Hindawi
Scientific Programming
Volume 2021, Article ID 6505253, 16 pages
https://doi.org/10.1155/2021/6505253

Research Article

Adaptive Spiral Flying Sparrow Search Algorithm

Chengtian Ouyang, Yaxian Qiu, and Donglin Zhu

School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China

Correspondence should be addressed to Yaxian Qiu; qiuyaxian@mail.jxust.edu.cn

Received 1 July 2021; Revised 30 July 2021; Accepted 13 August 2021; Published 27 August 2021

Academic Editor: Punit Gupta

The sparrow search algorithm is a new type of swarm intelligence optimization algorithm with better effect, but it still has shortcomings such as easy to fall into local optimality and large randomness. In order to solve these problems, this paper proposes an adaptive spiral flying sparrow search algorithm (ASFSSA), which reduces the probability of getting stuck into local optimum, has stronger optimization ability than other algorithms, and also finds the shortest and more stable path in robot path planning. First, the tent mapping based on random variables is used to initialize the population, which makes the individual position distribution more uniform, enlarges the workspace, and improves the diversity of the population. Then, in the discoverer stage, the adaptive weight strategy is integrated with Levy flight mechanism, and the fusion search method becomes extensive and flexible. Finally, in the follower stage, a variable spiral search strategy is used to make the search scope of the algorithm more detailed and increase the search accuracy. The effectiveness of the improved algorithm ASFSSA is verified by 18 standard test functions. At the same time, ASFSSA is applied to robot path planning. The feasibility and practicability of ASFSSA are verified by comparing the algorithms in the raster map planning routes of two models.

1. Introduction

With the continuous development of scientific research, more and more swarm intelligence optimization algorithms have been proposed. The swarm intelligence optimization algorithm basically abstracts a series of formulas based on the life characteristics or behavior rules of organisms or things and finds high-quality solutions in a certain solution space based on these formulas. Scholars have proposed a series of swarm intelligence optimization algorithms through the behaviors of ant, cat, whale, wolf, and others. Among them, the sparrow search algorithm (SSA) [1], which finds optimal solutions through the sparrow foraging process, is a novel swarm intelligence optimization algorithm proposed by two scholars, Xue and Shen in 2020. This algorithm has fewer population roles and simple principles, and it is easy to understand. Compared with gray wolf optimizer (GWO) [2] and particle swarm optimization (PSO) [3], SSA has higher accuracy, stronger convergence, and better optimization capabilities. Moreover, SSA is now more and more applied in practical engineering. However, SSA also has the common shortcomings of swarm intelligence algorithms, such as easy to get stuck in local and excessive randomness during initialization.

In order to overcome these shortcomings of the swarm intelligence optimization algorithm, scholars have studied the algorithm and successfully applied to practical engineering problems. Lei et al. [4] applied the Levi flight strategy in SSA to avoid local convergence and improve global optimization capabilities. They combine the improved sparrow search algorithm with DV-Hop to enhance the positioning accuracy in WSN. Xin et al. [5] used tent chaotic sequence to initialize the population in turn, and Gaussian mutation operator and tent chaotic disturbance balance the global individual position. A series of strategies increase the population diversity of the algorithm, strengthen the local search ability of the algorithm, and improve the possibility of jumping out of the local area. Tang et al. [6] introduced a cubic mapping strategy to initialize the population, avoiding the decrease of population diversity in the later iterations. The reverse learning strategy and elite particles are added to effectively prevent the algorithm from falling into a local optimum. The sine and cosine optimization algorithm is incorporated into the follower position update to reduce search blind spots and balance algorithm development and search capabilities. Finally, a Gaussian walk strategy is used to help the algorithm leave the stagnant state. The good search ability of the improved algorithm is tested through experiments. Xin et al. [7] also proposed an improved algorithm, adding the flying thoughts in the bird swarm algorithm to the location update of the discoverer and the follower, ensuring global convergence, increasing the diversity of the population, and being able to jump out of the local optimum. In this paper, the improved algorithm is also applied to the multithreshold image segmentation method, which solves the original low segmentation accuracy and slow segmentation speed. Mao et al. [8] used sin-cosine algorithm, nonlinear dynamic learning factor, and Levy flight strategy to improve the algorithm, speed up the convergence speed, and strengthen the local escape ability. Song et al. [9] proposed to introduce the skew tent map-based chaotic method to initialize the population and strengthen the individual quality of the population. After that, the nonlinear decreasing weight and mutation operator strategies were combined in each stage to dynamically change the search space, increase the diversity of the population, and improve the ability to jump out of the local optimum. Wu and others [10] used tent chaotic mapping, adaptive weight control step size, Cauchy mutation, and Gaussian mutation to improve the original algorithm in turn. Such a series of strategies balance the global search and local search capabilities, enhance the local development capabilities of the algorithm, and improve the search accuracy. Finally, the improved algorithm is applied to the mine intelligent ventilation system, which proves the good optimization effect of the improved algorithm. Zhang and Ding [11] initialized the population by logistic mapping. In the discoverer phase and the early warning phase, adaptive parameters are introduced to expand the search range of the discoverer, dynamically adjust the number of early warning individuals, and better help the sparrow individuals find the best position. The mutation operator strategy is added to the global process to increase the population diversity. The whole improved algorithm enhances the global search capability of SSA. Finally, the improved CSSA is applied to the stochastic configuration network. The results show that CSSA has good performance and improves the regression performance of SCN in solving large-scale data problems. Zhang et al. [12] used the sine cosine strategy to change the way the discoverer updates the location, and improve global and local search capabilities of the algorithm. A labor cooperation structure is added to the discoverer stage and the early warning stage, allowing the discoverer and the early warning individual to share their locations to achieve cooperation, so as to converge to the global optimal solution faster and more stably.

Although these improved strategies can better improve the optimization ability of each algorithm, they do not change the search mechanism of each algorithm itself. And they only improved the search ability of the algorithm in the neighborhood space, and the learning rate was not high, resulting in the algorithm still having drawbacks. This means that when faced with high-dimensional complex problems, these algorithms may also be stuck into the local optimal situation. Therefore, based on the research of many scholars, this paper proposes the adaptive spiral flying sparrow search algorithm (ASFSSA).

In the following paper, Section 2 gives an introduction to the overall work and motivation of the algorithm. Section 3 introduces the sparrow search algorithm. Section 4 illustrates the improvements made by the adaptive spiral flying sparrow search algorithm. Section 5 compares and analyzes the improved algorithm with other intelligent algorithms. In Section 6, the improved algorithm is applied to path planning and compared with the other three algorithms in two different grids. Section 7 summarizes this paper.

2. Related Work and Motivation

In order to improve the search speed and convergence accuracy of individual sparrows and reduce the possibility of falling into the local optimum, adaptive weights and Levy flight strategies are used in the discoverer stage. In order to make the individual sparrow search more detailed and get a better position, a variable spiral search strategy is introduced in the follower stage. The higher the quality of the solution, the better the search ability.

To verify the convergence and optimization ability of the proposed algorithm, six algorithms are used to compare the optimization effects of 18 test functions. As can be seen from the experimental results, the adaptive spiral flying sparrow search algorithm has more significant advantages than other algorithms. Finally, for the purpose of verifying the practicability and performance of the algorithm better, the four algorithms are applied to the path planning of the robot. The results show that the path planning of the adaptive spiral flying sparrow search algorithm is relatively stable and the cost is the least, which reduces the constraints of the immature and randomness in the path planning of the previous algorithms.

3. Sparrow Search Algorithm

The sparrow population is divided into two roles, namely, the discoverer and the follower. They have three behaviors: foraging, following, and reconnaissance. The task of the discoverer is to find food and inform the follower of the location of the food. Therefore, the discoverer needs to search in a large area, and the foraging range of the follower is generally small. This is the formula for updating the position of the discoverer:

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

In formula (1), t and M, respectively, denote the current number of iterations and the maximum number of iterations. $X_{i,j}$ represents the location of the i-th sparrow, and j is the representative dimension. In the above formula, α

and R_2 are both a random number from 0 to 1 (not zero), and R_2 is an important parameter that controls the individual's flight behavior. ST ($\mathrm{ST} \in [0.5,\ 1]$) is a safety threshold and an important parameter to measure whether the location of the discoverer is safe. L is a $1 \times D$ matrix with all elements of 1. If $R_2 < \mathrm{ST}$, it means that the current location is temporarily safe, there are no predators in the surrounding environment, and the discoverer can search for food in a large area. If $R_2 \geq \mathrm{ST}$, it means that the predator's trace is found at the current location, and the discoverer needs to go to other safe areas for foraging activities at this time.

