

Research Article

Improved Sparrow Search Algorithm Based on Iterative Local Search

Shaoqiang Yan ¹, Ping Yang ¹, Donglin Zhu ², Wanli Zheng ¹, and Fengxuan Wu ¹

¹*Xi'an Research Institute of High Technology, Xi'an, Shaanxi 710025, China*

²*School of Information Engineering, Jiangxi University of Science and Technology, Ganzhou, Jiangxi 341000, China*

Correspondence should be addressed to Shaoqiang Yan; yanshaoqiang668@163.com

Received 11 October 2021; Accepted 23 November 2021; Published 15 December 2021

Academic Editor: Ripon Chakraborty

Copyright © 2021 Shaoqiang Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper solves the shortcomings of sparrow search algorithm in poor utilization to the current individual and lack of effective search, improves its search performance, achieves good results on 23 basic benchmark functions and CEC 2017, and effectively improves the problem that the algorithm falls into local optimal solution and has low search accuracy. This paper proposes an improved sparrow search algorithm based on iterative local search (ISSA). In the global search phase of the followers, the variable helix factor is introduced, which makes full use of the individual's opposite solution about the origin, reduces the number of individuals beyond the boundary, and ensures the algorithm has a detailed and flexible search ability. In the local search phase of the followers, an improved iterative local search strategy is adopted to increase the search accuracy and prevent the omission of the optimal solution. By adding the dimension by dimension lens learning strategy to scouters, the search range is more flexible and helps jump out of the local optimal solution by changing the focusing ability of the lens and the dynamic boundary of each dimension. Finally, the boundary control is improved to effectively utilize the individuals beyond the boundary while retaining the randomness of the individuals. The ISSA is compared with PSO, SCA, GWO, WOA, MWOA, SSA, BSSA, CSSA, and LSSA on 23 basic functions to verify the optimization performance of the algorithm. In addition, in order to further verify the optimization performance of the algorithm when the optimal solution is not 0, the above algorithms are compared in CEC 2017 test function. The simulation results show that the ISSA has good universality. Finally, this paper applies ISSA to PID parameter tuning and robot path planning, and the results show that the algorithm has good practicability and effect.

1. Introduction

With the continuous emergence of various optimization problems, various algorithms and improved algorithms are emerging [1–4]. The emergence of swarm intelligence algorithm provides new ideas for solving various optimization problems. The swarm intelligence optimization algorithm is a meta-heuristic optimization algorithm that imitates the behavior of biological populations or biological behaviors and natural phenomena in the natural world. As the optimization effect of swarm intelligence algorithms is recognized by the public, swarm intelligence algorithms develop continuously, and more and more new swarm intelligence algorithms are proposed, such as firefly algorithm (FA) [5],

ant lion optimizer (ALO) [6], whale optimization algorithm (WOA) [7], sine cosine algorithm (SCA) [8], crow search algorithm (CSA) [9], Harris hawks optimization algorithm (HHO) [10], slime mould algorithm (SMA) [11], hunger games search (HGS) [12], Runge–Kutta method (RUN) [13], and colony predation algorithm (CPA) [14].

Sparrow search algorithm (SSA) is a new group intelligence optimization algorithm proposed by Xue and Shen [15] in 2020. Inspired by sparrow foraging behavior, the algorithm has obvious advantages over traditional intelligent optimization algorithms, such as grey wolf Optimizer (GWO) [16], particle swarm optimization (PSO) [17], and genetic algorithm [18], with high stability, good search accuracy, and fast convergence [19]. Despite its fast

convergence rate, the algorithm is prone to fall into local optimum and the results of optimization are random. To overcome this shortcoming, many scholars have proposed improved algorithms based on different strategies for the sparrow search algorithm and successfully solved many engineering problems. Based on the principle and model of sparrow search algorithm, Lv et al. proposed a fusion algorithm of bird swarm algorithm and sparrow search algorithm [20] and a chaotic sparrow search algorithm [21]. The former uses the search mechanism of bird swarm algorithm to add to the discoverers and followers of the sparrow search algorithm, which changes the update strategy of "full dimension reduction," effectively breaks through the local restriction of search, and strengthens the global search ability; and the latter uses tent chaotic map to initialize the population, which makes the population more uniform. After one iteration, the second iteration of chaotic disturbance and Gaussian variation is carried out according to individual fitness and average fitness, which prevents local aggregation in the optimization process, enhances its ability to jump out of local optimization, and achieves good results in the application of image segmentation. A chaotic sparrow algorithm based on cubic mapping and elite reverse learning to initialize population is presented by Tang et al. [22]. At the same time, the sinusoidal algorithm is introduced, which balances the development and exploring ability of the algorithm. At the same time, when the algorithm comes to a standstill, the Gauss Walk strategy is used to jump out of the standstill, and its optimization performance is verified in 15 benchmark functions. Finally, the UAV track planning simulation is carried out in the case of threat. Compared with other optimization algorithms, the algorithm obtains the safe and feasible track with the best cost and meets the constraints. Ouyang et al. proposed a learning sparrow algorithm [23] adding lens reverse learning during the discoverer search phase makes the search more flexible and increases the diversity of the population. A spiral guidance mechanism is introduced to make the discoverer search more precise. Then, a local search mechanism is added to prevent the omission of high-quality solutions, and compared with other swarm intelligence algorithms in 12 basic test functions and CEC 2017 test sets, this shows good optimization ability. Finally, the improved sparrow search algorithm is validated in the robot path planning, and a stable and safe optimal path is planned.

The above algorithms have made some improvements on the basis of sparrow algorithm, but there are still some shortcomings:

- (1) There is still some randomness in the improved method of population initialization, which does not guarantee absolute uniformity of the population each time it is initialized.
- (2) The selected improved search strategy is subject to regional limitations, is easy to exceed the boundaries, and fails to perform effective global search in the whole space, resulting in a large number of individuals exceeding the boundaries and still trapped in local optimum.

- (3) By jumping directly to the discoverer, it is easy to miss the optimal solution. The improvement of local search accuracy is not significant, and there can be more improvement in search accuracy.
- (4) In terms of boundary control, the strategy of updating to boundary is adopted for individuals beyond the boundary, which does not make good use of the individual location and reduces the diversity of the population.

In order to solve the above problems, the sparrow search algorithm based on iterative local search is proposed in this paper. By variable helix factor and improved iterative local search, the effective utilization and search of individuals are improved. By adding a dimension by dimension lens learning strategy and change the focusing ability of the lens, the algorithm converges faster while helping to jump out of the local optimum, and improving the boundary strategy, the population diversity is increased. To verify the optimization performance of the algorithm, ISSA PSO, SCA, GWO [16], WOA, MWOA [24], SSA, BSSA, CSSA, and LSSA are tested and analyzed on 23 basic functions. To further verify the universality of the algorithm, the above algorithms are tested and analyzed in CEC 2017 test function [25,26]. Finally, ISSA is applied to PID parameter tuning [27]. The accuracy and convergence speed of the tuned results are improved compared with the SSA, which shows that the algorithm has good practicability. The main contributions of this paper are as follows:

- (1) In order to improve the effective use and search of individuals, a variable helix factor strategy is proposed and boundary control is improved
- (2) An improved iterative local search strategy is presented to improve the problem of low accuracy and missing better solutions in the search process
- (3) In order to improve the ability of the algorithm to jump out of local optimization, a dimension by dimension lens learning method is proposed to change the lens focusing ability
- (4) The versatility and flexibility of the algorithm using benchmark functions and CEC 2017 functions are validated
- (5) ISSA is used to optimize PID parameters to help quickly complete PID parameter tuning
- (6) ISSA is used to optimize the robot path planning problem and help get fast and stable results

The main work arrangement of this paper is as follows: Section 2 introduces the basic sparrow algorithm. Section 3 introduces and analyses the ISSA. Section 4 compares and analyzes the algorithm on the basic test function. Section 5 compares and analyzes the algorithm on CEC 2017. Section 6 applies the algorithm to PID parameter tuning. Section 7 applies the algorithm to robot path planning. Section 8 discusses and provides future research directions.

2. Sparrow Search Algorithm

In the process of sparrow foraging, there are two behavioral strategies: discoverer and follower. The individuals with better positions in the sparrow population generally take 10%–20% of the total population as discoverers, while the remaining individuals take part in the process. At the same time, 10%–20% of individuals are randomly assigned as scouters. The discoverer is responsible for leading the population in search direction and finding food, while the follower follows the discoverer to obtain food, and the scouter is alert to environmental threats and warns the sparrow population to move closer to a safe area.

In order to describe the process of sparrow foraging through mathematical models, it is necessary to formulate rules to simplify various behaviors of sparrows. The specific rules are as follows:

- (1) The individual energy of sparrow population depends on individual fitness evaluation, and the individual energy of discoverer is higher than that of discoverers.
- (2) Once the scouters in the sparrow population find the threat of the external environment, they begin to send out an alarm signal. When the alert value is greater than the security threshold, the discoverers direct the population to the security zone.
- (3) Sparrows have flexible individual behavior strategies and can switch between discoverers and followers. As long as the individual energy reaches a certain degree, they can become discoverers, but the proportion between discoverers and followers in the population remains unchanged.
- (4) Sparrows with low energy may fly to other places for feeding in order to obtain higher energy.
- (5) When there is an external environmental threat, the sparrows at the edge of the population will quickly move to the safe area, and the sparrows in the middle of the population will immediately swim away to get close to other sparrows.

The discoverer is responsible for guiding the population to forage or to the location of the safe zone. The location update is described below:

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

Among them, t represents the current number of iterations, and M is the maximum number of iterations. $X_{i,j}$ represents the current position of the i -th sparrow in the j -th dimension. $\alpha \in [0, 1]$ and is a random number. R_2 represents an early warning value, ST is the security threshold, and $R_2 \in [0, 1]$, $ST \in [0.5, 1]$. Q represents a random number that follows a normal distribution. L represents a $1 \times d$ matrix with all elements 1. When $R_2 < ST$, this indicates that the population environment is safe at this time, no predators are

found around them, and the discoverers can conduct extensive searches to guide the population to higher energy levels. When $R_2 \geq ST$, this indicates that an individual within a population has discovered a predator and issued an alert, and that the discoverer quickly adjusts the search strategy to flee the current location, leading the population to a safe location.

In order to obtain high-quality food, followers follow the discoverer or forage alone, so the location of followers is updated as follows:

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

Among them, X_p is the best position currently occupied by the discoverer, and X_{worst} represents the worst position currently. A is a $1 \times d$ matrix with only 1 or -1 elements, where $A^+ = A^T (AA^T)^{-1}$. When $i > n/2$, this indicates that the less adaptable 1st participant is not getting food, is very hungry, and needs to fly elsewhere to get more energy; when $i < n/2$, followers monitor the finder and compete for food with the finder with a higher predator, thereby increasing their energy.

When aware of the danger, the sparrow population will make antipredation behavior, and its mathematical expression is as follows

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

where X_{best} represents the current global optimal position. β is the control step parameter and is a normally distributed random number with a mean value of 0 and a variance of 1. K is a random number belonging to $[-1, 1]$, which controls the direction of the sparrow's movement as well as the step. f_i represents the fitness value of the current sparrow individual. f_g and f_w are the current optimal and worst fitness values, respectively. ε is a very small real number that prevents the denominator from being zero. When $f_i \neq f_g$, this indicates that the current sparrow is at the edge of the population and that an individual is vulnerable to predators, and it is necessary to approach other individuals in the population center to reduce the risk of predation. When $f_i = f_g$, this indicates that individuals in the center of the population are aware of the danger and need to flee from their current location in order to avoid it.

3. Sparrow Search Algorithm Based on Iterative Local Search (ISSA)

3.1. Variable Helix Factor. Followers occupy the majority of individuals in the population. When $i < (N/2)$, they have a unique update mechanism that draw closer to the

discoverers quickly; it results in a fast convergence rate. When $i > (N/2)$, they have the ability to search globally. However, the global search ability is not strong, limited by the boundary of the search area, which tends to cause aggregation at the boundary in the early stage, resulting in loss of population diversity, easy to fall into the local extreme phenomenon, and poor ability to jump out of the local optimal.

The location update method of follower adopts the random coefficient that obeys the normal distribution. Without considering the boundary, the coefficient has strong global search ability. However, when considering the boundary, this mechanism is detrimental to the individuals who are at or near the boundary. Many absolute coefficients exceed 1. When the boundary is exceeded, it causes the population individuals to aggregate at the boundary, does not make full use of the current location, and results in a significant decrease in population diversity and overall algorithm performance. Based on this, a variable helix factor is proposed to reduce the number of individuals beyond the boundary, control the search step and direction, make full use of the whole population space, the space for early search is large, maintain the diversity of the population, and help jump out of the trap of local optimum. Later local search is more detailed, which greatly improves the search ability of the algorithm, as shown in Figure 1.

The formula for the variable helix factor works as follows:

$$H = a \cdot \cos(k \cdot l \cdot \pi), \quad (4)$$

$$a = \begin{cases} 1, & t < \frac{M}{2}, \\ e^{5 \cdot l}, & \text{otherwise,} \end{cases} \quad (5)$$

$$x_{i,j}^{t+1} = \begin{cases} \cos(a \cdot l \cdot \pi) \cdot \exp\left(\frac{x_{\text{worst}}^t - x_{i,j}^t}{t^2}\right), & i > \frac{N}{2} \text{ and } t < \frac{M}{2}, \\ e^{5 \cdot l} \cdot \cos(a \cdot l \cdot \pi) \cdot \exp\left(\frac{x_{\text{worst}}^t - x_{i,j}^t}{t^2}\right), & i > \frac{N}{2} \text{ and } t > \frac{M}{2}. \end{cases} \quad (7)$$

3.2. Improved Iterative Local Search. When $i < (N/2)$, the followers have a unique update mechanism that quickly closes to the discoverer's optimal solution, which results in fast convergence of the algorithm. The followers jump directly to the neighborhood of the discoverer's optimal solution. Although they have some development ability near the current optimal solution, they do not make enough use of the current solution and have poor stability. They cannot guarantee the quality and accuracy of the solution and have poor local development ability. Once trapped and unable to jump out of the local extreme state, the overall performance of the algorithm will be limited. Inspired by [28–30], this paper presents an improved iterative local search.

$$l = 1 - 2 \cdot \frac{t}{M}. \quad (6)$$

H is a variable helix factor, a is a parameter used to control the helix, with a value of 1 in the earlier period and a decreasing number of iterations in the later period; k is a parameter representing the helix cycle, $(M/10)$ in general; and l is a parameter that decreases linearly from 1 to -1 in terms of the number of iterations.

In this paper, we select a high-dimensional test function, step function, and a low-dimensional test function, Shekel function, to test the original model and the improved model for individuals beyond the boundary, as shown in Figure 2. Set the population individual as 30 and the number of iterations as 500, run each for 10 times, and take the average value. The test results are shown in Table 1.

According to Table 1, when the followers conduct extensive search, the number beyond the boundary accounts for the vast majority of the total number beyond the boundary of the sparrow population, resulting in the loss of a large number of individuals. Therefore, it is necessary to restrict the extensive search scope of the followers; based on the variable helix factor, the number of times the followers exceeding the boundary is greatly reduced to 0, which fully retains the favorable position information of the current individual.

Improvements to the extensive search of the followers make it possible for the followers to make full use of the whole search space, get rid of the attraction of the local optimal solution more easily, strengthen the search for the whole space, maintain the diversity of the population, enhance the ability of algorithm exploration in the early stage, and enhance the ability of algorithm development in the later stage. Based on this, the formula is updated as follows:

Local search algorithm [31] is a simple greedy search algorithm that is improved from the hill-climbing method. Local search starts from an initial solution, then searches the neighborhood of the solution, and updates the solution if there is a better solution or returns to the current solution. Iterative local search is an exploratory method that adds perturbation to the local optimal solution obtained by local search and then re-searches the local solution.

The improved iterative local search first performs a local search near the initial solution, then disturbs the initial solution by updating the location of the followers closer to the discoverer, and then searches the updated location again. It makes full use of the location information of the current individual

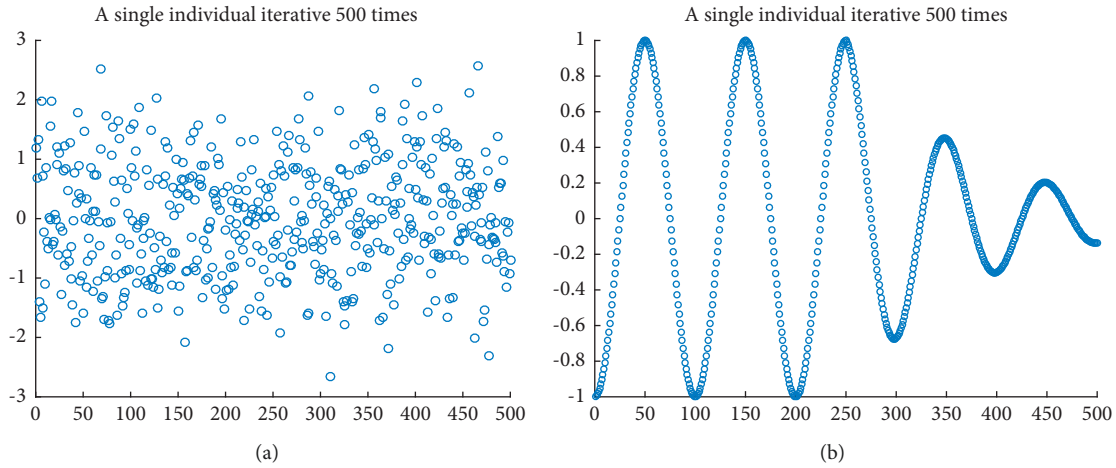


FIGURE 1: Coefficient model. (a) Original random coefficient. (b) Variable helix factor.

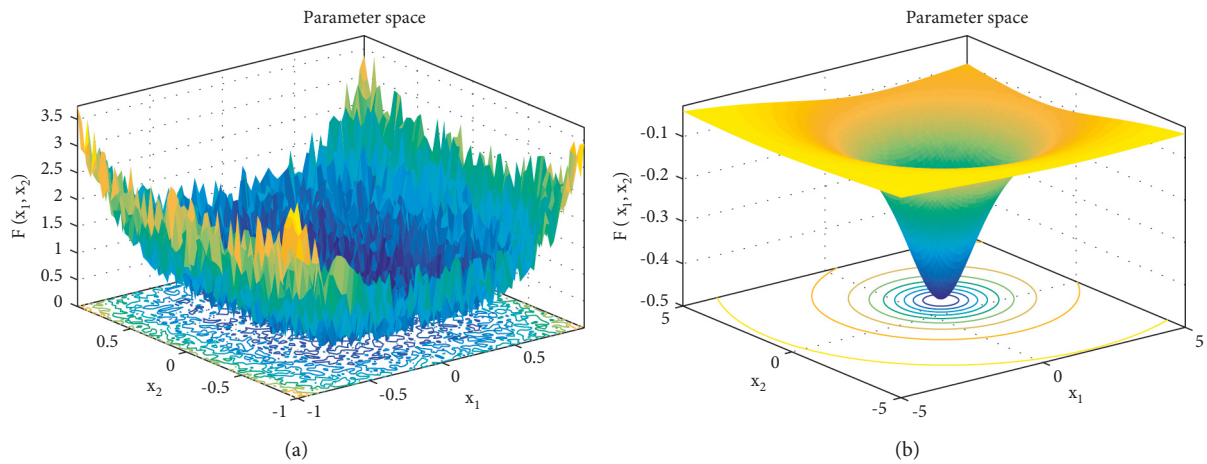


FIGURE 2: Parameter space. (a) Step function. (b) Shekel function.

TABLE 1: Statistics on the number of individuals beyond the boundary.

| Function | Step | Shekel |
|---|------------------|------------|
| Dimension | 30 dim | 4 dim |
| Boundary | $[-1.28 \ 1.28]$ | $[0 \ 10]$ |
| Number of times the original algorithm follower exceeded the boundary | 1505 | 3739 |
| Number of times the follower exceeded boundaries after improvement | 0 | 0 |
| Total number of times the original algorithm exceeded the boundary | 1696 | 4381 |
| Total number of times the improved algorithm exceeded the boundary | 161 | 1095 |

and the location information of the current optimal solution to make the search more flexible. The first modification uses the mechanism by which the follower approaches the discoverer to disturb, so that the individual jumps near the current global optimal position to jump out of the local optimal position; the second modification uses the local search to obtain stable optimization results and improve the accuracy of the local search; and the third modification uses the local search result to obtain the optimal local search result to ensure the quality of the solution, as shown in Figure 3.

