In this paper, a lens learning sparrow search algorithm (LLSSA) is proposed to improve the defects of the new sparrow search algorithm, which is random and easy to fall into local optimum. The algorithm has achieved good results in function optimization and has planned as a safer and less costly path to the three-dimensional UAV path planning. In the discoverer stage, the algorithm introduces the reverse learning strategy based on the lens principle to improve the search range of sparrow individuals and then proposes a variable spiral search strategy to make the follower’s search more detailed and flexible. Finally, it combines the simulated annealing algorithm to judge and obtain the optimal solution. Through 15 standard test functions, it is verified that the improved algorithm has strong search ability and mining ability. At the same time, the improved algorithm is applied to the path planning of 3D complex terrain, and a clear, simple, and safe route is found, which verifies the effectiveness and practicability of the improved algorithm.

1. Introduction

In recent decades, swarm intelligence algorithms have been gradually explored by researchers. The main principles of these algorithms are mostly to solve some optimization problems by simulating or revealing some natural phenomena or processes, such as Particle Swarm optimization (PSO) [1], Grey Wolf Optimizer (GWO) [2], Beetle Swarm Optimization (BSO) [3]. The sparrow search algorithm (SSA) is a group of intelligence algorithms proposed by Xue and Shen [4] in 2020 to simulate the feeding process of sparrows. It has few parameters, simple principles, and easy implementation and is superior to other algorithms in many optimization functions. However, for some complex optimization problems, the algorithm has some problems, such as low search accuracy and slow convergence rate. After the sparrow search algorithm was put forward, many scholars studied and applied it carefully and made improvements. Xin et al. [5] proposed a chaotic sparrow search algorithm (CSSA), which uses tent mapping based on random variables to initialize the population, so as to make the individual position of sparrows more evenly distributed and improve the diversity of the population; in the process of optimization, disturbance and mutation mechanism are used to prevent the algorithm from falling into the local extremum state and improve the global search performance of the algorithm. The effectiveness and feasibility of the algorithm are verified by 12 test functions and image segmentation problems. In addition, they also proposed an improved sparrow search algorithm for image segmentation [6]. Bird swarm algorithm (BSA) [7] is used to optimize the sparrow search algorithm. The benchmark function and multithreshold image segmentation based on interclass variance and Kapur’s entropy verify that the improved algorithm has strong search ability and development ability. Mao et al. [8] proposed an improved sparrow search algorithm which combined Cauchy mutation and reverse learning. At first, a sin chaotic mechanism with infinite folding times is used to initialize the population, and the global optimal solution of the previous generation is introduced into the position updating formula, which speeds up the information exchange of the algorithm. At the same time, the adaptive weight strategy is introduced to coordinate the local and global search ability. Then, the fusion of Cauchy mutation and reverse learning strategy is introduced to make disturbance mutation in the optimal position to
produce new solutions with higher quality, which improves the ability of the algorithm to jump out of the local optimum. Eight test functions are used to verify that the algorithm has been greatly improved in global optimization. Liu et al. [9] proposed a modified sparrow search algorithm with application for 3D route planning for UAV. Chaos strategy is introduced to enhance the diversity of the population, and adaptive inertia weight is used to balance the convergence speed and exploration ability of the algorithm. Finally, Cauchy Gauss mutation strategy is used to get rid of the stagnation ability of the later algorithm. Through the comparison of several algorithms in 3D path planning, the improved sparrow search algorithm has strong search ability and the planned route is relatively safe.

Currently, research on sparrow search algorithm has been proposed successively, but the proposed algorithms lack a global improvement mechanism. Although the mapping mechanism is uniform, it is not stable under the sparrow search algorithm with good global results, and it increases the workload of the population. Therefore, a search mechanism with global vision and flexibility is needed to improve the search ability of the sparrow search algorithm. For the above analysis, a lens learning sparrow search algorithm (LLSSA) is proposed. This algorithm uses a reverse learning strategy based on lens learning strategy in the discoverer location update formula to improve the algorithm’s vision in the optimal space. A variable spiral search strategy is proposed to enable followers to search in more detail and flexibility. Finally, the current solution is refined by fusing simulated annealing algorithm to find a higher quality solution. Through 15 standard test functions, the effectiveness of LLSSA is verified, and a distinct, simple, and satisfying route is found in the complex three-dimensional path planning.

2. Sparrow Search Algorithm

During the food search process, the sparrow population is divided into two roles: discoverer and follower, which conduct behavioral strategies separately. Discoverers are generally 0.2 of the population size and are the guides of the individual population, leading other individuals in the search for things, so the role of the discoverer is crucial in the population. They have a wide search range. The location update formula for the discoverer is as follows:

\[
X_{t+1}^{d_{ij}} = \begin{cases} 
X_{t}^{d_{ij}} \cdot \exp\left(\frac{-h}{a \cdot M}\right), & \text{if } R_2 < ST, \\
X_{t}^{d_{ij}} + Q \cdot L, & \text{if } R_2 \geq ST.
\end{cases}
\]  

In formula (1), \( h \) represents the current number of iterations and \( M \) is the maximum number of iterations. \( X_{t}^{d_{ij}} \) denotes the current position of the \( i \)-th sparrow in the \( j \)-th dimension. \( a \) is a random number between 0 and 1. \( Q \) is a random number from a normal distribution. \( L \) represents a \( 1 \times D \) matrix with all elements 1. \( R_2 \) and \( ST \) respectively represent warning and safety values, and \( R_2 \in [0, 1], ST \in [0.5, 1] \). When \( R_2 < ST \), this indicates that the community environment is in a safe state at this time, no predators are found around, and the discoverers can search for food on a wide scale. When \( R_2 \) is greater than or equal to \( ST \), this indicates that the individuals within the cluster have found a predator and issued an alarm so that all individuals of the population will make antipredation behavior and the discoverer will lead the follower to a safe location.