The location update of followers is described as follows:

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

In formula (2), X_P^{t+1} refers to the optimal position occupied by the discoverer in the $t+1$-th iteration so far (the $t+1$-th iteration has not ended) and X_{worst}^{t} represents the worst position occupied by the group in the t-th iteration. A is a $1 \times D$ matrix, and the elements in the matrix are randomly assigned values of 1 or −1, and $A^{+} = A^{T}(AA^{T})^{-1}$. If $i > n/2$, this means that the current follower is at the edge of the entire population and has no food. At this time, the follower needs to go elsewhere for food. Otherwise, the follower will go after the pace of the discoverer for food.

When being aware of the danger, the sparrow population will make antipredatory behavior:

$$X_{i,j}^{t+1} = \begin{cases} X_{\text{best}}^{t} + \beta \cdot \left|X_{i,j}^{t} - X_{\text{best}}^{t}\right|, & \text{if } f_i \neq f_g, \\ X_{i,j}^{t} + K \cdot \left(\dfrac{\left|X_{i,j}^{t} - X_{\text{worst}}^{t}\right|}{(f_i - f_w) + \varepsilon}\right), & \text{if } f_i = f_g. \end{cases} \quad (3)$$

In formula (3), X_{best}^{t} is the global optimal solution obtained in the t-th iteration. In the above formula, β, K, and ε are all parameters in the formula. β is a random number that obeys the standard normal distribution, and it is used to control the step size. K is an arbitrary random number from -1 to 1, it represents the direction in which the individual sparrow moves, and it is also a step control parameter. ε is a relatively small constant, and its function is to prevent the denominator from being zero. f_i represents the fitness value of the i-th individual, and f_g and f_w are the best fitness value and the worst fitness value of the current iteration number, respectively.

If $f_i > f_g$, then this indicates that the individual is at the edge of the population and is easily preyed by natural enemies. If $f_i = f_g$, then this means that the individual is located in the center of the population. At this time, the sparrow needs to be close to other individuals to reduce the probability of being captured.

4. Adaptive Spiral Flying Sparrow Search Algorithm

4.1. Tent Chaotic Mapping Based on Random Variables. Because SSA has the shortcoming of large randomness, it is decided to introduce orderly and uniform tent mapping to improve it. Many scholars have applied tent mapping to solve the optimization problem [13]. However, tent mapping is not very stable. In [7], in order to reduce this influence, tent mapping based on random variables was adopted, which has a good effect. Therefore, in this paper, the tent mapping strategy based on random variables is introduced to improve the initialization of SSA, so that the initialization of the population is more orderly, and the controllability of the algorithm is enhanced. Its specific formula is as follows:

$$z_{i+1} = \begin{cases} 2z_i + \operatorname{rand}(0,1) \times \dfrac{1}{N}, & 0 \leq z \leq \dfrac{1}{2}, \\ 2\left(1 - z_i\right) + \operatorname{rand}(0,1) \times \dfrac{1}{N}, & \dfrac{1}{2} \leq z \leq 1. \end{cases} \quad (4)$$

The expression after the Bernoulli transformation is

$$z_{i+1} = (2z_i)\text{mod}1 + \operatorname{rand}(0,1) \times \frac{1}{N}. \quad (5)$$

In formula (5), N is the number of particles in the chaotic sequence.

According to the characteristics of the tent mapping, the sequence flow for generating chaos in the feasible domain is as follows:

(1) Randomly generate the initial value z_0 in (0, 1), and let $i = 1$.

(2) Perform iteration by using that formula (5) to generate a z sequence, and i is increased by 1.

(3) Stop if the number of iterations reaches the maximum, and store the generated z sequence.

4.2. Adaptive Weighting. Weight strategies are common in particle swarm optimization algorithms [14]. Generally, the particle swarm algorithm reduces to some extent the trapping into local optimum by adaptively changing between the set maximum and minimum values. Inspired by this, this paper adds an inertia weight w which varies with the number of iterations in the discoverer stage of sparrow optimization. In the initial stage of the algorithm, it weakens the influence of random initialization and balances the Levy flight mechanism below, so as to enhance the local search and global search of the algorithm.

The discoverer guides other individuals in the population to search for food, so the introduction of adaptive weights improves the quality of individual locations, enabling other individuals to converge faster to optimal locations, and overall accelerates the convergence rate. Based on the characteristics of sparrows, the formula for adaptive weights is as follows:

$$w(t) = 0.2\cos\left(\frac{\pi}{2}\cdot\left(1-\frac{t}{\text{iter}_{\max}}\right)\right). \tag{6}$$

The meaning of formula (6) is that w has the property of nonlinear change between [0, 1]. According to the characteristics of cos function, the weight value is smaller at the beginning of the algorithm, but the optimization speed is faster and the later weight value is larger, but the change speed is slower, so the convergence property of the algorithm is balanced. The improved discoverer location is updated as follows:

$$X_{i,j}^{t+1} = \begin{cases} w(t)\cdot X_{i,j}^{t}\cdot \exp\left(\dfrac{-i}{\alpha\cdot \text{iter}_{\max}}\right), & \text{if } R_2 < \text{ST}, \\ w(t)\cdot X_{i,j}^{t} + Q\cdot L, & \text{if } R_2 \geq \text{ST}. \end{cases} \tag{7}$$

By introducing adaptive weights to dynamically adjust the position changes of sparrows, different guidance modes for the discoverer at different times make the algorithm search flexible. As the number of iterations increases, the individual sparrows converge toward the optimal position, and a larger weight makes the individual move faster, thus increased the convergence speed of the algorithm.

4.3. Levy Flight Mechanism. In SSA, there are few roles in the population, and the same role update position formula is the same, which will result in multiple individuals in the same optimal position. Too high solution repetition rate will reduce the efficiency of the algorithm, which is not conducive to the optimization of the algorithm. The discoverer has a wide search range and globality, and the adaptive weighting strategy is introduced to effectively improve the convergence effect. However, when facing the high-dimensional complex problems, there is still a probability of falling into local optimum. Therefore, the Levy flight strategy is introduced to improve the randomness of the algorithm solution, thereby enriching the diversity of population positions. This can also effectively improve the operating efficiency of the algorithm.

Levy flight obeys the Levy distribution. Principle [15] and the Levy distribution is shown in Figure 1. It uses a random long- and short-distance mechanism to cover a large area. After adding the Levy flight mechanism, the performance of the proposed algorithm can be improved.

The location update format [16] for joining Levy flight strategy is as follows:

$$x_i'(t) = x_i(t) + l \oplus \text{levy}(\lambda). \tag{8}$$

In formula (8), $x_i(t)$ represents the position of the i-th individual in the t-th iteration, $\oplus$ is an arithmetic symbol representing point-to-point multiplication. l denotes a step length control parameter, which is obtained by this formula: $l = 0.01(x_i(t) - x_p)$. levy(λ) is a path that obeys the Levy distribution, which represents the introduced Levy flight strategy and satisfies the following: levy $u = t^{-\lambda}$, $1 < \lambda \leq 3$.

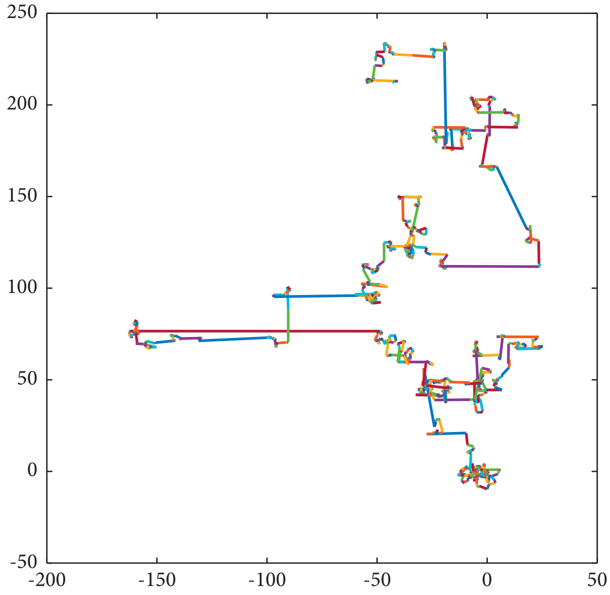


Figure 1: Levy flight diagram.

Because Levy distribution is very complex, Mantegna algorithm is usually used to simulate it [17, 18]. The formula for calculating the step size is as follows:

$$\begin{aligned} s &= \frac{\mu}{|v|^{1/\gamma}}, \\ \mu &\sim N\left(0,\sigma_\mu^2\right), \\ v &\sim N\left(0,\sigma_v^2\right), \\ \sigma_\mu &= \left\{\frac{\Gamma(1+\gamma)\sin(\pi\gamma/2)}{\gamma\cdot\Gamma[(\gamma+1)/2]\cdot 2^{(\gamma+1)/2}}\right\}^{1/\gamma}. \end{aligned} \tag{9}$$

Among them, $\sigma_v = 1$, and γ is generally 1.5.

The introduction of the Levy flight strategy makes the sparrows more flexible at this stage and can also lead other individuals to find a better location, free from the constraints of local extremes. Therefore, the combination of Levy flight mechanism and adaptive weights balances the search method, and the quality of each solution obtained is improved to a certain extent, which greatly improves the search ability of the algorithm.