At the same time, two cases are illustrated in this paper. Figure 3(a) indicates that after individual disturbance, a better solution can be found near the current global optimal

solution; and Figure 3(b) indicates that an individual can find a better solution using his own favorable position, both of which help to jump out of the local optimal solution. This paper has not been replaced by the current global optimal solution to maintain population diversity and prevent premature convergence.

3.2.1. *The First Modification (SSA Method)*. The improved iterative local search strategy is more suitable for this algorithm than the original iterative local search. It works by first disturbing the initial solution to get an intermediate solution (when $i < n/2$, the followers of the SSA are disturbed

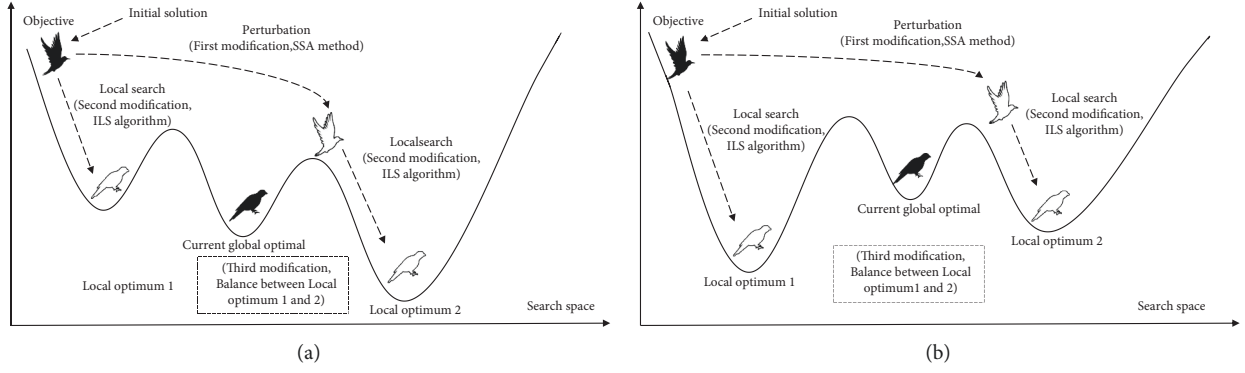


FIGURE 3: Main idea of improved iterative local search.

by a unique update mechanism that closes quickly to the finder's optimal solution). The initial and intermediate solutions are then searched again for a better solution. The first modification is used to solve the problem that the SSA is easy to fall into local optimal solution. The algorithm flow is as follows:

Step 1. The initial solution X^* is perturbed by the current optimal solution X_{best} to obtain the intermediate solution X^{**} . The perturbation formula is as follows:

$$X^{**} = X_{\text{best}} + |X^* - X_{\text{best}}| \cdot A^+ \cdot L. \quad (8)$$

The formula for the original followers is as follows:

$$x_{i,j}^{t+1} = x_p^{t+1} + |x_{i,j}^t - x_p^{t+1}| \cdot A^+ \cdot L, \quad i < \frac{N}{2}, \quad (9)$$

X_{best} is x_p^{t+1} , which is the best position the discoverer currently occupies; X^* is $x_{i,j}^t$, which is the current position; X^{**} is $x_{i,j}^{t+1}$, which is the updated location.

3.2.2. The Second Modification (ILS Algorithm). The second modification solves the problem of unstable and inaccurate optimization results of the SSA. The ILS algorithm here represents the local search stage in the ILS algorithm. It works by searching the initial and intermediate solutions locally to get a better solution and effectively utilizes the current position of the initial solution to prevent the individual from missing the better solution in the process of jumping directly to the current optimal solution. At the same time, local search near the current optimal solution helps to improve the accuracy of the solution and jump out of the local optimum in a small range. The algorithm flow is as follows:

Step 2. Initial solution X^* starts local search, and the formula is as follows:

$$X^1 = X^* \cdot \text{rand}(), \quad (10)$$

$\text{rand}()$ is a random number between 0 and 1.

Step 3. Intermediate solution X^{**} starts local search, and the formula is as follows:

$$X^2 = X^{**} \cdot \text{rand}(). \quad (11)$$

3.2.3. The Third Modification. The third modification is to optimize the local search results to ensure the quality of the solution. The working principle is to use greedy strategy to compare the local search results of the initial and intermediate solutions and select the better value as the final solution X . The algorithm flow is as follows:

Step 4. Calculate fitness $f(X^1)$ of X^1 .

Step 5. Calculate fitness $f(X^2)$ of X^2

Step 6. Compare $f(X^1)$ and $f(X^2)$ to select the best individuals for location updates; that is,

$$\begin{aligned} &\text{If } f(X^1) \leq f(X^2) \\ &X = X^1 \\ &\text{else } X = X^2 \\ &\text{end} \end{aligned}$$

Based on the SSA and the three modifications above, the formula of the follower ($i < N/2$) in the SSA is updated as follows:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t \cdot \text{rand}(), & i < \frac{N}{2} \text{ and } f(X^1) \leq f(X^2), \\ \left(x_p^{t+1} + |x_{i,j}^t - x_p^{t+1}| \cdot A^+ \cdot L \right) \cdot \text{rand}(), & i < \frac{N}{2} \text{ and } f(X^1) > f(X^2). \end{cases} \quad (12)$$

3.3. Dimension by Dimension Lens Imaging Learning. Swarm intelligence algorithms have the disadvantage of easily falling into local optimum. In this regard, relevant scholars have proposed the method of adding reverse learning to swarm intelligence algorithm [32–34]. The solution after reverse learning can be closer to the optimal solution. Generally, reverse learning can only search for the optimal solution in a certain space, but it still has monotonicity and the possibility of falling into local optimum. Lens learning [35, 36] has better optimization ability than general direction learning and can continuously converge to the optimal solution in a certain space. However, once there is no optimal solution in the selected space, it will still lead to local optimization in the end. At the same time, scouts have antipredator behavior to help the population to jump out of the local optimum, but their ability to jump out of the local optimum is unstable, resulting in sometimes unable to jump out of the local optimum. In view of this phenomenon, this paper proposes a dimension by dimension lens imaging learning strategy to change the focusing ability of the lens, which is used to strengthen the scouts' ability to jump out of the local optimum, lens learning for each dimension, and reduce the mutual interference between the dimensions. In the early stage, the lens with poor focusing ability is selected for reverse learning. At this time, the imaging is divergent and far away from the lens, which can help jump out of the local optimum, as shown in Figure 4(a); in the later stage, the lens with strong focusing ability is selected for reverse learning. The imaging is concentrated and close to the lens, which can accelerate the convergence, as shown in Figure 4(b). Comparison of three kinds of reverse learning is shown in Table 2.

The principle of lens imaging is as follows:

As shown in Figure 4, taking one-dimensional space as an example, it is assumed that there is an individual with a height of h at the position. Under the action of the lens, it forms an image with a height of h^* at the position x^* . a, b is the boundary, and the lens position is the midpoint of $[a, b]$. According to the principle of lens imaging and triangle similarity principle:

$$\frac{((a+b)/2) - x}{x^* - ((a+b)/2)} = \frac{h}{h^*} = k. \quad (13)$$

By transforming the above formula, we can get

$$x^* = \frac{a+b}{2} + \frac{a+b}{2k} - \frac{x}{k}, \quad (14)$$

k is used to indicate the focusing ability of the lens, that is, the imaging size. When $k = 1$, it can be simplified as follows:

$$x^* = a + b - x. \quad (15)$$

This is the general reverse learning strategy. It can be seen that the general reverse learning strategy is a special case of lens imaging. The general reverse learning strategy k is fixed, and the obtained individuals are also fixed. Lens learning can change the position of individuals by adjusting k , so as to further enhance the diversity of groups. Generally, it takes a constant that is not equal to 1. This paper proposes

a strategy of linearly increasing K according to the number of iterations, that is

$$k = a + \frac{t}{M}, \quad (16)$$

where a is a small constant to prevent the previous iterative imaging from being too large, which is taken as 0.1 in this paper. The early k is small, and the imaging is large; in the later stage, k is near 1, and the imaging is slightly smaller, which can help convergence.

At the same time, lens imaging is extended to each dimension, and lens imaging reverse learning is performed for each dimension. The formula is extended as follows:

$$x_j^* = \frac{a_j + b_j}{2} + \frac{a_j + b_j}{2k} - \frac{x_j}{k}, \quad (17)$$

where j is the current dimension, a_j is the lower bound of the j -th dimension, and b_j is the upper bound of the j -th dimension. At the same time, this paper adopts dynamic boundary:

$$\begin{aligned} a_j &= \min(x_j), \\ b_j &= \max(x_j), \end{aligned} \quad (18)$$

where $\min(x_j)$ is the minimum value of the j -th dimension in all individuals, and $\max(x_j)$ is the maximum value of the j -th dimension in all individuals. Because a_j and b_j do not represent the boundary of the whole search space, when the imaging exceeds the boundary $[a_j, b_j]$, it may not exceed the boundary of the whole search space. Therefore, when k is small in the current period, the imaging will exceed the boundary of the current j -th dimension, which helps to expand the search range, reduces the possibility of premature stagnation in the early stage, and helps to jump out of the local optimum. Finally, the greedy strategy is adopted. If the fitness value of the reverse solution is small and better than the original solution, the solution is updated and applied to the algorithm as follows:

$$x_{i,j}^{t+1} = \begin{cases} \frac{a_j^t + b_j^t}{2} + \frac{a_j^t + b_j^t}{2k} - \frac{x_{i,j}^t}{k}, & f(x^*) < f(x), \\ x_{i,j}^{t+1}, & \text{otherwise.} \end{cases} \quad (19)$$

3.3.1. Verify the Ability to Jump Out of Local Optimization.

In this paper, the Shekel function mentioned above is selected as an example to draw the individual distribution diagram between the improved algorithm and the original algorithm to verify the ability of the above strategy to jump out of local optimization. The function image is shown in Figure 2. The selected population is 100, and the maximum number of iterations is 20. The individual distribution of the two algorithms are shown in Figures 5 and 6.

As can be seen from Figure 5, most individuals in the original algorithm have local aggregation and fall into local optimization. As can be seen from Figure 6, the improved

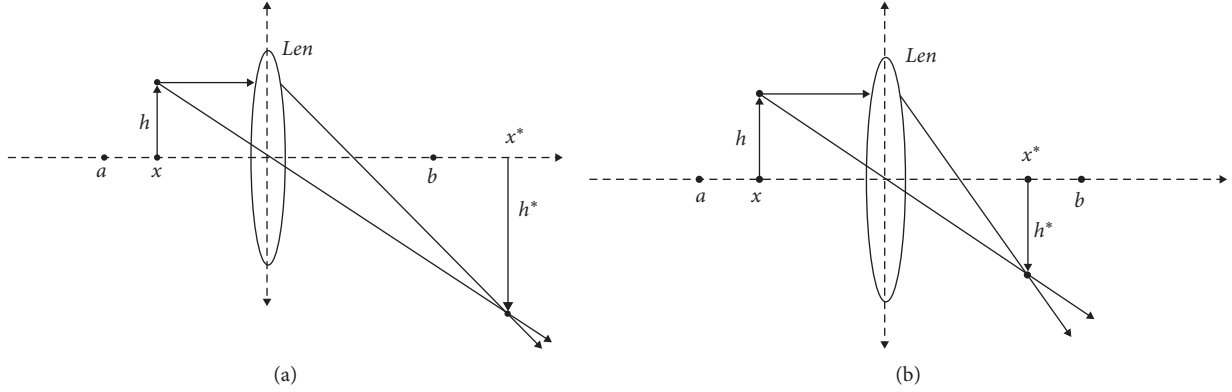


FIGURE 4: Main principles of lens imaging.

TABLE 2: Comparison of three kinds of reverse learning.

| | Boundary | Focusing ability (k) | Reverse solution position | Effect |
|--|--|---|---|---|
| Reverse learning | Unchanged | 1 | On boundary midpoint symmetry | Accelerating convergence |
| Lens imaging learning | Dynamic change according to the maximum and minimum of individual position of population | A large constant, which is more than 1 | About boundary midpoint reduction imaging | Accelerating convergence |
| Dimension by dimension lens imaging learning | The maximum and minimum values of each dimension change dynamically according to the individual position of the population | According to the dynamic change of iteration times, the first and middle stages are less than 1, and the later stage is more than 1 | In the early and middle stages, the image is enlarged at the midpoint of the boundary, and in the later stage, the image is smaller | Jump out of the local optimum in the first and middle stage and accelerate the convergence in the later stage |

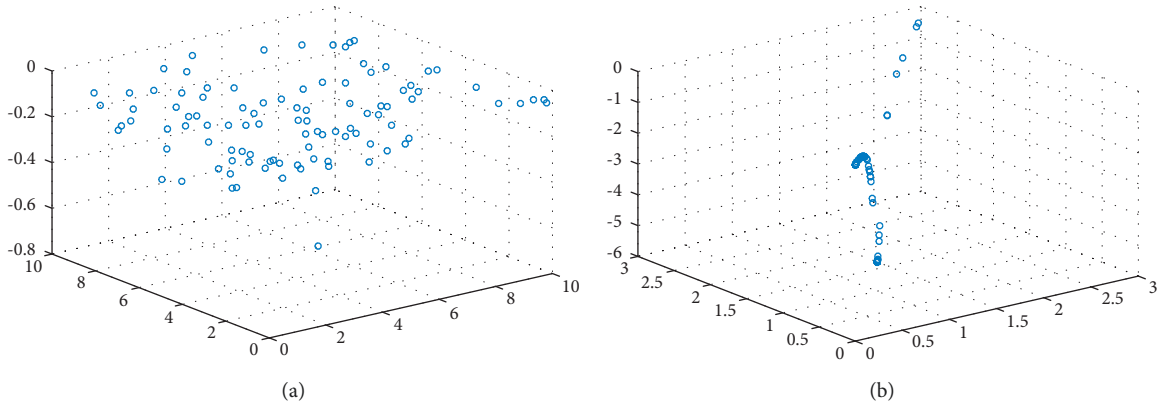


FIGURE 5: Individual distribution of SSA. (a) SSA individual initialization map. (b) Individual distribution of SSA in 20 generations.

algorithm has a larger search space, and most individuals converge near the optimal solution, effectively jumping out of the local optimal solution.

3.3.2. Proof of Convergence of Dimension by Dimension Lens Imaging Learning. The proof of convergence of general refraction reverse learning swarm intelligence algorithm is given in reference [37]. Here, its conclusion is introduced to prove the convergence of the ISSA for dimensional lens back learning. It should be pointed out that the proof of convergence does not necessarily ensure that the algorithm

converges to the global optimal solution. Since the SSA is also a swarm intelligence search algorithm, there are the following theorems:

Theorem 1. *If the SSA algorithm based on general back learning converges, the ISSA algorithm also converges.*

Proof. Let $x_i(t)$ and $x_i^*(t)$ be the current solution and reverse solution of generation t , $x_{i,j}(t)$ and $x_{i,j}^*(t)$ be the values of $x_i(t)$ and $x_i^*(t)$ in the j -th dimension, respectively, and the global optimal solution is x^{best} . From Theorem 1,

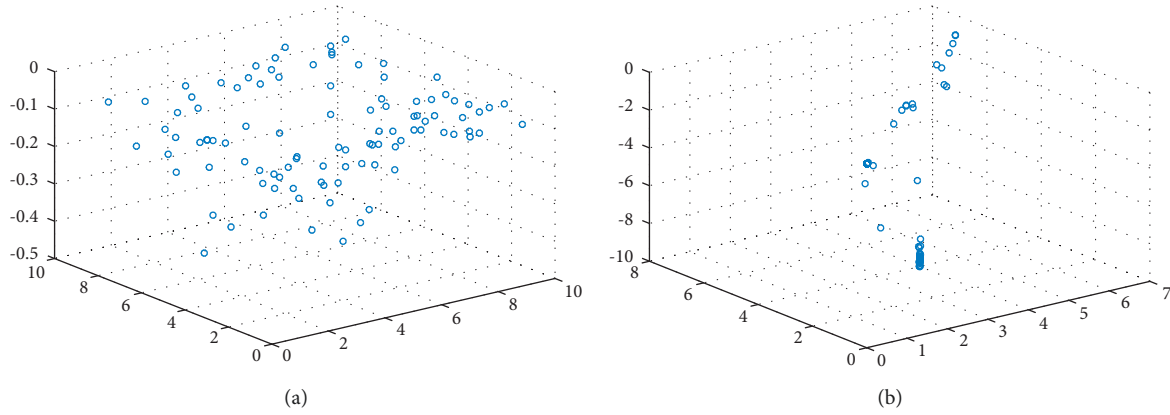


FIGURE 6: Individual distribution of ISSA. (a) ISSA individual initialization map. (b) Individual distribution of ISSA in 20 generations.

$$\lim_{t \rightarrow \infty} x_i(t) = x^{\text{best}}. \quad (20)$$

Extended to the j -th dimension:

$$\lim_{t \rightarrow \infty} x_{i,j}(t) = x_j^{\text{best}}. \quad (21)$$

Due to $a_j(t) = \min(x_j(t))$, $b_j(t) = \max(x_j(t))$, we get

$$\lim_{t \rightarrow \infty} a_j(t) = \lim_{t \rightarrow \infty} b_j(t) = x_j^{\text{best}}. \quad (22)$$

In the t -th generation, the current solution generated by the reverse learning strategy based on the lens imaging principle is

$$x_{i,j}^*(t) = \frac{a_j(t) + b_j(t)}{2} + \frac{a_j(t) + b_j(t)}{2k} - \frac{x_{i,j}(t)}{k}. \quad (23)$$

When $t \rightarrow \infty$, from the above formula, we get

$$\begin{aligned} \lim_{t \rightarrow \infty} x_{i,j}^*(t) &= \lim_{t \rightarrow \infty} \frac{a_j(t) + b_j(t)}{2} + \frac{a_j(t) + b_j(t)}{2k} - \frac{x_{i,j}(t)}{k} \\ &= \lim_{t \rightarrow \infty} \frac{x_j^{\text{best}}(t) + x_j^{\text{best}}(t)}{2} + \frac{x_j^{\text{best}}(t) + x_j^{\text{best}}(t)}{2k} \\ &\quad - \frac{x_j^{\text{best}}}{k} = x_j^{\text{best}}. \end{aligned} \quad (24)$$

Return to the whole dimension:

$$\lim_{t \rightarrow \infty} x_i^*(t) = x^{\text{best}}. \quad (25)$$

It can be seen that when $x_i(t)$ converges to x^{best} , the inverse solution $x_i^*(t)$ generated by lens learning also converges to x^{best} . Therefore, if the SSA based on general reverse learning converges, the ISSA also converges. \square

3.4. Improved Boundary Control. In the standard SSA, when an individual in the optimization process exceeds the boundary, the boundary control will be carried out for the individual. The principle is as follows:

$$x_{i,j}^{t+1} = \begin{cases} \text{Ub}, & x_{i,j}^{t+1} > \text{Ub}, \\ \text{Lb}, & x_{i,j}^{t+1} < \text{Lb}. \end{cases} \quad (26)$$

Ub and Lb are the upper and lower bounds of space, respectively. In this method, the strategy of turning the individuals beyond the boundary into the boundary will lead to the aggregation of individuals at the boundary and reduce the diversity of the population. It can be seen from Table 1 that although the improved strategy reduces the number of individuals beyond the boundary, some individuals still exceed the boundary and gather at the boundary. Literature [38] adopts the strategy of the current optimal solution for individuals beyond the boundary, which will be difficult to get rid of the local optimization. Therefore, this paper proposes a simple boundary treatment method, namely,

$$x_{i,j}^{t+1} = \text{Lb} + (\text{Ub} - \text{Lb}) \cdot \text{rand}(), \quad x_{i,j}^{t+1} > \text{Ub} \text{ and } x_{i,j}^{t+1} < \text{Lb}. \quad (27)$$

In this method, the individuals beyond the boundary are randomly assigned to the search space, which makes more effective use of the population individuals and increases the diversity of the population more than that of the original algorithm.