To obtain good quality food, the followers would closely follow the discoverers. Some of these followers oversee the discoverers and those discoverers with high rates of predation for food, thus increasing their own nutrition. The positional update for the followers is described below:

\[
X_{t+1}^{f_{ij}} = \begin{cases} 
Q \cdot \exp\left(\frac{X_{t}^{w_{ij}} - X_{t}^{d_{ij}}}{\varepsilon}\right), & \text{if } i > \frac{n}{2}, \\
X_{t}^{f_{ij}} + |X_{t}^{d_{ij}} - X_{t}^{f_{ij}}| \cdot A^* \cdot L, & \text{otherwise}.
\end{cases}
\]  

In (2), \( X_p \) is the optimal position currently occupied by the discoverer, and \( X_{t}^{w_{ij}} \) denotes the current worst position. \( A \) is a matrix of \( 1 \times D \) in which an element is only 1 or -1, with \( A^* = A^T(AA^T)^{-1} \). When \( i > (n/2) \), it indicates that the sparrow population is aware of danger, at which point they make antipredation behavior, and the mathematical expression is as follows:

\[
X_{t+1}^{f_{ij}} = \begin{cases} 
X_{t}^{f_{ij}} + \beta \cdot |X_{t}^{f_{ij}} - X_{t}^{f_{ij}}|, & \text{if } f_i \neq f_g, \\
X_{t}^{f_{ij}} + K \cdot \left(\frac{X_{t}^{w_{ij}} - X_{t}^{f_{ij}}}{f_i - f_g} + \varepsilon\right), & \text{if } f_i = f_g.
\end{cases}
\]  

In (3), \( X_{t}^{f_{ij}} \) is the current global optimum. \( \beta \) is the control step size parameter and is a random number obeying a normal distribution with mean 0 and variance 1. \( K \in [-1, 1] \) is a random number, and \( K \) represents the direction in which the sparrow moves while also controlling the moving step size. \( f_i \) indicates the fitness value of the current sparrow individuals. \( f_p \) and \( f_w \) are the optimal and worst fitness values within the current search scope, respectively. \( \varepsilon \) is the smallest real number, preventing the occurrence of 0 in the denominator. When \( f_i \neq f_g \) indicates that the current sparrow is at the boundary of the population and vulnerable to predator attack, the location needs to be adjusted. \( f_i = f_g \), which indicates that individual sparrow in the interior of the population is aware of the danger and need to be close to other sparrows to avoid the danger.

3. Inverse Learning Strategy Based on Lens Imaging

It is because of the broad search range and flexible search methods that discoverers can lead the group to find good food. Once the discoverer is trapped in the local optimum, the performance of the overall algorithm will be degraded, so improving the search horizon of the discoverer is particularly important. General learning strategies have achieved good results in some optimization algorithms, but have little
effect on the performance of the algorithm, because the general learning strategies carry out reverse solutions in local space, which enriches the diversity of the population, but the search scope is narrow and loses flexibility. In order to improve the searching ability of the discoverer, a reverse learning strategy based on lens imaging [10] is proposed and applied to the individual update position formula of the discoverer. The application of this strategy allows new individuals to be acquired for each iteration, resolving the problems left over from previous general learning strategies. The principle is described as follows.

**Definition 1** (reverse point). Set \( X = (x_1, x_2, \ldots, x_D) \) as a point in D-dimensional space, and \( x_j \in [a_j, b_j], j = 1, 2, \ldots, D; \) then, the reverse point of \( X \) is \( X' = (x'_1, x'_2, \ldots, x'_D) \), and \( x'_j = a_j + b_j + x_j \).

**Definition 2** (base point). Suppose there are several points \( O_1, O_2, \ldots, O_m \), for any point \( X = (x_1, x_2, \ldots, x_D) \) with its reverse point \( X' = (x'_1, x'_2, \ldots, x'_D) \) to \( O_i (i = 1, 2, \ldots, m) \). Euclidean distances are \( d_j \) and \( d'_j \), so \( k = (d_i/d'_i) \), and \( k = 1, 2, \ldots, n \), and \( O_i \) is called the base point of \( X \) and \( X' \) when \( k = i \).

Using one-dimensional space as an example, assume that an individual \( P \) with a height of \( h \) is projected as \( x_p \) (the globally optimal individual) on the coordinate axis, and a lens with a focal length of \( f \) is placed at the base position \( O \) (the midpoint of \( [a, b] \) in the figure). An image \( P' \) with a height of \( h' \) can be obtained through the lens imaging process, and its projection on the coordinate axis is \( x'_p \). At this time, \( x'_p \) is such that \( x_p \) gives rise to new individuals through a back learning strategy based on the principles of lens imaging. The diagram is shown in Figure 1.

As shown in Figure 1, the globally optimal individual \( x_p \) is giving its corresponding reversal point \( x'_p \) obtained from the principle of lens imaging:

\[
(a + b/2) - x_p \quad x'_p = \frac{(a + b/2) - x_p}{h'} \quad (4)
\]

Let \( (h/h') = k \), \( k \) be the scaling factor, and the inverse point obtained upon transformation is

\[
x'_p = \frac{a + b}{2} + \frac{a + b}{2k} - \frac{x_p}{k} \quad (5)
\]

It follows that when \( k = 1 \),

\[
x'_p = a + b - x_p \quad (6)
\]

Form (6) is called general reverse learning strategy. From the above formulas, it can be seen that general learning strategy is only a specific case of lens imaging learning strategy, and the new individuals obtained by general reverse learning strategy are fixed each time. In high-dimensional complex functions, new individuals with fixed ranges are also likely to fall into local optimum with monotonicity; by adjusting the parameter \( k \), new individuals based on lenses imaging learning strategies are dynamic, which improves population diversity. Extending the formula to D-dimensional space is available:

\[
x'_p = \frac{a_j + b_j}{2} + \frac{a_j + b_j}{2k} - \frac{x'_p}{k} \quad (7)
\]

In (7), \( x'_p \) versus \( x'_p \) are the \( j \)-th dimension components of \( x_p \) versus \( x'_p \), and \( a_j \) and \( b_j \) represent the \( j \)-th dimension components at the upper and lower bounds of the decision variable, respectively.