4.4. Variable Spiral Search Strategy. Followers update dynamically with the location of the discoverer, which leads to blindness and singularity in the way they search. Inspired by the rotation operation of the whale algorithm [19, 20], a variable spiral location update strategy is introduced to make follower location updates more flexible, develop a variety of search paths for location updates, and balance 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

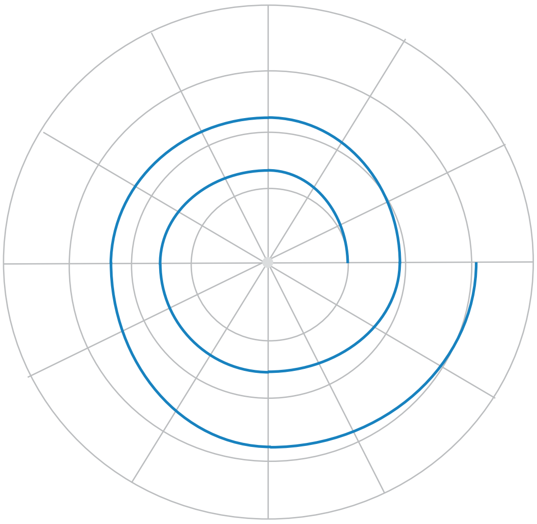


Figure 2: Schematic diagram of spiral search.

local optimum, thus weakens the search ability of the algorithm. The parameter z is designed as an adaptive variable to dynamically adjust the spiral shape of the follower search, which broadens the ability of the follower to explore the unknown region, improves both the search efficiency and the global search performance of the algorithm. The formula for the variable spiral position update strategy is as follows:

$$X_{i,j}^{t+1} = \begin{cases} e^{zl} \cdot \cos(2\pi l) \cdot Q \cdot \exp\left(\dfrac{X_{\text{worst}}^{t} - X_{i,j}^{t}}{i^2}\right), & \text{if } i > \dfrac{n}{2}, \\ X_{P}^{t+1} + \left|X_{i,j}^{t} - X_{P}^{t+1}\right| \cdot A^{+} \cdot L \cdot e^{zl} \cdot \cos(2\pi l), & \text{otherwise}, \\ z = e^{k\cdot\cos\left(\pi\cdot\left(1-\left(i/i_{\max}\right)\right)\right)}. \end{cases} \tag{10}$$

The parameter z varies according to the number of iterations and is composed of an exponential function based on e. The size and amplitude of the helix are dynamically adjusted according to the properties of cos function. k is the coefficient of change. According to the optimization characteristics of each function, in order to make the algorithm have a suitable search range, $k = 5$. L is a uniformly distributed random number of [−1, 1]. With the range of follower position updating is from large to small, more quality solutions are found in the early stage, and late optimization reduces the increase of idle work, which improves 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 to a certain extent.

4.5. Process for Improving Sparrow Search Algorithm. The sparrow search algorithm has better optimization performance than other algorithms, but it still depends on the initial population so that it is easy to fall into a local optimal state. In order to improve these shortcomings, this paper proposes an adaptive spiral flying sparrow search algorithm. Initially, the population was initialized by tent chaotic mapping based on random variables to provide adequate preparation for the discoverer's optimization. Then, adaptive weights and Levy flight strategies were introduced to make the discoverer's position update method more extensive and flexible, and then a variable spiral was proposed. The strategy makes the follower's search more detailed, avoids premature phenomenon, and speeds up the optimization speed of the algorithm.

The specific implementation steps of ASFSSA are as follows:

Step 1: initialize the sparrow population parameters, for example, the total population pop, the total number of discoverers pNum, the total number of iterations iter, and the solution accuracy ε.

Step 2: use tent mapping to initialize the position of population individuals, and generate pop sparrow individuals.

Step 3: use the correlation function to calculate the fitness value f_i of each population individual, and find the maximum fitness value f_g and the minimum fitness value f_w.

Step 4: sort the population according to the fitness value.

Step 5: select the individuals with the top pNum fitness as the discoverer, and the rest are followers, and use formulas (7) and (8) after adding the strategy to update the position of the discoverer.

Step 6: use formula (10) to update the positions of pop-pNum followers.

Step 7: use formula (3) to update the position of the sparrow that is aware of the danger.

Step 8: after one iteration is completed, recalculate the fitness value f_i of each individual, and update the maximum fitness value f_g, the minimum fitness value f_w, and the corresponding position.

Step 9: judge whether the algorithm has reached the maximum number of iterations or the accuracy of the solution. If it has reached, the optimization result will be output; otherwise, it will return to Step 4.

The specific flowchart is shown in Figure 3.

5. Algorithm Performance Test

To test the optimization capability of the improved algorithm, 18 standard test functions are selected to verify the performance. The test functions are listed in Table 1. The first 10 are unimodal functions, the middle 4 are complex multimodal functions, and the last 4 are fixed-dimension function. In order to increase the experimental convincing, it is necessary to compare the particle swarm algorithm (PSO), gray wolf algorithm (GWO), beetle swarm optimization (BSO), SSA, CSSA, and the proposed algorithm optimization effect. Among them, CSSA is an improved sparrow search algorithm in the literature [5], and BSO [21] is a new

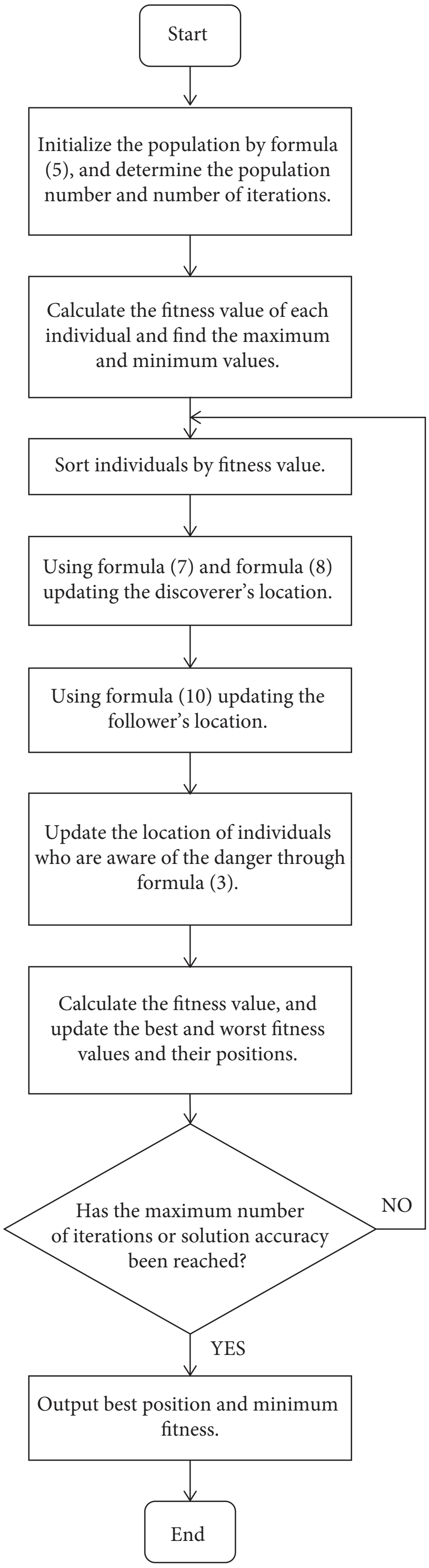

Start
Initialize the population by formula (5), and determine the population number and number of iterations.
Calculate the fitness value of each individual and find the maximum and minimum values.
Sort individuals by fitness value.
Using formula (7) and formula (8) updating the discoverer's location.
Using formula (10) updating the follower's location.
Update the location of individuals who are aware of the danger through formula (3).
Calculate the fitness value, and update the best and worst fitness values and their positions.
Has the maximum number of iterations or solution accuracy been reached?
NO
YES
Output best position and minimum fitness.
End

Figure 3: Algorithm flowchart.

algorithm which combines beetle antennae search algorithm (BAS) [22] and PSO, and the research on this algorithm is very popular. The complexity and parameters of all algorithms are shown in Table 2.

The population number of all algorithms is 100; the number of iterations is 200. In PSO, c1 = c2 = 2, $w = 0.728$. The BSO parameter settings are as follows: eta = 0.95, $c = 2$, $k = 0.4$, step0 = 0.9, and step1 = 0.2. The upper and lower limits of all algorithms are given in the test function table. The experimental environment is MATLAB 2018b, the Windows 10 operating system, and the running memory is 8G. In order to enhance the experimental persuasion, exclude the influence of accidental events, and count the minimum (min) value, average (aver) value, and standard (std) deviation of each algorithm, which reflect the optimization ability and stability of each algorithm. The comparison table of optimization effect of each algorithm is shown in Table 3, and the convergence of each algorithm function is shown in Figure 4.