3.5. Improved Sparrow Search Algorithm Flow. In this paper, an improved sparrow search algorithm based on iterative local search is proposed. Firstly, the variable helix factor is used to improve the extensive search of followers, which reduces the individuals beyond the boundary and speeds up the convergence speed in the later stage. Secondly, the improved iterative local search is used to improve the local search of the followers. The initial solution is subject to local search and iterative local search after disturbance, which makes full use of the current position information to prevent premature convergence and improve the quality and accuracy of understanding. Then, the lens with changed focusing ability is used to carry out dimension by dimension lens imaging learning for the scouter, and it increases the search space and helps the population jump out of the local optimum. Finally, the boundary control strategy is improved

to reduce the aggregation of individuals at the boundary and increase the diversity of the population. The introduction of various strategies makes the algorithm more flexible in the optimization ability, makes the population more diverse, strengthens its ability to get rid of local optimization, balances the global search ability and local search ability of the algorithm, and is conducive to finding reliable solutions. The specific algorithm flow is shown in Figure 7. The specific pseudo code is as follows:

3.6. Time Complexity Analysis. Time complexity is an important index to measure the performance of the algorithm, which is used to measure the running time of the algorithm. Assuming that the population size of the algorithm is P , the maximum number of iterations is M and the dimension is D , and the time complexity of the sparrow search algorithm is $O(P \cdot M \cdot D)$. From a macro point of view, the improved sparrow search algorithm does not change the structure and cycle times of the algorithm, so its time complexity is also $O(P \cdot M \cdot D)$, which is consistent with the original algorithm. From the microscopic point of view, the greedy strategy is adopted for iterative local search and dimension by dimension lens learning, which increases the algorithm complexity of some followers and all scouts to a certain extent, but the introduction of the improved strategy does not improve the order of magnitude of the algorithm, so the time complexity is still $O(P \cdot M \cdot D)$.

4. Benchmark Function Test

In order to better verify the performance of ISSA, this paper selects 23 common basic test functions for verification and tests and compares them with 10 algorithms including PSO, SCA, GWO, WOA, MWOA, SSA, BSSA, CSSA, and LSSA. The specific parameter settings are shown in Table 3, and the test function information is shown in Table 4. F1–F7 is the high-dimensional single-peak benchmark function, F8–F13 is the high-dimensional multi-peak benchmark function, and F14–F23 is the low-dimensional multi-peak benchmark function. F1–F13 is tested in 100 dimensions to verify the performance of the algorithm in higher dimensions. For the sake of fairness, the population size and maximum number of iterations of each algorithm are 30 and 500, respectively, and each algorithm is run independently for 30 times to calculate its best value (best), worst value (worst), average value (AVE), and standard deviation (STD), and the optimal value of each index is processed in bold. Finally, each algorithm will be ranked according to the average value of the algorithm in the function. When the average values of the two algorithms are equal, the standard deviation will be compared. For performance evaluation, simulations are performed on Windows 10 Matlab 2016a, AMD Ryzen 7 4800U with Radeon Graphics @1.80 GHz with 16 GB RAM.

ISSA ranks first in most functions, and its average ranking is better than that of other algorithms. It can be seen from Tables 5 and 6 that, compared with other algorithms, ISSA has found the theoretical optimal value except F7, F10, F12, F13, and F15 and has found the optimal value in all

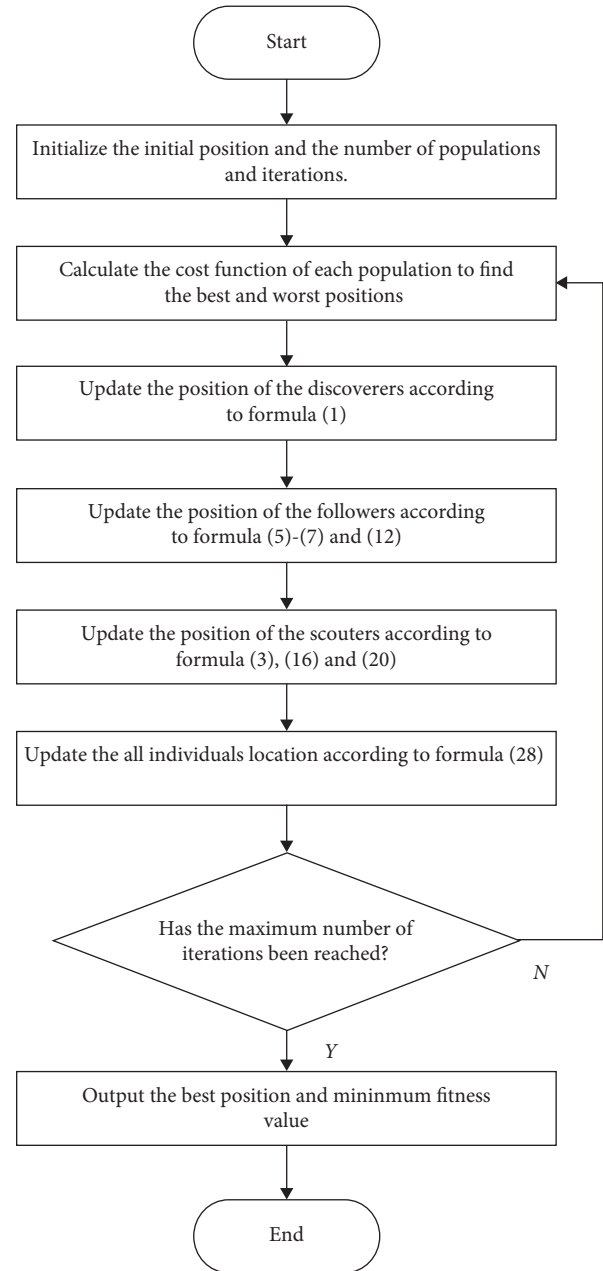


FIGURE 7: Algorithm flow chart.

algorithms in the above five functions. It can be seen that the ISSA has good ability to find the optimal solution. Among them, only the ISSA in F5 finds the theoretical optimal value, and only SSA and ISSA in F6 find the optimal value. The optimization accuracy of ISSA in F12 and F13 is a large number of orders of magnitude higher than that of other algorithms. When the dimension increases to 100, the optimization performance of ISSA is still very stable, while PSO, SCA, WOA, GOA, and SSA have a great impact on the optimization ability. It is worth mentioning that in F6 and F12, ISSA is only better than SSA and SSA improved the algorithm in terms of optimal value, but other indicators are better than other swarm intelligence algorithms. In F1–F13 function, ISSA finds a better solution than or the same

```

Input:
M: maximum number of iterations
PD: proportion of discoverer
SD: proportion of scouters
R2: alert value
N: population sparrows
Output: Xbest, fg
Initialize population
t = 1;
While (t < M)
  Find the position of the best and worst sparrow individuals according to fitness values.
  R2 = rand (1)
  For i = 1 : PD
    Update the location of the discoverers according to formula (1);
  End for
  For i = (PD + 1) : N
    Update the location of the followers according to formulas (5)–(7) and (12);
  End for
  For l = 1 : SD
    Get the individual position of a sparrow that is aware of danger according to formulas (3), (16), and (19);
  End for
  Get the location of the new optimal individual;
  Update the all individuals' location according to formula (27);
  t = t + 1
End while
Return: Xbest, fg

```

ALGORITHM 1: The framework of the ISSA.

TABLE 3: Parameter.

| Algorithm | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|-----------|--|-------|-------------|-------|--------------------|----------------------------------|----------------------------------|----------------------------------|----------------------------------|---|
| Parameter | c1 = 2 c2 = 2 Wmix = 0.2 Wmax = 0.9 | a = 2 | a = (2 → 0) | b = 1 | b = 1 n = 12000 | ST = 0.8 PD = 0.2 SD = 0.2 | ST = 0.8 PD = 0.2 SD = 0.2 | ST = 0.8 PD = 0.2 SD = 0.2 | ST = 0.8 PD = 0.2 SD = 0.2 | ST = 0.8 PD = 0.2 SD = 0.2 a = 0.1 |

solution as SSA, indicating that ISSA does not reduce the optimization ability of the algorithm. In the fixed dimension F14–F23, ISSA algorithm has good stability and the ability to jump out of the local optimum. It can find the solution close to the theoretical optimum almost every time.

In order to better describe the optimization ability, difference, and convergence speed of the algorithm with other algorithms, the Wilcoxon rank sum test results with other algorithms is given in Table 7 according to ISSA, and the convergence diagram of each algorithm is given in Figure 8. At the same time, in order to test the contribution of the four strategies to the algorithm, this paper selects a basic function to test the four components of the improved algorithm, as shown in the first figure in Figure 8. The above is based on the test results of 23 benchmark functions.

In this paper, the four strategies and the original SSA and ISSA are tested in the F1 function of the basic test function, as shown in the first figure in Figure 8. ISSA1, ISSA2, ISSA3, and ISSA4 represent the algorithms improved separately by the four improved strategies proposed before. Among them, ISSA2, ISSA3, and ISSA have found the theoretical optimal value. Compared with the original SSA, ISSA2 has

significantly improved the convergence speed, reflecting its excellent search ability; ISSA3 suddenly converges to the theoretical optimal value in the middle of the iteration, indicating its excellent ability to jump out of the local optimal. Compared with the original SSA algorithm, the optimization results and convergence speed of ISSA1 and ISSA4 are improved by dozens of orders of magnitude, which can improve to a certain extent. It can be seen that improved iterative local search and dimension by dimension lens imaging learning play a more critical role in improving the ability of the algorithm. The combination of the above four strategies enables us to obtain an ISSA with faster convergence speed, more accurate results, and more stability.

In the Wilcoxon rank sum test, when the value is less than 0.05, it can be considered that there is a significant difference between the two. In Table 7, NaN indicates that their performance is equivalent and cannot be compared. It can be seen that most values are less than 0.05, indicating that the optimization performance of ISSA is significantly different from other algorithms, among which the difference between ISSA and CSSA is the

TABLE 4: Test function.

| Function | Dimensions | Interval | Min |
|--|------------|-------------------|----------|
| $F_1(x) = \sum_{i=1}^n x_i^2$ | 30/100 | [-100, 100] | 0 |
| $F_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $ | 30/100 | [-100, 100] | 0 |
| $F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$ | 30/100 | [-100, 100] | 0 |
| $F_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$ | 30/100 | [-100, 100] | 0 |
| $F_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | 30/100 | [-30, 30] | 0 |
| $F_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$ | 30/100 | [-100, 100] | 0 |
| $F_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$ | 30/100 | [-1.28, 1.28] | 0 |
| $F_8(x) = \sum_{i=1}^n [-x_i \sin(\sqrt{ x_i })]$ | 30/100 | [-500, 500] | -418.98n |
| $F_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | 30/100 | [-5.12, 5.12] | 0 |
| $F_{10}(x) = -20 \exp(-0.2 \sqrt{(1/n) \sum_{i=1}^n x_i^2}) - \exp[(1/n) \sum_{i=1}^n \cos(2\pi x_i)] + 20 + e$ | 30/100 | [-32, 32] | 0 |
| $F_{11}(x) = (1/4000) \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$ | 30/100 | [-600, 600] | 0 |
| $F_{12}(x) = (\pi/n) \{10 \sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^n u(x_i, 10, 100, 4)$ | 30/100 | [-50, 50] | 0 |
| $y_i = 1 + ((x_i + 1)/4)$ | | | |
| $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$ | | | |
| $F_{13}(x) = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$ | 30/100 | [-50, 50] | 0 |
| $F_{14}(x) = ((1/500) + \sum_{j=1}^{25} 1/(j + \sum_{i=1}^{25} (x_i - a_{ij})^6))^{-1}$ | 2 | [-65.536, 65.536] | 0.998 |
| $F_{15}(x) = \sum_{i=1}^{11} (a_i - (x_1 (b_i^2 + b_1 x_2) / (b_i^2 + b_1 x_3 + x_4)))^2$ | 4 | [-5, 5] | 0.0003 |
| $F_{16}(x) = 4x_1^2 - 2.1x_1^4 + (1/3)x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$ | 2 | [-5, 5] | -1.032 |
| $F_{17}(x) = (x_2 - (5.1/4\pi^2)x_1 - 6)^2 + 10(1 - (1/8\pi))\cos x_1 + 10$ | 2 | [-5, 5] | 0.3979 |
| $F_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$ | 2 | [-2, 2] | 3 |
| $F_{19}(x) = \sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij} x_j - p_{ij})^2$ | 3 | [0, 1] | -3.863 |
| $F_{20}(x) = \sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2)$ | 6 | [0, 1] | -3.32 |
| $F_{21}(x) = \sum_{i=1}^5 [(X - a_i)(X - a_i)^7 + c_i]^{-1}$ | 4 | [0, 10] | -10.1532 |
| $F_{22}(x) = \sum_{i=1}^7 [(X - a_i)(X - a_i)^7 + c_i]^{-1}$ | 4 | [0, 10] | -10.4029 |
| $F_{23}(x) = \sum_{i=1}^{10} [(X - a_i)(X - a_i)^7 + c_i]^{-1}$ | 4 | [0, 10] | -10.5364 |

TABLE 5: Comparison table of the optimization effect of each algorithm (30 dimensions and fixed dimensions).

| F | Index | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|----|-------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| F1 | Best | 3.039E-08 | 2.140E-04 | 5.309E-40 | 3.217E-106 | 2.455E-106 | 0 | 5.308E-269 | 0 | 0 | 0 |
| | Worst | 2.030E-06 | 1.076E+01 | 9.250E-38 | 3.331E-94 | 1.392E-95 | 0 | 9.853E-249 | 0 | 0 | 0 |
| | Ave | 2.356E-07 | 8.143E-01 | 1.680E-38 | 1.167E-95 | 1.526E-96 | 0 | 3.337E-250 | 0 | 0 | 0 |
| | Std | 3.697E-07 | 2.128E+00 | 2.379E-38 | 6.075E-95 | 3.710E-96 | 0 | 0 | 0 | 0 | 0 |
| | Rank | 9 | 10 | 8 | 7 | 6 | 1 | 5 | 1 | 1 | 1 |
| F2 | Best | 5.760E-05 | 4.632E-05 | 3.354E-23 | 2.808E-63 | 1.441E-62 | 0 | 1.827E-136 | 0 | 2.123E-263 | 0 |
| | Worst | 2.620E-03 | 1.803E-02 | 5.443E-22 | 7.961E-57 | 3.663E-55 | 5.557E-231 | 1.955E-126 | 0 | 2.174E-169 | 0 |
| | Ave | 4.306E-04 | 4.924E-03 | 1.410E-22 | 1.014E-57 | 1.594E-56 | 2.250E-232 | 1.332E-127 | 0 | 7.253E-171 | 0 |
| | Std | 4.897E-04 | 4.613E-03 | 1.008E-22 | 2.073E-57 | 6.709E-56 | 0 | 4.462E-127 | 0 | 0 | 0 |
| | Rank | 9 | 10 | 8 | 6 | 7 | 3 | 5 | 1 | 4 | 1 |
| F3 | Best | 4.148E+01 | 7.288E+02 | 1.144E-07 | 1.056E+04 | 2.370E+04 | 0 | 6.239E-263 | 0 | 0 | 0 |
| | Worst | 1.465E+02 | 1.962E+04 | 3.338E-03 | 5.602E+04 | 6.525E+04 | 6.949E-59 | 3.651E-232 | 0 | 1.751E-244 | 0 |
| | Ave | 9.521E+01 | 7.423E+03 | 1.535E-04 | 3.870E+04 | 4.355E+04 | 2.316E-60 | 1.217E-233 | 0 | 7.297E-246 | 0 |
| | Std | 3.092E+01 | 4.667E+03 | 6.126E-04 | 1.181E+04 | 1.007E+04 | 1.269E-59 | 0 | 0 | 0 | 0 |
| | Rank | 7 | 8 | 6 | 9 | 10 | 5 | 4 | 1 | 3 | 1 |
| F4 | Best | 6.129E-01 | 1.282E+01 | 9.960E-08 | 3.426E+00 | 4.515E-02 | 2.102E-293 | 3.783E-130 | 0 | 1.942E-210 | 0 |
| | Worst | 1.677E+00 | 6.014E+01 | 5.466E-06 | 8.940E+01 | 8.833E+01 | 8.277E-36 | 6.088E-120 | 1.338E-182 | 1.789E-142 | 0 |
| | Ave | 1.077E+00 | 3.790E+01 | 1.135E-06 | 4.785E+01 | 4.205E+01 | 2.774E-37 | 4.300E-121 | 4.720E-184 | 6.026E-144 | 0 |
| | Std | 2.676E-01 | 1.281E+01 | 1.271E-06 | 2.867E+01 | 3.029E+01 | 1.511E-36 | 1.332E-120 | 0 | 3.265E-143 | 0 |
| | Rank | 7 | 8 | 6 | 10 | 9 | 5 | 4 | 2 | 3 | 1 |
| F5 | Best | 2.564E+01 | 4.001E+01 | 2.581E+01 | 2.707E+01 | 2.701E+01 | 2.124E-14 | 2.502E-08 | 1.176E-09 | 8.738E-10 | 0 |
| | Worst | 1.817E+02 | 2.485E+05 | 2.874E+01 | 2.877E+01 | 2.876E+01 | 3.498E-05 | 2.572E-04 | 5.266E-04 | 1.097E-03 | 0 |
| | Ave | 8.783E+01 | 2.463E+04 | 2.704E+01 | 2.794E+01 | 2.795E+01 | 4.175E-06 | 5.466E-05 | 2.864E-05 | 5.689E-05 | 0 |
| | Std | 4.634E+01 | 4.827E+04 | 6.734E-01 | 4.754E-01 | 4.112E-01 | 9.187E-06 | 7.765E-05 | 1.005E-04 | 2.064E-04 | 0 |
| | Rank | 9 | 10 | 6 | 7 | 8 | 2 | 4 | 3 | 5 | 1 |
| F6 | Best | 7.636E-06 | 5.178E+00 | 6.012E-07 | 4.835E-02 | 6.880E-02 | 0 | 3.524E-11 | 1.360E-12 | 3.385E-10 | 0 |
| | Worst | 1.739E-03 | 5.039E+01 | 1.255E+00 | 9.709E-01 | 1.047E+00 | 7.534E-06 | 1.103E-07 | 5.188E-08 | 3.010E-07 | 7.059E-04 |
| | Ave | 2.531E-04 | 1.420E+01 | 7.218E-01 | 4.287E-01 | 3.361E-01 | 3.019E-07 | 1.788E-08 | 6.178E-09 | 3.970E-08 | 1.409E-04 |
| | Std | 3.327E-04 | 1.305E+01 | 3.225E-01 | 2.176E-01 | 2.303E-01 | 1.372E-06 | 2.881E-08 | 1.170E-08 | 7.458E-08 | 1.965E-04 |
| | Rank | 6 | 10 | 9 | 8 | 7 | 4 | 2 | 1 | 3 | 5 |
| F7 | Best | 7.982E-02 | 1.117E-02 | 3.244E-04 | 5.252E-05 | 5.025E-05 | 1.287E-05 | 2.264E-05 | 1.056E-05 | 1.083E-05 | 8.231E-06 |
| | Worst | 3.788E-01 | 2.485E-01 | 4.318E-03 | 1.108E-02 | 1.460E-02 | 1.062E-03 | 9.727E-04 | 4.982E-04 | 8.427E-04 | 5.115E-04 |
| | Ave | 1.778E-01 | 8.566E-02 | 1.795E-03 | 2.821E-03 | 3.241E-03 | 2.440E-04 | 2.722E-04 | 1.632E-04 | 2.355E-04 | 1.506E-04 |
| | Std | 6.799E-02 | 6.210E-02 | 1.129E-03 | 2.905E-03 | 3.662E-03 | 2.364E-04 | 2.438E-04 | 1.211E-04 | 1.908E-04 | 1.196E-04 |
| | Rank | 10 | 9 | 6 | 7 | 8 | 4 | 5 | 2 | 3 | 1 |
| F8 | Best | -8.128E+03 | -4.963E+03 | -7.948E+03 | -1.257E+04 | -1.257E+04 | -1.257E+04 | -1.030E+04 | -1.257E+04 | -1.257E+04 | -1.257E+04 |
| | Worst | -3.967E+03 | -3.578E+03 | -3.642E+03 | -8.304E+03 | -7.837E+03 | -7.673E+03 | -7.671E+03 | -8.952E+03 | -1.033E+04 | -1.083E+04 |
| | Ave | -6.667E+03 | -4.023E+03 | -6.361E+03 | -1.153E+04 | -1.142E+04 | -1.060E+04 | -8.891E+03 | -1.128E+04 | -1.182E+04 | -1.245E+04 |
| | Std | 9.060E+02 | 3.243E+02 | 7.713E+02 | 1.520E+03 | 1.230E+03 | 1.948E+03 | 5.289E+02 | 8.299E+02 | 6.529E+02 | 3.565E+02 |
| | Rank | 8 | 10 | 9 | 3 | 4 | 6 | 7 | 5 | 2 | 1 |

