Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 19(11): 11422–11452.
DOI: 10.3934/mbe.2022533
Received: 18 April 2022
Revised: 03 August 2022
Accepted: 04 August 2022
Published: 10 August 2022

Research article

Design and application of improved sparrow search algorithm based on sine cosine and firefly perturbation

Xiangyang Ren[1], Shuai Chen[1], Kunyuan Wang[2,*] and Juan Tan[1]

[1] School of Management Engineering and Business, Hebei University of Engineering, Handan 056038, China
[2] School of Marxism, Hebei University of Engineering, Handan 056038, China

* **Correspondence:** Email: wangkunyuan@hebeu.edu.cn.

Abstract: Swarm intelligence algorithms are relatively simple and highly applicable algorithms, especially for solving optimization problems with high reentrancy, high stochasticity, large scale, multi-objective and multi-constraint characteristics. The sparrow search algorithm (SSA) is a kind of swarm intelligence algorithm with strong search capability, but SSA has the drawback of easily falling into local optimum in the iterative process. Therefore, a sine cosine and firefly perturbed sparrow search algorithm (SFSSA) is proposed for addressing this deficiency. Firstly, the Tent chaos mapping is invoked in the initialization population stage to improve the population diversity; secondly, the positive cosine algorithm incorporating random inertia weights is introduced in the discoverer position update, so as to improve the probability of the algorithm jumping out of the local optimum and speed up the convergence; finally, the firefly perturbation is used to firefly perturb the sparrows, and all sparrows are updated with the optimal sparrows using the firefly perturbation method to improve their search-ability. Thirteen benchmark test functions were chosen to evaluate SFSSA, and the results were compared to those computed by existing swarm intelligence algorithms, as well as the proposed method was submitted to the Wilcoxon rank sum test. Furthermore, the aforesaid methods were evaluated in the CEC 2017 test functions to further validate the optimization efficiency of the algorithm when the optimal solution is not zero. The findings show that SFSSA is more favorable in terms of algorithm performance, and the method's searchability is boosted. Finally, the suggested algorithm is used to the locating problem of emergency material distribution centers to further validate the feasibility and efficacy of SFSSA.

Keywords: sparrow search algorithm; Tent chaotic mapping; positive cosine algorithm; firefly

perturbation strategy; contingency site selection

1. Introduction

Swarm intelligence optimization algorithms have been developing rapidly in the last decades, and have been innovated in solving optimization problems. Meanwhile, the algorithm technology is being updated and iterated with the development [1]. Most of the swarm intelligence optimization algorithms are applied to the algorithms by simulating the characteristics of some animals in nature, through their survival ability and some laws of living habits [2]. For example, the most classical Particle Swarm Optimization (PSO) algorithm was proposed by Eberhart and Kennedy in 1995, and the source of the algorithm was based on the process of simulating the flight of a flock of birds in search of food [3,4]. Ant Colony Optimization (ACO) was proposed by Dorigo (Italy) in 1992 in his Ph.D. thesis and was based on the study of the biological characteristics of ants by leaving pheromones in the process of searching for food [5,6]. PSO and ACO are the most classical intelligent optimization algorithms, there are also Artificial Fish Swarms Algorithm (AFSA) [7], Grey Wolf Optimization Algorithm (GWO) [8], Whale Optimization Algorithm (WOA) [9], Bat Algorithm (BA) [10], Firefly Algorithm (FA) [11], Chimp Optimization Algorithm(ChOA) [12], Sparrow Search Algorithm (SSA) [13], and so on.

SSA is a new swarm intelligence optimization algorithm proposed by Xu et al. [13] in 2020, which is inspired by the foraging and anti-predatory behavior of sparrows, and the authors conducted some comparative experiments to test the effectiveness and performance of the algorithm they proposed, and the simulation results prove that SSA outperforms other existing algorithms in terms of search accuracy, convergence speed, and stability. However, when searching for the global optimal solution, SSA, like other optimization algorithms, suffers from the problem of reduced population diversity and easily falls into the local optimum.