From Table 3, we can see that the ASFSSA has the best performance in function optimization, especially in F_1–F_4, F_7–F_9, F_{16}, and F_{18}, which can find the optimal value each time. The results of CSSA come later, and PSO and GWO are the worst, especially when the boundary is complex. The BSO has good optimization characteristics in the F_{10}, F_{11}, and F_{17} functions, but the effect in other algorithms has great shortcomings, and the effect is the worst. It can be seen that BSO has the limitation of the optimization ability. Among these three functions, the optimization effect of ASFSSA is second only to it. Generally speaking, ASFSSA has a good convergence speed and accuracy on unimodal functions, and it has a strong ability to resist local extremes on multimodal functions. Therefore, the introduction of multiple strategies significantly improves the stability and searching ability of the algorithm.

From Figure 4, we can see that ASFSSA has a significant improvement in convergence speed and accuracy. Especially, the ASFSSA has very fast convergence speed and very high convergence precision in the functions of F_1–F_5, F_7–F_{10}, F_{15}-F_{16} and has obvious ability of resisting local attraction in the functions of F_5, F_{10}, and F_{15}. ASFSSA has good convergence effect in F_6, F_{13}, F_{14}, and F_{17} functions, but there a few differences with other algorithms. BSO has better performance in the three functions of F_{11}-F_{12} and F_{18}, and ASFSSA has the second best performance, but BSO has worse performance in the other functions, and some functions have worse performance than PSO. The convergence effect of other algorithms is insufficient. Thus, it can be seen that the introduction of multistrategy makes the algorithm get rid of the insufficient search mode in the optimization process, open up a more flexible and detailed search, and improve the convergence ability of the algorithm.

5.1. Wilcoxon Rank Sum Test. It is not comprehensive to simply calculate the indexes of the running results of each algorithm. In order to highlight the superiority of the ASFSSA algorithm, it is necessary to carry out the statistical test. In order to reflect the fairness, the Wilcoxon rank sum test [20] is used to verify whether the results of each run of ASFSSA are significantly different from those of other algorithms at the $P = 5\%$ significance level. If $P < 5\%$, it can be considered that there is a significant difference between the two algorithms; if $P > 5\%$, it means that the difference between the two algorithms is not obvious; that is, the

Table 1: Test function table.

Function	Dim (D)	Interval	Min
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
$F_2(x) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} x_i$	30	[−10, 10]	0
$F_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{n} x_j)^2$	30	[−100, 100]	0
$F_4(x) = \max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	30	[−100, 100]	0
$F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	[−30, 30]	0
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[−100, 100]	0
$F_7 = \sum_{i=1}^{d} x_i^2 + (\sum_{i=1}^{d} 0.5ix_i)^2 + (\sum_{i=1}^{d} 0.5ix_i)^4$	30	[−5, 10]	0
$F_8(x) = \sum_{i=1}^{d/4} [(x_{4i-3} + 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^4]$	30	[−4, 5]	0
$F_9(x) = \sum_{i=1}^{d} \lvert x_i\rvert^{i+1}$	30	[−100, 100]d	0
$F_{10}(x) = \sum_{i=1}^{n} ix_i^4 + \text{random}[0, 1)$	30	[−1.28, 1.28]	0
$F_{11}(x) = \sum_{i=1}^{n} -x_i \sin(\sqrt{\lvert x_i\rvert})$	30	[−500, 500]	-418.9829n
$F_{12} = 418.9829n - \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i\rvert})$	30	[−500, 500]	0
$F_{13} = \sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 \sin^2(\pi w_i + 1) + (w_d - 1)^2] [1 + \sin^2(2\pi w_d)]$ while $w_i = 1 + (x_i - 1/4)$, $\forall i = 1, \ldots, d$	30	[−10, 10]d	0
$F_{14}(x) = \sum_{i=1}^{d-1} ((100x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$	30	[−10, 10]d	0
$F_{15} = \sin^2(3\pi x_1) + (x_1 - 1)^2 [1 + \sin^2(3\pi x_2)] + (x_2 - 1)^2 [1 + \sin^2(2\pi x_2)]$	2	[−10, 10]	0
$F_{16}(x) = 2x_1^2 - 1.05x_1^4 + (x_1^6/6) + x_1x_2 + x_2^2$	2	[−5, 5]	0
$F_{17}(x) = (1.5 - x_1 + x_1x_2)^2 + (2.25 - x_1 + x_1x_2^2)^2 + (2.625 - x_1 + x_1x_2^3)^2$	2	[−4.5, 4.5]	0
$F_{18}(x) = (0.002 + \sum_{i=1}^{25} 1/(i + (x_1 - a_{1i})^6 + (x_2 - a_{2i})^6))^{-1}$, where $a = \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \ldots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \ldots & 32 & 32 & 32 \end{pmatrix}$	2	[−65.536.65.536]	0.998

Table 2: The complexity and parameters of each algorithm.

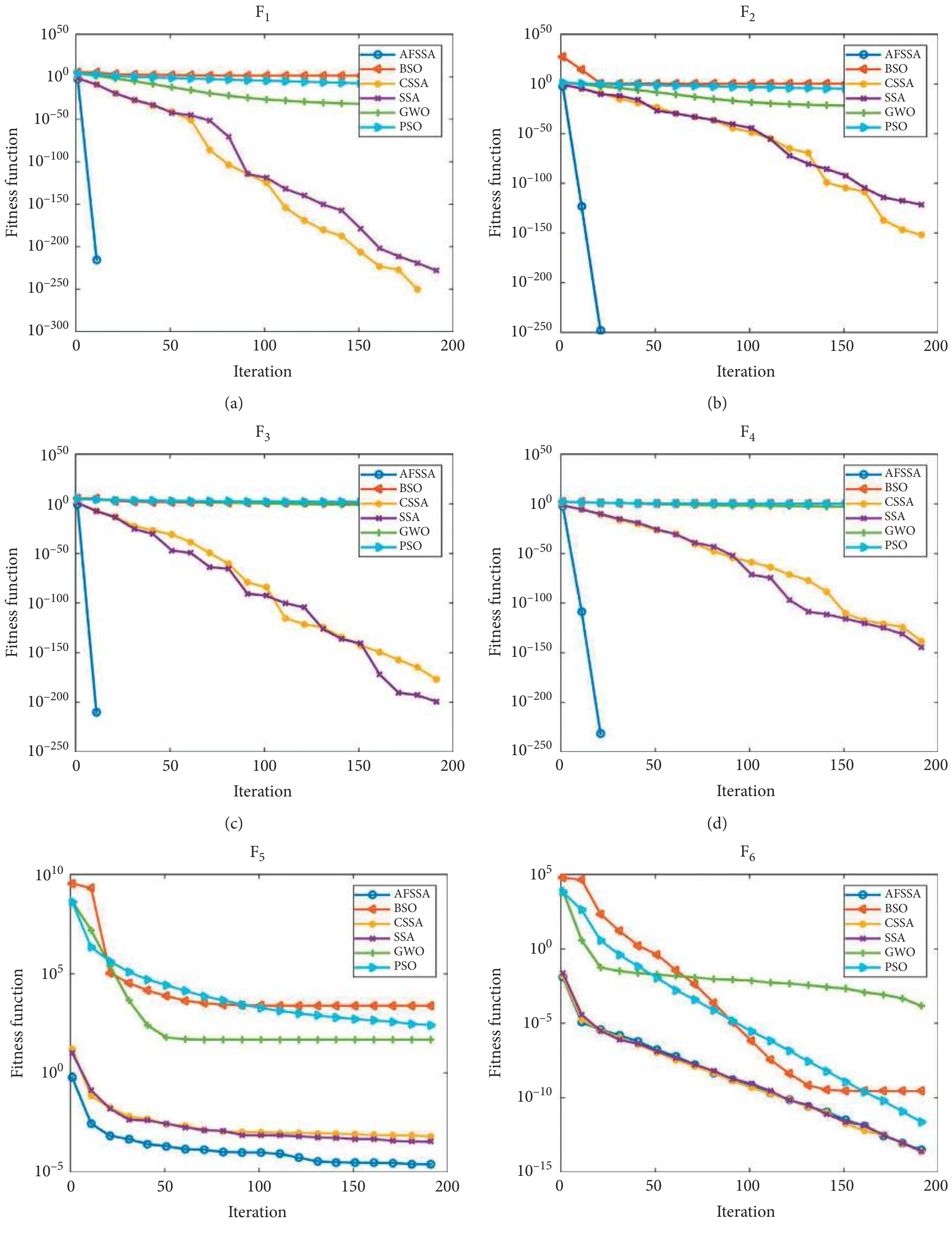
Algorithm	Time complexity	Space complexity	Iteration and population	Parameter setting
ASFSSA	O $(P \times M \times D)$	O $(N \times D)$	Iteration = 200, population = 100	$\varepsilon = 1e-50$
BSO				eta = 0.95, $c = 2$, $k = 0.4$, step0 = 0.9, step1 = 0.2
CSSA				$\varepsilon = 1e-50$
SSA				$\varepsilon = 1e-50$
GWO				—
PSO				c1 = c2 = 2, $w = 0.728$

Table 3: Table of optimization results of each algorithm.