TABLE 5: Continued.

| F | Index | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|-----|-------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| F9 | Best | 2.609E+01 | 8.825E-01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Worst | 8.661E+01 | 2.056E+02 | 1.045E+01 | 0 | 1.137E-13 | 0 | 0 | 0 | 0 | 0 |
| | Ave | 5.757E+01 | 3.868E+01 | 1.057E+00 | 0 | 3.790E-15 | 0 | 0 | 0 | 0 | 0 |
| | Std | 1.410E+01 | 4.056E+01 | 2.499E+00 | 0 | 2.076E-14 | 0 | 0 | 0 | 0 | 0 |
| | Rank | 10 | 9 | 8 | 1 | 7 | 1 | 1 | 1 | 1 | 1 |
| F10 | Best | 2.946E-03 | 2.093E-02 | 7.905E-14 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 |
| | Worst | 1.341E+00 | 2.035E+01 | 2.780E-13 | 7.994E-15 | 7.994E-15 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 |
| | Ave | 2.508E-01 | 1.104E+01 | 1.554E-13 | 3.849E-15 | 5.033E-15 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 |
| | Std | 4.770E-01 | 9.347E+00 | 4.636E-14 | 2.483E-15 | 2.653E-15 | 0 | 0 | 0 | 0 | 0 |
| | Rank | 9 | 10 | 8 | 6 | 7 | 1 | 1 | 1 | 1 | 1 |
| F11 | Best | 1.262E-06 | 4.291E-01 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Worst | 3.694E-02 | 2.192E+00 | 2.243E-02 | 2.284E-01 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Ave | 9.120E-03 | 1.013E+00 | 4.588E-03 | 7.613E-03 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Std | 1.036E-02 | 2.950E-01 | 7.488E-03 | 4.170E-02 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Rank | 9 | 10 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 |
| F12 | Best | 4.722E-08 | 1.373E+00 | 6.907E-03 | 4.668E-03 | 2.905E-03 | 5.172E-19 | 3.481E-13 | 4.456E-13 | 2.676E-09 | 1.571E-32 |
| | Worst | 1.037E-01 | 3.061E+06 | 1.258E-01 | 3.701E-01 | 2.246E-01 | 1.549E-07 | 8.831E-09 | 3.088E-09 | 6.638E-07 | 9.777E-05 |
| | Ave | 6.914E-03 | 1.092E+05 | 4.088E-02 | 3.888E-02 | 2.754E-02 | 7.364E-09 | 8.349E-10 | 2.263E-10 | 1.373E-07 | 2.558E-05 |
| | Std | 2.630E-02 | 5.587E+05 | 2.534E-02 | 6.678E-02 | 3.888E-02 | 2.905E-08 | 1.843E-09 | 5.697E-10 | 1.673E-07 | 3.011E-05 |
| | Rank | 6 | 10 | 9 | 8 | 7 | 3 | 2 | 1 | 4 | 5 |
| F13 | Best | 4.356E-06 | 5.547E+00 | 1.045E-01 | 2.076E-01 | 1.165E-01 | 1.010E-21 | 1.163E-11 | 2.611E-16 | 1.351E-15 | 1.350E-32 |
| | Worst | 1.152E-02 | 3.410E+06 | 9.183E-01 | 1.069E+00 | 1.303E+00 | 2.333E-07 | 2.900E-07 | 1.083E-10 | 3.592E-11 | 1.350E-32 |
| | Ave | 5.321E-03 | 2.239E+05 | 5.196E-01 | 6.395E-01 | 5.154E-01 | 2.072E-08 | 2.126E-08 | 1.612E-11 | 3.754E-12 | 1.350E-32 |
| | Std | 5.498E-03 | 7.069E+05 | 1.948E-01 | 2.354E-01 | 2.762E-01 | 5.998E-08 | 6.089E-08 | 2.840E-11 | 8.284E-12 | 0 |
| | Rank | 6 | 10 | 8 | 9 | 7 | 4 | 5 | 3 | 2 | 1 |
| F14 | Best | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 | 9.980E-01 |
| | Worst | 1.172E+01 | 1.076E+01 | 1.267E+01 | 1.076E+01 | 1.076E+01 | 1.267E+01 | 1.267E+01 | 9.980E-01 | 1.267E+01 | 1.267E+01 |
| | Ave | 4.147E+00 | 2.251E+00 | 4.065E+00 | 2.767E+00 | 3.907E+00 | 8.622E+00 | 1.086E+01 | 9.980E-01 | 1.585E+00 | 1.453E+00 |
| | Std | 3.028E+00 | 1.888E+00 | 4.163E+00 | 2.953E+00 | 3.791E+00 | 5.438E+00 | 4.143E+00 | 4.517E-16 | 2.179E+00 | 2.149E+00 |
| | Rank | 8 | 4 | 7 | 5 | 6 | 9 | 10 | 1 | 3 | 2 |
| F15 | Best | 3.084E-04 | 4.873E-04 | 3.075E-04 | 3.081E-04 | 3.217E-04 | 3.075E-04 | 3.075E-04 | 3.075E-04 | 3.075E-04 | 3.075E-04 |
| | Worst | 1.093E-03 | 1.506E-03 | 2.036E-02 | 5.941E-03 | 2.252E-03 | 3.447E-04 | 6.256E-04 | 3.205E-04 | 3.155E-04 | 3.319E-04 |
| | Ave | 8.461E-04 | 1.002E-03 | 4.462E-03 | 1.216E-03 | 6.588E-04 | 3.098E-04 | 3.204E-04 | 3.079E-04 | 3.080E-04 | 3.100E-04 |
| | Std | 1.588E-04 | 3.287E-04 | 8.091E-03 | 1.271E-03 | 4.499E-04 | 7.828E-06 | 5.824E-05 | 2.380E-06 | 1.552E-06 | 6.408E-06 |
| | Rank | 7 | 8 | 10 | 9 | 6 | 3 | 5 | 1 | 2 | 4 |
| F16 | Best | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 |
| | Worst | -1.032E+00 | -1.031E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -2.155E-01 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 |
| | Ave | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.004E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 | -1.032E+00 |
| | Std | 0 | 7.279E-05 | 0 | 0 | 0 | 1.490E-01 | 0 | 0 | 0 | 0 |
| | Rank | 1 | 9 | 1 | 1 | 1 | 10 | 1 | 1 | 1 | 1 |

TABLE 5: Continued.

| F | Index | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|--------------|--------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| F17 | Best | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 |
| | Worst | 3.979E-01 | 4.108E-01 | 3.985E-01 | 3.980E-01 | 3.980E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 |
| | Ave | 3.979E-01 | 3.994E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 | 3.979E-01 |
| | Std | 0 | 2.363E-03 | 1.095E-04 | 1.569E-05 | 2.801E-05 | 0 | 0 | 0 | 0 | 0 |
| F18 | Rank | 1 | 10 | 9 | 7 | 8 | 1 | 1 | 1 | 1 | 1 |
| | Best | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 |
| | Worst | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+01 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 |
| | Ave | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 4.800E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 | 3.000E+00 |
| F19 | Std | 0 | 0 | 0 | 0 | 0 | 6.850E+00 | 0 | 0 | 0 | 0 |
| | Rank | 1 | 1 | 1 | 1 | 1 | 10 | 1 | 1 | 1 | 1 |
| | Best | -3.863E+00 | -3.862E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 |
| | Worst | -3.863E+00 | -3.854E+00 | -3.856E+00 | -3.861E+00 | -3.860E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 |
| F20 | Ave | -3.863E+00 | -3.855E+00 | -3.862E+00 | -3.862E+00 | -3.862E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 | -3.863E+00 |
| | Std | 3.162E-15 | 1.785E-03 | 1.528E-03 | 6.637E-04 | 6.423E-04 | 0 | 0 | 0 | 0 | 0 |
| | Rank | 6 | 10 | 7 | 8 | 9 | 1 | 1 | 1 | 1 | 1 |
| | Best | -3.322E+00 | -3.224E+00 | -3.322E+00 | -3.322E+00 | -3.322E+00 | -3.322E+00 | -3.322E+00 | -3.322E+00 | -3.322E+00 | -3.322E+00 |
| F21 | Worst | -3.203E+00 | -2.991E+00 | -3.138E+00 | -3.082E+00 | -3.087E+00 | -3.203E+00 | -3.203E+00 | -3.203E+00 | -3.203E+00 | -3.203E+00 |
| | Ave | -3.270E+00 | -3.078E+00 | -3.247E+00 | -3.258E+00 | -3.243E+00 | -3.263E+00 | -3.286E+00 | -3.263E+00 | -3.247E+00 | -3.318E+00 |
| | Std | 5.993E-02 | 6.050E-02 | 6.344E-02 | 7.915E-02 | 6.898E-02 | 6.047E-02 | 5.542E-02 | 6.047E-02 | 5.828E-02 | 2.171E-02 |
| | Rank | 3 | 10 | 7 | 6 | 9 | 4 | 2 | 5 | 8 | 1 |
| F22 | Best | -1.015E+01 | -5.819E+00 | -1.015E+01 | -1.015E+01 | -1.015E+01 | -1.015E+01 | -1.015E+01 | -1.015E+01 | -1.015E+01 | -1.015E+01 |
| | Worst | -2.631E+00 | -4.973E-01 | -2.683E+00 | -4.984E+00 | -2.615E+00 | -5.055E+00 | -9.996E+00 | -1.015E+01 | -1.015E+01 | -1.015E+01 |
| | Ave | -7.630E+00 | -2.164E+00 | -9.088E+00 | -8.265E+00 | -8.515E+00 | -9.303E+00 | -1.013E+01 | -1.015E+01 | -1.015E+01 | -1.015E+01 |
| | Std | 3.218E+00 | 1.734E+00 | 2.466E+00 | 2.490E+00 | 2.753E+00 | 1.932E+00 | 3.978E-02 | 0 | 0 | 0 |
| F23 | Rank | 9 | 10 | 6 | 8 | 7 | 5 | 4 | 1 | 1 | 1 |
| | Best | -6.207E+00 | -1.040E+01 | -1.040E+01 | -1.040E+01 | -1.040E+01 | -1.040E+01 | -1.040E+01 | -1.040E+01 | -1.040E+01 | -1.040E+01 |
| | Worst | -9.028E-01 | -5.129E+00 | -1.838E+00 | -1.835E+00 | -5.088E+00 | -5.088E+00 | -1.040E+01 | -1.040E+01 | -5.088E+00 | -1.040E+01 |
| | Ave | -2.986E+00 | -1.023E+01 | -7.664E+00 | -7.782E+00 | -8.985E+00 | -9.850E+00 | -1.040E+01 | -1.040E+01 | -9.340E+00 | -1.040E+01 |
| Average rank | Std | 1.774E+00 | 9.629E-01 | 3.007E+00 | 3.125E+00 | 2.391E+00 | 1.615E+00 | 0 | 0 | 2.162E+00 | 0 |
| | Rank | 10 | 4 | 9 | 8 | 7 | 5 | 1 | 1 | 6 | 1 |
| | Best | -1.054E+01 | -8.411E+00 | -1.054E+01 | -1.054E+01 | -1.054E+01 | -1.054E+01 | -1.054E+01 | -1.054E+01 | -1.054E+01 | -1.054E+01 |
| | Worst | -2.422E+00 | -9.439E-01 | -5.129E+00 | -2.422E+00 | -2.803E+00 | -5.129E+00 | -1.026E+01 | -1.054E+01 | -1.054E+01 | -1.054E+01 |
| Average rank | Ave | -9.395E+00 | -3.822E+00 | -1.018E+01 | -7.221E+00 | -8.252E+00 | -9.815E+00 | -1.051E+01 | -1.054E+01 | -1.054E+01 | -1.054E+01 |
| | Std | 2.644E+00 | 1.569E+00 | 1.366E+00 | 3.235E+00 | 2.843E+00 | 1.870E+00 | 7.007E-02 | 0 | 0 | 0 |
| | Rank | 7 | 10 | 5 | 9 | 8 | 6 | 4 | 1 | 1 | 1 |
| | Average rank | 4.60 | 5.93 | 4.45 | 4.38 | 2.93 | 2.14 | 1.17 | 1.72 | 1.10 | 1.10 |

TABLE 6: Comparison table of the optimization effect of each algorithm (100 dimensions).

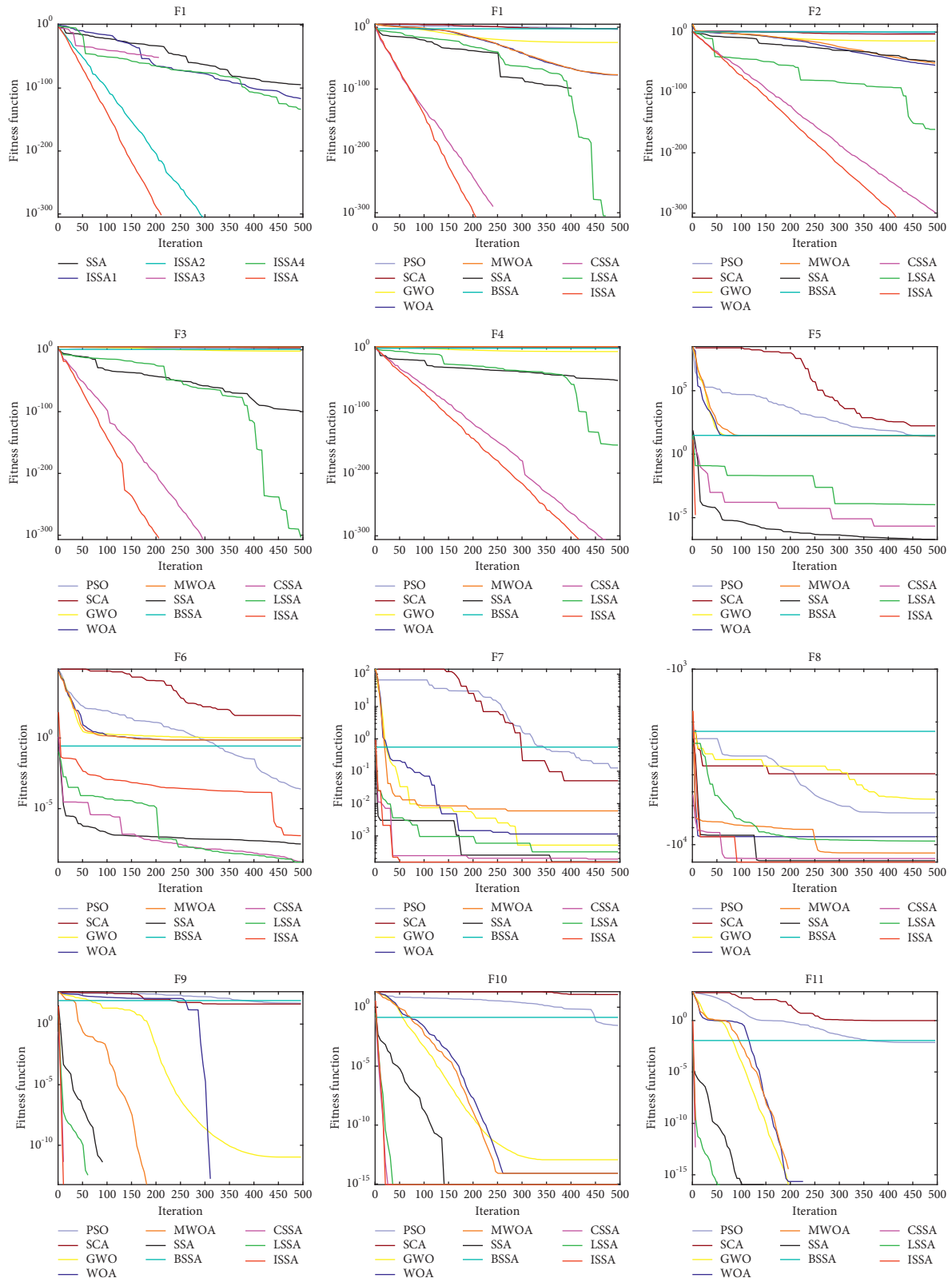
| F | Index | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|----|-------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| F1 | Best | 9.872E+00 | 9.308E+02 | 2.693E-12 | 4.175E-83 | 2.894E-84 | 0 | 2.714E-255 | 0 | 0 | 0 |
| | Worst | 3.569E+01 | 2.981E+04 | 1.824E-10 | 8.142E-70 | 4.168E-70 | 2.928E-66 | 9.366E-235 | 0 | 1.063E-272 | 0 |
| | Ave | 2.206E+01 | 1.131E+04 | 4.086E-11 | 4.068E-71 | 1.464E-71 | 1.010E-67 | 3.248E-236 | 0 | 3.935E-274 | 0 |
| | Std | 5.407E+00 | 7.239E+03 | 4.038E-11 | 1.574E-70 | 7.601E-71 | 5.438E-67 | 0 | 0 | 0 | 0 |
| | Rank | 9 | 10 | 8 | 6 | 5 | 7 | 4 | 1 | 1 | 3 |
| F2 | Best | 2.295E+01 | 1.120E-01 | 1.379E-07 | 4.706E-55 | 4.940E-57 | 0 | 1.226E-127 | 0 | 9.989E-215 | 0 |
| | Worst | 5.920E+01 | 2.205E+01 | 5.637E-07 | 3.020E-47 | 1.360E-48 | 8.218E-35 | 1.609E-115 | 0 | 4.528E-140 | 0 |
| | Ave | 3.901E+01 | 7.362E+00 | 2.368E-07 | 1.053E-48 | 1.017E-49 | 3.229E-36 | 5.982E-117 | 0 | 1.509E-141 | 0 |
| | Std | 9.152E+00 | 6.263E+00 | 8.618E-08 | 5.508E-48 | 3.266E-49 | 1.515E-35 | 2.945E-116 | 0 | 8.268E-141 | 0 |
| | Rank | 10 | 9 | 8 | 6 | 5 | 7 | 4 | 1 | 3 | 1 |
| F3 | Best | 1.196E+04 | 1.540E+05 | 2.370E+01 | 4.152E+05 | 5.261E+05 | 0 | 9.631E-256 | 0 | 0 | 0 |
| | Worst | 2.610E+04 | 4.486E+05 | 3.307E+03 | 1.682E+06 | 1.491E+06 | 5.484E-51 | 1.912E-231 | 0 | 0 | 0 |
| | Ave | 1.660E+04 | 2.370E+05 | 1.037E+03 | 1.007E+06 | 1.040E+06 | 1.906E-52 | 6.898E-233 | 0 | 0 | 0 |
| | Std | 3.793E+03 | 6.543E+04 | 8.578E+02 | 3.146E+05 | 2.624E+05 | 1.018E-51 | 0 | 0 | 0 | 0 |
| | Rank | 7 | 8 | 6 | 9 | 10 | 5 | 4 | 1 | 1 | 3 |
| F4 | Best | 9.191E+00 | 8.607E+01 | 7.311E-01 | 2.040E+01 | 1.568E+00 | 2.570E-282 | 1.681E-130 | 0 | 3.573E-261 | 0 |
| | Worst | 1.493E+01 | 9.584E+01 | 8.765E+00 | 9.697E+01 | 9.638E+01 | 7.474E-32 | 4.501E-118 | 0 | 4.421E-158 | 0 |
| | Ave | 1.220E+01 | 9.061E+01 | 2.931E+00 | 7.732E+01 | 7.614E+01 | 2.491E-33 | 2.897E-119 | 0 | 1.474E-159 | 0 |
| | Std | 1.550E+00 | 2.355E+00 | 1.931E+00 | 2.188E+01 | 2.359E+01 | 1.365E-32 | 9.208E-119 | 0 | 0 | 0 |
| | Rank | 7 | 10 | 6 | 9 | 8 | 5 | 4 | 1 | 3 | 1 |
| F5 | Best | 7.371E+03 | 2.331E+07 | 9.657E+01 | 9.745E+01 | 9.752E+01 | 1.531E-14 | 1.382E-07 | 2.260E-08 | 8.400E-10 | 0 |
| | Worst | 3.114E+04 | 2.554E+08 | 9.849E+01 | 9.858E+01 | 9.846E+01 | 3.712E-03 | 2.409E-03 | 3.776E-04 | 1.007E-03 | 0 |
| | Ave | 1.571E+04 | 1.182E+08 | 9.778E+01 | 9.819E+01 | 9.814E+01 | 1.501E-04 | 3.859E-04 | 8.782E-05 | 2.173E-04 | 0 |
| | Std | 5.206E+03 | 6.138E+07 | 5.923E-01 | 2.454E-01 | 2.609E-01 | 6.750E-04 | 6.669E-04 | 1.081E-04 | 2.744E-04 | 0 |
| | Rank | 9 | 10 | 6 | 8 | 7 | 3 | 5 | 2 | 4 | 1 |
| F6 | Best | 1.156E+01 | 2.016E+03 | 7.714E+00 | 2.371E+00 | 1.979E+00 | 0 | 7.437E-10 | 1.845E-09 | 8.530E-10 | 0 |
| | Worst | 3.269E+01 | 3.267E+04 | 1.146E+01 | 8.063E+00 | 6.289E+00 | 2.874E-06 | 1.995E-05 | 4.954E-06 | 1.287E-04 | 7.775E-03 |
| | Ave | 2.134E+01 | 1.089E+04 | 9.632E+00 | 4.301E+00 | 3.952E+00 | 1.845E-07 | 1.796E-06 | 6.391E-07 | 7.234E-06 | 1.427E-03 |
| | Std | 5.340E+00 | 6.941E+03 | 1.016E+00 | 1.423E+00 | 1.118E+00 | 5.339E-07 | 3.757E-06 | 1.034E-06 | 2.359E-05 | 2.389E-03 |
| | Rank | 9 | 10 | 8 | 7 | 6 | 1 | 3 | 2 | 4 | 5 |
| F7 | Best | 1.198E+03 | 4.974E+01 | 2.106E-03 | 2.143E-04 | 1.476E-04 | 2.729E-05 | 9.448E-06 | 1.484E-05 | 9.049E-06 | 2.371E-06 |
| | Worst | 2.001E+03 | 4.241E+02 | 1.629E-02 | 1.567E-02 | 2.330E-02 | 1.299E-03 | 1.096E-03 | 3.918E-04 | 1.621E-03 | 3.207E-04 |
| | Ave | 1.518E+03 | 1.609E+02 | 7.634E-03 | 4.709E-03 | 5.183E-03 | 3.217E-04 | 3.569E-04 | 1.502E-04 | 4.565E-04 | 1.394E-04 |
| | Std | 1.889E+02 | 9.512E+01 | 3.085E-03 | 5.310E-03 | 5.977E-03 | 2.954E-04 | 2.842E-04 | 1.087E-04 | 4.189E-04 | 9.063E-05 |
| | Rank | 10 | 9 | 8 | 6 | 7 | 3 | 4 | 2 | 5 | 1 |
| F8 | Best | -1.781E+04 | -7.934E+03 | -1.999E+04 | -4.187E+04 | -4.190E+04 | -4.190E+04 | -2.698E+04 | -4.190E+04 | -4.190E+04 | -4.190E+04 |
| | Worst | -5.016E+03 | -6.076E+03 | -5.790E+03 | -2.378E+04 | -2.746E+04 | -2.346E+04 | -1.761E+04 | -2.955E+04 | -2.184E+04 | -3.456E+04 |
| | Ave | -1.050E+04 | -6.759E+03 | -1.623E+04 | -3.407E+04 | -3.358E+04 | -3.150E+04 | -2.276E+04 | -3.604E+04 | -3.379E+04 | -4.050E+04 |
| | Std | 3.860E+03 | 4.827E+02 | 2.907E+03 | 5.642E+03 | 5.637E+03 | 6.075E+03 | 2.130E+03 | 4.310E+03 | 7.876E+03 | 2.173E+03 |
| | Rank | 9 | 10 | 8 | 3 | 5 | 6 | 7 | 2 | 4 | 1 |

