path search through path initialization. The discoverer position update rule is improved to enhance the search near the start-end line. Adaptive variable speed escape search is used to improve the search efficiency of paths when obstacles are encountered. Adopting adaptive oscillation optimization reduces path fluctuations and improves path smoothing. Finally, the path smoothness is improved by simplifying the nodes and smoothing process, which is more suitable for path planning in the real environment. It is also verified that the SPSA has a shorter convergence path and lower energy consumption than other algorithms.

The structure of this study is as follows: Section 2 establishes the environment model that matches the actual situation. Section 3 describes the specific implementation of the sparrow search algorithm and the particle swarm algorithm. Section 4 describes the specific implementation of the SPSA. Section 5 draws the corresponding structure and flow chart for the structure of the SPSA. Section 6 analyzes the experimental results for the algorithm complexity analysis and experimental results. Section 7 concludes the SPSA.

2. Environment Modelling

In the actual complex environment, the UAV will encounter multiple threats such as obstacles and trees in reaching the target set from the start point, and sometimes, it also needs to consider the climate, the energy of the UAV, and other factors. The main problem of UAV path planning is the path optimization problem considering multiple disturbance factors. By building a mathematical model, the optimal flight path is calculated.

2.1. Environmental Model Building. The movement of the UAV is less restricted by the terrain and can reach the target point quickly in a complex environment, but the trajectory of the UAV's movement is in a three-dimensional environment. The three-dimensional environment is more complicated than the two-dimensional environment. The amount of calculation will also be tremendous. To reduce the time of UAV path calculation, the establishment of a suitable model has an important influence on the calculation of the optimal path. This study adopts the modelling method in the literature [35]. Each time the position is selected, it will move forward by one unit in the x-axis direction, simplifying the three-dimensional path problem into a two-dimensional path planning problem.

Assuming that the velocity of the UAV flight is variable and disregarding the effect of the structure of the UAV on the flight, the square tall building in the real environment is equated to a square obstacle, and the cylindrical building is equated to a cylindrical obstacle, as shown in Figure 1.

2.2. Adaptability Calculation. The trajectory of the UAV flight is usually close to the line between the start point and the endpoint (start-end line) when avoiding obstacles, while the path should be as smooth as possible. The distance and the safety of UAV from the obstacles are considered in the literature [14]. However, the computational effort also increases with the increase in obstacles, which is not conducive to finding the globally optimal path. Considering the energy loss of UAV, a new fitness calculation is proposed as follows:

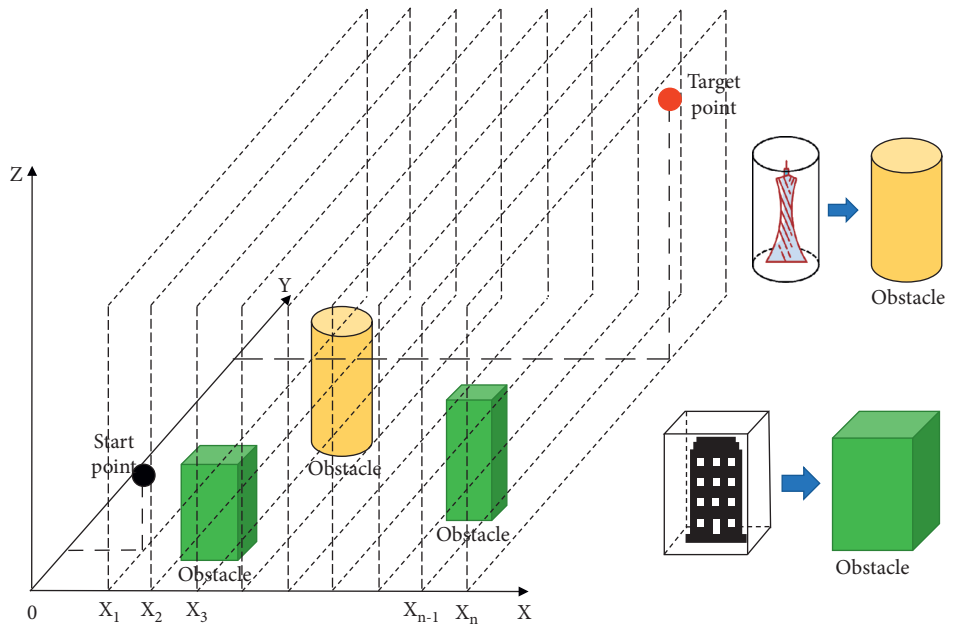

Figure 1: Environment modelling.

$$E = E_{\text{path}} + E_{\text{turning}} + E_{\text{climb}}. \tag{1}$$

The fitness value of UAV flight E is the total energy required for UAV path planning. E_{path} is the energy consumed for all planned paths assuming that the UAV flies at a constant speed. E_{turning} indicates the additional energy consumed for UAV flight turning. E_{climb} indicates the energy is required for UAV climbing and landing. E_{climb} represents the energy consumption for climbing and landing. It is assumed that the energy of the UAV flight consists of three parts, and one part is the energy of uniform flight, and it is assumed that the energy consumption of this part is proportional to the distance flown. The second part is that when turning, there will be additional turning energy loss in addition to the first part of energy. When the turning angle is larger, as the speed remains the same centripetal force is needed to do work, more energy is needed to maintain a constant speed and complete the flight turning. Similarly, more work is required to move the aircraft in the vertical direction, assuming that the energy required by the UAV is proportional to the distance travelled in the vertical direction. Each part of the solution equation is as follows:

$$\begin{aligned} E_{\text{path}} &= k_0 \cdot \sum_{n=1}^{s-1} \sqrt{(x_n - x_{n+1})^2 + (y_n - y_{n+1})^2 + (z_n - z_{n+1})^2}, \\ E_{\text{turning}} &= k_1 \cdot \sum_{n=1}^{s-1} \mu_0 * (1 + \cos\theta), \\ E_{\text{climb}} &= k_2 \cdot \sum_{s=0}^{s-1} |\Delta z|. \end{aligned} \tag{2}$$

Here, x_s, y_s, and z_s denote the coordinates of x, y, and z axes of the nodes, respectively, μ_0 denotes the turning angle coefficient, and θ denotes the angle between two adjacent line segments. When the angle is larger, the trajectory is smoother, and the fuel cost is lower. When the turn is smaller, the UAV needs to slow down, steer, and accelerate the process, which will consume more fuel. $|\Delta z|$ denotes the variation on the z-axis between two adjacent nodes, where k_0, k_1, and k_2 are scale factors. The fitness is performed by cubic spline interpolation, and the summation is performed for each slice of the fitness, which is similar to the energy consumption in the real situation.

As the UAV flight in the actual situation of variable speed flight needs to adjust the route to avoid trees, buildings, and other obstacles, UAV flight steering requires more energy E_{turning}, while for the flight trajectory without substantial height oscillation, the flight altitude constantly changes will also consume more energy. In general, the optimal adaptation in UAV flight corresponds to a smooth and smooth flight trajectory.

3. Basic Theoretical Algorithms

The choice of the algorithm has an essential impact on path planning. A suitable algorithm can improve the convergence of the optimal path and avoid local optima. This section describes the specific implementation of the involved sparrow search algorithm and particle swarm algorithm.

3.1. Basic Sparrow Search Algorithm. The sparrow search algorithm is a novel bionic algorithm with fast convergence and merit-seeking solid ability. Sparrows are divided into producers and scroungers. The identities of producers and scroungers can be interchanged, but the weight remains the same in the population. The producer with good adaptation in the sparrow search algorithm will get food first. The producers' positions are updated by the following equation

$$P_{i,j}^{t+1} = \begin{cases} P_{i,j}^{t} \cdot \exp\left(\dfrac{-i}{\alpha \cdot \text{iter}_{\max}}\right), & \text{if } R_2 < \text{ST}, \\ P_{i,j}^{t} + Q \cdot L, & \text{if } R_2 \geq \text{ST}. \end{cases} \tag{3}$$

Here, t denotes the current iteration, i denotes the serial number of the current scrounger, j denotes the dimension, α is a random number, and $\text{iter}_{\max}$ denotes the maximum number of iterations. r_2 and ST denote the warning value and safety value, which are constants, Q is a random number, and L is a $1 \times d$ matrix, and each element is 1. When $R_2 < \text{ST}$, it means there are no predators around the foraging environment, and the scrounger can conduct an extensive search. If $R_2 \geq \text{ST}$, it means that some sparrows in the population found the predator and alerted other sparrows, and then, the sparrows needed to fly to other places for foraging.