The defect of easily falling into local optimum when searching for the global optimal solution of swarm intelligence optimization algorithms reflects its limitation in searching for optimal solutions. As a result, scholars both domestic and abroad put forward corresponding improvement methods. Duan et al. [14] improved the convergence speed and accuracy by considering the introduction of an extended memory factor into the PSO algorithm and then applied this extended memory-based PSO algorithm to AFSA. Han et al. [15] studied the multi-constrained unmanned helicopter high-quality flight path problem and proposed a method based on integrating group intelligence and brain-like cognitive learning artificial bee swarm algorithm (ABCA) path planning method to improve the traditional ABCA evolutionary method. Dereli et al. [16] improved the WOA algorithm convergence speed and overcame the problem of frequently falling into optimum by improving the equations of WOA adding two phases of search and wrap-around to make the population value better than the individual value. Chaudhary et al. [17] proposed a swarm bat algorithm (BA) for improving the searchability based on the two problems that BA tended to fall into local optimum and the speed of convergence decreased with iteration to the end. Wu et al. [18] provided an adaptive logarithmic spiral-Levy FA (AD-IFA) to solve the problem that LF-FA was underdeveloped locally and did not converge quickly, their work proved that AD-IFA outperformed both standard FA and LF-FA in terms of computational speed and derived optimal values. Xin Lu et al. [19] improved some shortcomings of SSA by introducing the Tent chaos model and Gaussian variation method. Ma et al. [20] applied a large number of variants combined with SSA to verify the performance of SSA variants compared with other intelligent algorithms and proved the effectiveness and stability of their algorithm. Ma et al. [21] studied and proposed an enhanced multi-strategy sparrow search algorithm (EMSSA) with algorithm improvements to incorporate adaptive tent chaos theory and weighted positive cosine algorithm, which in turn improved the population diversity and local optimum problems. Wu et al. [22] solved the travel quotient problem through the improvement of SSA using positive cosine and greedy algorithms and compared it with other intelligent algorithms, which proved that the improved algorithm was feasible. Zhang et al. [23] proposed an improved SSA by three strategies and applied it to mobile robot bionic path planning, results showed that the improved algorithm was more feasible compared with other studies.

SSA is continuously being improved since its introduction, and it is still in the exploration stage. To improve the search ability and convergence accuracy of SSA, this paper continues to explore new improvement strategies based on previous work and proposes an improved SSA that incorporates the positive cosine algorithm with random inertia weights and the firefly perturbation strategy.

The innovation points on the original sparrow search algorithm in this paper can be summarized as follows:

1) Tent chaos mapping is invoked in the initial population to make the population distribution more uniform and improve the population diversity.

2) A positive cosine approach with random inertia weights is proposed to update the discoverer's position. The strategy successfully reduces the likelihood that the algorithm would enter a local optimum solution and improves convergence accuracy thanks to the improved discoverer position.

3) Add firefly perturbation to update the position of all sparrows, update the optimal position, improve the search performance of the algorithm, compare the sparrows after perturbation with those before perturbation, and select the better position if it is better to update the sparrow position.

The remaining of this paper is structured as follows. In the second part, the mathematical model of the original sparrow search algorithm is described. In the third part, the improvement points of this paper are analyzed and the improved SFSSA algorithm is proposed. In the fourth part, the improved algorithm is compared with the other four algorithms, and thirteen benchmark test functions are chosen to prove the advantage of SFSSA in terms of algorithm performance. In the fifth part, to further illustrate the algorithm's capacity to identify the ideal value when the optimal value is not zero, the CEC2017 test function is used to compare the method. In the sixth part, the SFSSA algorithm is applied to the emergency material location problem, and the running results further prove the feasibility and effectiveness of the SFSSA algorithm. In the seventh part, the work of this paper is briefly summarized and some directions are made for future research work.