TABLE 6: Continued.

| F | Index | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|--------------|-------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| F9 | Best | 4.779E+02 | 5.265E+01 | 2.724E-09 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Worst | 7.998E+02 | 4.466E+02 | 4.533E+01 | 0 | 1.137E-13 | 0 | 0 | 0 | 0 | 0 |
| | Ave | 6.303E+02 | 2.595E+02 | 3.979E+00 | 0 | 3.790E-15 | 0 | 0 | 0 | 0 | 0 |
| | Std | 8.286E+01 | 1.086E+02 | 8.599E+00 | 0 | 2.076E-14 | 0 | 0 | 0 | 0 | 0 |
| | Rank | 10 | 9 | 8 | 1 | 7 | 1 | 1 | 1 | 1 | 1 |
| F10 | Best | 3.132E+00 | 9.221E+00 | 3.430E-07 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 |
| | Worst | 4.305E+00 | 2.067E+01 | 1.375E-06 | 7.994E-15 | 7.994E-15 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 | 8.882E-16 |
| | Ave | 3.726E+00 | 1.899E+01 | 6.452E-07 | 4.322E-15 | 4.204E-15 | 0 | 0 | 0 | 0 | 0 |
| | Std | 2.946E-01 | 3.752E+00 | 2.326E-07 | 2.873E-15 | 2.273E-15 | 4.012E-31 | 4.012E-31 | 4.012E-31 | 4.012E-31 | 4.012E-31 |
| | Rank | 9 | 10 | 8 | 7 | 6 | 1 | 1 | 1 | 1 | 1 |
| F11 | Best | 2.761E-01 | 9.224E+00 | 4.273E-12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Worst | 6.430E-01 | 1.889E+02 | 2.807E-02 | 3.137E-01 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Ave | 4.252E-01 | 8.541E+01 | 2.511E-03 | 1.046E-02 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Std | 9.838E-02 | 5.334E+01 | 7.694E-03 | 5.728E-02 | 0 | 0 | 0 | 0 | 0 | 0 |
| | Rank | 9 | 10 | 7 | 8 | 1 | 1 | 1 | 1 | 1 | 1 |
| F12 | Best | 9.845E-01 | 7.795E+07 | 1.679E-01 | 1.867E-02 | 1.876E-02 | 2.330E-20 | 1.123E-12 | 2.531E-12 | 3.574E-13 | 4.712E-33 |
| | Worst | 9.589E+00 | 6.179E+08 | 3.925E-01 | 9.377E-02 | 1.440E-01 | 8.878E-09 | 6.918E-08 | 1.562E-07 | 1.101E-08 | 9.480E-05 |
| | Ave | 4.297E+00 | 2.889E+08 | 2.624E-01 | 4.903E-02 | 4.792E-02 | 9.451E-10 | 1.021E-08 | 1.261E-08 | 2.028E-09 | 2.722E-05 |
| | Std | 1.884E+00 | 1.390E+08 | 6.522E-02 | 1.946E-02 | 2.491E-02 | 1.909E-09 | 1.627E-08 | 2.920E-08 | 2.960E-09 | 3.211E-05 |
| | Rank | 9 | 10 | 8 | 7 | 6 | 1 | 3 | 4 | 2 | 5 |
| F13 | Best | 2.737E+01 | 2.004E+08 | 5.195E+00 | 1.224E+00 | 1.242E+00 | 6.058E-14 | 7.614E-10 | 6.218E-10 | 1.775E-10 | 1.350E-32 |
| | Worst | 9.778E+01 | 1.593E+09 | 6.920E+00 | 5.151E+00 | 5.166E+00 | 4.220E-07 | 3.989E-06 | 1.629E-06 | 6.166E-06 | 1.350E-32 |
| | Ave | 5.972E+01 | 6.758E+08 | 6.397E+00 | 2.787E+00 | 3.138E+00 | 4.654E-08 | 4.157E-07 | 2.250E-07 | 5.104E-07 | 1.350E-32 |
| | Std | 1.998E+01 | 3.405E+08 | 4.285E-01 | 9.742E-01 | 8.932E-01 | 9.147E-08 | 8.210E-07 | 4.018E-07 | 1.269E-06 | 0 |
| | Rank | 9 | 10 | 8 | 6 | 7 | 2 | 4 | 3 | 5 | 1 |
| Average rank | | 8.92 | 9.62 | 7.46 | 6.38 | 6.15 | 3.31 | 3.46 | 1.70 | 2.85 | 1.61 |

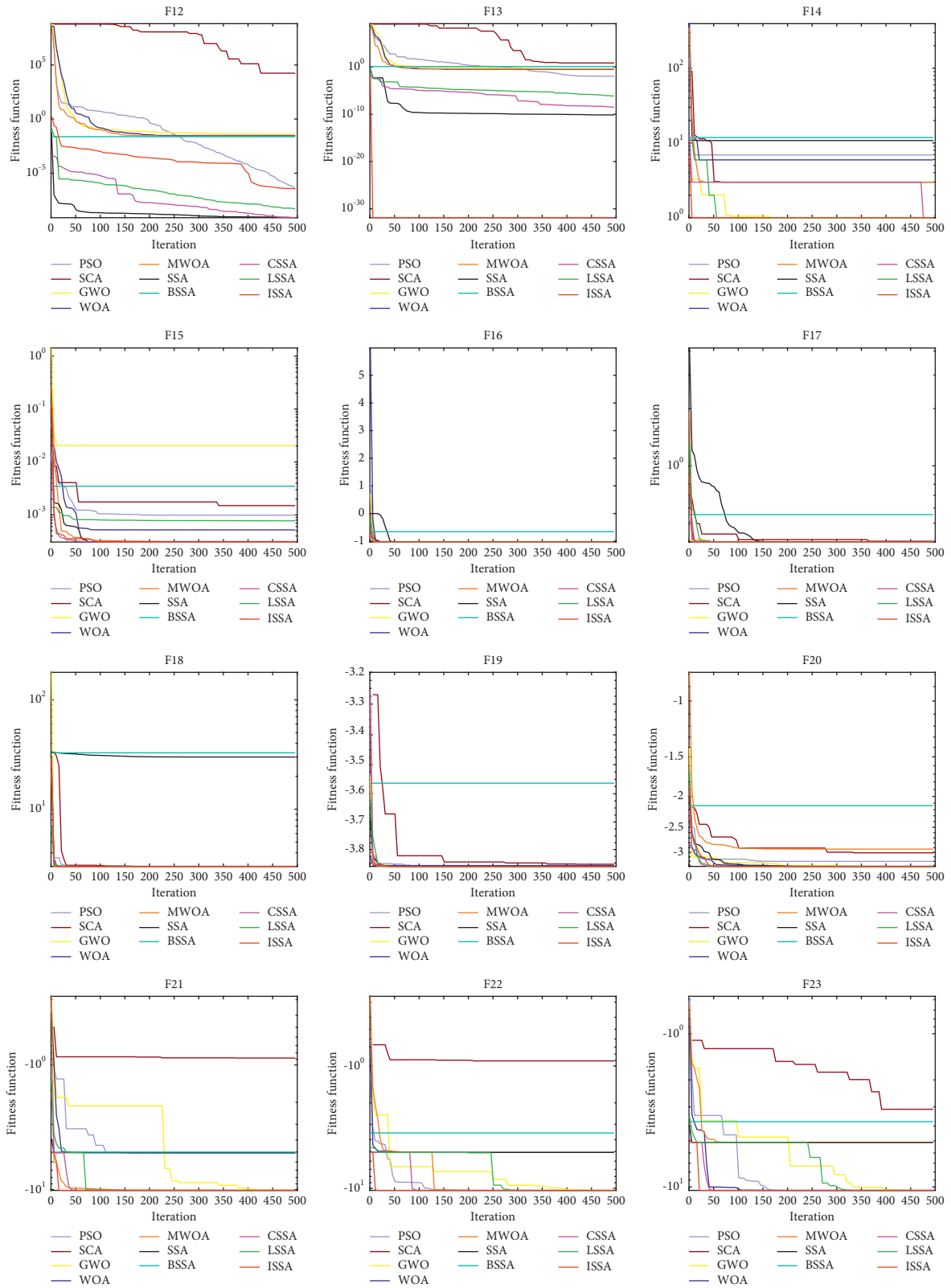
TABLE 7: Wilcoxon rank sum test results of each algorithm (30 dimensions).

| F | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA |
|-----|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| F1 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | NaN | 1.212E-12 | NaN | NaN |
| F2 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 4.057E-03 | 1.212E-12 | NaN | 1.212E-12 |
| F3 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 5.772E-11 | 1.212E-12 | NaN | 4.714E-05 |
| F4 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 4.574E-12 | 1.212E-12 |
| F5 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.211E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 |
| F6 | 1.167E-02 | 2.954E-11 | 2.814E-10 | 2.954E-11 | 2.954E-11 | 1.957E-04 | 3.962E-04 | 3.540E-04 | 3.962E-04 |
| F7 | 3.020E-11 | 3.020E-11 | 4.504E-11 | 9.063E-08 | 4.616E-10 | 1.023E-01 | 9.626E-02 | 7.062E-01 | 7.483E-02 |
| F8 | 2.392E-11 | 2.392E-11 | 2.392E-11 | 2.831E-07 | 3.102E-08 | 1.783E-06 | 2.392E-11 | 3.327E-09 | 4.347E-08 |
| F9 | 1.212E-12 | 1.212E-12 | 3.818E-12 | NaN | 3.337E-01 | NaN | NaN | NaN | NaN |
| F10 | 1.211E-12 | 1.212E-12 | 1.199E-12 | 9.318E-08 | 1.317E-09 | NaN | NaN | NaN | NaN |
| F11 | 1.212E-12 | 1.212E-12 | 6.617E-04 | 3.337E-01 | NaN | NaN | NaN | NaN | NaN |
| F12 | 1.259E-01 | 2.982E-11 | 2.982E-11 | 2.982E-11 | 2.982E-11 | 9.460E-06 | 9.460E-06 | 9.460E-06 | 2.420E-05 |
| F13 | 1.211E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 | 1.212E-12 |
| F14 | 4.126E-07 | 3.800E-09 | 1.642E-05 | 2.328E-04 | 3.743E-05 | 5.429E-08 | 1.058E-10 | 1.608E-01 | 4.181E-01 |
| F15 | 5.992E-11 | 2.982E-11 | 8.620E-09 | 1.081E-10 | 4.027E-11 | 9.080E-03 | 1.598E-01 | 3.241E-06 | 3.455E-03 |
| F16 | 1.685E-14 | 3.620E-13 | 1.685E-14 | 1.685E-14 | 1.685E-14 | 2.708E-14 | 1.685E-14 | 1.685E-14 | 1.685E-14 |
| F17 | NaN | 4.566E-12 | 3.337E-01 | 6.519E-04 | 6.500E-05 | NaN | NaN | NaN | NaN |
| F18 | NaN | NaN | NaN | NaN | NaN | 1.607E-01 | NaN | NaN | NaN |
| F19 | NaN | 1.129E-12 | 2.158E-02 | 1.828E-09 | 5.600E-10 | NaN | NaN | NaN | NaN |
| F20 | 6.138E-01 | 6.738E-11 | 2.457E-02 | 1.021E-01 | 4.245E-04 | 1.000E+00 | 1.189E-01 | 5.271E-05 | 3.055E-01 |
| F21 | 1.173E-05 | 1.212E-12 | 1.101E-02 | 1.878E-09 | 5.404E-11 | NaN | NaN | NaN | NaN |
| F22 | 4.721E-03 | 1.720E-12 | 1.000E+00 | 1.704E-12 | 1.715E-12 | 2.416E-04 | 1.039E-10 | 3.337E-01 | 3.337E-01 |
| F23 | 2.157E-02 | 1.212E-12 | 1.608E-01 | 1.208E-12 | 1.208E-12 | 2.773E-03 | 5.808E-09 | NaN | NaN |



(a)

FIGURE 8: Continued.



(b)

FIGURE 8: Convergence diagram of each algorithm.

smallest, followed by LSSA, BSSA, and SSA. As shown in Figure 8, the ISSA shows excellent optimization speed and convergence accuracy, and the convergence speed is fastest in most functions. The ISSA converges faster in the single-peak benchmark function and has better ability to resist the attraction local optimum in the multipeak benchmark function.

5. CEC 2017 Function Test

In order to better illustrate the generality and effectiveness of the algorithm and avoid that the ISSA is only applicable to the case where the optimal value is 0, the algorithm is tested on the CEC 2017 test function. The evaluation times are $10000 * \text{dim}$, the number of population individuals is 30, the dimension is 30, SD is set to 0.6, and other parameters remain unchanged. In this paper, the above algorithms are run independently for 30 times, and five indexes of each algorithm are calculated according to the results, namely, the best value (Best), the worst value (Worst), the median (Med), the average value (Ave), and the standard deviation (Std). Finally, each algorithm will be ranked according to the average value of the algorithm in the function. The optimal value of each index is treated in bold. Due to the defects of F2 function, it will not be tested in this paper. The specific test results are shown in Table 8. At the same time, six functions F4, F7, F14, F17, F24, and F27 are selected to draw the box diagram of the results, as shown in Figure 9.

It can be seen from the data in Table 8 that, in the 29 functions, ISSA ranks first in most functions and its average ranking is better than other algorithms. ISSA has good optimization effect and can be close to the theoretical optimal value of each function. In F1, F3, F6, F7, F8, F13, F14, F18, and F29, each index of ISSA is the best in the algorithm. Among the 20 functions, ISSA finds the optimal value of all algorithms, which shows that ISSA has strong optimization ability. When the optimal solution is not 0, ISSA shows better optimization performance than SSA, and PSO and CSSA also show good performance.

As can be seen from the box diagram in Figure 9, the ISSA has strong search ability and is closest to the theoretical optimal value among all algorithms. And ISSA's box graph has shorter length and stronger stability than other algorithms. Among them, the realization of SSA is poor because the individual in the SSA is directly jumping when approaching the current optimal solution, rather than moving to the current optimal solution like PSO. This problem leads to the rapid convergence of SSA, but it is easy to miss the high-quality solution and fall into local optimization. The ISSA uses the improved strategy to make up for this disadvantage, makes full use of the current solution, ensures the convergence speed, and increases its ability to jump out of the local optimum. But overall, the ISSA has better optimization performance than other algorithms, has good universality and effectiveness, and can adapt to some complex optimization problems.

6. PID Parameter Tuning

A PID controller is the most widely used controller in the industry (accounting for about 90% of the controller). The PID controller is composed of three basic gain parameters to control the controlled object. It is mainly applicable to the system whose basic linear and dynamic characteristics do not change with time. Its structure is shown in Figure 10. When the proportional parameter K_p increases, the rise time and steady-state error decrease. When the integral parameter K_i increases, the rise time is smaller, but the stability time and overshoot increase. The negative effect of the K_i increase can be overcome by adjusting the differential parameter K_d . The relationship between output and input of the PID controller is as follows:

$$u(t) = K_p e(t) + K_i \int e(t) + K_d \frac{de(t)}{dt}. \quad (28)$$

Manual PID parameter tuning is a time-consuming process. Generally, it is tried through the experience and skills of engineers and the intelligent algorithm can complete the parameter tuning in a short time. In order to verify the practicability of ISSA, this paper uses ISSA to optimize PID parameters, simulates under unit step response and sinusoidal input response respectively, and tests with SSA to prove its optimization performance.

In this paper, the objective function [39] is set as follows:

$$F = \int_0^{\infty} (w_1 |e(t)| + w_2 u^2(t)) dt, \quad (29)$$

$e(t)$ is the error between the input value and the output value. Considering the dynamic characteristics of the iterative process, the integral of its absolute value is adopted; $u(t)$ is the control value, which is added to avoid excessive control range; w_1 and w_2 are weights, and the value range is $[0, 1]$. In addition, measures shall be taken to prevent overshoot, that is, when overshoot occurs, an additional overshoot item shall be introduced into the objective function. At this time, the settings are as follows:

$$F = \int_0^{\infty} (w_1 |e(t)| + w_2 u^2(t) + w_3 |e(t)|) dt, \quad e(t) < 0, \quad (30)$$

where w_3 is the weight and $w_3 \gg w_1$. Generally, $w_1 = 0.999$, $w_2 = 0.001$, $w_3 = 100$. Therefore, the goal of ISSA is to find a set of PID parameters to minimize the error of objective function. In this paper, the number of individuals and the number of iterations of the population are set to be 30 and 50, respectively. Other parameters are consistent with those of Table 2. The test is carried out under the condition of unit leap forward and sinusoidal input respectively and run independently for 10 times. The test results are shown in Figure 11, and the optimization results are shown in Table 9.