**4. Variable Spiral Search Strategy**

While it is convenient for followers to follow the discoverer in updating their location, it is easy for followers to search blindly and singularly. Inspired by the rotation of the whale algorithm, this paper introduces a variable spiral location update strategy, which allows followers to have a variety of search paths to better update the location, while balancing the global and local search of the algorithm. The spiral search diagram is shown in Figure 2.

In the follower location update process, the helix parameter \( z \) cannot be a fixed shape, which results in monotonous search methods and the possibility of falling into local optimum, thus weakening the search ability of the algorithm. The parameter \( z \) is designed as an adaptive variable to dynamically adjust the spiral shape of follower search, which expands the follower’s ability to explore unknown areas, makes the algorithm search more efficient, and improves the algorithm’s global search performance. The formula for the variable spiral position update strategy is as follows:

\[
x'_i = \begin{cases} 
\exp \left( \frac{X'_{i,n} - X_{i,j}}{2} \right), & \text{if } i > \frac{n}{2} \\
X'_{i,j} + X_{i,j} - X_{i,j+1} \cdot A & \text{otherwise.}
\end{cases}
\]

\[
z = \exp \left( \frac{1}{\epsilon (1 - (i/u_{\text{max}}))} \right)
\]

The parameter \( Z \) changes according to the number of iterations. The change of \( Z \) is determined by the exponential function based on \( \epsilon \) in formula (8), and the size and amplitude of the helix are dynamically adjusted according to the
nature of cos function. \( k \) is the coefficient of change, which is 5 in this paper. Here, \( L \) is a uniformly distributed random number of \([-1, 1]\). Follower location updates range from large to small, find more quality solutions in the early stage, reduce the increase of search useless work in the later stage, and improve the global optimal search performance of the algorithm. At the same time, according to the spiral characteristics, the optimization accuracy of the algorithm is improved.

5. Simulated Annealing Algorithm

Simulated annealing algorithm, derived from the principle of solid annealing, is a probability-based algorithm and a stochastic global optimization algorithm proposed by Metropolis et al. [11]. The simulated annealing algorithm is a classical algorithm and has been studied extensively. Its principle is not described here. The simulated annealing algorithm combines with the improved sparrow algorithm. The simulated annealing algorithm is used to refine the solution found by the sparrow search algorithm each time. It uses the sudden jump of the simulated annealing to get rid of the interference of local optimum and extracts the solution each time. The algorithm has good judgment ability and improves the quality of each solution and tends to be global.

6. Lens Learning Sparrow Search Algorithm

Sparrow search algorithm has deficiencies in the case of multidimensional complex functions. Therefore, this paper presents a lens learning sparrow search algorithm. This algorithm applies the reverse learning strategy based on the principle of lenses to the discoverer stage of sparrow search, improving the diversity of the population and expanding the search range of individual sparrows. In the follower search phase, a variable spiral search strategy is introduced, which makes the follower search more detailed and flexible. Finally, this algorithm is combined with the simulated annealing algorithm, and the previously found solutions are filtered again to obtain the optimal solution. The algorithm works as follows:

1. Initialize the population location, number, and number of iterations.
2. Calculate the fitness function of each group to get the corresponding maximum and minimum values to determine the best and worst position.
3. Calculate the alert value and update the location of the discoverer based on the alert value.
4. Use Levy flight to update the location of the discoverer.
5. Update the position of followers.
6. Perform another sine-cosine search of the follower’s location and update the location.
7. Update the locations of sparrows that are aware of danger.
8. Perform simulated annealing operation, as follows:
   \[
   T = \frac{\text{fit}(fg)}{\ln 5},
   \]
   \[
   T_{k+1} = \lambda T_k.
   \]
9. Determine the \( P_i \) of each particle at the current temperature \( T \), and the formula is as follows:
   \[
   T_{\text{fit}}(p_i) = \frac{e^{-\left(\frac{\text{fit}(p_i) - \text{fit}(fg)}{\alpha}\right)}}{\sum_{i=1}^{N} e^{-\left(\frac{\text{fit}(p_i) - \text{fit}(fg)}{\alpha}\right)}}.
   \]
10. Determine a new global optimal value \( f_g \) from the current sparrow individual \( P_i \).
11. Determine whether the termination condition is met or not; if not, return to step (2). If so, proceed to the next step.
12. Output best position and minimum cost. The specific pseudocode is as follows (Algorithm 1).

7. Effectiveness Analysis

In order to better see the advantages of the improved strategy, the SPHEREFUNCTION function is taken as an example to analyze the effectiveness of the strategy of the LLSSA algorithm. Let the number of populations and the number of iterations be 50 and 10, respectively, and the individual distribution map after 10 iterations of each algorithm is obtained. The specific distribution maps are shown in Figures 3 and 4.

It can be seen from Figures 3 and 4 that the particles of LLSSA converge faster, most of the individuals are close to the optimal value, and the search range is wider, which is beneficial to jump out of the local optimum, while the particles of SSA converge slower, and the search range is smaller, which is easy to fall into the local optimum state in
the high-dimensional complex function. This shows that LLSSA has flexibility in the optimization mechanism and verifies the effectiveness of the three strategies.

8. Time Complexity Analysis

For an algorithm, time complexity is an important consideration, and it is also an important means of judging the amount of calculations of an algorithm. Suppose the population size of the algorithm is $P$, the maximum number of iterations is $M$, and the dimension of the problem is $D$. From a macro point of view, the SSA algorithm is the same as other intelligent optimization algorithms, with a time complexity of $O(P \times M \times D)$, while the LLSSA algorithm does not add extra cycles in the optimization process, so in the entire algorithm, time complexity did not increase.