2. Sparrow search algorithm

2.1. Predatory and anti-predatory behavior of sparrows

In nature, most animals are predators, but they are also been predated. Under the conditions of natural selection, animals will form a set of defensive measures to prevent anti-predatory behavior during foraging, and sparrows have such typical characteristics in the foraging process. Sparrows are flock animals, in each foraging process, each sparrow will have a clear division of labor, and sparrows in the foraging process has a keen perception and scouting ability, sparrows responsible for finding food can provide search lines and directions to their companions to help quickly find food, sparrows responsible for scouting, after the discovery of natural enemies, will promptly send a signal to their companions to quickly leave to avoid being predated.

2.2. Description of the sparrow search algorithm

Sparrows are born with high alertness and keen perception of surroundings, sparrows are divided into discoverer, joiner, and scouter, the sparrow search algorithm is introduced based on the predatory characteristics of sparrows.

Suppose there is a population of *N* sparrows in a *D*-dimensional space, and the initial location of this population is as follows:

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,d} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,d} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n,1} & X_{n,2} & \cdots & X_{n,d} \end{bmatrix} \tag{1}$$

where *d* represents the dimension of the problem variable, and *n* is the total number of sparrows.

Sparrow adaptation values are as follows:

$$F_X = \begin{bmatrix} f(X_{1,1} & X_{1,2} & \cdots & X_{1,d}) \\ f(X_{2,1} & X_{2,2} & \cdots & X_{2,d}) \\ \vdots & \vdots & \ddots & \vdots \\ f(X_{n,1} & X_{n,2} & \cdots & X_{n,d}) \end{bmatrix} \tag{2}$$

where *f* represents the adaptation value.

In the process of each iteration, some sparrows with high fitness are selected as discoverers in the population, with the percentage of generally 10 to 20%, and the location of discoverers is updated with the following equation:

$$X_{i,j}^{t+1} = \begin{cases} X_{i,j}^{t} \cdot \exp\left(-\frac{i}{\alpha \cdot I}\right) & , R_2 < ST \\ X_{i,j}^{t} + Q \cdot L & , R_2 \geq ST \end{cases} \tag{3}$$

where t represents the number of current iterations, $j = 1,2,3 \cdots d$. I denotes the maximum number of iterations. $X_{i,j}^{t+1}$ indicates the position of the i^{th} sparrow in the j^{th} dimension. $\alpha \in (0,1)$ is a random number. R_2 is the alarm value, $R_2 \in [0,1]$. ST is the safety value, $ST \in [0.5,1]$. Q is a random number that obeys the standard normal distribution. L is a $1 \times d$ matrix and the elements in the matrix are all 1. When $R_2 < ST$, means that the surrounding environment is safe without natural enemies, and the discoverer will conduct an extensive search. When $R_2 \geq ST$, it means that natural enemies appear, and the scouter needs to send an early warning signal to the population, at this point all sparrows need to fly to other safe places to search for food.

The position of the joiners is updated with the following equation:

$$X_{i,j}^{t+1} = \begin{cases} Q \cdot \exp\left(\frac{X_w^t - X_{i,j}^t}{i^2}\right) & , i > \frac{n}{2} \\ X_p^{t+1} + \left|X_{i,j}^t - X_P^{t+1}\right| \cdot B^{+} \cdot L & , \text{otherwise} \end{cases} \tag{4}$$

where X_w^t denotes the global worst position for the t^{th} iteration. X_p^{t+1} indicates the optimal position of the discoverer in the $(t+1)^{th}$ iteration. B represents a $1 \times d$ matrix, elements in this matrix are randomly assigned 1 or -1, and $B^+ = B^T(BB^T)^{-1}$. When $i > n/2$, the i^{th} joiner is in a poor position and therefore in a very hungry state, then the joiner needs to fly to other places to search for food.

The proportion of scouts is 10 to 20% of the sparrow population, and the location of scouts is updated with the following equation:

$$X_{i,j}^{t+1} = \begin{cases} X_b^t + \beta \cdot \left|X_{i,j}^t - X_{\mathrm{b}}^t\right| & , f_i > f_g \\ X_{i,j}^t + K \cdot \left(\frac{\left|X_{i,j}^t - X_w^t\right|}{(f_i - f_w) + \varepsilon}\right) & , f_i = f_g \end{cases} \tag{5}$$

where X_w^t is the current global optimal position. β is the step control parameter, it is a random number that obeys $N(0,1)$. K is a random number and $K \in [1, -1]$. f_i indicates the current individual fitness value, f_g is the current global optimal fitness value, f_w is the current global worst fitness value. ε is the minimum constant to avoid a denominator of 0. When $f_i > f_g$, it means that the sparrow is at the edge of population danger and is extremely vulnerable to being attacked by natural enemies. When $f_i = f_g$, it means that the scout is aware of the danger and needs to move closer to other joiners to avoid being predated by natural enemies, where K is the direction in which the scout moves.