Function	Algorithm	Best	Aver	Std
$F_1(x)$	ASFSSA	0	0	0
	BSO	8.15851	26.3851	14.9876
	CSSA	0	0	0
	SSA	0	$5.192E-251$	0
	GWO	$1.4406E-103$	$1.56961E-97$	$5.36616E-97$
	PSO	$3.8599E-12$	$3.0697E-11$	$2.7076E-11$
$F_2(x)$	ASFSSA	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	$8.9056E-08$	$2.5055E-07$	$1.4584E-07$
$F_3(X)$	ASFSSA	0	0	0
	BSO	0.0004945	15.1459	34.9405
	CSSA	0	$4.2858E-192$	0
	SSA	0	$7.7137E-210$	0
	GWO	$2.2250E-04$	0.029431	0.070791
	PSO	22.2592	49.9596	18.1286
$F_4(X)$	ASFSSA	0	0	0
	BSO	0.01503	0.015026	1.40522
	CSSA	0	$4.1077E-141$	$2.2499E-140$
	SSA	$4.0636E-14$	$1.9036E-07$	$2.9266E-07$
	GWO	$2.4178E-06$	$6.4757E-06$	$2.1850E-06$
	PSO	$8.1218E-14$	$6.9044E-13$	$7.0838E-13$
$F_5(X)$	ASFSSA	$2.5510E-09$	$9.6950E-06$	$3.977E-05$
	BSO	553.7041	2543.2161	1657.5282
	CSSA	$1.3357E-08$	$2.6062E-04$	0.0015114
	SSA	$4.7653E-09$	0.00028325	0.00049636
	GWO	45.8565	47.3855	0.89355
	PSO	89.4507	234.8606	104.6002
$F_6(X)$	ASFSSA	$1.2670E-17$	$2.5726E-15$	$4.5874E-15$
	BSO	$1.4135E-19$	$2.7997E-10$	$1.4985E-09$
	CSSA	$1.2204E-17$	$8.1489E-15$	$1.1839E-14$
	SSA	$4.0636E-14$	$1.0936E-07$	$2.9266E-07$
	GWO	$2.4178E-06$	$6.4757E-06$	$2.185E-06$
	PSO	$8.1218E-14$	$6.9044E-13$	$7.0838E-13$
$F_7(X)$	ASFSSA	0	0	0
	BSO	10.9679	76.8897	44.1926
	CSSA	0	$2.6949E-153$	$1.476E-152$
	SSA	0	$6.9176E-139$	$3.7889E-138$
	GWO	$1.1787E-06$	$2.2057E-05$	$2.7228E-05$
	PSO	30.8725	58.7189	18.8337
$F_8(X)$	ASFSSA	0	0	0
	BSO	1.4696	9.2489	5.9505
	CSSA	0	$1.7622E-279$	0
	SSA	0	$2.3385E-270$	0
	GWO	$6.7719E-06$	$5.7427E-05$	$5.9614E-05$
	PSO	0.017546	0.18078	0.14242

Table 3: Continued.

Function	Algorithm	Best	Aver	Std
$F_9(X)$	ASFSSA	0	0	0
	BSO	$7.8221E-46$	$2.0384E-39$	$3.2288E-39$
	CSSA	0	$6.07E-95$	$3.3246E-94$
	SSA	0	$8.7599E-97$	$4.798E-96$
	GWO	0	$9.7069E-251$	0
	PSO	$1.6341E-29$	$6.6714E-24$	$1.2043E-23$
$F_{10}(X)$	ASFSSA	$1.0044E-05$	$9.5155E-05$	$8.3380E-05$
	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	0.030015	0.00096086
$F_{11}(X)$	ASFSSA	−12569.3148	−11107.0762	796.9218
	BSO	−12569.4418	−11820.3055	973.8153
	CSSA	−9859.7543	−8700.9989	615.6127
	SSA	−9374.4498	−8208.1416	501.1037
	GWO	−9055.703	−6452.354	1033.4608
	PSO	−8405.1316	−6176.2342	793.1086
$F_{12}(X)$	ASFSSA	0.490047	1270.9199	964.1809
	BSO	0.03331	848.2818	1061.7445
	CSSA	2529.3974	3661.9962	725.5963
	SSA	3471.670439	4292.617883	529.1990723
	GWO	4866.1835	6134.3633	604.5177
	PSO	5205.7803	7638.9387	1024.9673
$F_{13}(X)$	ASFSSA	$7.2046E-11$	$3.1056E-07$	$6.4439E-07$
	BSO	3.5095	$8.4787E+54$	$3.4748E+55$
	CSSA	$3.6686E-11$	$1.6611E-05$	$6.0978E-05$
	SSA	$1.1633E-09$	$6.0348E-07$	$1.8076E-06$
	GWO	$1.0500E+54$	$1.4300E+57$	$5.7000E+57$
	PSO	$1.5273E+60$	$1.9700E+60$	$2.1400E+59$
$F_{14}(X)$	ASFSSA	0	$4.9628E-05$	$2.5146E-04$
	BSO	$1.3048E-23$	$4.77072E-21$	$6.0171E-21$
	CSSA	$6.2203E-08$	0.0012751	0.0037596
	SSA	$2.9797E-08$	0.00034106	0.00062708
	GWO	$1.0500E+54$	$1.4300E+57$	$5.6800E+57$
	PSO	$2.0728E+87$	$3.3100E+87$	$5.6800E+86$
$F_{15}(X)$	ASFSSA	$1.3498E-31$	$1.3498E-31$	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	$1.30489E-23$	$4.7707E-21$	$6.0171E-21$
$F_{16}(X)$	ASFSSA	0	0	0
	BSO	$1.2475E-42$	$1.8013E-41$	$1.9512E-41$
	CSSA	0	$4.9534E-285$	0
	SSA	0	$3.7547E-237$	0
	GWO	$2.6095E-166$	$2.1062E-131$	$1.0027E-130$
	PSO	$4.0625E-25$	$6.4627E-22$	$1.8985E-21$
$F_{17}(X)$	ASFSSA	$2.9038E-25$	$2.3672E-19$	$7.0572E-19$
	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	$1.1013E-22$	$8.1661E-20$	$1.7949E-19$
$F_{18}(X)$	ASFSSA	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



(a)

(b)

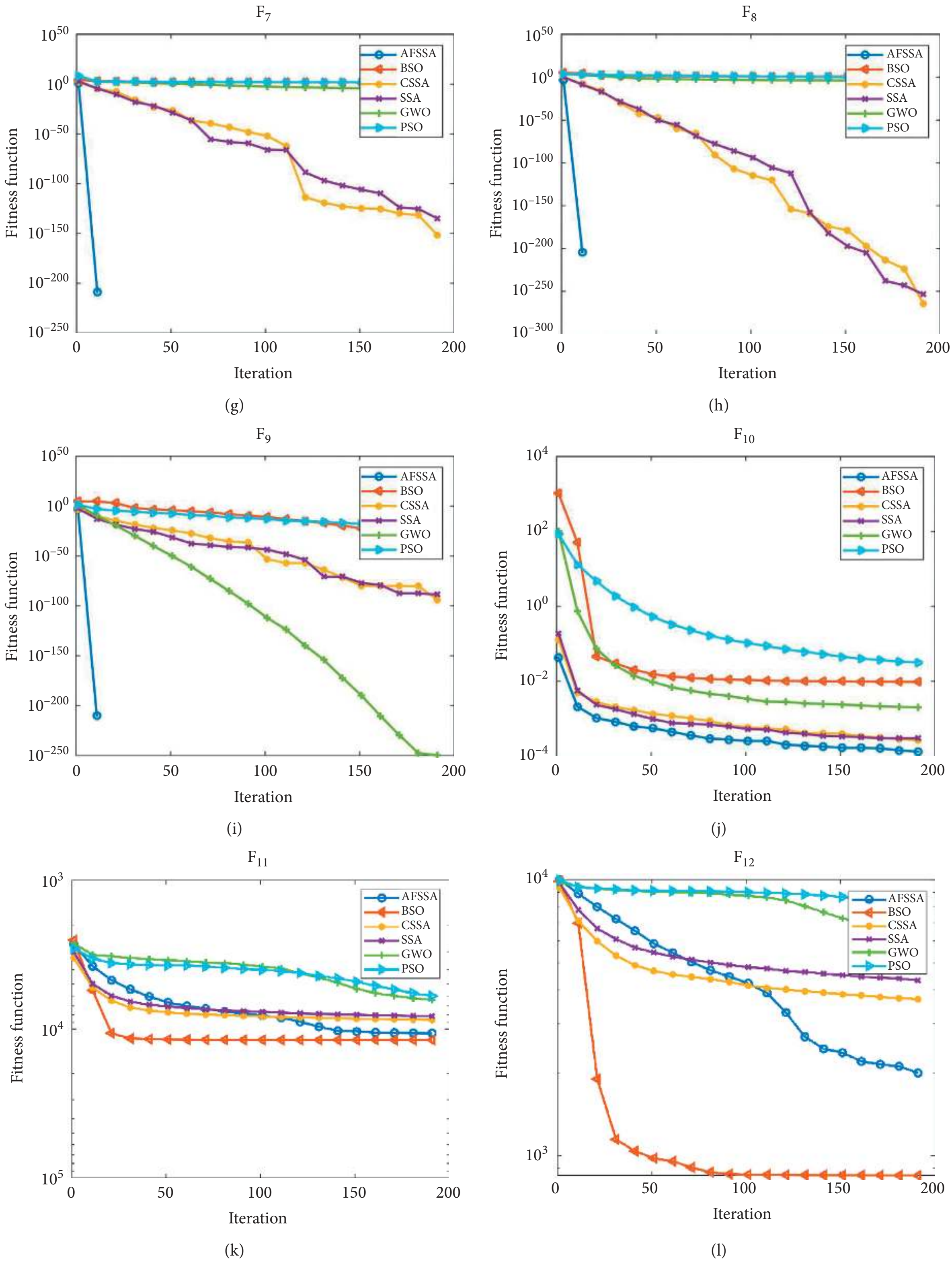
(c)