The scrounger will watch the scrounger, and if it perceives that the scrounger has found better food, it will leave its present position to grab the food, and if it wins, it gets the scrounger's food. The scrounger's position is updated as follows:

$$P_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\dfrac{P_{\text{worst}}^{t} - P_{i,j}^{t}}{t^2}\right), & \text{if } i > n/2, \\ P_{i,j}^{t+1} + \left|P_{i,j}^{t} - P_{P}^{t+1}\right| \cdot A^{+} \cdot L, & \text{otherwise}. \end{cases} \tag{4}$$

Here, $P_{i,j}^{t+1}$ denotes the optimal position of the next iteration, P_{worst}^{t} is the worst position in the current iteration, and n is the number of scroungers, where A denotes a $1 \times d$ matrix where each element is randomly assigned to 1 or -1, and $A^{+} = A^{T}(AA^{T})^{-1}$. When $i > n/2$, the less adapted scroungers need to fly elsewhere in search of food.

Sparrows aware of the danger account for 10% to 20% of the total population. The locations of these sparrows are updated as follows:

$$P_{i,j}^{t+1} = \begin{cases} P_{\text{best}}^{t} + \beta \cdot \left|P_{i,j}^{t} - P_{\text{best}}^{t}\right|, & \text{if } f_i > f_g, \\ P_{i,j}^{t+1} + K \cdot \left(\dfrac{P_{i,j}^{t} - P_{\text{best}}^{t}}{f_i - f_w + \varepsilon}\right), & \text{if } f_i = f_g. \end{cases} \tag{5}$$

Here, P_{best}^{t} is the current global best position, β and K are random numbers, f_g and f_w are the current global best and worst fitness values, and ε is a constant to avoid the denominator being zero.

The presence of obstacles in the environment of UAV path planning can interfere with the sparrow search algorithm to search the optimal and worst paths, resulting in the presence of unsearchable target points and loss of good search routes. Since the absolute value of each element in A^{+} is the same, the presence of obstacles in the planned path loses the number of individuals.

3.2. Basic Particle Swarm Algorithm. The particle swarm algorithm is an algorithm that is studied by imitating the predatory behaviour of a flock of birds. The particle swarm algorithm uses massless particles to simulate the birds in a flock, and the particles have velocity and position properties. The optimal solution obtained by each particle searching for a single individual in space is used as the individual's extremum, and the extremum of all individuals is used as the optimal global solution. All particles in the swarm adjust their velocity and position according to the current individual extreme value and the optimal global solution.

The particle swarm algorithm first initializes the position of the particle itself and adjusts its position by iteration. In each iteration, the particle adjusts its speed by the extreme individual value and the optimal global solution, thus changing its position. The particle swarm algorithm velocity and position updates are shown in equations (6) and (7) as follows:

$$\begin{aligned} V_{ij}(t+1) = V_{ij}(t+1) + c1 \cdot \text{Rand} \cdot \left(P_{\text{1best}\_ij} - P_{ij}\right) \\ + c2 \cdot \text{Rand} \cdot \left(P_{\text{g best}\_j} - P_{ij}\right), \end{aligned} \tag{6}$$

$$P_{ij}(t+1) = P_{ij}(t) + V_{ij}(t+1). \tag{7}$$

Here, V_{ij} denotes the velocity in j-dimensional space of the particle with index i, P_{1best_ij} denotes the position in j-dimension corresponding to the particle with index i obtaining the local extremum, and $P_{\text{g best}\_j}$ denotes the position in j-dimension corresponding to the optimal global solution obtained by all particles. Rand is a random number from 0 to 1, and $c1$ and $c2$ are learning factors.

The corresponding pseudocode for the basic particle swarm algorithm 1 is shown below.

4. Sparrow Particle Swarm Algorithm

4.1. Path Initialization. UAV path planning is a pathfinding optimization that considers multiple environmental constraints and the UAV's dynamics. The selection of the next adjacent path node is affected by obstacles and the current node position constraints, and from safety considerations, the UAV cannot cross obstacles from the current node to the next node. The path initialization of the sparrow particle swarm algorithm has an essential impact on the convergence of the subsequent sparrow search algorithm. A good path initialization can effectively improve the search efficiency and prevent the path from generating large oscillations that do not conform to the characteristics of the smooth movement of the UAV in the real environment, for which a path initialization method is proposed.

Assuming a variable speed of the UAV, the UAV should tend to fly to the target point if the target point is known. As shown in Figure 2, P_a denotes the current position and P_{Target} denotes the position of the target point. X_a denotes the x-axis coordinates corresponding to the current position, and P_{a+1} is the next point to be determined, and it is the orange point in Figure 2. P_t is the centre of the following available coordinate range determined from the current

```
Initialize
  Set the basic parameters
    For each particle i
    Initialize velocity and position of particles
  Calculate fitness f_i and set p Best_i = f_i
  End For
  g Best = f_i
  For each iteration
    For each particle
      Update the position and velocity using by (6) and (7)
      Calculate the fitness f_i
      If f_i < p Best_i
        p Best_i = f_i
      End If
      If p Best_i < g Best
        g Best = p Best_i
      End If
      End For
  End For
Return results
Terminate
```

Algorithm 1: Basic particle swarm algorithm pseudocode.

point P_a and the coordinates of the target point. P_t is calculated according to the following equation:

$$P_t = P_a + \frac{P_{\text{Target}} - P_a}{m}. \tag{8}$$

Here, m is the difference between the horizontal coordinate X_{a+m} of the target point and the horizontal coordinate X_a of the current point P_a. P_{a+1} is chosen by the following formula:

$$P_{a+1} = \text{round}\left(P_t + 2 * \text{rand} * \Delta h - \Delta h\right), \tag{9}$$

$$P_t - \Delta h \le P_{a+1} \le P_t + \Delta h. \tag{10}$$

Here, round denotes rounding. rand is a random value from 0 to 1. Δh denotes half of the square length of the range of selectable coordinate points.

During the initialization process, each time the individual confirms the following position based on the current point and the target point, this improved method can avoid too much difference between two adjacent nodes during the UAV movement, interfering with the subsequent convergence.

4.2. Improving the Producer Location Update Formula. The traditional sparrow search algorithm with better pathfinding capability can lead to stagnant search in an environment with more obstacles, thus losing more individuals. The intelligence body has weaker search capability in an environment with more obstacles, leading to local convergence of the path search. The current speed of the particle swarm algorithm is affected by the globally optimal path, assuming that there are no obstacles between the start point and the target point. The optimal path is start-end line. If there is an obstacle, the shortest path should be closer to the start-end line. Using the start-end line as an idealized global optimal path increases the influence of this optimal global

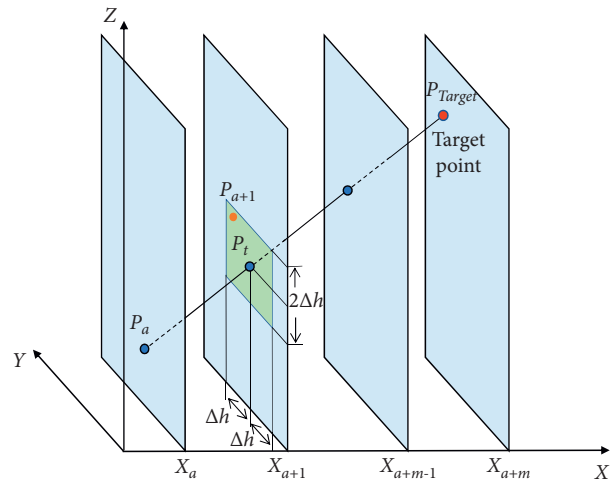

Figure 2: Path selection schematic.

path on the producer and enhances the producer's ability to find the optimal solution in the start-end line and enhance the ability to search near the start-end line by changing the producer's function.

$$P_{i,j}^{t+1} = \begin{cases} P_{i,j}^{t} + \exp\left(\dfrac{-i}{\alpha \cdot \text{inter}_{\max}}\right) + \text{rand} \cdot \left(P_{\text{ST}} - P_{i,j}^{t}\right), & \text{if } f_i > f_g, \\ P_{i,j}^{t+1} + \text{rand} \cdot \left(P_{\text{ST}} - P_{i,j}^{t}\right), & \text{if } f_i = f_g, \end{cases} \tag{11}$$

Here, P_{ST} is the corresponding coordinate position on the start-end line. rand is a random number from 0 to 1. By increasing the difference between the current path and the start-end line, the searchability around the start-end line is strengthened to avoid falling into the local optimum.