3. Improved sparrow search algorithm

3.1. Initializing population-based on tent chaotic mapping

Chaos is a sequence of randomness generated by a simple deterministic system. Chaotic mapping plays a more prominent role in stochastic optimization algorithms, where their properties allow swarm intelligence algorithms to avoid falling into local optima due to the ergodic and semi-random nature of chaos. Chaotic sequences can influence the optimization results of the whole population during population initialization, selection, crossover, and mutation [24]. The common chaotic sequences are Logistic mapping, Tent mapping, Sine mapping, Circle mapping, Singer mapping, and Chebyshev mapping et al. [25]. In the process of initializing the population for the swarm intelligence algorithm, Tent chaotic sequences and logistic mapping are frequently used. By utilizing the properties of chaotic sequences to improve population diversity and make the population distribution more uniform, Tent chaotic sequences also maximize the avoidance of reducing the population later on when searching for the local optimum problem probability [26].The Tent mapping has merits of uniformity in population distribution and better searchability than the Logistic mapping [27].

Tent mapping is a segmented linear mapping function [28]. It is called a "Tent" because the shape of the mapping morphology is similar to a tent, as shown in Figure 1 [29].

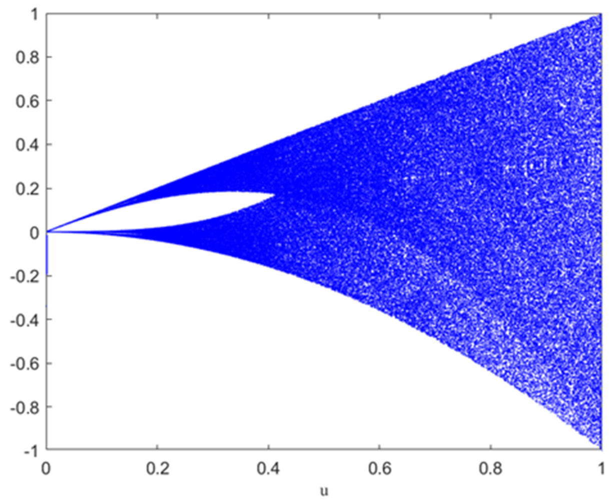

Figure 1. Tent chaotic mapping bifurcation diagram.

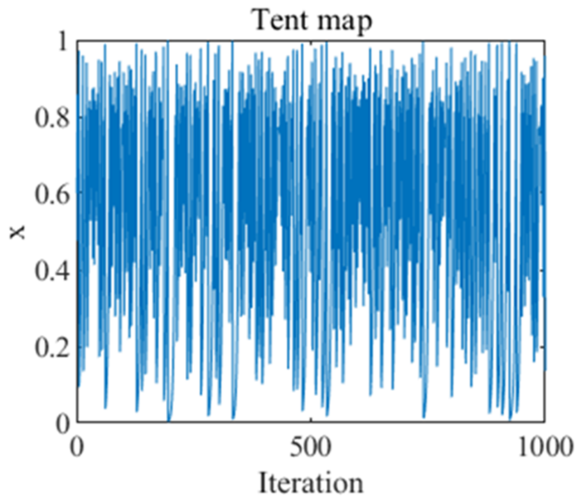

Figure 2. Distribution of Tent chaotic mapping.