(d)

(e)

(f)

FIGURE 4: Continued.



(g)

(h)

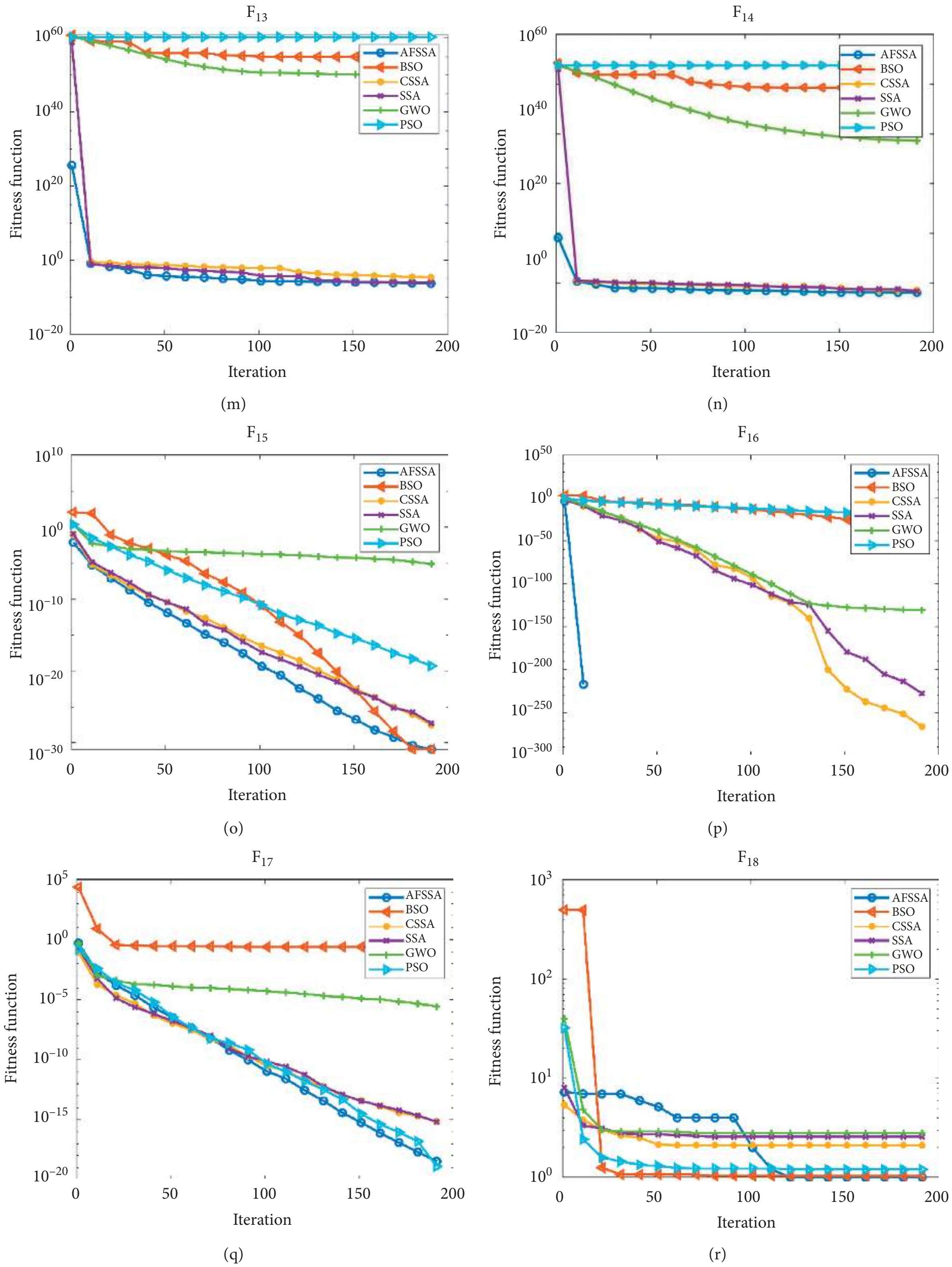
(i)

(j)

(k)

(l)

Figure 4: Continued.



(m)

(n)

(o)

(p)

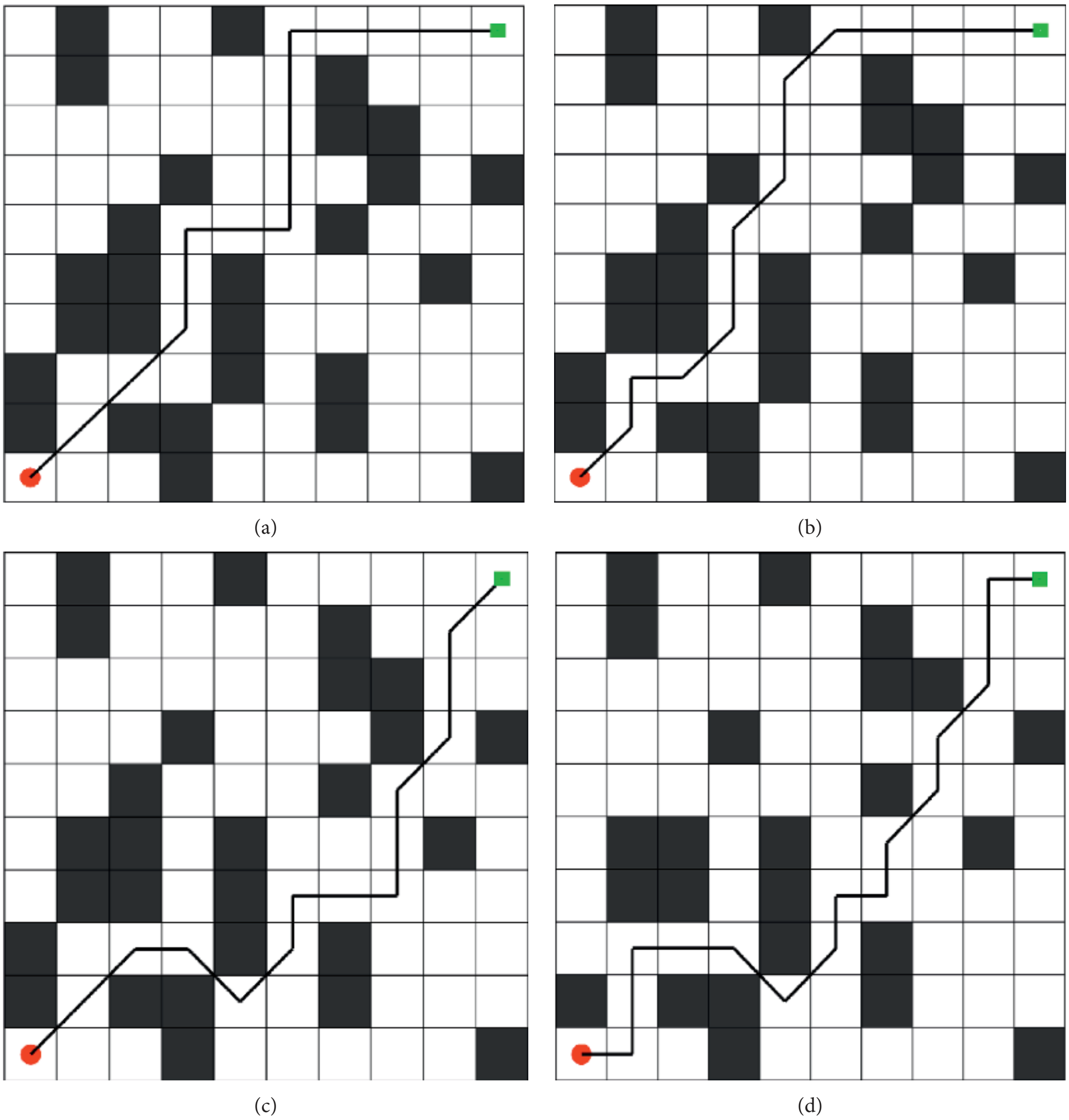
(q)

(r)

Figure 4: 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}. (p) F_{16}. (q) F_{17}. (r) F_{18}.