4.3. Adaptive Variable Speed Escape Search. In the sparrow search algorithm, when $i > n/2$, it indicates that the sparrow needs to fly to other places to find food for more energy, since each element in A^+ has the same absolute value and is a number from 0 to 1 (excluding 0 and 1). Since the values of formula (4) running to $P_{i,j}^{t+1}$, P_P^{t+1}, A^+, and L have been fixed, when the next node is unreachable due to obstacles, even if the search is carried out again, it still cannot reach the next node, which is the same as the last one, and at this time, the path search is stalled. For this stagnation, the adaptive variable speed escape exploration is inspired by the particle swarm algorithm.

When the scrounger flies to other places in search of food, if the next node is unreachable, the scrounger searches at the lowest speed and searches ten times, and if a moveable node cannot be found, the speed of movement is increased. If there is a moveable node, the search stops, and the point is taken as the current point. Every ten searches are no moveable point, and then, the speed increases by one until it reaches the maximum search speed. If it still cannot find a moveable node, then the search ends.

$$P_{i,j}^{t+1} = P_{i,j}^{t} + 2 \cdot \text{rand} \cdot \leq V_{\text{cur}} - V_{\text{cur}}. \tag{12}$$

Here, V_{cur} denotes the current velocity, increasing from the lowest velocity, since the globally optimal path corresponds to the smallest possible distance between two adjacent points.

The pseudocode for the variable speed escape search Algorithm 2 is shown below.

4.4. Adaptive Oscillation Optimization. Since there are random numbers in the selection of paths and significant differences between other paths and the current path, it leads to the planning with oscillation. The oscillation of the drone trajectory will interfere with the judgment of the optimal path adaptation value and easily fall into local minima. The oscillating trajectory will also consume more energy of the UAV and is not suitable as the actual UAV flight trajectory.

For addressing this problem, an adaptive oscillation optimization method is proposed to optimize the planned original path with adaptive oscillation to reduce the trajectory oscillation of the UAV. To reduce the path oscillation at certain points, the overall trajectory trend is judged by its multiple adjacent points to reduce the path oscillation better. The trajectory optimization equation is as follows:

$$P'_{i,j} = \text{round}\left(\frac{\left(P_{i,j-1} + P_{i,j} + \ldots + P_{i,j+N_P-2}\right)}{N_P}\right). \tag{13}$$

Here, round means rounding the value. N_P is the number of adjacent nodes that will have an impact on the current position (including the node itself), the maximum value of N_P is 5, and the minimum value is 3. Initially, the value of N_P is 5. The optimization of the current position is influenced by the position of the previous node, this node, and the next three nodes. If the optimized position $P'_{i,j}$ has obstacles, then the value of N_P is subtracted by 1 and re-optimized. If N_P is less than 3, no position optimization is performed for the current point. As shown in Figure 3, the original trajectory is blue, and the path after adaptive oscillation optimization is red, which shows that the smooth trajectory has better smoothness.

4.5. Simplification of Path Nodes and Smoothing. The paths after adaptive oscillation optimization have lower oscillation compared with the original paths. The UAV should move in a straight line or smoothly as much as possible in the actual environment. To make the path more realistic, the SPSA finally converges and simplifies the optimal path nodes found to facilitate the subsequent smoothing process with the third spline interpolation.

In Figure 4(a), a simplified diagram of the nodes is shown, where the black line represents the planned path and the blue line represents the simplified path nodes. Point A can reach points B, C, and D, and cannot reach points E and F. Assuming that the UAV reaches point F from point A, which subsequent nodes can be reached from point A is first determined. The corresponding vector is (1, 1, 1, 0, 0), where 1 represents that it can be reached safely and 0 means that it cannot be reached safely, and there are obstacles in the linear path. The last one among these nodes sorted to represent the node that can be reached is D, and it is taken as the current node. When the current node is D, judging the subsequent nodes is continued, assuming that point D can reach point E and point F, the corresponding path vector is (1, 1), and then F is selected as the next node.

This method can effectively avoid the situation that the path is not streamlined enough due to path oscillation, etc. For example, point A′ in Figure 4(b) can reach points B′, C′, D′, and F′, corresponding to the vectors (1, 1, 1, 0, 1). Due to the presence of obstacles, point A′ cannot reach point E′, so directly choose F′ as the next node. Points A′, D′, and F′ form a triangle in space, by the triangle. The sum of line segment A′D′ and line segment D′F′ is greater than A′F′ by the law, from which it can be concluded that this optimization method can minimize the path.

The triple spline interpolation function can improve the path's smoothness and reduce the path's twists and turns, and the path is more consistent with the flight path of the UAV as a real environment. The simplified path can be smoothed better by the cubic spline interpolation function, which reduces the turn amplitude in the path and saves the energy consumption of the UAV.

5. Structure and Flow chart of the Sparrow Particle Swarm Algorithm

5.1. Structure of the Sparrow Particle Swarm Algorithm. The pseudocode of the sparrow particle swarm algorithm Algorithm 3 is shown below.

5.2. Flow chart of the Sparrow Particle Swarm Algorithm. The sparrow particle swarm algorithm flow chart is shown in Figure 5.

Update the position by equation (4)
Initialize the velocity V_{cur} of individual: $V_{cur} = V_{min}$
Set $T = 0$
While the position can not reach
  If $T =$ Tmax
    $V_{cur} = V_{cur} + 1$
    If $V_{cur} > V_{max}$
      break
    End If
    Set $T = 0$
  End If
  $T = T + 1$
  If $V_{cur} \leq V_{max}$
    Update the position by equation (12)
  End If
End While

Algorithm 2: Variable speed escape search algorithm pseudocode.