The structure of Tent mapping is simple, and the distribution density of the results is relatively uniform with good ergodicity [30], the equation is displayed below.

$$x_{n+1} = \begin{cases} 2x_n & , \ 0 \le x_n < 0.5 \\ 2(1 - x_n) & , \ 0.5 \le x_n \le 1 \end{cases} \tag{6}$$

Equation (6) is the expression when parameter a is set to be 0.5. The original mathematical expression of the Tent mapping is as follows [31].

$$x_{n+1} = \begin{cases} \frac{x_n}{a} & , \ (0 < x_n < a) \\ \frac{1-x_n}{1-a} & , \ (0 \le x_n \le 1) \end{cases} \tag{7}$$

where the system is in a chaotic state when $a \in (0,1)$ and $x_n \in (0,1)$. It can be seen from its expression that the Tent algorithm involves fewer parameters and is relatively simple to operate. Here, the coefficient $a = 0.7$ and the initial value $x = 0.6$, the Tent mapping is iterated 1000 times, and the distribution of the values is in the interval [0,1], as shown in Figure 2 [32]. Figure 2, clearly demonstrates that the Tent mapping is distributed more uniformly, and the values are taken in the interval (0,1).

3.2. Sine cosine algorithm

Sine Cosine Algorithm (SCA) is a swarm intelligence optimization algorithm proposed by Seyedali Mirjalili, an Australian scholar, in 2016 [33]. It uses the oscillatory variation property feature of the sine cosine function to make the solution converge to the global optimal position. Sparrows' positions are constantly updated during the process of seeking food, and when the discoverer finds the optimal position, it attracts a large number of follower sparrows to concentrate together, increasing the probability that the population falls into a local solution. Therefore, based on the sine cosine algorithm, random inertia weights [34] are introduced as shown in Eq (8) to update the discoverer position, and then the new discoverer position update formula is obtained as shown in Eq (9). The introduction of the random inertia weights can balance the global convergence ability and facilitate the algorithm to find the global optimal solution, and the random variables can be used to adjust the random weights to induce the algorithm to avoid local optima and improve the stability of the searching results.

$$\omega = \omega_{min} + (\omega_{max} - \omega_{min}) \sin\left(\frac{t\pi}{iter_{max}}\right) \tag{8}$$

where, ω_{max} is the maximum value of the random inertia weight. ω_{min} is the minimum value of the random inertia weight.

$$X_{i,j}^{t+1} = \begin{cases} (1-\omega) \cdot X_{i,j}^{t} + \omega \cdot \sin r_0 \cdot \left| r_1 \cdot X_{\text{best}} - X_{i,j}^{t} \right|, & R_2 < ST \\ (1-\omega) \cdot X_{i,j}^{t} + \omega \cdot \cos r_0 \cdot \left| r_1 \cdot X_{\text{best}} - X_{i,j}^{t} \right|, & R_2 \geqslant \mathrm{ST} \end{cases} \tag{9}$$

where $r_0 \in (0,2\pi)$ for random numbers. $r_1 \in (0,2\pi)$ for random numbers.

3.3. Firefly perturbation strategy

The Firefly Algorithm (FA) was proposed by Professor Xin-She Yang at the University of Cambridge [35]. The origin of the FA algorithm comes from the luminous properties of fireflies to attract the opposite sex, and the luminous biological properties can warn potential predators [36]. Such swarm-living animals as bees and ants, for example, can communicate among their group members, which is the main reason why group intelligence is characterized by organization and decentralized decision-making. Fireflies attract each other through their luminescence system, and the strength of luminescence becomes a signal to attract other fireflies so that their luminescence weakness will move towards the side with strong luminescence [37]. Therefore, based on the change in light intensity of fireflies and the attraction design formula, the assumptions are listed below.

1) Fireflies not only attract the opposite sex but also move toward the side of the stronger luminous fireflies.

2) The attraction of fireflies is proportional to their brightness, for any two fireflies, one will move towards the other brighter than it, however, the brightness is decreasing with the increase of distance.

3) If no one brighter than the given firefly is found, it will move randomly.

The equation for the degree of luminescence of fireflies is as follows:

$$I = I_0 . e^{-\gamma r_{i,j}^2} \tag{10}$$

where I_0 is the intensity of the maximum light source of fireflies, and related to the target value of the function, the better the target value of the function, the greater the intensity of firefly light. γ is the light absorption coefficient, firefly light degree will change with the distance and propagation medium. $r_{i,j}^2$ is the spatial distance between firefly i and firefly j.