From a micropoint of view, suppose the ratio of discoverers is $r$, the calculation time of introducing reverse learning based on the lens principle is $t_1$, the calculation time of the variable spiral search strategy is $t_2$, and the calculation time of using the simulated annealing algorithm to update the optimal solution is $t_3$. It can be seen from the pseudocode

```plaintext
Input
M: the maximum iterations
PD: the number of producers
SD: the number of sparrows who perceive the danger
$R_2$: the alarm value
N: the number of sparrows
$\lambda$: attenuation parameter
Initialize a population of $N$ sparrows and define its relevant
Output: $X_{\text{best}}, f_g$
Initialize the population according to equation (5)
$t = 1$
While ($t < M$)
Rank the fitness values and find the current best individual and the current worst individual.
$R_3 = \text{rand}(1)$
For $i = 1: PD$
Using equation (7) and equation (8) update the finder’s location;
End for
For $i = (PD + 1): N$
Using equation (13) update the follower’s location;
End for
For $l = 1: SD$
Updating the position of a sparrow individual who is aware of danger according to equation (3);
End for
Perform simulated annealing
Get the current new location;
If the new location is better than before, update it.
$t = t + 1$
End while
Return: $X_{\text{best}}, f_g$
```

**Algorithm 1:** The framework of the LLSSA.
of the algorithm that the LLSSA algorithm adds $O(M \times P \times r \times t_1 \times D)$ in the discoverer phase and $O(M \times (1 - r) \times P \times t_2 \times D)$ in the follower position update phase. $O(M \times t_3)$ is added in the update stage of the optimal solution. It can be seen that the LLSSA algorithm has increased $O(M \times (P \times D(r(t_1 - t_2) + t_2) + t_3)$ compared with the SSA algorithm. However, it has not been improved in the order of magnitude, and the optimization efficiency and accuracy of the algorithm can be effectively improved. Therefore, the increased time complexity is significant and worthwhile.

9. Performance Testing

In order to further verify the feasibility and effectiveness of the improved sparrow search algorithm, this paper selects 15 standard test functions to verify its optimization performance. The specific information is shown in Table 1. In order to ensure the fairness of the experiment, all the algorithms are run on the computer with MATLAB 2018b, Windows 10 operating system, and 8G memory. The number of iterations of each algorithm is 200 and the number of population is 50. The initial temperature is $t = 50$, $\lambda = 0.75$, and the parameters of PSO algorithm are $C_1 = C_2 = 2$, $w = 0.729$.

At the same time, in order to exclude the contingency of each algorithm and enhance persuasion, the three indexes of optimal value, average value, and standard deviation of each algorithm’s search result are counted, and the three indexes are used to judge the optimization ability and stability of each algorithm.

As can be seen from Table 2, the LLSSA algorithm has the best search performance in $F_{1-7}, F_{10-11}, F_{15}$ and has better search ability than other algorithms. In the $F_9$ function, BSO is the best. In $F_{13-14}$, BSO can find the best one, but its stability is poor, and the optimization results on other functions are extremely poor. Therefore, BSO has some limitations in function optimization. LLSSA achieves good results on many function problems. It is inferior only to BSO in $F_{9}$, which shows that LLSSA has strong optimization ability. Other algorithms have achieved poor optimization results. Therefore, the introduction and fusion of simulated annealing make the sparrow search algorithm have strong local judgment ability, balance the local and global search, and improve the search ability of the algorithm.

To describe the convergence of each algorithm in each function, the average convergence of each algorithm in each function is counted, and the specific convergence effect is shown in Figure 5. From Figure 5, it can be clearly seen that LLSSA has excellent convergence performance in $F_{1-5}, F_8, F_{11-12}, F_{14}$ functions, which all have advantages in accuracy and speed. However, among $F_{6-7}, F_{9-10}, F_{15}$ functions, the convergence accuracy is high and the speed is slow, while it has advantages in other functions, but it is not significant. Generally speaking, LLSSA overcomes the limitations of the original algorithm search mechanism, has a reliable optimization scheme, and has greater advantages in speed and accuracy.

Comparing algorithms based on mean and standard deviation alone is not enough. To be more persuasive, statistical tests are needed to verify that the proposed improved algorithm has significant improvement advantages over other existing algorithms. To determine whether each result of LLSSA is statistically significantly different from the best results of other algorithms, the Wilcoxon rank sum test was used at the 5% significance level. The $P$ values calculated in the Wilcoxon rank sum test between LLSSA and other algorithms are given in Table 3 for all the benchmark functions. When $P < 5\%$, the HO hypothesis can be considered rejected, indicating a significant difference between the two algorithms. When $P < 5\%$, it can be assumed that the HO hypothesis is accepted, indicating that the difference between the two algorithms is not obvious; that is, the optimization performance of the two algorithms is equal. Therefore, an algorithm with equivalent performance in each function is marked as N/A, indicating that it is not applicable. The algorithm differences table is shown in Table 3.

From Table 3, there is a big difference between each algorithm and LLSSA as a whole. Only on the first two unimodal functions, the basic SSA itself can find the optimal value. Therefore, there is no significant difference between LLSSA, SSA, and CSSA, but only on the $F_{10}$ function, the $P$ value is greater than 0.05. The experimental results show that BSO has better optimization ability on this function.

10. Application and Analysis

The proposal and improvement of an algorithm are ultimately to be implemented in the actual project. Since the development of intelligent optimization algorithms, the improvement of multiple algorithms has been successfully applied to specific applications; for example: literature [12] proposed a novel metaheuristic based on integrating chaotic maps into a Henry gas solubility optimization algorithm, which achieved good results in welding beams and cantilever beams. Reference [13] proposed an adaptive metaheuristic algorithm based on decomposition in order to solve the multitarget complex problem in UAV flight. The above authors have all proposed suitable algorithms in actual problems. Therefore, this paper applies the LLSSA algorithm to the UAV path planning of three-dimensional complex terrain to verify the feasibility and practicability of the LLSSA algorithm.