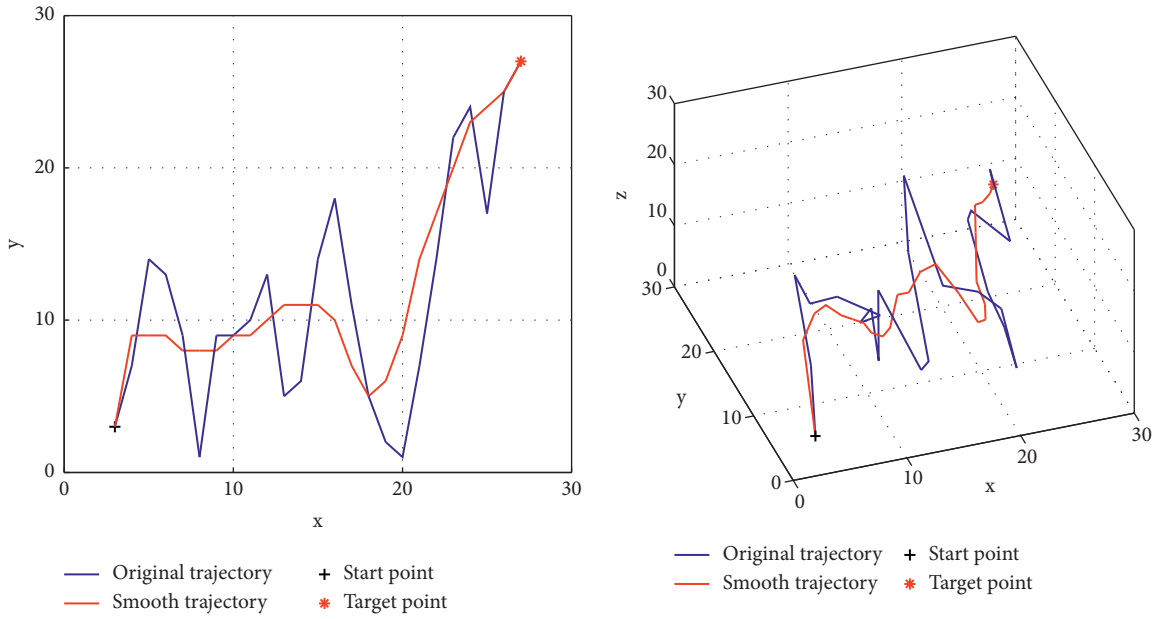

Figure 3: Comparison of before and after adaptive trajectory optimization.

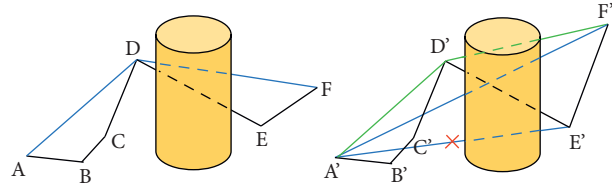

Figure 4: Simplified schematic diagram of UAV trajectory nodes.

6. Experimental Simulation and Result Analysis

To verify the feasibility of the sparrow particle swarm algorithm for path planning, experimental simulations are conducted in three different environments with increasing complexity, and each algorithm is simulated 30 times in each case. The equipment parameters of this simulation experiment are as follows: CPU is Intel(R) Xeon(R) E5-2450H @ 2.10 GHz; graphic card is GTX 1050 Ti; memory space is 32G; and simulation software is MATLAB 2020b.

To verify the performance of the algorithms, we need to avoid the effect of path initialization and perform the same path initialization for all the compared algorithms. In the

```
Initialize
Set the basic parameters
Set the start point and target point
Initialize the position P_i of each individual in the population using equations (10)–(12)
Oscillation optimization of all individuals trajectories
For each iteration
  Initialize optimal fitness and worst fitness
  For each producer
    For each dimension
      Update the position of P_i by the equation (11)
      Set T = 0
      While the position can not reach and T < Tmax
        Update the position of P_i by the equation (11)
        T = T + 1
      End While
    End For
  End For
  For each scrounger
    For each dimension
      Update the position of P_i by the equation (4)
      Set T = 0
      While the position can not reach and T < Tmax
        Update the position of P_i by the equation (4)
        T = T + 1
      End While
      If T ≥ Tmax
        Search for the next position by adaptive escape using equation (12)
      End If
    End For
  End For
For each individual that finds danger
    For each dimension
      Update the position of P_i by the equation (5)
      Set T = 0
      While the position can not reach and T < Tmax
        Update the position of P_i by the equation (5)
        T = T + 1
      End While
    End For
  End For
  Optimize adaptive oscillation using equation (13)
  Update position of all individuals
  Calculate and sort fitness values
  End For
Perform node optimization on the optimal path and smooth optimization
Return results
Terminate
```

Algorithm 3: Sparrow particle swarm algorithm (SPSA) pseudocode.

initialization of the search path, the path shifted one unit in the x-axis direction (corresponding to a horizontal movement of 20 m) and by ten units in the y-axis and z-axis (corresponding to a movement of 200 m in the y-axis direction and 100 m in the z-axis direction, respectively). In Table 1, Tmax is the maximum number of cycles. For improving the efficiency of all path searches, when the search reaches the next path point, it is determined whether the next path point is an obstacle, and if it is an obstacle, it is reselected. If the reselected times reached Tmax, then the search is deadlocked, and the search for that individual ends, and the path search continues for the next individual. For reflecting the fairness of each algorithm comparison, the method is used for each algorithm individual if it encounters an obstacle when searching for the next path. The results of the initialization of the experimental data are shown in the following table.

6.1. Algorithm Complexity Analysis. The complexity analysis of the algorithm is one of the criteria to evaluate the performance of the algorithm. SPSA is divided into the

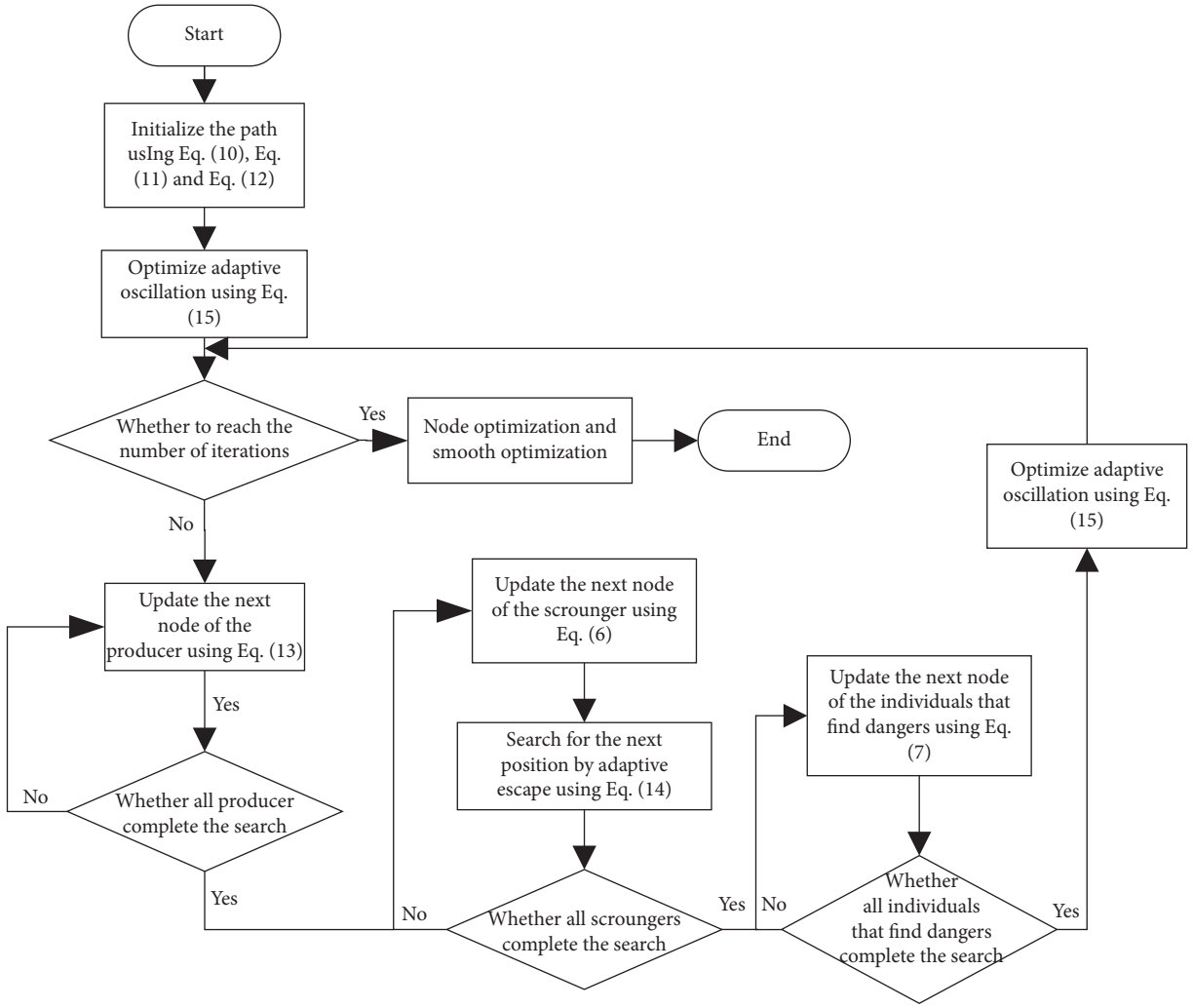

Figure 5: Sparrow particle swarm algorithm flow chart.