Table 4: Wilcoxon rank sum test P value.

Function	BSO	CSSA	SSA	PSO	GWO
$F_1(X)$	$1.2118e-12$	N/A	N/A	$1.2118e-12$	$1.2118e-12$
$F_2(X)$	$1.2118e-12$	$8.8658e-07$	$5.3750e-06$	$1.2118e-12$	$1.2118e-12$
$F_3(X)$	$1.2118e-12$	0.0216	N/A	$1.2118e-12$	$1.2118e-12$
$F_4(X)$	$1.2118e-12$	$6.6096e-05$	$6.6096e-05$	$1.2118e-12$	$1.2118e-12$
$F_5(X)$	$3.0199e-11$	0.0184	N/A	$3.0199e-11$	$3.0199e-11$
$F_6(X)$	0.0010	$1.0937e-10$	$1.6132e-10$	$3.0199e-11$	$3.0199e-11$
$F_7(X)$	$1.2118e-12$	0.0056	$1.4552e-04$	$1.2118e-12$	$1.2118e-12$
$F_8(X)$	$1.2118e-12$	0.0419	0.0419	$1.2118e-12$	$1.2118e-12$
$F_9(X)$	$1.2118e-12$	$8.8658e-07$	$3.4526e-07$	$1.2118e-12$	$1.6572e-11$
$F_{10}(X)$	$3.0199e-11$	0.0378	0.0242	$3.0199e-11$	$3.1589e-10$
$F_{11}(X)$	$1.1023e-08$	0.0025	$5.5999e-07$	$4.5043e-11$	$1.2057e-10$
$F_{12}(X)$	$4.8011e-07$	0.0020	$4.5726e-09$	$3.0199e-11$	$3.0199e-11$
$F_{13}(X)$	$3.0199e-11$	N/A	N/A	$3.0199e-11$	$3.0199e-11$
$F_{14}(X)$	$3.0199e-11$	N/A	N/A	$3.0199e-11$	$3.0199e-11$
$F_{15}(X)$	N/A	$1.1970e-12$	$1.6572e-11$	$1.2118e-12$	$1.2118e-12$
$F_{16}(X)$	$1.2118e-12$	N/A	N/A	$1.2118e-12$	$1.2118e-12$
$F_{17}(X)$	N/A	$2.9468e-11$	$2.9468e-11$	$2.9468e-11$	$2.9468e-11$
$F_{18}(X)$	$4.4787e-06$	0.0071	0.0236	0.0200	$1.4930e-11$



(a) (b) (c) (d)

Figure 5: 10 ∗ 10 shortest path planning diagram. (a) ASFSSA. (b) CSSA. (c) SSA. (d) PSO.

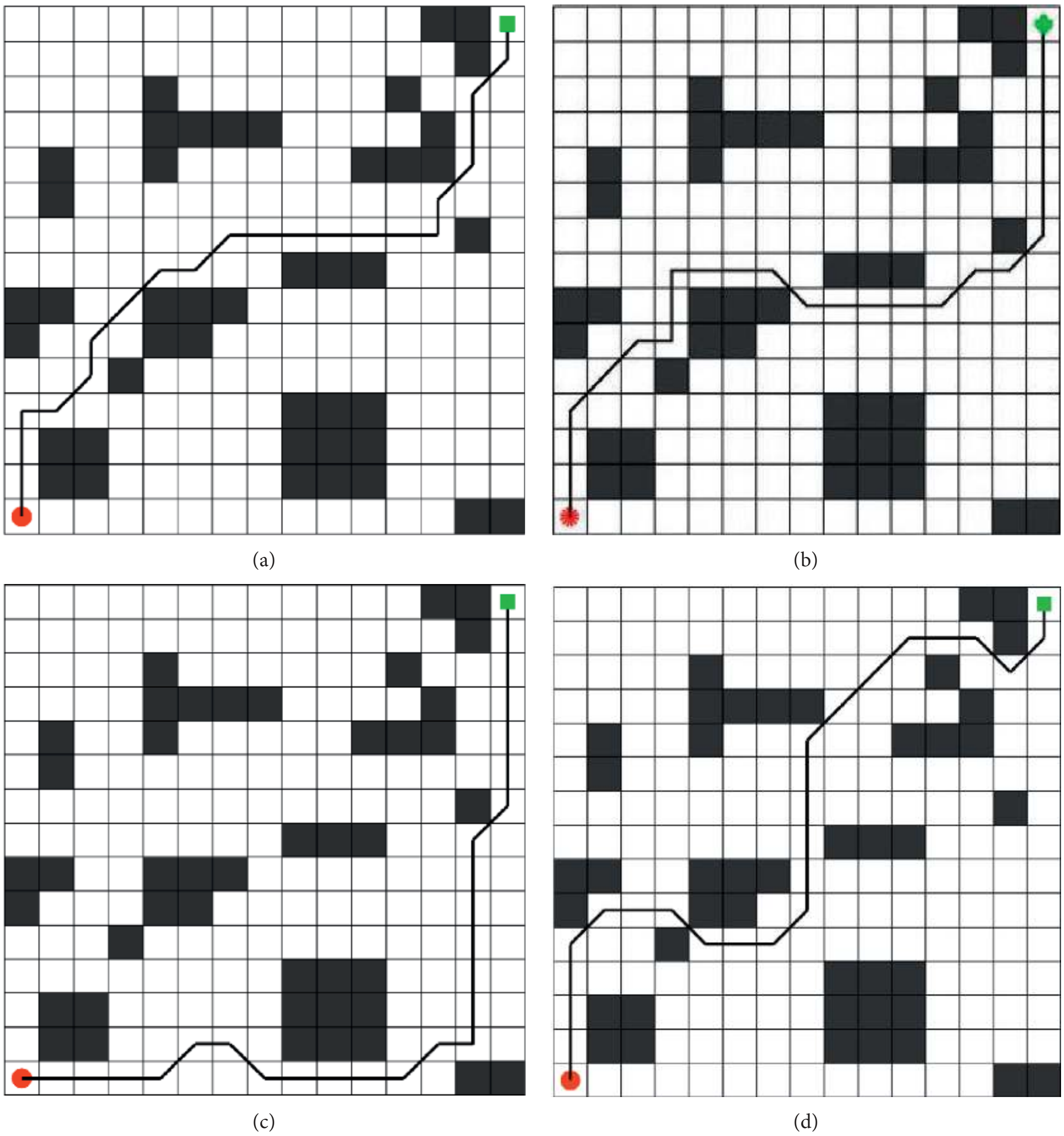


Figure 6: 15 ∗ 15 shortest path planning diagram. (a) ASFSSA. (b) CSSA. (c) SSA. (d) PSO.

optimization performance of the two algorithms is equivalent. Table 4 shows the results for ASFSSA and PSO, GWO, BSO, SSA, and CSSA are at a significance level of $P = 5\%$, where N/A indicates that the performance between the two is similar and not comparable.

As can be seen from Table 3, each algorithm is significantly different from the ASFSSA algorithm, with only a few cases of comparable performance. It can be seen that the introduction of multiple strategies improves the disadvantages of the original algorithm and enhances the optimization ability of the algorithm.