Since the results of 10 independent runs are consistent, only one is shown here. It can be seen from Figure 11 and Table 9 that in the unit step response, ISSA can complete the parameter setting in a very short time, and the convergence

TABLE 8: Test results of each algorithm in CEC 2017.

| F | Index | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| F1 | Best | 1.52E+02 | 9.06E+09 | 5.10E+08 | 4.33E+06 | 4.28E+09 | 1.09E+02 | 1.65E+07 | 1.23E+02 | 1.06E+02 | 1.00E+02 |
| | Worst | 9.74E+03 | 1.84E+10 | 3.22E+09 | 1.17E+08 | 1.04E+10 | 1.99E+04 | 1.66E+08 | 1.20E+04 | 2.08E+04 | 9.24E+03 |
| | Med | 1.68E+03 | 1.19E+10 | 1.05E+09 | 2.06E+07 | 6.36E+09 | 6.79E+02 | 8.20E+07 | 1.48E+03 | 5.36E+03 | 5.23E+02 |
| | Ave | 2.53E+03 | 1.20E+10 | 1.35E+09 | 2.99E+07 | 6.43E+09 | 3.47E+02 | 9.87E+07 | 3.18E+03 | 8.01E+03 | 1.78E+03 |
| | Std | 2.61E+03 | 2.06E+09 | 7.53E+08 | 2.74E+07 | 1.21E+09 | 5.82E+03 | 4.14E+07 | 3.29E+03 | 7.96E+03 | 2.32E+03 |
| F3 | Rank | 2 | 10 | 8 | 6 | 9 | 4 | 7 | 3 | 5 | 1 |
| | Best | 1.95E+03 | 2.30E+04 | 2.87E+04 | 8.52E+04 | 4.82E+04 | 1.26E+04 | 4.67E+04 | 5.46E+03 | 2.15E+04 | 3.07E+02 |
| | Worst | 1.19E+04 | 4.89E+04 | 6.27E+04 | 3.02E+05 | 6.49E+04 | 2.98E+04 | 6.87E+04 | 1.44E+04 | 4.99E+04 | 7.66E+02 |
| | Med | 5.12E+03 | 3.46E+04 | 5.06E+04 | 1.89E+05 | 5.56E+04 | 1.67E+04 | 5.67E+04 | 1.02E+04 | 3.30E+04 | 4.90E+02 |
| | Ave | 5.49E+03 | 3.39E+04 | 4.92E+04 | 2.00E+05 | 5.53E+04 | 1.78E+04 | 5.69E+04 | 9.94E+03 | 3.21E+04 | 4.64E+02 |
| F4 | Std | 1.58E+03 | 8.48E+03 | 1.15E+04 | 7.55E+04 | 3.44E+03 | 4.49E+03 | 8.03E+03 | 2.35E+03 | 7.20E+03 | 1.20E+02 |
| | Rank | 2 | 6 | 7 | 10 | 8 | 4 | 9 | 3 | 5 | 1 |
| | Best | 4.18E+02 | 8.86E+02 | 5.20E+02 | 5.03E+02 | 1.23E+03 | 4.59E+02 | 5.10E+02 | 4.59E+02 | 4.08E+02 | 4.04E+02 |
| | Worst | 4.82E+02 | 1.89E+03 | 6.49E+02 | 6.48E+02 | 2.73E+03 | 5.38E+02 | 6.93E+02 | 5.35E+02 | 5.40E+02 | 6.09E+02 |
| | Med | 4.79E+02 | 1.53E+03 | 6.18E+02 | 5.58E+02 | 2.13E+03 | 5.10E+02 | 5.80E+02 | 4.88E+02 | 4.72E+02 | 4.77E+02 |
| F5 | Ave | 4.63E+02 | 1.48E+03 | 6.09E+02 | 5.61E+02 | 2.08E+03 | 5.08E+02 | 5.78E+02 | 4.83E+02 | 4.82E+02 | 4.66E+02 |
| | Std | 2.67E+01 | 2.49E+02 | 3.67E+01 | 4.26E+01 | 4.82E+02 | 1.79E+01 | 5.62E+01 | 1.56E+01 | 2.91E+01 | 4.15E+01 |
| | Rank | 1 | 9 | 8 | 6 | 10 | 5 | 7 | 4 | 3 | 2 |
| | Best | 6.31E+02 | 7.39E+02 | 5.53E+02 | 7.05E+02 | 7.99E+02 | 7.34E+02 | 5.91E+02 | 6.56E+02 | 6.21E+02 | 6.49E+02 |
| | Worst | 7.30E+02 | 8.27E+02 | 7.47E+02 | 9.26E+02 | 9.37E+02 | 8.20E+02 | 6.70E+02 | 8.07E+02 | 8.21E+02 | 8.16E+02 |
| F6 | Med | 7.00E+02 | 7.62E+02 | 5.94E+02 | 7.57E+02 | 8.49E+02 | 8.15E+02 | 6.45E+02 | 7.40E+02 | 7.18E+02 | 7.80E+02 |
| | Ave | 6.90E+02 | 7.68E+02 | 6.03E+02 | 7.80E+02 | 8.69E+02 | 8.08E+02 | 6.34E+02 | 7.40E+02 | 7.32E+02 | 7.62E+02 |
| | Std | 3.06E+01 | 1.73E+01 | 3.50E+01 | 7.36E+01 | 4.91E+01 | 1.79E+01 | 2.40E+01 | 4.57E+01 | 5.45E+01 | 3.41E+01 |
| | Rank | 3 | 7 | 1 | 8 | 10 | 9 | 2 | 5 | 4 | 6 |
| | Best | 6.34E+02 | 6.42E+02 | 6.38E+02 | 6.61E+02 | 6.65E+02 | 6.57E+02 | 6.08E+02 | 6.21E+02 | 6.22E+02 | 6.02E+02 |
| F7 | Worst | 6.57E+02 | 6.61E+02 | 6.68E+02 | 6.83E+02 | 6.88E+02 | 6.78E+02 | 6.65E+02 | 6.57E+02 | 6.58E+02 | 6.12E+02 |
| | Med | 6.45E+02 | 6.50E+02 | 6.57E+02 | 6.70E+02 | 6.76E+02 | 6.64E+02 | 6.29E+02 | 6.27E+02 | 6.41E+02 | 6.11E+02 |
| | Ave | 6.46E+02 | 6.49E+02 | 6.56E+02 | 6.72E+02 | 6.74E+02 | 6.64E+02 | 6.38E+02 | 6.33E+02 | 6.41E+02 | 6.08E+02 |
| | Std | 7.80E+00 | 5.44E+00 | 6.90E+00 | 8.86E+00 | 5.46E+00 | 6.39E+00 | 2.16E+01 | 1.08E+01 | 1.02E+01 | 4.37E+00 |
| | Rank | 5 | 6 | 7 | 9 | 10 | 8 | 3 | 2 | 4 | 1 |
| F8 | Best | 1.07E+03 | 1.08E+03 | 8.20E+02 | 1.06E+03 | 1.19E+03 | 1.18E+03 | 8.74E+02 | 9.28E+02 | 8.64E+02 | 7.99E+02 |
| | Worst | 1.37E+03 | 1.18E+03 | 9.91E+02 | 1.44E+03 | 1.41E+03 | 1.35E+03 | 1.33E+03 | 1.34E+03 | 1.33E+03 | 8.76E+02 |
| | Med | 1.32E+03 | 1.10E+03 | 8.91E+02 | 1.24E+03 | 1.27E+03 | 1.33E+03 | 9.82E+02 | 1.17E+03 | 1.14E+03 | 8.51E+02 |
| | Ave | 1.29E+03 | 1.11E+03 | 8.89E+02 | 1.28E+03 | 1.29E+03 | 1.32E+03 | 1.00E+03 | 1.17E+03 | 1.14E+03 | 8.46E+02 |
| | Std | 6.72E+01 | 2.99E+01 | 4.61E+01 | 1.15E+02 | 7.20E+01 | 3.85E+01 | 8.42E+01 | 1.09E+02 | 1.44E+02 | 1.66E+01 |
| F8 | Rank | 8 | 4 | 2 | 7 | 9 | 10 | 3 | 6 | 5 | 1 |
| | Best | 8.74E+02 | 1.02E+03 | 9.08E+02 | 9.42E+02 | 1.04E+03 | 9.38E+02 | 9.15E+02 | 8.60E+02 | 8.84E+02 | 8.55E+02 |
| | Worst | 9.62E+02 | 1.09E+03 | 1.01E+03 | 1.08E+03 | 1.11E+03 | 1.05E+03 | 9.79E+02 | 9.82E+02 | 1.01E+03 | 9.14E+02 |
| | Med | 9.15E+02 | 1.04E+03 | 9.52E+02 | 9.72E+02 | 1.09E+03 | 9.92E+02 | 9.59E+02 | 9.43E+02 | 9.66E+02 | 8.80E+02 |
| | Ave | 9.21E+02 | 1.05E+03 | 9.57E+02 | 9.89E+02 | 1.08E+03 | 9.94E+02 | 9.53E+02 | 9.48E+02 | 9.58E+02 | 8.82E+02 |
| Std | 2.58E+01 | 2.28E+01 | 2.31E+01 | 3.69E+01 | 1.63E+01 | 2.56E+01 | 1.44E+01 | 2.79E+01 | 3.98E+01 | 1.37E+01 | |
| Rank | 2 | 9 | 5 | 7 | 10 | 8 | 4 | 3 | 6 | 1 | |

TABLE 8: Continued.

| F | Index | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|-----|-------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| F9 | Best | 2.45E+03 | 3.74E+03 | 3.57E+03 | 3.60E+03 | 6.34E+03 | 4.01E+03 | 2.04E+03 | 3.08E+03 | 5.06E+03 | 1.06E+03 |
| | Worst | 5.01E+03 | 7.85E+03 | 5.79E+03 | 1.35E+04 | 1.15E+04 | 5.58E+03 | 9.94E+03 | 5.44E+03 | 5.56E+03 | 1.85E+03 |
| | Med | 3.99E+03 | 4.43E+03 | 4.97E+03 | 6.12E+03 | 8.81E+03 | 5.41E+03 | 5.85E+03 | 5.35E+03 | 5.39E+03 | 1.44E+03 |
| | Ave | 3.84E+03 | 4.95E+03 | 4.95E+03 | 6.82E+03 | 8.73E+03 | 5.30E+03 | 6.02E+03 | 4.88E+03 | 5.37E+03 | 1.42E+03 |
| | Std | 5.34E+02 | 1.09E+03 | 4.57E+02 | 1.82E+03 | 1.34E+03 | 3.75E+02 | 2.32E+03 | 8.19E+02 | 1.23E+02 | 1.87E+02 |
| | Rank | 2 | 4 | 5 | 9 | 10 | 6 | 8 | 3 | 7 | 1 |
| F10 | Best | 3.71E+03 | 7.33E+03 | 3.15E+03 | 5.34E+03 | 5.74E+03 | 4.57E+03 | 4.82E+03 | 4.09E+03 | 3.91E+03 | 4.06E+03 |
| | Worst | 5.45E+03 | 8.48E+03 | 5.06E+03 | 8.13E+03 | 8.59E+03 | 9.10E+03 | 9.80E+03 | 6.33E+03 | 6.53E+03 | 8.89E+03 |
| | Med | 4.27E+03 | 8.17E+03 | 4.20E+03 | 7.29E+03 | 7.34E+03 | 5.95E+03 | 6.57E+03 | 5.39E+03 | 4.58E+03 | 6.24E+03 |
| | Ave | 4.47E+03 | 8.13E+03 | 4.05E+03 | 7.00E+03 | 7.54E+03 | 6.43E+03 | 6.71E+03 | 5.42E+03 | 4.83E+03 | 6.09E+03 |
| | Std | 5.13E+02 | 2.79E+02 | 6.38E+02 | 9.77E+02 | 6.40E+02 | 1.63E+03 | 1.60E+03 | 5.13E+02 | 6.99E+02 | 1.09E+03 |
| | Rank | 2 | 10 | 1 | 8 | 9 | 6 | 7 | 4 | 3 | 5 |
| F11 | Best | 1.16E+03 | 1.71E+03 | 1.34E+03 | 1.47E+03 | 2.57E+03 | 1.19E+03 | 1.31E+03 | 1.16E+03 | 1.25E+03 | 1.14E+03 |
| | Worst | 1.25E+03 | 3.73E+03 | 2.12E+03 | 2.49E+03 | 5.28E+03 | 1.39E+03 | 1.88E+03 | 1.32E+03 | 1.59E+03 | 1.29E+03 |
| | Med | 1.19E+03 | 1.90E+03 | 1.44E+03 | 2.30E+03 | 4.37E+03 | 1.23E+03 | 1.43E+03 | 1.26E+03 | 1.44E+03 | 1.19E+03 |
| | Ave | 1.19E+03 | 2.02E+03 | 1.59E+03 | 2.12E+03 | 4.23E+03 | 1.25E+03 | 1.45E+03 | 1.25E+03 | 1.43E+03 | 1.19E+03 |
| | Std | 2.46E+01 | 4.01E+02 | 2.58E+02 | 3.84E+02 | 6.64E+02 | 4.68E+01 | 1.36E+02 | 4.06E+01 | 9.29E+01 | 3.12E+01 |
| | Rank | 1 | 8 | 7 | 9 | 10 | 3 | 6 | 4 | 5 | 2 |
| F12 | Best | 2.20E+04 | 6.42E+08 | 4.00E+06 | 1.03E+07 | 6.90E+08 | 3.78E+04 | 9.49E+05 | 1.09E+05 | 1.25E+05 | 2.90E+04 |
| | Worst | 1.25E+06 | 1.99E+09 | 3.22E+08 | 1.08E+08 | 2.59E+09 | 1.13E+07 | 1.22E+07 | 6.57E+05 | 9.09E+05 | 1.91E+07 |
| | Med | 3.54E+05 | 8.82E+08 | 3.93E+07 | 6.62E+07 | 9.19E+08 | 2.07E+06 | 2.91E+06 | 2.08E+05 | 4.45E+05 | 1.54E+06 |
| | Ave | 4.22E+05 | 1.03E+09 | 6.63E+07 | 6.01E+07 | 1.11E+09 | 2.45E+06 | 4.41E+06 | 2.23E+05 | 4.57E+05 | 2.11E+06 |
| | Std | 3.55E+05 | 4.03E+08 | 8.44E+07 | 2.84E+07 | 5.40E+08 | 2.66E+06 | 3.09E+06 | 1.11E+05 | 2.80E+05 | 3.55E+06 |
| | Rank | 2 | 9 | 8 | 7 | 10 | 5 | 6 | 1 | 3 | 4 |
| F13 | Best | 2.90E+03 | 2.13E+08 | 3.83E+04 | 3.85E+04 | 6.22E+07 | 3.05E+03 | 8.09E+03 | 2.97E+03 | 5.95E+03 | 1.68E+03 |
| | Worst | 5.62E+04 | 6.13E+08 | 1.40E+08 | 2.82E+05 | 4.32E+08 | 7.25E+04 | 4.51E+06 | 7.12E+04 | 5.45E+04 | 1.96E+04 |
| | Med | 8.39E+03 | 3.48E+08 | 1.84E+05 | 1.22E+05 | 1.53E+08 | 1.39E+04 | 2.34E+04 | 9.69E+03 | 2.75E+04 | 9.36E+03 |
| | Ave | 1.21E+04 | 3.69E+08 | 3.90E+07 | 1.19E+05 | 1.63E+08 | 2.56E+04 | 5.41E+05 | 1.38E+04 | 2.92E+04 | 9.11E+03 |
| | Std | 1.07E+04 | 9.87E+07 | 5.71E+07 | 5.05E+04 | 8.14E+07 | 2.50E+04 | 1.36E+06 | 1.58E+04 | 2.09E+04 | 4.37E+03 |
| | Rank | 2 | 10 | 8 | 6 | 9 | 4 | 7 | 3 | 5 | 1 |
| F14 | Best | 4.96E+03 | 4.08E+04 | 2.93E+03 | 1.37E+04 | 1.41E+06 | 5.09E+03 | 7.46E+03 | 1.67E+03 | 2.50E+03 | 1.66E+03 |
| | Worst | 1.12E+05 | 2.71E+05 | 8.84E+05 | 7.05E+06 | 4.40E+06 | 1.64E+05 | 1.79E+06 | 7.66E+04 | 1.09E+05 | 5.18E+04 |
| | Med | 2.71E+04 | 8.97E+04 | 4.63E+04 | 1.11E+06 | 2.68E+06 | 3.07E+04 | 3.35E+05 | 1.39E+04 | 2.86E+04 | 5.97E+03 |
| | Ave | 3.93E+04 | 1.25E+05 | 2.31E+05 | 1.86E+06 | 2.54E+06 | 3.37E+04 | 4.24E+05 | 1.69E+04 | 3.86E+04 | 1.52E+04 |
| | Std | 2.72E+04 | 6.22E+04 | 2.90E+05 | 1.72E+06 | 9.81E+05 | 2.81E+04 | 4.18E+05 | 1.72E+04 | 3.18E+04 | 1.71E+04 |
| | Rank | 5 | 6 | 7 | 9 | 10 | 3 | 8 | 2 | 4 | 1 |
| F15 | Best | 1.65E+03 | 6.16E+05 | 1.39E+04 | 2.15E+04 | 3.22E+06 | 2.08E+03 | 2.30E+03 | 1.85E+03 | 1.86E+03 | 1.57E+03 |
| | Worst | 2.12E+04 | 5.07E+07 | 5.42E+05 | 1.83E+05 | 5.30E+07 | 2.29E+04 | 2.04E+04 | 1.50E+04 | 4.43E+04 | 3.46E+04 |
| | Med | 4.05E+03 | 5.21E+06 | 1.03E+05 | 4.40E+04 | 7.88E+06 | 5.81E+03 | 6.81E+03 | 3.46E+03 | 2.92E+04 | 2.38E+03 |
| | Ave | 5.39E+03 | 1.11E+07 | 1.12E+05 | 6.36E+04 | 1.36E+07 | 7.88E+03 | 6.92E+03 | 4.75E+03 | 2.35E+04 | 4.92E+03 |
| | Std | 5.43E+03 | 1.08E+07 | 1.23E+05 | 4.40E+04 | 1.44E+07 | 6.35E+03 | 3.96E+03 | 3.14E+03 | 1.80E+04 | 6.67E+03 |
| | Rank | 3 | 9 | 8 | 7 | 10 | 5 | 4 | 1 | 6 | 2 |

TABLE 8: Continued.