Table 1: Initialization of SPSA parameters.

Parameters	Symbol	Value
Warning value	ST	0.6
Scrounger ratio	JD	0.3
Producer ratio	PD	0.7
The proportion of sparrows who are aware of the danger	SD	0.2
Total group number	G	30
Number of iterations	M	1000
Maximum speed	V	5
Minimum speed	V	1
Maximum number of cycles	$T_{\max}$	10

initialization phase and the algorithm iteration phase: the initialization phase is executed only once and the algorithm iteration phase is executed according to the iteration cycle. The dimension to be calculated for each individual is D, the number of individuals is N, the maximum number of iterations is M, the range of movement in the y-axis and z-axis directions selected during initialization is A and B, respectively, and SD is the proportion of sparrows found to be in danger. The structure of the sparrow search algorithm for path planning in three-dimensional space is $O(N \times D \times A \times B + M \times N \times D)$, where the complexity of the initialized path is $O(N \times D \times A \times B)$. The computational complexity of the path iterative search optimization is $O(M \times N \times D)$. The SPSA increases some computational complexity due to the enhanced ability to optimize the search path, and the complexity of the adaptive oscillation

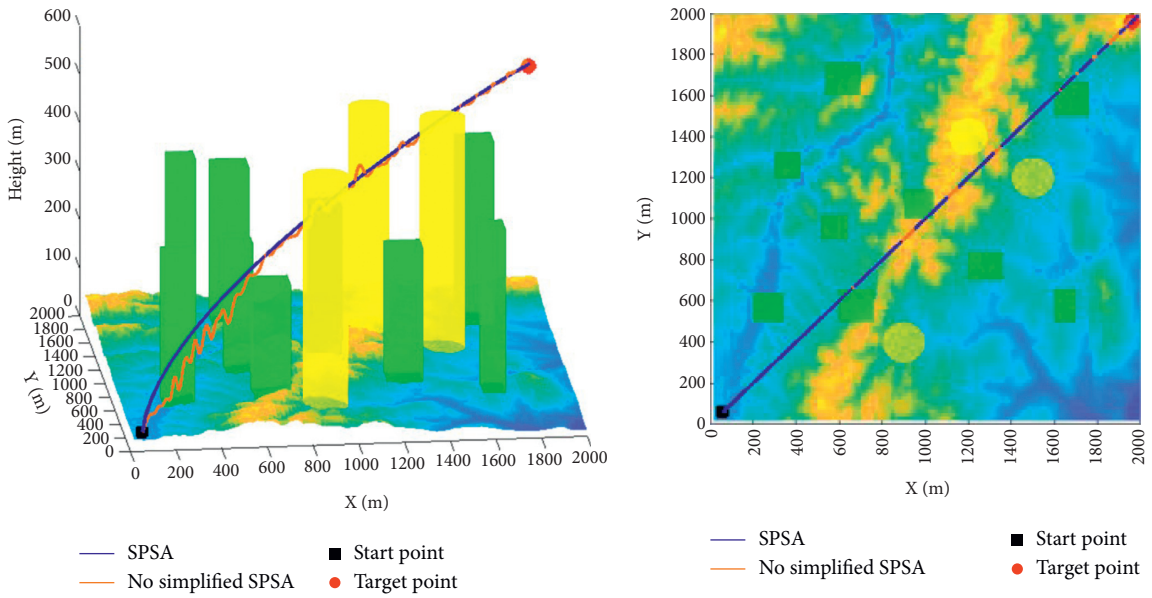

Figure 6: Comparison before and after simplifying nodes.

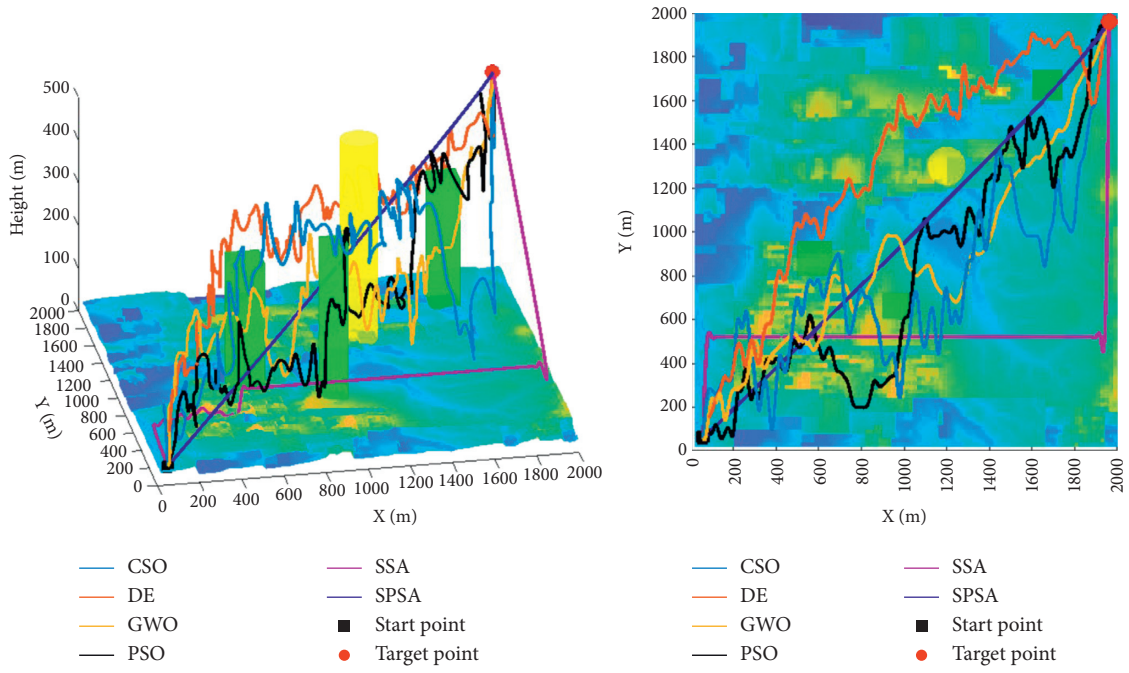

Figure 7: Comparison of path planning trajectories of different algorithms in environment 1.

optimization algorithm is $O(M \times N \times D)$ considering the worst case. The complexity of adaptive variable speed escape search is $O(S\ D \times M \times N \times D)$, and the computational complexity of path simplification and smoothing is $O(D)$ only for the final converged minimum value path. Since the SPSA does not improve the structure of the sparrow algorithm, the maximum computational complexity is $O(N \times D \times A \times B + M \times N \times D)$, so the computational complexity of this SPSA is $O(N \times D \times A \times B + M \times N \times D)$. It shows that the SPSA also has a fast convergence rate.