In recent years, the problem of UAV path planning has been widely favored by researchers. UAV path planning is also one of the problems of automatic control. Common path planning methods include genetic algorithm [14], particle swarm and ant colony algorithm [15], artificial potential field method [16], and gray wolf algorithm, but for complex high-dimensional problems such as path planning, the functions of these algorithms are limited. Many factors considered are difficult for each algorithm. For complex path planning problems, the algorithm is very easy to meet the current minimum cost and give up the overall plan, lack of autonomous learning. Therefore, researchers have made different improvement measures for the deficiencies of each algorithm to make the optimized path more ideal. Aiming at the problem of path planning in three-dimensional complex
### Table 1: Test function table

| Function | DIM | Section | MIN   |
|----------|-----|---------|-------|
| $F_1(x) = \sum_{i=1}^{n} x_i^2$ | 30  | $[-100, 100]$ | 0     |
| $F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} x_i$ | 30  | $[-10, 10]$ | 0     |
| $F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$ | 30  | $[-100, 100]$ | 0     |
| $F_4(x) = \max_i \{\|x_i\|, 1 \leq i \leq n\}$ | 30  | $[-100, 100]$ | 0     |
| $F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$ | 30  | $[-30, 30]$ | 0     |
| $F_6 = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5|x_i|)^2 + (\sum_{i=1}^{n} 0.5|x_i|)^4$ | 30  | $[-5, 10]$ | 0     |
| $F_7(x) = \sum_{i=1}^{n} \|x_i\|$ | 30  | $[-100, 100]d$ | 0     |
| $F_8(x) = \sum_{i=1}^{n} x_i^2 + \text{random}[0, 1]$ | 30  | $[-1.28, 1.28]$ | 0     |
| $F_9(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{|x_i|})$ | 30  | $[-500, 500]$ | 0     |
| $F_{10} = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{|x_i|})$ | 30  | $[-500, 500]$ | 0     |

\begin{align*}
F_{11} &= (n/\pi) \left\{ 10 \sin(y_i) + \sum_{i=1}^{n-1} [y_i - 1]^2 [1 + 10 \sin^2(\pi y_i)] + (y_n - 1)^2 \right\} + y_i = 1 + (x_i + 1/4) \\
&\quad + \frac{1}{y_i - a,} \quad x_i > a, \\
&\quad \frac{1}{k(-x_i - a)^m,} \quad x_i < -a.
\end{align*}

\begin{align*}
u(x_i, a, k, m) &= \left\{ \begin{array}{ll}
0, & -a < x_i < a, \\
\frac{k(-x_i - a)^m,} & x_i < -a.
\end{array} \right.
\end{align*}

$F_{12} = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2 [1 + \sin^2(2\pi x_2)]$ | 2   | $[-10, 10]$ | 0     |

$F_{13}(x) = (1.5 - x_1 + x_1 x_2)^2 + (2.25 - x_1 + x_1 x_2)^2 + (2.625 - x_1 + x_1 x_2)^2$ | 2   | $[-4.5, 4.5]$ | 0     |

$F_{14} = 100x_1^2 - x_2^2 + (x_1 - 1)^2 + (x_1 - 1)^2 + 90(x_2 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$ | 4   | $[-10, 10]$ | 0     |

$F_{15}(x) = (0.002 + \sum_{i=1}^{25} (1/\pi + (x_1 - a_i)^2 + (x_2 - a_2)^2))^{-1}$ where \(a = \begin{pmatrix}
-32 & -16 & 0 & 16 & 32 & -32 & \ldots & 0 & 32 & 32 \\
-32 & -32 & -32 & -32 & -32 & -16 & \ldots & 32 & 32 & 32
\end{pmatrix}
$ | 2   | $[-65.536, 65.536]$ | 0.998 |

Figure 5: Continued.
Figure 5: Continued.
Figure 5: Convergence effect diagram of each algorithm. (a) $F_1$, (b) $F_2$, (c) $F_3$, (d) $F_4$, (e) $F_5$, (f) $F_6$, (g) $F_7$, (h) $F_8$, (i) $F_9$, (j) $F_{10}$, (k) $F_{11}$, (l) $F_{12}$, (m) $F_{13}$, (n) $F_{14}$, (o) $F_{15}$.

Table 2: Table of optimization results of each algorithm.