6. Robot Path Planning Based on ASFSSA

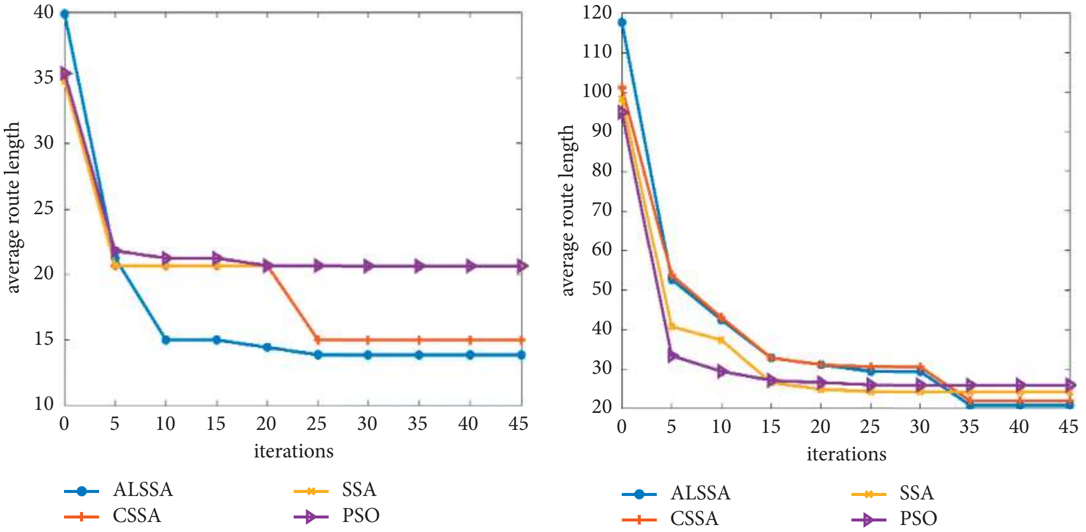
To verify the feasibility and practicability of the improved algorithm, this paper takes a classic case of robot route planning to explore it. Each individual sparrow is a viable path in routing. Assuming there are N possible paths, dimension D is determined by the number of connections from the starting point to the destination point. Using the raster method to model the environment, the raster method is to use 1 ∗ 1 raster to construct the equivalent working environment, and use the raster value to equivalent the obstacles in the location [23, 24]. This effectively equates the working environment of the robot to a plane, similar to the lattice effect, and then determines the feasible and obstacle zones based on the raster values.

The raster number defines 0 as the feasible domain and 1 as the obstacle zone, so the robot can plan its path on the raster with a value of 0. Dimension D is the number of columns in the raster map and the cost function of the path length of the i-th sparrow individual, as shown in the following equation:

$$f(x_i) = \sum_{j=1}^{D-1} \sqrt{\left(x_{j+1} - x_j\right)^2 + \left(y_{j+1} - y_j\right)^2}. \qquad (11)$$

Table 5: Statistics table of optimization route by algorithms.

Map size	Performance index	ASFSSA	CSSA	SSA	PSO
10 ∗ 10	Shortest	12.7279	12.7279	15.5564	15.5564
	Average	13.8593	14.4250	20.6475	21.2132
	Worse	15.5564	20.6475	21.2132	41.0122
20 ∗ 20	Shortest	19.7990	22.6274	22.6274	25.4559
	Average	20.9304	23.7588	24.3244	26.0215
	Worst	22.6274	25.4559	31.1127	31.1127



(a)

(b)

Figure 7: The average route convergence graph of each algorithm. (a) 10 ∗ 10. (b) 15 ∗ 15.

In equation (11), j is the j-th dimension of a sparrow individual.

6.1. Experimental Environment Settings. To better verify the practicability and feasibility of the improved algorithm, the improved algorithm is used to route the raster maps of the two models. Compared with CSSA, SSA, and PSO, the number of population is 20 and the number of iterations is 100. Other environmental parameters are consistent with the above.

6.2. Simulation Results and Analysis. The optimal route of each algorithm in the two model graphs is shown in Figures 5 and 6. In order to eliminate the influence of chance, each algorithm is tested for 10 times, and the shortest, average, and worst route of each algorithm are counted. Three indicators are used to measure the stability and feasibility of each algorithm in this experiment. The optimization statistical table of each algorithm is shown in Table 5, and the average route convergence chart is shown in Figure 7.

As shown in Figures 5 and 6, the ASFSSA route is the simplest and clearest, followed by CSSA, while PSO and SSA are obviously trapped in local optimum. From Table 5, we can see that ASFSSA has the best searching ability and stability in both of raster graphs, and CSSA has better stability. However, from Figure 7, we can see that the convergence of SSA and PSO is insufficient, and the optimization results of SSA and PSO are extremely unstable and poor. Therefore, in the two model diagrams, the introduction of multiple strategies makes the algorithm flexible in the search, greatly improves the search ability of the algorithm, and plans a route with the least cost.

7. Conclusion

The sparrow search algorithm has better performance than other algorithms in optimization, but it is still easy to get stuck in local and rely on population initialization. This paper analyzes these defects and proposes an adaptive spiral flying sparrow search algorithm. Firstly, tent mapping based on random variables is used to initialize the population, which makes the distribution of sparrow individuals uniform and helps the individuals to work better. Secondly, the adaptive weights strategy and Levy flight mechanism are used in the discoverer stage, which makes the discoverer flexible and adaptable in the optimization process and reduces the traditional regular strategy. Finally, the variable spiral search strategy is used in the follower stage, which makes the follower search method more detailed and in-depth, and improves the convergence accuracy of the algorithm. The effectiveness of the improved algorithm is verified by 18 test functions and the comparison of other algorithms. Compared with other algorithms, ASFSSA has significant optimization capabilities. At the same time, the feasibility and practicability of the improved algorithm are verified by the robot path planning study. The path planned by the improved algorithm is clear, and the cost function is the smallest. It can be seen that the introduction of multiple strategies has effectively improved the optimization ability of the basic sparrow search algorithm. However, in some functions, the optimization effect of ASFSSA is second only to the BSO algorithm. In the future research phase, how to improve these optimization effect is the research focus, and it needs to be upgraded to more complex applications.

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. 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.

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

[3] 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. Computational Cybernetics and Simulation*, vol. 5, IEEE, Orlando, FL, USA, October 1997.

[4] Y. Lei, G. De, and L. Fei, "Improved sparrow search algorithm based DV-hop localization in WSN," in *Proceedings of the 2020 Chinese Automation Congress (CAC)*, Shanghai, China, November 2020.

[5] L. Xin, X. Mu, J. Zhang, and Z. Wang, "Chaotic sparrow search optimization algorithm," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 1–10, 2021.

[6] A. Tang, T. Han, D. Xu et al., "Path planning method of unmanned aerial vehicle based on chaos sparrow search algorithm," *Systems Engineering and Electronics*, vol. 41, no. 7, pp. 2128–2136, 2020, https://kns.cnki.net/kcms/detail/51.1307.TP.20201124.1519.002.html.

[7] L. Xin, 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.

[8] Q. Mao, Q. Zhang, C. Mao, and J. Bai, "Hybrid sine cosine algorithm and Levy's flying sparrow algorithm," *Journal of Shanxi University (Natural Science Edition)*, pp. 1–6, 2021.

[9] W. Song, S. Liu, X. Wang, and W. Wu, "An improved sparrow search algorithm," in *Proceedings of the 2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, England, UK, December 2020.

[10] X. Wu, Z. Han, L. Wei et al., "Intelligent on-demand adjustment algorithm and key technology of mine air flow," *Journal of China University of Mining & Technology*, vol. 50, no. 4, pp. 725–734, 2021.

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

[12] J. Zhang, K. Xia, Z. He et al., "Semi-supervised ensemble classifier with improved sparrow search algorithm and its application in pulmonary nodule detection," *Mathematical Problems in Engineering*, vol. 2021, Article ID 6622935, 18 pages, 2021.

[13] Y. Li, M. Han, and Q. Guo, "Modified whale optimization algorithm based on tent chaotic mapping and its application in structural optimization," *KSCE Journal of Civil Engineering*, vol. 24, no. 12, pp. 3703–3713, 2020.

[14] X. Liang, W. Li, Y. Zhang, and M. Zhou, "An adaptive particle swarm optimization method based on clustering," *Soft Computing*, vol. 19, no. 2, pp. 431–448, 2015.

[15] S. N. Chegini, A. Bagheri, and F. Najafi, "PSOSCALF: a new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems," *Applied Soft Computing*, vol. 73, pp. 697–726, 2018.

[16] X. Chen, X. Huang, D. Zhu, and Y. Qiu, "Research on chaotic flying sparrow search algorithm," *Journal of Physics: Conference Series*, vol. 1848, no. 1, Article ID 012044, 2021.

[17] H. Haklı and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight," *Applied Soft Computing*, vol. 23, pp. 333–345, 2014.

[18] A. A. Heidari and P. Pahlavani, "An efficient modified grey wolf optimizer with Lévy flight for optimization tasks," *Applied Soft Computing*, vol. 60, pp. 115–134, 2017.

[19] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[20] X. Tao, W. Guo, Q. Li, C. Ren, and R. Liu, "Multiple scale self-adaptive cooperation mutation strategy-based particle swarm optimization," *Applied Soft Computing*, vol. 89, Article ID 106124, 2020.

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

[22] X. Jiang and S. Li, "BAS: Beetle Antennae Search Algorithm for Optimization Problems," https://arxiv.org/abs/1710.10724.

[23] F. Xu, H. Li, C. M. Pun, and H. Hu, Y. Li, Y. Song, H. Gao, A new global best guided artificial bee colony algorithm with application in robot path planning," *Applied Soft Computing*, vol. 88, Article ID 106037, 2020.

[24] J. Li, "Robot path planning based on improved sparrow algorithm," *Journal of Physics: Conference Series*, vol. 1861, no. 1, Article ID 012017, 2021.