6.2. Analysis of Experimental Results. To verify the effect of the simplified node on the path, the experimental results are shown in Figure 6. The red line is the trajectory curve of the path without the simplified node after three times of spline interpolation, and the blue line is the curve after three times

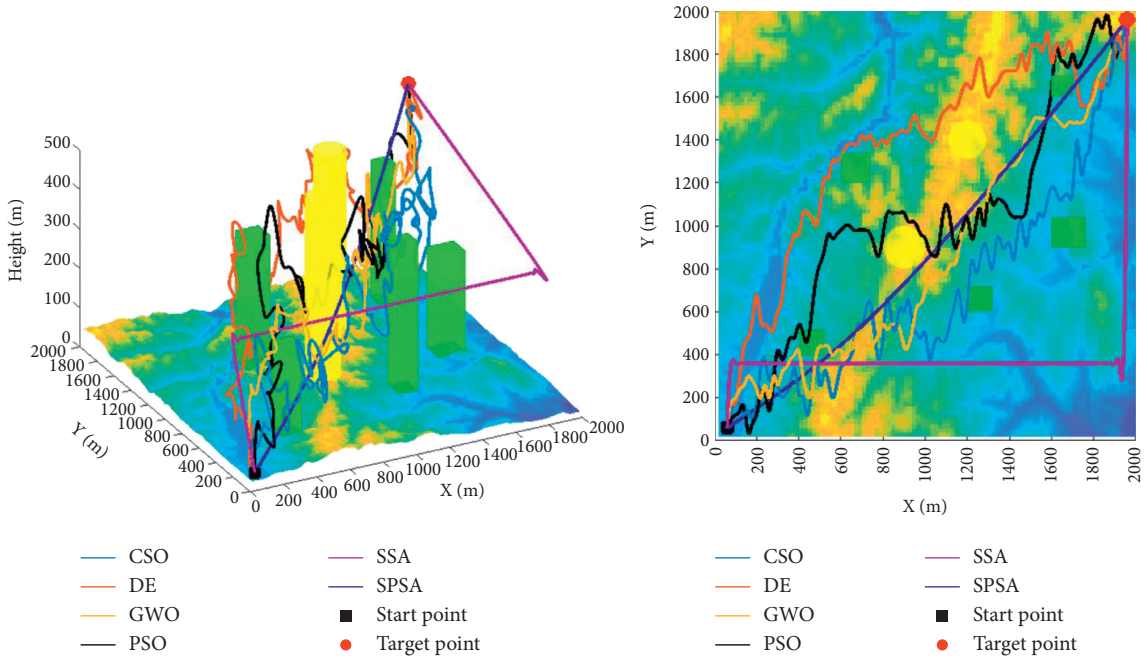

Figure 8: Comparison of path planning trajectories of different algorithms in environment 2.

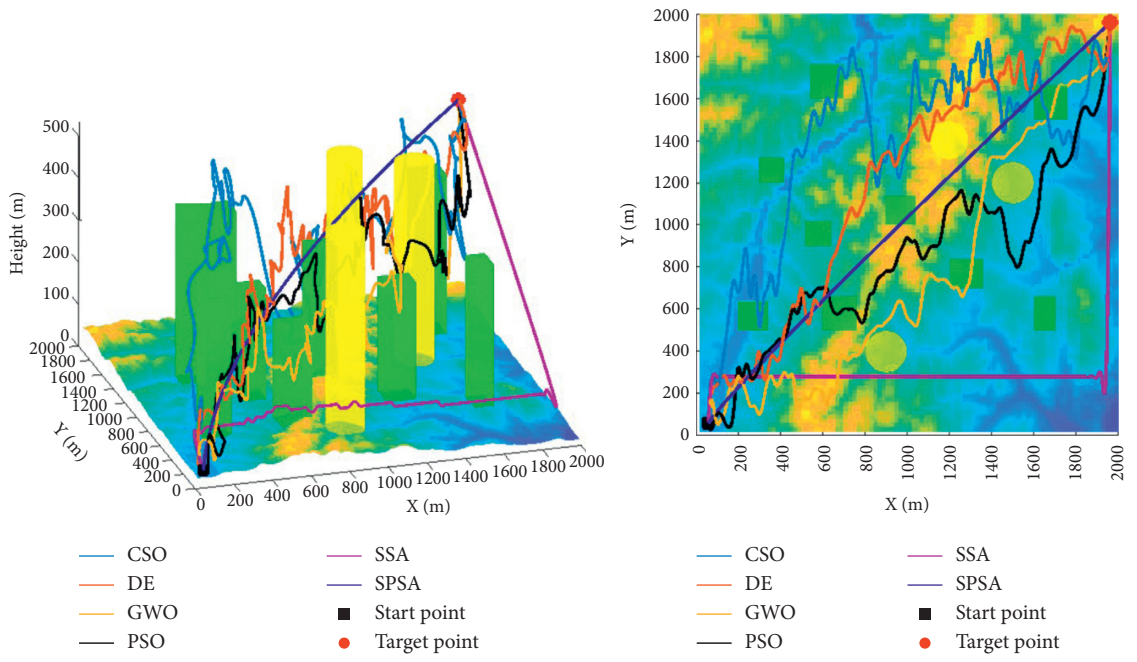

Figure 9: Comparison of path planning trajectories of different algorithms in environment 3.

of spline interpolation after the simplified node. It can be seen that there is almost no oscillation in the x-axis and y-axis directions, and there is oscillation in the z-axis direction for the nodes of the un-simplified path, which shows that the simplified path is more suitable as the flight path of the UAV.

It can be seen in Figures 7–9 that the SPSA has better convergence than other algorithms. The SSA has better smoothness than the paths planned by CSO, DE, GWO, and PSO algorithms, which all show different degrees of oscillation, and the altitude of the flight trajectory keeps changing with less smoothness. Oscillation optimizes and simplifies the nodes, improves the smoothness of the trajectory, and is more suitable for smooth operation as a UAV. The fitness values of SPSA in Figures 10–12 can converge to the optimal value faster than other algorithms. For the presence of different obstacles in the UAV-specific constrained

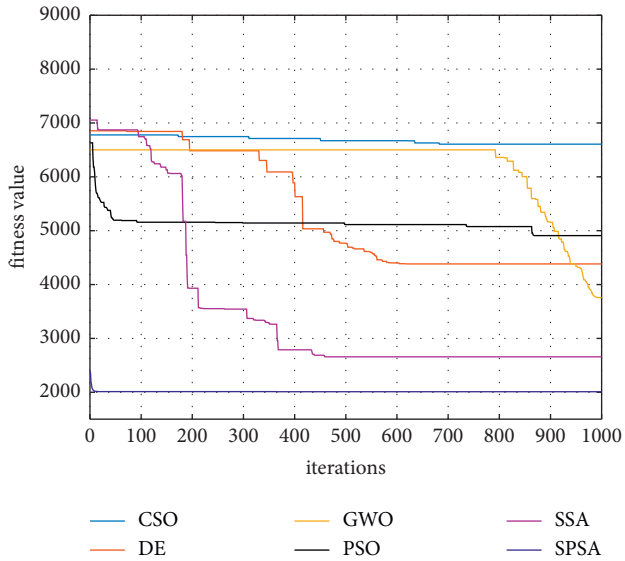

Figure 10: Adaptation of different algorithms in environment 1 with iterations.

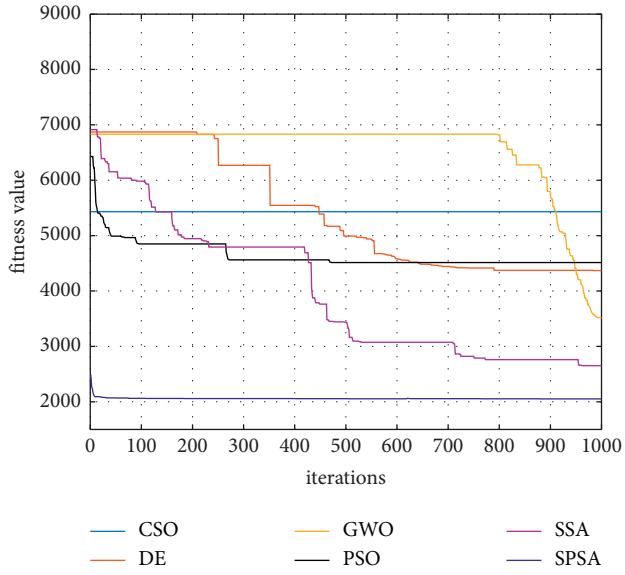

Figure 11: Adaptation of different algorithms in environment 2 with iterations.