| Function  | Algorithm | Best     | Ave     | Std       |
|-----------|-----------|----------|---------|-----------|
| $F_1(x)$  | LLSSA     | 0        | 0       | 0         |
|           | BSO       | 8.15851  | 26.3851 | 14.9876   |
|           | CSSA      | 0        | 0       | 0         |
|           | SSA       | 0        | 5.192E−251 | 0       |
|           | GWO       | 1.5739E−36 | 4.5232E−34 | 7.3785E−34 |
|           | PSO       | 3.8599E−12 | 3.0697E−11 | 2.7076E−11 |
| $F_2(x)$  | LLSSA     | 0        | 0       | 0         |
|           | BSO       | 0.212219 | 1.8836  | 2.13278   |
|           | CSSA      | 0        | 3.8973E−160 | 2.1346E−159 |
|           | SSA       | 0        | 2.2022E−144 | 8.4437E−144 |
|           | GWO       | 1.6295E−25 | 2.9050E−23 | 4.4358E−23 |
|           | PSO       | 8.9056E−08 | 2.5055E−07 | 1.4584E−07 |
| Function | Algorithm | Best | Ave | Std |
|----------|-----------|------|-----|-----|
| $F_3(X)$ | LLSSA | 0 | 0 | 0 |
|  | BSO | 0.0004945 | 15.1459 | 34.9405 |
|  | CSSA | 0 | 4.2858E–192 | 0 |
|  | SSA | 0 | 7.7137E–210 | 0 |
|  | GWO | 2.225E–04 | 0.029431 | 0.070791 |
|  | PSO | 22.2592 | 49.9596 | 18.1286 |
| $F_4(X)$ | LLSSA | 0 | 1.008E–163 | 0 |
|  | BSO | 0.01503 | 0.015026 | 1.40522 |
|  | CSSA | 0 | 4.107E–141 | 2.249E–140 |
|  | SSA | 4.063E–14 | 7.7137E–07 | 2.926E–07 |
|  | GWO | 2.2250 | 0.029431 | 0.070791 |
|  | PSO | 8.1218 | 6.9044 | 7.0838 |
| $F_5(X)$ | LLSSA | 2.2981E–13 | 2.3943E–05 | 3.8273E–05 |
|  | BSO | 553.7041 | 2543.2161 | 1657.5282 |
|  | CSSA | 0 | 2.6949E–141 | 1.657E–138 |
|  | SSA | 4.7653E–09 | 0.00028325 | 0.00049636 |
|  | GWO | 2.4178 | 6.4757 | 2.1850 |
|  | PSO | 8.1218 | 6.9044 | 7.0838 |
| $F_6(X)$ | LLSSA | 0 | 0 | 0 |
|  | BSO | 10.9679 | 76.8897 | 44.1926 |
|  | CSSA | 1.3357E–08 | 2.6062E–04 | 0.0015114 |
|  | SSA | 4.7653E–09 | 0.00028325 | 0.00049636 |
|  | GWO | 1.1787 | 2.2057 | 2.7228 |
|  | PSO | 30.8725 | 58.7189 | 18.8337 |
| $F_7(X)$ | LLSSA | 0 | 0 | 0 |
|  | BSO | 7.8221E–46 | 2.0384E–39 | 3.228E–39 |
|  | CSSA | 0 | 6.07E–95 | 3.324E–94 |
|  | SSA | 0 | 8.7599E–97 | 4.79E–96 |
|  | GWO | 1.1787E–06 | 2.2057E–05 | 2.7228E–05 |
|  | PSO | 0 | 6.6714E–24 | 1.2043E–23 |
| $F_8(X)$ | LLSSA | 8.0687 | 0.0001561 | 0.00012633 |
|  | BSO | 0.0038813 | 0.009674457 | 0.004701 |
|  | CSSA | 9.9226E–06 | 0.00026062 | 0.00016246 |
|  | SSA | 5.318E–05 | 0.00029582 | 0.00025735 |
|  | GWO | 0.00040374 | 0.0020252 | 0.89355 |
|  | PSO | 0.010201 | 6.6714E–24 | 1.2043E–23 |
| $F_9(X)$ | LLSSA | 1.0255E–12 | 10931.8556 | 1203.7127 |
|  | BSO | 1.0255E–12 | 11820.3055 | 973.8153 |
|  | CSSA | 9.859E–13 | 797.8153 | 615.627 |
|  | SSA | 9.3744E–09 | 501.0137 | 1033.4608 |
|  | GWO | 9.0557 | 6452.354 | 793.1086 |
|  | PSO | 8.40513 | 2.9712 | 1.2043E–23 |
| $F_{10}(X)$ | LLSSA | 1.04486E–09 | 4.06779E–10 | 9.39488E–10 |
|  | BSO | 0.104491 | 1.053227 | 0.53947 |
|  | CSSA | 1.90657E–12 | 1.93721E–08 | 3.43612E–08 |
|  | SSA | 2.36977E–12 | 2.7896E–09 | 5.1766E–09 |
|  | GWO | 9.2201E–06 | 0.01983839 | 0.013634913 |
|  | PSO | 2.35982E–06 | 0.041670227 | 0.071831524 |
terrain, this paper proposes to integrate the improved sparrow search algorithm with the simulated annealing algorithm, so that the algorithm avoids the local optimal solution in the solving process and finds a clear and simple path with the smallest cost function.

10.1. Cost Function Design. Cost function is an indispensable part of route planning. The more reasonable the cost function is set, the closer it is to the real demand. The cost function designed in this paper is as follows:

\[
J_{\text{cost}} = w_1 J_{\text{length}} + w_2 J_{\text{height}} + w_3 J_{\text{smooth}},
\]

\[
\begin{aligned}
&w_1 \geq 0, \\
&\sum_{i=1}^{3} w_i = 1.
\end{aligned}
\]  

\(J_{\text{cost}}\) represents the total cost of the path, and \(J_{\text{length}}, J_{\text{height}},\) and \(J_{\text{smooth}}\) represent the total length cost, total height cost, and smoothness cost of the path, respectively. \(w_1, w_2,\) and \(w_3\) are the corresponding weights. In this paper, \(w_1 = 0.5, w_2 = 0.3,\) and \(w_3 = 0.2.\)

The cost function can be used to evaluate the quality of the generated path, which is the basis of iterative evolution of algorithm population. The performance of cost function determines the efficiency and quality of algorithm implementation, and it is also the performance index of path planning. In order to better evaluate the path quality, this paper constructs the fitness function \([17–19]\) by comprehensively considering the height cost, length cost, and smoothness cost of the path. Suppose each path is composed of \(n\) points. \(x, y\) are the point coordinates of the terrain projected on the horizontal plane, and \(z\) is the height corresponding to the point coordinates on the horizontal plane.