| F | Index | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|-----|-------|-----------------|-----------------|-----------------|----------|----------|----------|-----------------|-----------------|-----------------|-----------------|
| F16 | Best | 2.24E+03 | 3.37E+03 | 1.98E+03 | 2.93E+03 | 3.86E+03 | 2.83E+03 | 2.19E+03 | 2.42E+03 | 2.33E+03 | 1.88E+03 |
| | Worst | 3.06E+03 | 4.09E+03 | 3.12E+03 | 4.56E+03 | 5.69E+03 | 8.55E+03 | 3.60E+03 | 3.30E+03 | 3.33E+03 | 5.57E+03 |
| | Med | 2.85E+03 | 3.66E+03 | 2.77E+03 | 3.33E+03 | 4.15E+03 | 3.79E+03 | 2.81E+03 | 3.10E+03 | 2.89E+03 | 3.23E+03 |
| | Ave | 2.80E+03 | 3.68E+03 | 2.56E+03 | 3.44E+03 | 4.31E+03 | 4.12E+03 | 2.84E+03 | 3.00E+03 | 2.87E+03 | 3.35E+03 |
| | Std | 2.70E+02 | 1.55E+02 | 4.02E+02 | 4.03E+02 | 4.33E+02 | 1.23E+03 | 3.20E+02 | 2.71E+02 | 2.51E+02 | 1.09E+03 |
| | Rank | 2 | 8 | 1 | 7 | 10 | 9 | 3 | 5 | 4 | 6 |
| F17 | Best | 1.94E+03 | 2.11E+03 | 2.07E+03 | 2.15E+03 | 2.22E+03 | 1.99E+03 | 1.96E+03 | 2.05E+03 | 2.13E+03 | 1.83E+03 |
| | Worst | 2.75E+03 | 2.69E+03 | 3.81E+03 | 3.22E+03 | 3.15E+03 | 3.36E+03 | 2.92E+03 | 3.01E+03 | 3.02E+03 | 2.25E+03 |
| | Med | 2.32E+03 | 2.30E+03 | 3.08E+03 | 2.56E+03 | 2.95E+03 | 2.68E+03 | 2.40E+03 | 2.44E+03 | 2.65E+03 | 1.99E+03 |
| | Ave | 2.33E+03 | 2.37E+03 | 3.00E+03 | 2.65E+03 | 2.89E+03 | 2.81E+03 | 2.42E+03 | 2.46E+03 | 2.58E+03 | 2.01E+03 |
| | Std | 2.55E+02 | 1.44E+02 | 4.67E+02 | 2.65E+02 | 2.49E+02 | 3.40E+02 | 2.45E+02 | 2.11E+02 | 2.25E+02 | 1.48E+02 |
| | Rank | 2 | 3 | 10 | 7 | 9 | 8 | 4 | 5 | 6 | 1 |
| F18 | Best | 5.23E+04 | 3.20E+05 | 4.38E+04 | 2.75E+05 | 2.86E+06 | 1.19E+04 | 5.88E+04 | 3.78E+04 | 1.09E+05 | 6.74E+03 |
| | Worst | 5.93E+05 | 6.66E+06 | 1.06E+07 | 8.14E+06 | 2.41E+07 | 1.54E+06 | 7.28E+06 | 4.14E+05 | 1.47E+06 | 1.74E+05 |
| | Med | 3.46E+05 | 1.48E+06 | 4.59E+05 | 2.96E+06 | 8.73E+06 | 7.18E+04 | 5.67E+05 | 8.42E+04 | 2.53E+05 | 4.00E+04 |
| | Ave | 2.74E+05 | 1.83E+06 | 1.55E+06 | 2.81E+06 | 8.77E+06 | 1.80E+05 | 1.79E+06 | 1.43E+05 | 3.56E+05 | 7.38E+04 |
| | Std | 1.84E+05 | 1.65E+06 | 2.78E+06 | 2.01E+06 | 3.77E+06 | 3.80E+05 | 2.81E+06 | 1.22E+05 | 3.37E+05 | 6.25E+04 |
| | Rank | 4 | 8 | 6 | 9 | 10 | 3 | 7 | 2 | 5 | 1 |
| F19 | Best | 2.04E+03 | 4.15E+06 | 2.06E+04 | 9.08E+05 | 6.49E+06 | 2.53E+03 | 2.57E+03 | 2.17E+03 | 2.36E+03 | 2.02E+03 |
| | Worst | 2.18E+04 | 3.99E+07 | 4.22E+06 | 1.30E+07 | 6.12E+07 | 1.63E+04 | 2.10E+04 | 9.56E+03 | 5.64E+04 | 2.64E+04 |
| | Med | 3.82E+03 | 2.04E+07 | 5.12E+05 | 3.18E+06 | 1.39E+07 | 5.51E+03 | 4.44E+03 | 5.15E+03 | 6.78E+03 | 6.49E+03 |
| | Ave | 5.54E+03 | 2.38E+07 | 9.64E+05 | 4.52E+06 | 2.01E+07 | 6.91E+03 | 7.75E+03 | 5.63E+03 | 1.09E+04 | 7.79E+03 |
| | Std | 4.49E+03 | 9.96E+06 | 1.34E+06 | 4.12E+06 | 1.30E+07 | 3.76E+03 | 6.79E+03 | 2.43E+03 | 1.36E+04 | 5.90E+03 |
| | Rank | 1 | 10 | 7 | 8 | 9 | 5 | 4 | 2 | 6 | 5 |
| F20 | Best | 2.26E+03 | 2.38E+03 | 2.27E+03 | 2.38E+03 | 2.49E+03 | 2.39E+03 | 2.14E+03 | 2.29E+03 | 2.30E+03 | 2.15E+03 |
| | Worst | 2.89E+03 | 2.73E+03 | 3.39E+03 | 3.15E+03 | 2.99E+03 | 3.43E+03 | 3.22E+03 | 2.95E+03 | 2.91E+03 | 2.77E+03 |
| | Med | 2.44E+03 | 2.58E+03 | 3.04E+03 | 2.80E+03 | 2.78E+03 | 2.75E+03 | 2.66E+03 | 2.54E+03 | 2.71E+03 | 2.27E+03 |
| | Ave | 2.49E+03 | 2.59E+03 | 2.97E+03 | 2.74E+03 | 2.78E+03 | 2.84E+03 | 2.60E+03 | 2.52E+03 | 2.69E+03 | 2.30E+03 |
| | Std | 1.63E+02 | 8.77E+01 | 2.43E+02 | 2.01E+02 | 1.16E+02 | 2.39E+02 | 1.92E+02 | 1.83E+02 | 1.27E+02 | 1.30E+02 |
| | Rank | 2 | 4 | 10 | 7 | 8 | 9 | 5 | 3 | 6 | 1 |
| F21 | Best | 2.39E+03 | 2.52E+03 | 2.42E+03 | 2.50E+03 | 2.56E+03 | 2.49E+03 | 2.40E+03 | 2.20E+03 | 2.42E+03 | 2.36E+03 |
| | Worst | 2.51E+03 | 2.60E+03 | 2.68E+03 | 2.68E+03 | 2.74E+03 | 2.73E+03 | 2.49E+03 | 2.58E+03 | 2.58E+03 | 2.46E+03 |
| | Med | 2.45E+03 | 2.53E+03 | 2.54E+03 | 2.57E+03 | 2.63E+03 | 2.62E+03 | 2.43E+03 | 2.49E+03 | 2.46E+03 | 2.39E+03 |
| | Ave | 2.45E+03 | 2.54E+03 | 2.53E+03 | 2.58E+03 | 2.63E+03 | 2.59E+03 | 2.44E+03 | 2.48E+03 | 2.48E+03 | 2.41E+03 |
| | Std | 2.32E+01 | 1.77E+01 | 5.36E+01 | 5.01E+01 | 3.40E+01 | 5.86E+01 | 3.25E+01 | 6.21E+01 | 4.72E+01 | 3.81E+01 |
| | Rank | 3 | 7 | 6 | 8 | 10 | 9 | 2 | 4 | 5 | 1 |
| F22 | Best | 2.30E+03 | 3.96E+03 | 2.50E+03 | 2.33E+03 | 3.59E+03 | 2.38E+03 | 2.34E+03 | 2.30E+03 | 2.30E+03 | 2.38E+03 |
| | Worst | 6.81E+03 | 9.97E+03 | 1.00E+04 | 8.80E+03 | 1.03E+04 | 1.04E+04 | 8.79E+03 | 7.87E+03 | 7.58E+03 | 9.42E+03 |
| | Med | 5.86E+03 | 9.60E+03 | 4.99E+03 | 7.17E+03 | 4.34E+03 | 7.44E+03 | 2.38E+03 | 2.30E+03 | 5.73E+03 | 7.31E+03 |
| | Ave | 4.91E+03 | 9.00E+03 | 4.48E+03 | 6.68E+03 | 6.02E+03 | 7.56E+03 | 3.30E+03 | 3.34E+03 | 5.46E+03 | 7.27E+03 |
| | Std | 1.90E+03 | 1.71E+03 | 1.98E+03 | 1.76E+03 | 2.55E+03 | 1.42E+03 | 2.12E+03 | 1.94E+03 | 2.08E+03 | 1.30E+03 |
| | Rank | 4 | 10 | 3 | 7 | 6 | 9 | 1 | 2 | 5 | 8 |

TABLE 8: Continued.

| F | Index | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|-----|-------|-----------------|-----------------|-----------------|----------|----------|-----------------|-----------------|-----------------|-----------------|-----------------|
| F23 | Best | 2.70E+03 | 2.95E+03 | 2.71E+03 | 2.93E+03 | 2.99E+03 | 2.99E+03 | 2.76E+03 | 2.83E+03 | 2.78E+03 | 2.85E+03 |
| | Worst | 3.36E+03 | 3.05E+03 | 2.95E+03 | 3.29E+03 | 3.25E+03 | 3.57E+03 | 2.91E+03 | 3.06E+03 | 2.95E+03 | 3.59E+03 |
| | Med | 3.15E+03 | 2.97E+03 | 2.75E+03 | 3.03E+03 | 3.05E+03 | 3.33E+03 | 2.83E+03 | 2.92E+03 | 2.82E+03 | 3.31E+03 |
| | Ave | 3.16E+03 | 2.98E+03 | 2.77E+03 | 3.02E+03 | 3.10E+03 | 3.33E+03 | 2.82E+03 | 2.93E+03 | 2.83E+03 | 3.26E+03 |
| | Std | 1.45E+02 | 3.02E+01 | 5.08E+01 | 9.83E+01 | 9.88E+01 | 1.35E+02 | 4.96E+01 | 6.17E+01 | 4.42E+01 | 1.70E+02 |
| | Rank | 8 | 5 | 1 | 6 | 7 | 10 | 2 | 4 | 3 | 9 |
| F24 | Best | 3.07E+03 | 3.13E+03 | 3.11E+03 | 3.00E+03 | 3.13E+03 | 3.19E+03 | 2.91E+03 | 2.98E+03 | 2.93E+03 | 2.87E+03 |
| | Worst | 3.40E+03 | 3.20E+03 | 3.77E+03 | 3.36E+03 | 3.31E+03 | 3.69E+03 | 3.06E+03 | 3.37E+03 | 3.10E+03 | 3.08E+03 |
| | Med | 3.23E+03 | 3.17E+03 | 3.34E+03 | 3.19E+03 | 3.20E+03 | 3.34E+03 | 3.01E+03 | 3.13E+03 | 3.01E+03 | 2.91E+03 |
| | Ave | 3.24E+03 | 3.17E+03 | 3.40E+03 | 3.17E+03 | 3.21E+03 | 3.33E+03 | 2.99E+03 | 3.14E+03 | 3.01E+03 | 2.93E+03 |
| | Std | 1.33E+02 | 1.72E+01 | 1.94E+02 | 7.64E+01 | 5.73E+01 | 1.05E+02 | 4.18E+01 | 1.03E+02 | 4.47E+01 | 5.27E+01 |
| | Rank | 8 | 5 | 10 | 6 | 7 | 9 | 2 | 4 | 3 | 1 |
| F25 | Best | 2.88E+03 | 3.11E+03 | 2.92E+03 | 2.91E+03 | 3.30E+03 | 2.88E+03 | 2.90E+03 | 2.88E+03 | 2.88E+03 | 2.88E+03 |
| | Worst | 2.90E+03 | 3.36E+03 | 3.03E+03 | 3.05E+03 | 3.55E+03 | 2.95E+03 | 3.02E+03 | 2.94E+03 | 2.89E+03 | 2.95E+03 |
| | Med | 2.88E+03 | 3.22E+03 | 2.98E+03 | 2.93E+03 | 3.39E+03 | 2.90E+03 | 2.96E+03 | 2.89E+03 | 2.89E+03 | 2.89E+03 |
| | Ave | 2.88E+03 | 3.23E+03 | 2.98E+03 | 2.95E+03 | 3.38E+03 | 2.91E+03 | 2.97E+03 | 2.90E+03 | 2.89E+03 | 2.90E+03 |
| | Std | 3.67E+00 | 6.72E+01 | 3.00E+01 | 3.94E+01 | 5.73E+01 | 2.12E+01 | 3.70E+01 | 1.78E+01 | 1.51E+00 | 2.86E+01 |
| | Rank | 1 | 9 | 8 | 6 | 10 | 5 | 7 | 3 | 2 | 4 |
| F26 | Best | 5.76E+03 | 6.49E+03 | 4.29E+03 | 6.03E+03 | 7.32E+03 | 3.01E+03 | 3.55E+03 | 2.90E+03 | 2.80E+03 | 2.80E+03 |
| | Worst | 8.91E+03 | 7.27E+03 | 5.10E+03 | 9.96E+03 | 1.04E+04 | 1.21E+04 | 7.60E+03 | 7.55E+03 | 6.22E+03 | 1.09E+04 |
| | Med | 6.60E+03 | 6.97E+03 | 4.63E+03 | 7.58E+03 | 8.11E+03 | 9.02E+03 | 5.23E+03 | 5.60E+03 | 5.13E+03 | 6.32E+03 |
| | Ave | 7.08E+03 | 6.92E+03 | 4.64E+03 | 7.58E+03 | 8.29E+03 | 8.76E+03 | 5.23E+03 | 5.65E+03 | 4.67E+03 | 6.69E+03 |
| | Std | 1.06E+03 | 1.75E+02 | 1.51E+02 | 6.53E+02 | 6.15E+02 | 1.80E+03 | 7.03E+02 | 1.07E+03 | 1.35E+03 | 2.08E+03 |
| | Rank | 7 | 6 | 1 | 8 | 9 | 10 | 5 | 4 | 2 | 5 |
| F27 | Best | 3.17E+03 | 3.34E+03 | 3.22E+03 | 3.27E+03 | 3.38E+03 | 3.33E+03 | 3.22E+03 | 3.23E+03 | 3.22E+03 | 3.20E+03 |
| | Worst | 3.42E+03 | 3.49E+03 | 3.33E+03 | 3.68E+03 | 4.00E+03 | 4.14E+03 | 3.40E+03 | 3.41E+03 | 3.35E+03 | 3.20E+03 |
| | Med | 3.19E+03 | 3.41E+03 | 3.25E+03 | 3.33E+03 | 3.60E+03 | 3.80E+03 | 3.28E+03 | 3.30E+03 | 3.23E+03 | 3.20E+03 |
| | Ave | 3.21E+03 | 3.41E+03 | 3.26E+03 | 3.41E+03 | 3.61E+03 | 3.71E+03 | 3.29E+03 | 3.32E+03 | 3.25E+03 | 3.20E+03 |
| | Std | 6.02E+01 | 3.97E+01 | 2.94E+01 | 1.39E+02 | 1.26E+02 | 2.21E+02 | 6.50E+01 | 6.10E+01 | 4.13E+01 | 1.09E-04 |
| | Rank | 2 | 7 | 4 | 8 | 9 | 10 | 5 | 6 | 3 | 1 |
| F28 | Best | 3.10E+03 | 3.65E+03 | 3.30E+03 | 3.27E+03 | 3.66E+03 | 3.20E+03 | 3.26E+03 | 3.10E+03 | 3.12E+03 | 3.10E+03 |
| | Worst | 3.27E+03 | 4.12E+03 | 3.62E+03 | 3.40E+03 | 4.26E+03 | 3.29E+03 | 3.37E+03 | 3.26E+03 | 3.26E+03 | 3.30E+03 |
| | Med | 3.21E+03 | 3.82E+03 | 3.41E+03 | 3.33E+03 | 3.87E+03 | 3.22E+03 | 3.33E+03 | 3.11E+03 | 3.21E+03 | 3.30E+03 |
| | Ave | 3.21E+03 | 3.85E+03 | 3.40E+03 | 3.33E+03 | 3.95E+03 | 3.22E+03 | 3.33E+03 | 3.15E+03 | 3.22E+03 | 3.28E+03 |
| | Std | 4.84E+01 | 1.55E+02 | 7.88E+01 | 2.67E+01 | 1.72E+02 | 2.45E+01 | 3.22E+01 | 6.51E+01 | 3.33E+01 | 4.67E+01 |
| | Rank | 2 | 9 | 8 | 6 | 10 | 3 | 7 | 1 | 4 | 5 |
| F29 | Best | 3.60E+03 | 4.24E+03 | 3.94E+03 | 4.17E+03 | 4.73E+03 | 4.27E+03 | 3.59E+03 | 3.42E+03 | 3.52E+03 | 3.58E+03 |
| | Worst | 4.30E+03 | 5.05E+03 | 8.78E+03 | 5.73E+03 | 6.02E+03 | 7.31E+03 | 4.26E+03 | 4.82E+03 | 4.40E+03 | 4.06E+03 |
| | Med | 3.95E+03 | 4.53E+03 | 4.92E+03 | 4.97E+03 | 5.11E+03 | 5.24E+03 | 4.16E+03 | 4.09E+03 | 4.05E+03 | 3.78E+03 |
| | Ave | 3.94E+03 | 4.59E+03 | 5.22E+03 | 4.88E+03 | 5.22E+03 | 5.27E+03 | 4.06E+03 | 4.27E+03 | 3.97E+03 | 3.79E+03 |
| | Std | 1.63E+02 | 2.42E+02 | 9.10E+02 | 3.98E+02 | 3.54E+02 | 6.08E+02 | 1.77E+02 | 4.26E+02 | 2.14E+02 | 1.27E+02 |
| | Rank | 2 | 6 | 8 | 7 | 9 | 10 | 4 | 5 | 3 | 1 |

TABLE 8: Continued.

| F | Index | PSO | SCA | GWO | WOA | MWOA | SSA | BSSA | CSSA | LSSA | ISSA |
|---|--------------|-----------------|----------|----------|----------|----------|----------|----------|----------|----------|-----------------|
| | Best | 5.40E+03 | 4.89E+07 | 1.44E+06 | 4.12E+06 | 2.32E+07 | 1.21E+04 | 1.11E+04 | 5.82E+03 | 6.80E+03 | 3.61E+03 |
| | Worst | 1.07E+04 | 1.15E+08 | 3.20E+07 | 4.34E+07 | 3.98E+08 | 6.06E+06 | 3.05E+05 | 1.88E+04 | 2.42E+04 | 1.29E+05 |
| | Med | 6.25E+03 | 7.19E+07 | 7.15E+06 | 1.34E+07 | 1.38E+08 | 3.41E+04 | 4.66E+04 | 1.05E+04 | 1.97E+04 | 1.25E+04 |
| | Ave | 6.76E+03 | 7.64E+07 | 1.60E+07 | 1.49E+07 | 1.54E+08 | 1.25E+06 | 5.65E+04 | 1.06E+04 | 1.72E+04 | 1.68E+04 |
| | Std | 1.69E+03 | 1.76E+07 | 1.35E+07 | 1.22E+07 | 9.75E+07 | 2.45E+06 | 5.64E+04 | 3.59E+03 | 6.03E+03 | 2.37E+04 |
| | Rank | 1 | 9 | 8 | 7 | 10 | 6 | 5 | 2 | 4 | 3 |
| | Average rank | 3.07 | 7.34 | 6.00 | 7.41 | 9.20 | 6.66 | 4.90 | 3.31 | 4.35 | 2.80 |

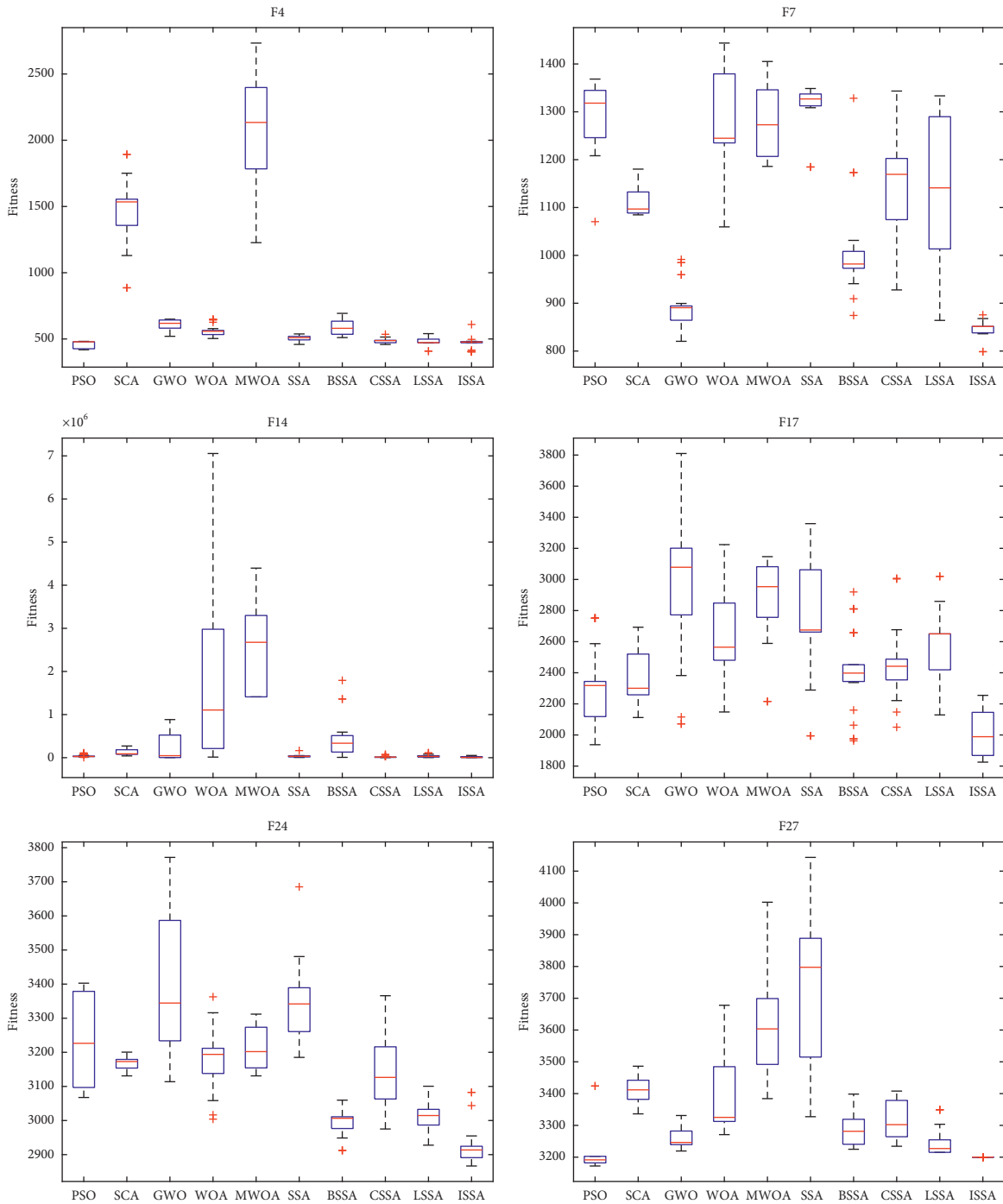


FIGURE 9: Box diagram of each algorithm.

speed and accuracy are better than SSA. In the sinusoidal input response, ISSA can almost coincide with r_{in} , and the tracking effect is significant. The tracking effect of SSA is slightly inferior. At the same time, the convergence speed and accuracy of ISSA are slightly better than SSA. The above results verify that ISSA has good algorithm performance, can quickly and accurately complete the PID parameter tuning, and help the system have shorter response time, higher system control accuracy, and better robustness. So far, the practicability has been proved.