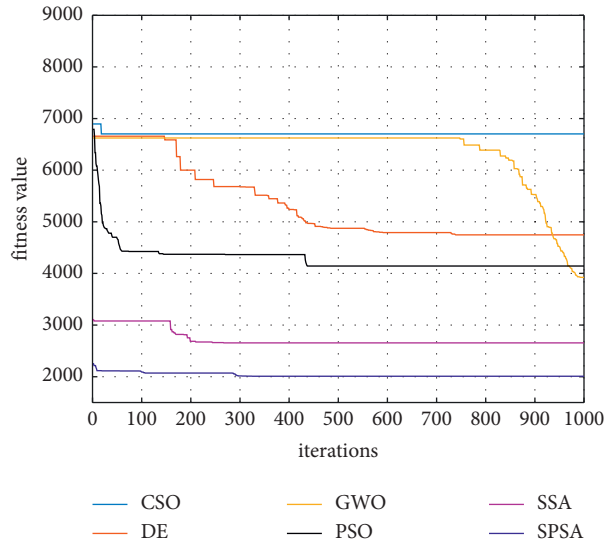

Figure 12: Adaptation of different algorithms in environment 3 with iterations.

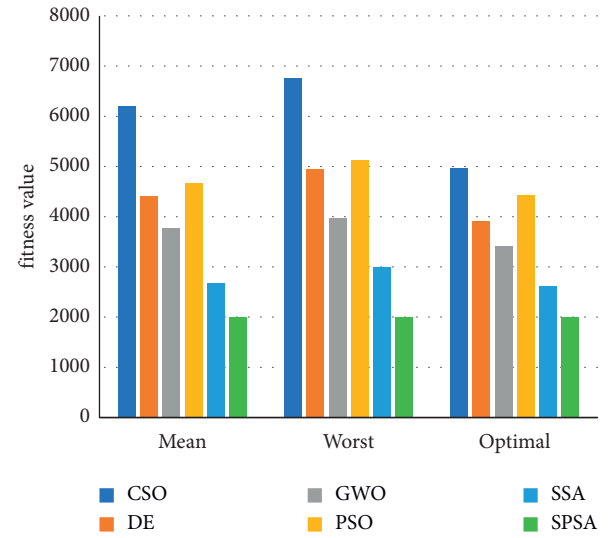

Figure 13: Comparison of the results of path planning in environment 1.

environment, increasing the influence of the start-end line on the producer strengthens the searchability of the individuals near the start-end line, leading to a path that can converge quickly. The SPSA has significantly better results in the first iteration due to the start-end line guidance, which makes the path shorter than other algorithms. For the stability of the actual UAV flight path, the adaptive oscillation optimization makes it possible that there is no excessive distance difference between two adjacent path nodes, making the path length much shorter and achieving better results than other algorithms. The exploration of the optimal path is avoided by oscillatory optimization to avoid missing the optimal path. Figures 13–15 show that the SPSA is highly robust in different environments.

From the mean, worst, optimal, and standard deviation of different algorithms in Table 2, we can see that the CSO algorithm has the worst convergence effect in path planning. Because in the CSO algorithm, there are differences in the routes between each individual, the smooth paths may be affected by other paths leading to oscillations instead of easy convergence. DE algorithm, PSO algorithm, and SSA have better convergence effect as the iteration increases, and the iterative oscillation of the path makes it difficult to converge

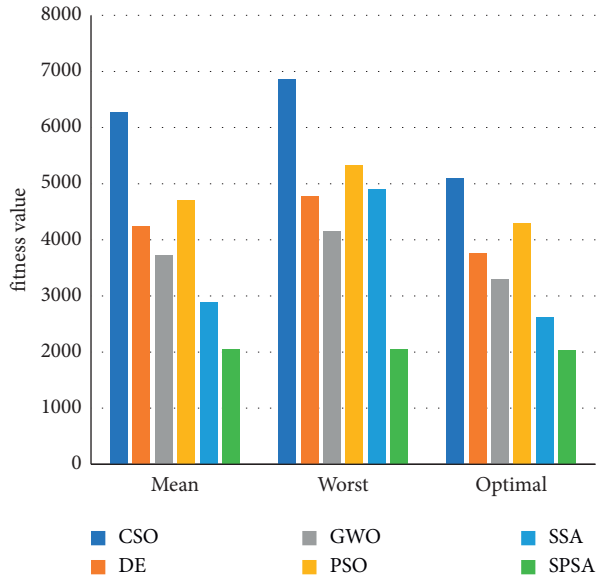

Figure 14: Comparison of the results of path planning in environment 2.

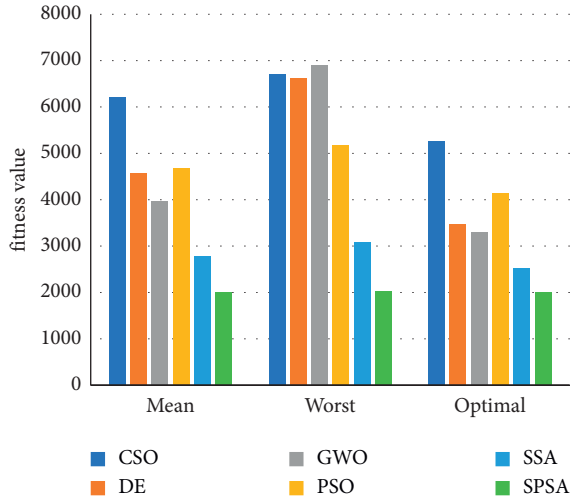

Figure 15: Comparison of the results of path planning in environment 3.

Table 2: Simulation results of various algorithms.

	CSO	DE	GWO	PSO	SSA	SPSA
Mean	6201.82	4408.46	3763.38	4676.34	2686.43	**2008.42**
Worst	6756.89	4938.99	3973.66	5128.23	2993.09	**2009.55**
Optimal	4970.68	3907.56	3421.28	4427.57	2614.56	**2006.73**
Standard	413.09	243.01	83.02	218.58	127.61	**0.89**
Mean	6271.97	4233.72	3718.73	4710.89	2882.40	**2043.68**
Worst	6854.16	4781.97	4160.03	5323.78	4895.46	**2057.16**
Optimal	5105.15	3757.25	3291.77	4304.02	2616.61	**2028.70**
Standard	465.75	244.00	177.03	233.50	631.23	**8.10**
Mean	6221.22	4562.37	3958.70	4683.83	2784.55	**2004.95**
Worst	6702.69	6625.81	6902.46	5182.99	3081.59	**2024.60**
Optimal	5251.67	3473.63	3294.27	4143.35	2525.77	**1999.51**
Standard	381.40	752.85	578.59	297.68	183.07	**5.84**

to a better path. By comparison, the SPSA has better convergence stability and the lowest fitness value due to the directional nature of the path search, which strengthens the ability of individuals to search near the start-end line. The smoothness of the paths is improved by continuously optimizing the paths and avoiding the oscillating paths from interfering with the convergence of subsequent paths.

7. Conclusions

The traditional SSA algorithm is suitable for the optimization search algorithm performed in an unconstrained environment. In the real environment, various environmental factors will affect the SSA algorithm planning path, resulting in poor convergence of the algorithm and loss of more individuals. To address this situation, strengthening the search efficiency and improving the path smoothing are the main problems that need to be solved for UAV path planning. In this study, we combine the sparrow search algorithm and particle swarm algorithm to propose the SPSA, and the main contributions of this study are as follows:

(1) Path initialization has a guiding role in the convergence of the subsequent algorithm. The next node's range is calculated according to the spatial geometry, and then, the node that can be moved within the range is randomly selected as the next node. This method avoids the interference of the initialization path trajectory oscillation on the convergence of the subsequent algorithm and can ensure the search capability of the algorithm.

(2) Position update of the producer. The start-end line as the globally optimal path, combined with the particle swarm algorithm, improves the position update of the producer in the sparrow particle swarm algorithm, strengthening the searchability of the algorithm near the start-end line.

(3) The searchability of discovering dangerous individuals. Due to the existence of A^+ in the position update formula of the sparrow found to be dangerous in the algorithm, the path may not reach the target point and lose the searched individuals. Inspired by the particle swarm algorithm, a variable speed escape search method is proposed to improve the sparrow search algorithm when the next node is unreachable, improve the path's exploration ability, and reduce the loss of the searched individuals.