10.2. Path Length Cost. Path length is one of the most important indicators to evaluate the quality of a path. The shorter the path is, the less energy and time consumption the UAV will take to fly. The cost of introducing path length is as follows:

\[
J_{\text{length}} = \sum_{j=0}^{n} \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2 + (z_{j+1} - z_j)^2}.
\]  

\(12\)

10.3. High Cost. The stable flight height of the UAV is also an important part of the UAV track planning process. For most aircraft, the flight height should not change much. Stable flight height helps to reduce the burden on the control system and save more fuel. Therefore, the cost of introducing track elevation is

\[
J_{\text{height}} = \sqrt{\frac{1}{n} \sum_{j=1}^{n} \left( z_j - \frac{1}{n} \sum_{j=1}^{n} z_j \right)^2}.
\]  

\(13\)

10.4. Smoothness Cost. When an UAV is making a turn, it needs to consume some energy due to the air resistance and at the same time exerts some pressure on the body. The smaller the turning angle, the greater the pressure generated and the more energy consumed, making the flight inefficient.

| Function | Algorithm | Best | Ave  | Std   |
|----------|-----------|------|------|-------|
| \(F_{12}(X)\) | LLSSA | 1.3498E−31 | 1.3498E−31 | 0 |
|  | BSO | 1.3498E−31 | 1.3498E−31 | 0 |
|  | CSSA | 1.4730E−31 | 2.3043E−29 | 3.9261E−29 |
|  | SSA | 1.3498E−31 | 3.9049E−29 | 5.2232E−29 |
|  | GWO | 1.0606E−08 | 8.9305E−07 | 7.8567E−07 |
|  | PSO | 1.30489E−23 | 4.7708E−21 | 6.0171E−21 |
| \(F_{13}(X)\) | LLSSA | 6.2457E−27 | 1.1032E−20 | 3.5156E−20 |
|  | BSO | 0 | 0.2540 | 0.3592 |
|  | CSSA | 4.3789E−21 | 3.2277E−16 | 1.5937E−15 |
|  | SSA | 2.7959E−21 | 1.2661E−16 | 2.7684E−16 |
|  | GWO | 1.2267E−08 | 4.0761E−07 | 4.2024E−07 |
|  | PSO | 1.1013E−22 | 8.1661E−20 | 1.7949E−19 |
| \(F_{14}(X)\) | LLSSA | 6.47687E−14 | 9.70058E−09 | 1.93534E−08 |
|  | BSO | 0 | 2.17439 | 2.93539 |
|  | CSSA | 1.2546E−14 | 5.2214E−07 | 1.3653E−06 |
|  | SSA | 9.50199E−13 | 1.02709E−06 | 2.50633E−06 |
|  | GWO | 0.00022591 | 1.142301517 | 2.11174329 |
|  | PSO | 0.022667333 | 0.026019705 | 0.001130521 |
| \(F_{15}(X)\) | ALSSA | 0.998 | 0.998 | 0 |
|  | BSO | 0.998 | 1.0311 | 0.17843 |
|  | CSSA | 0.998 | 2.1068 | 2.9273 |
|  | SSA | 0.998 | 2.5594 | 3.6421 |
|  | GWO | 0.998 | 2.8013 | 2.8092 |
|  | PSO | 0.998 | 1.1968 | 0.3976 |
Therefore, the smoothness of flight is also a key factor in the cost of flight.

\[
J_{\text{smooth}} = (x_n - x_1) \sum_{j=1}^{\Delta x} \arccos \left( \frac{\overrightarrow{\varphi}_j}{|\overrightarrow{\varphi}_j|} \frac{\overrightarrow{\varphi}_j}{|\overrightarrow{\varphi}_j|} \right) 
\]  

(14)

\[
\overrightarrow{\varphi}_j = (x_{j+1} - x_j, y_{j+1} - y_j, z_{j+1} - z_j)
\]

\(\overrightarrow{\varphi}_j\) represents the degree of deviation between the previous position and this position, and \(J_{\text{smooth}}\) describes the cumulative sum of the degree of deviation of the connected nodes in the entire path. It is used to simulate the turbulence during the flight of the UAV. The smaller the turbulence, the higher the safety inside and outside the UAV.

### 11. Simulation Experiment and Analysis

In order to verify the feasibility and practicability of the fusion algorithm to optimize the effect of UAV, it is compared with SAPSO, PSO, SSA, and CSSA. The parameters of each algorithm are set as follows: the number of population is 100, the number of iterations is 400, the initial temperature \(T\) is 25, \(\lambda = 0.99\), and the specific experimental model parameters are shown in Table 4. In order to enhance the persuasion of the experiment and reduce the interference of accidental events, each algorithm runs 10 times independently and counts the minimum value, average value, and the worst value of each planned route. The optimal roadmap of each algorithm is shown in Figure 6, and the average cost function convergence diagram is shown in Figure 7.

As you can see from Figures 6 and 7 and Table 4, the LLSSA and SAPSO algorithms optimize the UAV’s path to be the simplest and clearest. The CSSA, SSA, and PSO optimization routes appear to be distorted, obviously falling into a local optimal state. From the performance index table of each algorithm, LLSSA has better convergence accuracy and stability than SAPSO, and other algorithms have poorer stability, with the worst cost value exceeding 50. From the average cost convergence graph, it can be seen that the LLSSA algorithm has extremely fast convergence speed and accuracy. It found a path with the least cost before 50 iterations. The route found by the SSA algorithm is poor, complex, and costly, indicating that the SSA algorithm has limitations in solving such complex multidimensional functions. At the same time, it also shows that in complex path planning problems, the algorithm will meet the current minimum cost and fall into a local optimal state. Therefore, a flexible and judging search mechanism is needed to make the algorithm get rid of the constraints of the complex environment.

Similarly, in order to further verify the feasibility and significance of the algorithm on the UAV route, Wilcoxon test as described above is conducted among the algorithms, and the rank sum test results are shown in Table 6.