7. Robot Path Planning

In PID parameter tuning, the dimension of the practical application is low. Therefore, this paper selects the discrete problem of more complex path planning to further verify the practicability of ISSA and makes a comparative experiment with SSA. In path planning, each sparrow is a feasible path. The environment modeling adopts the grid method, and the obstacles at the equivalent position are calculated according to the grid value. Grid number 0 is defined as feasible area and 1 as obstacle area.

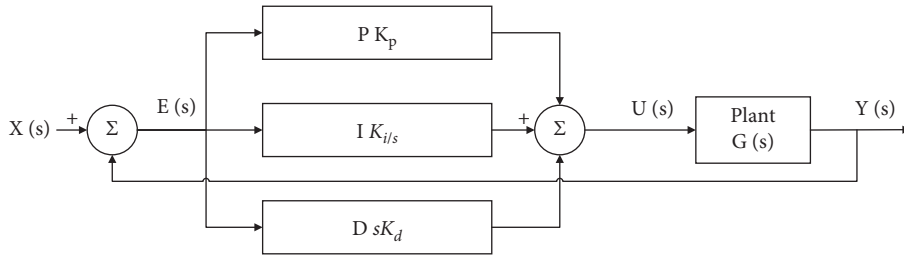


FIGURE 10: PID controller structure diagram.

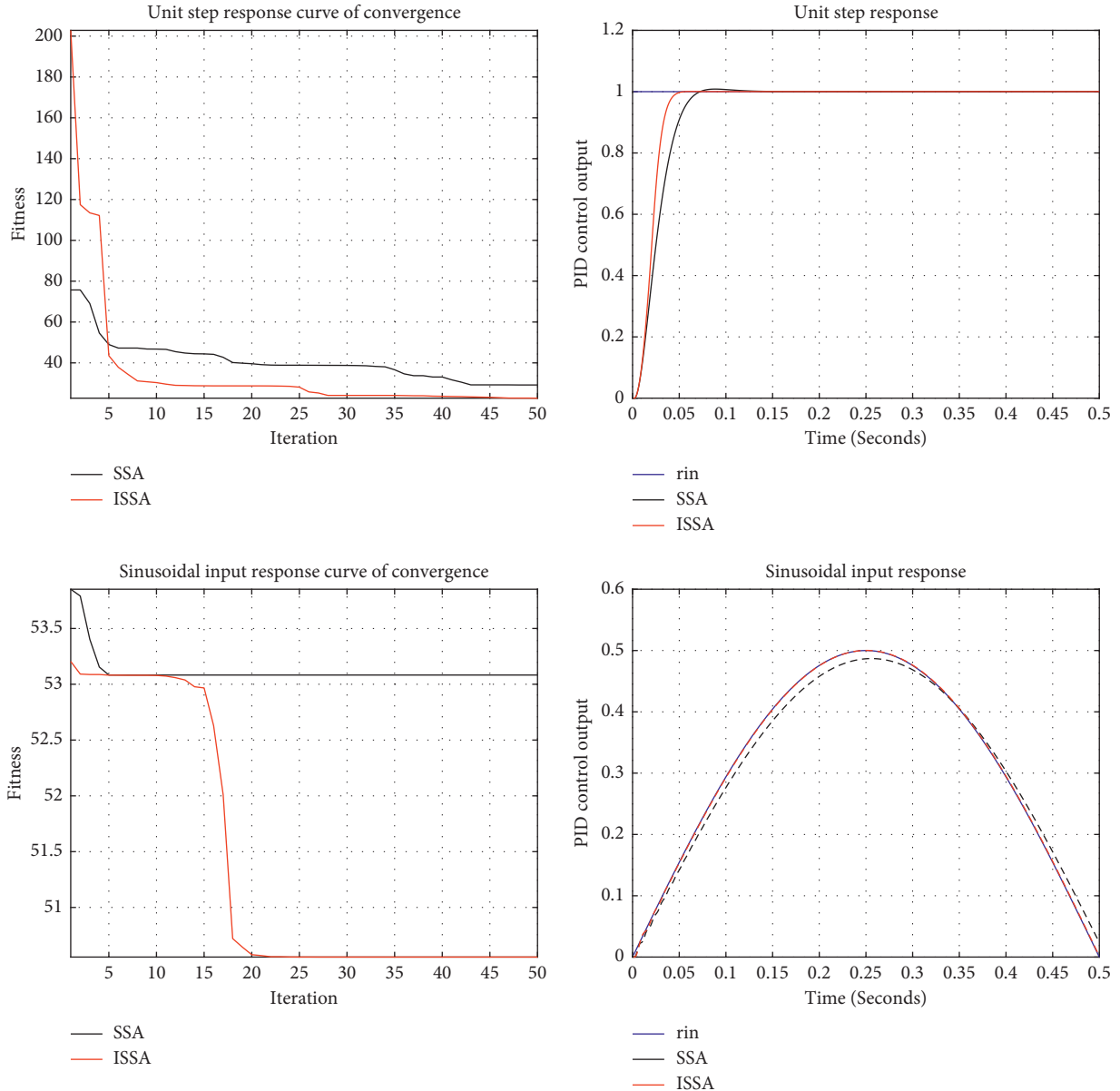


FIGURE 11: Convergence curve and PID control output.

Then, the robot can plan the path on the grid specified as 0, and dimension D is the column number of the grid map. The cost function of the path length of the i -th sparrow is shown in

$$f(x) = \sum_{j=1}^{D-1} \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2}. \quad (31)$$

TABLE 9: PID parameter setting results.

| Algorithm | Unit step | | | | Sinusoidal input | | | |
|-----------|----------------|---------|----------|----------|------------------|----------|----------|----------|
| | Fitness | K_p | K_i | K_d | Fitness | K_p | K_i | K_d |
| SSA | 29.1777 | 10 | 0.221046 | 0.126894 | 53.0827 | 10 | 10 | 1.90441 |
| ISSA | 22.7045 | 41.4593 | 0.596364 | 0.402723 | 50.5554 | 286.6486 | 21.19153 | 0.898048 |

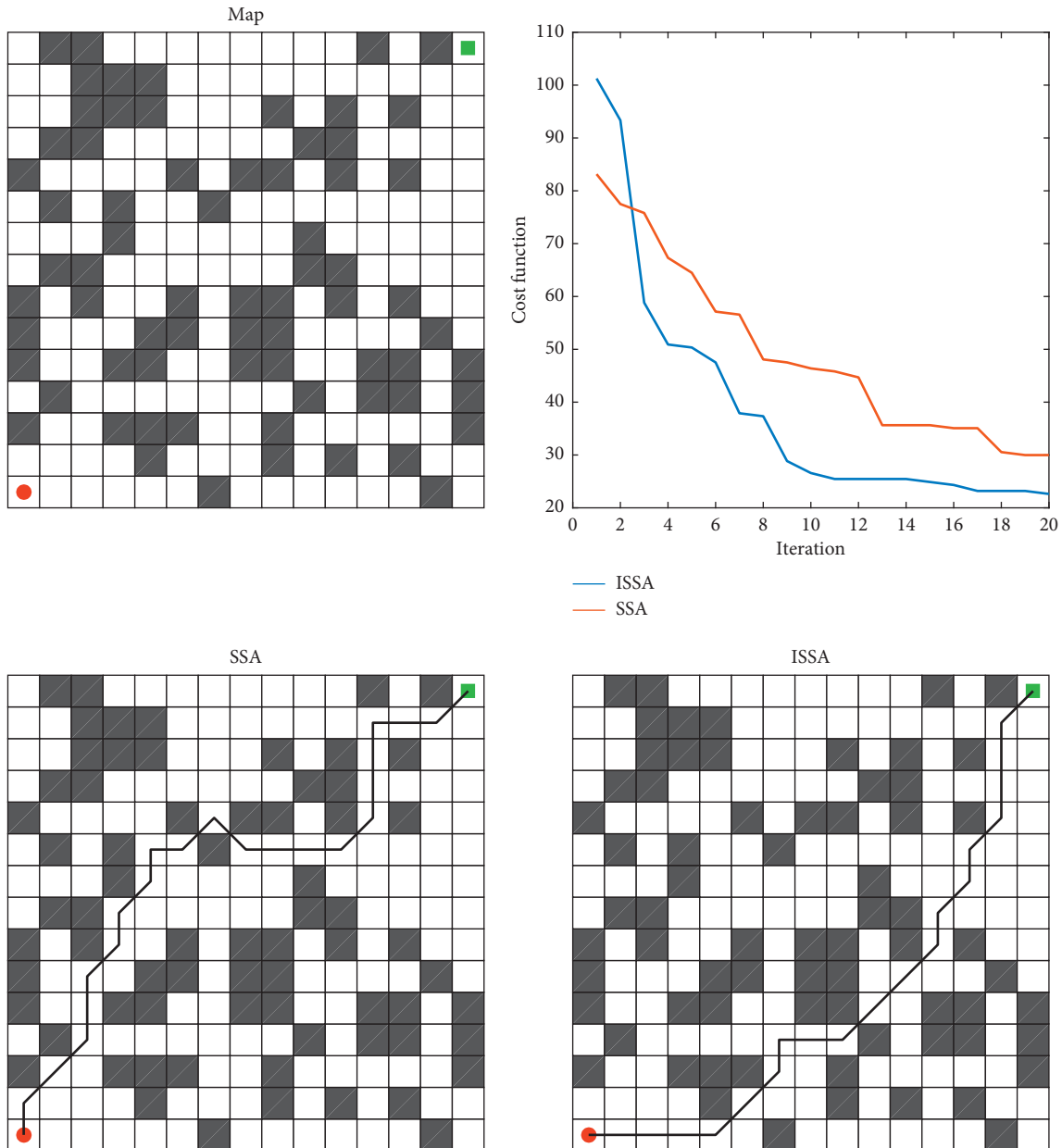


FIGURE 12: Convergence curve and optimal path.

In equation (31), j is the j -th dimension of a sparrow. Set each algorithm at 15×15 , and the best path is shown in Figure 12. In order to eliminate the contingency, each algorithm is tested for 10 times, the optimal value, worst value, average value, and standard deviation of the fitness value of each algorithm are calculated, and these four indexes are

used to measure the stability and feasibility of each algorithm. The optimization statistics of each algorithm are shown in Table 10.

It can be seen from the Table 10 and Figure 12, the minimum cost of ISSA planning is 19.7990, while the minimum cost of SSA is 22.6274. It can be seen that ISSA has

TABLE 10: Robot path planning results.

| Algorithm | Fitness | | | |
|-----------|----------------|----------------|----------------|---------------|
| | Best | Worst | Ave | Std |
| SSA | 22.6274 | 39.5980 | 29.9813 | 6.5115 |
| ISSA | 19.7990 | 28.2843 | 22.6274 | 3.4641 |

strong path planning ability. According to other indicators, it can be seen that the path planned by ISSA has good stability. Therefore, ISSA has a good effect in more complex robot path planning and can plan a more stable and safer path.

8. Conclusions

This paper proposes an improved sparrow search algorithm based on iterative local search strategy, which introduces four strategies: variable helix factor, improved iterative local search, the lens imaging with changing focusing ability, and improved boundary control. ISSA overcomes the shortcomings of poor utilization of current individuals and lack of effective search and effectively improves the problems of falling into local optimal solution and low optimization accuracy. The test function results show that ISSA has good optimization performance and universality. The results of PID parameter tuning and robot path planning show that ISSA algorithm has good practicability.

The improved ISSA has good optimization performance, but it also has some shortcomings. For example, it can only find the optimal value in some functions, and other performance indicators are poor and unstable; a certain amount of work is added, resulting in a longer consumption time of the algorithm; and it did not improve the search scope of discoverers. In view of the shortcomings, we still need to do some work in the future: first, how to improve the stability of the algorithm; second, how to improve the search ability of followers; third, how to balance the time and optimization ability of the algorithm; and fourth, how to improve the search ability of discoverers on the basis of discoverers.

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. 61703411).

References

- [1] W. L. Liu, Y. J. Gong, W. N. Chen, Z. Liu, H. Wang, and J. Zhang, "Coordinated charging scheduling of electric vehicles: a mixed-variable differential evolution approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 12, pp. 5094–5109, 2019.
- [2] S. Zhou, L. Xing, X. Zheng, N. Du, L. Wang, and Q. Zhang, "A self-adaptive differential evolution algorithm for scheduling a single batch-processing machine with arbitrary job sizes and release times," *IEEE Transactions on Cybernetics*, vol. 51, pp. 1430–1442, 2019.
- [3] F. Zhao, R. Ma, and L. Wang, "A self-learning discrete jaya algorithm for multiobjective energy-efficient distributed no-idle flow-shop scheduling problem in heterogeneous factory system," *IEEE Transactions on Cybernetics*, 2021.
- [4] F. Zhao, X. He, and L. Wang, "A two-stage cooperative evolutionary algorithm with problem-specific knowledge for energy-efficient scheduling of no-wait flow-shop problem," *IEEE Transactions on Cybernetics*, vol. 51, pp. 5291–5303, 2020.
- [5] X.-S. Yang, "Firefly algorithms for multimodal optimization," in *Proceedings of the 2009 International Symposium on Stochastic Algorithms*, Sapporo, Japan, 2009.
- [6] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, 2015.
- [7] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.
- [8] S. Mirjalili, "SCA: a sine cosine algorithm for solving optimization problems," *Knowledge-Based Systems*, vol. 96, pp. 120–133, 2016.
- [9] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm," *Computers & Structures*, vol. 169, pp. 1–12, 2016.
- [10] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: algorithm and applications," *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019.
- [11] R.-E. Precup, R.-C. David, R.-C. Roman, E. M. Petriu, and A.-I. Szedlak-Stinean, "Slime mould algorithm-based tuning of cost-effective fuzzy controllers for servo systems," *International Journal of Computational Intelligence Systems*, vol. 14, no. 1, pp. 1042–1052, 2021.
- [12] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, "Hunger games search: visions, conception, implementation, deep analysis, perspectives, and towards performance shifts," *Expert Systems with Applications*, vol. 177, Article ID 114864, 2021.
- [13] I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, and H. Chen, "RUN beyond the metaphor: an efficient optimization algorithm based on Runge Kutta method," *Expert Systems with Applications*, vol. 181, Article ID 115079, 2021.
- [14] J. Tu, H. Chen, M. Wang, and A. H. Gandomi, "The colony predation algorithm," *Journal of Bionic Engineering*, vol. 18, no. 3, pp. 674–710, 2021.
- [15] 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.
- [16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014.

- [17] 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*, vol. 5, Orlando, FL, USA, 1997.
- [18] C. R. Houck, J. Joines, and M. G. Kay, *A Genetic Algorithm for Function Optimization: A Matlab Implementation*, NCSU, Raleigh, NC, USA, 1995.
- [19] Y. Li, S. Wang, Q. Chen, and X. Wang, "Comparative study on some new swarm intelligence optimization algorithms," *Computer Engineering and Application*, vol. 56, no. 22, pp. 1–12, 2020.
- [20] X. Lv, X. Mu, and J. Zhang, "Multi-threshold image segmentation based on improved sparrow search algorithm," *Systems Engineering and Electronics*, vol. 43, no. 2, pp. 318–327, 2021.
- [21] X. Lv, X. Mu, J. Zhang, and Z. Wang, "Chaos sparrow search optimization algorithm," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 8, pp. 1712–1720, 2021.
- [22] A. Tang, T. Han, D. Xu, and L. Xie, "UAV track planning method based on chaotic sparrow search algorithm," *Computer Applications*, vol. 41, no. 7, pp. 2128–2136, 2021.
- [23] C. Ouyang, D. Zhu, and F. Wang, "A learning sparrow search algorithm," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 3946958, 23 pages, 2021.
- [24] D. Zhang, H. Xu, Y. Wang, T. Song, and L. Wang, "Whale optimization algorithm embedded in circle mapping and dimensional keyhole imaging reverse learning," *Control and Decision Making*, vol. 36, no. 5, pp. 1173–1180, 2021.
- [25] G. Wu, R. Mallipeddi, and P. N. Suganthan, *Problem Definitions and Evaluation Criteria for the CEC 2017 Competition on Constrained Real-Parameter Optimization*, National University of Defense Technology, Changsha, China, 2017.
- [26] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Technical report, Zhengzhou University, Zhengzhou, China, 2016.
- [27] Y. Chen, Y. Ma, and W. Yun, "Application of improved genetic algorithm in PID controller parameters optimization," *Telkommnika*, vol. 11, no. 3, pp. 1524–1530, 2013.
- [28] C. Gan, W. Cao, M. Wu, and X. Chen, "A new bat algorithm based on iterative local search and stochastic inertia weight," *Expert Systems with Applications*, vol. 104, pp. 202–212, 2018.
- [29] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, Springer, Berlin, Germany, pp. 65–74, 2010.
- [30] D. Molina, A. LaTorre, and F. Herrera, "SHADE with iterative local search for large-scale global optimization," in *Proceedings of the 2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018.
- [31] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *IEEE Transactions on Systems, Man and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 3, pp. 392–403, 1998.
- [32] X. Xia, J. Liu, and Y. Li, "Particle swarm optimization algorithm with reverse-learning and local-learning behavior," *Journal of Software*, vol. 9, no. 2, pp. 350–357, 2014.
- [33] J. He, Z. Peng, D. Cui, and Q. Li, "A teaching and learning optimization algorithm based on multi-inverse learning," *Engineering Science and Technology*, vol. 51, no. 6, pp. 159–167, 2019.
- [34] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.
- [35] L. Wen, T. Wu, M. Tang, M. Xu, and S. Cai, "Gray wolf optimization algorithm based on lens imaging learning strategy," *Acta Automatica Sinica*, vol. 46, no. 10, pp. 2148–2164, 2020.
- [36] C. Ouyang, D. Zhu, and Y. Qiu, "Lens learning sparrow search algorithm," *Mathematical Problems in Engineering*, vol. 2021, Article ID 9935090, 17 pages, 2021.
- [37] S. Peng, Z. Wu, X. Zhou, and C. Deng, "Improved particle swarm algorithm based on the reverse learning model of refraction principle," *Journal of Electronics*, vol. 43, no. 11, pp. 2137–2144, 2015.
- [38] Q. Liang, B. Chen, H. Wu, C. Ma, and S. Li, "A novel modified sparrow search algorithm with application in side lobe level reduction of linear antenna array," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 9915420, 25 pages, 2021.
- [39] S. He, Z. Cao, and S. Yu, "PID parameter optimization based on pollination algorithm," *Computer Engineering and Applications*, vol. 52, no. 17, pp. 59–62, 2016.