(4) Adaptive oscillation optimization. For the randomness on path selection resulting in oscillation, an adaptive oscillation optimization method is proposed to reduce the interference of path oscillation to the subsequent path search and improve the smoothness of the path.

(5) Simplify node and smoothing processing. Node optimization can reduce the path turning points and improve the smoothness of the path, making the path suitable as the actual flight trajectory of the UAV.

Through simulation experiments, it can be concluded that the sparrow particle swarm algorithm has better convergence and stronger searchability in complex environments, and the smoothed path is suitable as the flight path of the UAV.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partly supported by the National Natural Science Foundation of China (NSFC, Project no. 61973209) and Shanghai Municipal Information Development Special Fund-5G+ Intelligent Manufacturing Demonstration Line Innovation Laboratory Construction (Shanghai Municipal Commission of Economy and Informatization Project no. 202001008).

References

[1] F. Duchoň, A. Babinec, M. Kajan et al., "Path planning with modified a star algorithm for a mobile robot," *Procedia Engineering*, vol. 96, pp. 59–69, 2014.

[2] I. Noreen, A. Khan, and Z. Habib, "A comparison of RRT, RRT* and RRT*-Smart path planning algorithms," vol. 8, 2016.

[3] J. J. Kuffner and S. M. LaValle, "RRT-connect: an efficient approach to single-query path planning," in *Proceedings of the 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 2, pp. 995–1001, San Francisco, CA, USA, April 2000.

[4] Y.-b. Chen, G.-c. Luo, Y.-s. Mei, J.-q. Yu, and X.-l. Su, "UAV path planning using artificial potential field method updated by optimal control theory," *International Journal of Systems Science*, vol. 47, no. 6, pp. 1407–1420, 2016.

[5] Z. Wu, J. Li, J. Zuo, and S. Li, "Path planning of UAVs based on collision probability and kalman filter," *IEEE Access*, vol. 6, pp. 34237–34245, 2018.

[6] X. Hu, B. Pang, F. Dai, and K. H. Low, "Risk assessment model for UAV cost-effective path planning in urban environments," *IEEE Access*, vol. 8, Article ID 150162, 2020.

[7] R. K. Dewangan, A. Shukla, and W. W. Godfrey, "Three dimensional path planning using Grey wolf optimizer for UAVs," *Applied Intelligence*, vol. 49, no. 6, pp. 2201–2217, 2019.

[8] W. Deng, R. Chen, B. He, Y. Liu, L. Yin, and J. Guo, "A novel two-stage hybrid swarm intelligence optimization algorithm and application," *Soft Computing*, vol. 16, no. 10, pp. 1707–1722, 2012.

[9] Y.-N. Ma, Y.-J. Gong, C.-F. Xiao, Y. Gao, and J. Zhang, "Path planning for autonomous underwater vehicles: an ant colony algorithm incorporating alarm pheromone," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 141–154, 2019.

[10] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

[11] X. Meng, Y. Liu, X. Gao, and H. Zhang, "A new bionspired algorithm: chicken swarm optimization," Y. Tan, Y. Shi, and C. A. C. Coello, Eds., in *Proceedings of the Advances in Swarm Intelligence*, vol. 8794, pp. 86–94, Springer International Publishing, Hefei, China, October 2014.

[12] H.-b. Duan, X.-y. Zhang, J. Wu, and G.-j. Ma, "Max-min adaptive ant colony optimization approach to multi-UAVs coordinated trajectory replanning in dynamic and uncertain environments," *Journal of Bionics Engineering*, vol. 6, no. 2, pp. 161–173, 2009.

[13] H. Shiri, J. Park, and M. Bennis, "Remote UAV online path planning via neural network-based opportunistic control," *IEEE Wireless Communications Letters*, vol. 9, no. 6, pp. 861–865, 2020.

[14] C. Qu, W. Gai, M. Zhong, and J. Zhang, "A novel reinforcement learning based grey wolf optimizer algorithm for unmanned aerial vehicles (UAVs) path planning," *Applied Soft Computing*, vol. 89, Article ID 106099, 2020.

[15] C. Duan, J. Feng, and H. Chang, "Meteorology-aware path planning for the UAV based on the improved intelligent water drops algorithm," *IEEE Access*, vol. 9, pp. 49844–49856, 2021.

[16] X. Liu, X. Du, X. Zhang, Q. Zhu, and M. Guizani, "Evolution-algorithm-based unmanned aerial vehicles path planning in complex environment," *Computers & Electrical Engineering*, vol. 80, Article ID 106493, 2019.

[17] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, 2020.

[18] J. Yuan, Z. Zhao, Y. Liu et al., "DMPPT control of photovoltaic microgrid based on improved sparrow search algorithm," *IEEE Access*, vol. 9, pp. 16623–16629, 202.

[19] C. Zhang and S. Ding, "A stochastic configuration network based on chaotic sparrow search algorithm," *Knowledge-Based Systems*, vol. 220, Article ID 106924, 2021.

[20] Z. Zhang, R. He, and K. Yang, "A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm," *Advanced Manufacturing*, 2021.

[21] G. Liu, C. Shu, Z. Liang, B. Peng, and L. Cheng, "A modified sparrow search algorithm with application in 3d route planning for UAV," *Sensors*, vol. 21, no. 4, p. 1224, 2021.

[22] C. Ouyang, D. Zhu, and F. Wang, "A learning sparrow search algorithm," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 3946958, 23 pages, 2021.

[23] O. Y. Abdulhammed, "Load balancing of IoT tasks in the cloud computing by using sparrow search algorithm," *The Journal of Supercomputing*, 2021.

[24] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942–1948, Perth, WA, Australia, November 1995.

[25] H. S. Dewang, P. K. Mohanty, and S. Kundu, "A robust path planning for mobile robot using smart particle swarm optimization," *Procedia Computer Science*, vol. 133, pp. 290–297, 2018.

[26] P. K. Das, H. S. Behera, and B. K. Panigrahi, "A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning," *Swarm and Evolutionary Computation*, vol. 28, pp. 14–28, 2016.

[27] S. Shao, Y. Peng, C. He, and Y. Du, "Efficient path planning for UAV formation via comprehensively improved particle swarm optimization," *ISA Transactions*, vol. 97, pp. 415–430, 2020.

[28] Y. Zhang, D. Gong, and J. Zhang, "Robot path planning in uncertain environment using multi-objective particle swarm optimization," *Neurocomputing*, vol. 103, pp. 172–185, 2013.

[29] Y.-Q. Qin, De-B. Sun, Li Ning, and Yi-G. Cen, "Path planning for mobile robot using the particle swarm optimization with mutation operator," in *Proceedings of the 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.04EX826)*, vol. 4, pp. 2473–2478, Shanghai, China, August 2004.

[30] V. Roberge, M. Tarbouchi, and G. Labonte, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 132–141, 2013.

[31] G. Li and W. Chou, "Path planning for mobile robot using self-adaptive learning particle swarm optimization," *Science China Information Sciences*, vol. 61, no. 5, Article ID 052204, 2018.

[32] M. D. Phung, C. H. Quach, T. H. Dinh, and Q. Ha, "Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection," *Automation in Construction*, vol. 81, pp. 25–33, 2017.

[33] J. L. Foo, J. Knutzon, V. Kalivarapu, J. Oliver, and E. Winer, "Path planning of unmanned aerial vehicles using B-splines and particle swarm optimization," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 4, pp. 271–290, 2009.

[34] S. Thabit and A. Mohades, "Multi-robot path planning based on multi-objective particle swarm optimization," *IEEE Access*, vol. 7, pp. 2138–2147, 2019.

[35] Y. Chen, J. Yu, Y. Mei, Y. Wang, and X. Su, "Modified central force optimization (MCFO) algorithm for 3D UAV path planning," *Neurocomputing*, vol. 171, pp. 878–888, 2016.