It can be seen from Table 6 and the above comprehensive situation that the algorithm that introduces the simulated annealing strategy has a greater advantage in route planning. The performance of the PSO algorithm and the CSSA algorithm is equivalent, and the SSA algorithm is the worst. Applicable to SSA algorithm, a flexible search mechanism and judgment strategy should be added to make the improved SSA algorithm stand out.
Figure 6: Continued.
Figure 6: Optimal route planning for each algorithm. (a) LLSSA, (b) SAPSO, (c) CSSA, (d) SSA, and (e) PSO.

Table 4: Experimental environment parameter table.

| Name       | Coordinate | Height (radius) |
|------------|------------|-----------------|
| Start point| (5, 90)    | —               |
| End point  | (45, 15)   | 8               |
| Obstacle 1 | (15, 70)   | 4               |
| Obstacle 2 | (20, 60)   | 3               |
| Obstacle 3 | (30, 30)   | 4               |
| Obstacle 4 | (40, 45)   | 3               |

Table 5: Route statistics of each algorithm.

| Algorithm | MIN       | AVER      | Worst value |
|-----------|-----------|-----------|-------------|
| LLSSA     | 43.0889   | 43.1057   | 43.1205     |
| CSSA      | 46.3648   | 48.5135   | 50.1908     |
| SSA       | 50.1908   | 51.4657   | 53.3881     |
| SAPSO     | 43.1159   | 44.0132   | 46.0752     |
| PSO       | 48.5228   | 49.3312   | 50.8184     |
12. Conclusion

In this paper, aiming at the problem that the sparrow search algorithm falls into the local optimum and depends on the initial population in the optimization process, the reverse learning strategy based on the lens principle and the variable cosine search strategy are introduced, which are applied to the stage of discoverer and follower, respectively, making the search method more flexible and careful. Then the simulated annealing algorithm is fused to extract the optimal solution. Through the test of standard test function, it can be seen that LLSSA has good optimization ability and convergence effect. It is applied to the path planning of UAV in three-dimensional complex terrain. Through the comparison of CSSA, SSA, SAPSO, and PSO optimization UAV route planning, it shows that LLSSA optimization UAV path is the simplest and clearest, and the cost is the least. Through the planning route index table and the average cost function convergence diagram, it shows that LLSSA has good stability, so its application in UAV path planning has high reliability. It can be seen that in the process of sparrow optimization, if the leading discoverer is limited, the whole algorithm will be paralyzed, and the solution found each time is not necessarily reliable. Therefore, it is extremely important to effectively improve the discoverer’s search mechanism and judgment ability. It can be seen from the above experiments that LLSSA relies on the judgment ability of simulated annealing algorithm in complex terrain path planning problems. Although this method can improve the quality of route optimization, it does not necessarily have a good effect in simple path planning problems, and it increases the calculation time, which is still a challenge in multiple complex environments. In the next step, we apply it to the path planning of multimission UAV.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was financially supported by the Regional Foundation of the National Natural Science Foundation of China (no. 61561024).

References

[1] J. Kennedy and R. C. Eberhart, “A discrete binary version of the particle swarm algorithm,” in Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics.
[2] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” Advances in Engineering Software, vol. 69, pp. 46–61, 2014.

[3] T. Wang and L. Yang, “Beetle swarm optimization algorithm: theory and application,” 2018, https://arxiv.org/abs/1808.00206.

[4] 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.

[5] X. Lu, X. Mu, J. Zhang, and Z. Wang, “Chaos sparrow search optimization algorithm,” Journal of Beijing University of Aeronautics and Astronautics, vol. 1–10, 2021.

[6] X. Lu, X. Mu, and J. Zhang, “Multi threshold image segmentation based on improved sparrow search algorithm,” Systems Engineering and Electronic Technology, vol. 43, no. 2, pp. 318–327, 2021.

[7] X.-B. Meng, X. Z. Gao, L. Lu, Y. Liu, and H. Zhang, “A new bio-inspired optimisation algorithm: bird swarm algorithm,” Journal of Experimental & Theoretical Artificial Intelligence, vol. 28, no. 4, pp. 673–687, 2016.

[8] T. Mao and Q. Zhang, “Improved sparrow algorithm combining Cauchy mutation and reverse learning,” Computer Science and Exploration, pp. 1–12, 2021.

[9] 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.

[10] L. Wen, T. Wu, M. Tang et al., “Gray wolf optimization algorithm based on lens imaging learning strategy,” Acta Automatica Sinica, vol. 46, no. 10, pp. 2148–2164, 2020.

[11] P. J. M. Van Laarhoven and E. H. L. Aarts, “Simulated annealing,” Simulated Annealing: Theory and Applications, Springer, Dordrecht, The Netherlands, 1987.

[12] B. S. Yıldız, N. Pholdee, N. Panagant et al., “A novel chaotic Henry gas solubility optimization algorithm for solving real-world engineering problems,” Engineering with Computers, pp. 1–13, 2021.

[13] P. Champasak, N. Panagant, N. Pholdee, S. Bureerat, and A. R. Yıldız, “Self-adaptive many-objective meta-heuristic based on decomposition for many-objective conceptual design of a fixed wing unmanned aerial vehicle,” Aerospace Science and Technology, vol. 100, Article ID 105783, 2020.

[14] D. Whitley, “A genetic algorithm tutorial,” Statistics and Computing, vol. 4, no. 2, pp. 65–85, 1994.

[15] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” IEEE Computational Intelligence Magazine, vol. 1, no. 4, pp. 28–39, 2006.

[16] 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.

[17] C. YongBo, M. YueSong, Y. JianQiao, S. XiaoLong, and X. Nuo, “Three-dimensional unmanned aerial vehicle path planning using modified wolf pack search algorithm,” Neurocomputing, vol. 266, pp. 445–457, 2017.

[18] X. Zhang, X. Lu, S. Jia, and X. Li, “A novel phase angle-encoded fruit fly optimization algorithm with mutation adaptation mechanism applied to UAV path planning,” Applied Soft Computing, vol. 70, pp. 371–388, 2018.