The equation for the attraction of fireflies is as follows:

$$\beta = \beta_0 . e^{-\gamma r_{i,j}^2} \tag{11}$$

where β_0 is the maximum attraction strength.

The equation for updating the position of firefly i attracted to firefly j is shown in Eq (12):

$$f_i = f_i + \beta . (f_j - f_i) + \alpha . (\text{rand} - 0.5) \tag{12}$$

where f_i, f_j are the spatial locations of fireflies i and j respectively, $\alpha \in [0,1]$ is the step size factor, a rand is a random number on $[0,1]$ that obeys uniform distribution.

3.4. Flow Chart of SFSSA

Through the above analysis, it is found that the sparrow search algorithm has strong local search ability compared with other swarm intelligence algorithms, but there are still similar shortcomings as other intelligent algorithms, such as fast convergence speed and ease to fall into local optimum. Therefore, this paper proposes corresponding improvement measures: 1) the Tent chaotic map is used to improve the population diversity, 2) the improved sine and cosine algorithm is used to improve the searchability of the discoverer sparrow, 3) the sparrow position is updated by using the firefly disturbance, which can make the sparrow population jump out of the local optimum. The flow chart of SFSSA is shown in Figure 3, and the pseudo code of SFSSA is shown in Algorithm 1 below, and the specific improved process is as follows.

Step 1. Initialize parameters. Set population size *N*, the maximum number of iterations, discoverer *PD*, scout *SD*, warning value R_2, safety value ST, etc.

Step 2. Initialize population. The population is initialized using the Tent chaos mapping in Eq (6).

Step 3. Calculate the fitness value f_i of each sparrow and rank.

Step 4. According to the *PD* ratio, some sparrows with superior fitness values are selected as discoverers, and the discoverer positions are updated according to Eq (3).

Step 5. A positive cosine algorithm incorporating random inertia weights is introduced at the discoverer location to select the one with the best fitness value as the discoverer, and the discoverer location is updated according to Eq (9).

Step 6. The remaining populations are the joiners and the joiner positions are updated according to Eq (4).

Step 7. Some individuals in the population are randomly selected as scouts according to the proportion, the scout sparrow position is updated according to Eq (5), the new fitness value is calculated, and the update operation is performed if it is better than the current optimal value.

Step 8. The optimal position of the population is updated by adding firefly perturbation, and the intensity magnitude of the light source I_0 is used as the superiority of the fitness value, and the direction of movement of the population is determined according to the degree of firefly luminescence and attractiveness in Eqs (10) and (11).

Step 9. A perturbation strategy is applied to the population according to Eq (12) to update the location.

Step 10. Calculate the fitness value and determine the optimal position of the population.

Step 11. Observe whether the stop condition is satisfied, exit if it is satisfied, and output the result, otherwise, repeat from Step 3.

Algorithm 1. The framework of the SFSSA.

Input :
M = the maximum number of iterations
PD = the proportion of discoverer
SD: the proportion of scouts
R_2: the alarm value
N: the population sparrows
Output: X_{best}, f_g
Initializing populations with Tent chaos mapping
1: **while** (t <M)
2: Calculate the fitness value for each sparrow and rank them
3: R_2=$rand$(1)
4: **for** i=1:PD
5: Select the optimal discoverer according to Eq. (3)
6: Update the discoverer according to the improved sine-cosine Eq. (9)
7: **end for**
8: **for** i=(PD+1):N
9: Update accessions by Eq. (4)
10: **end for**
11: **for** l=1:SD
12: Update scouts by Eq. (5)
13: Using firefly luminescence and attraction strength to determine the direction of population movement by Eq.(10) and Eq.(11)
14: Update of population location by means of firefly disturbance by Eq.(12)
15: **end for**
16: Calculate the fitness value and determine the optimal position;
17: t=t+1
18: **end while**
19: **return**: X_{best}, f_g

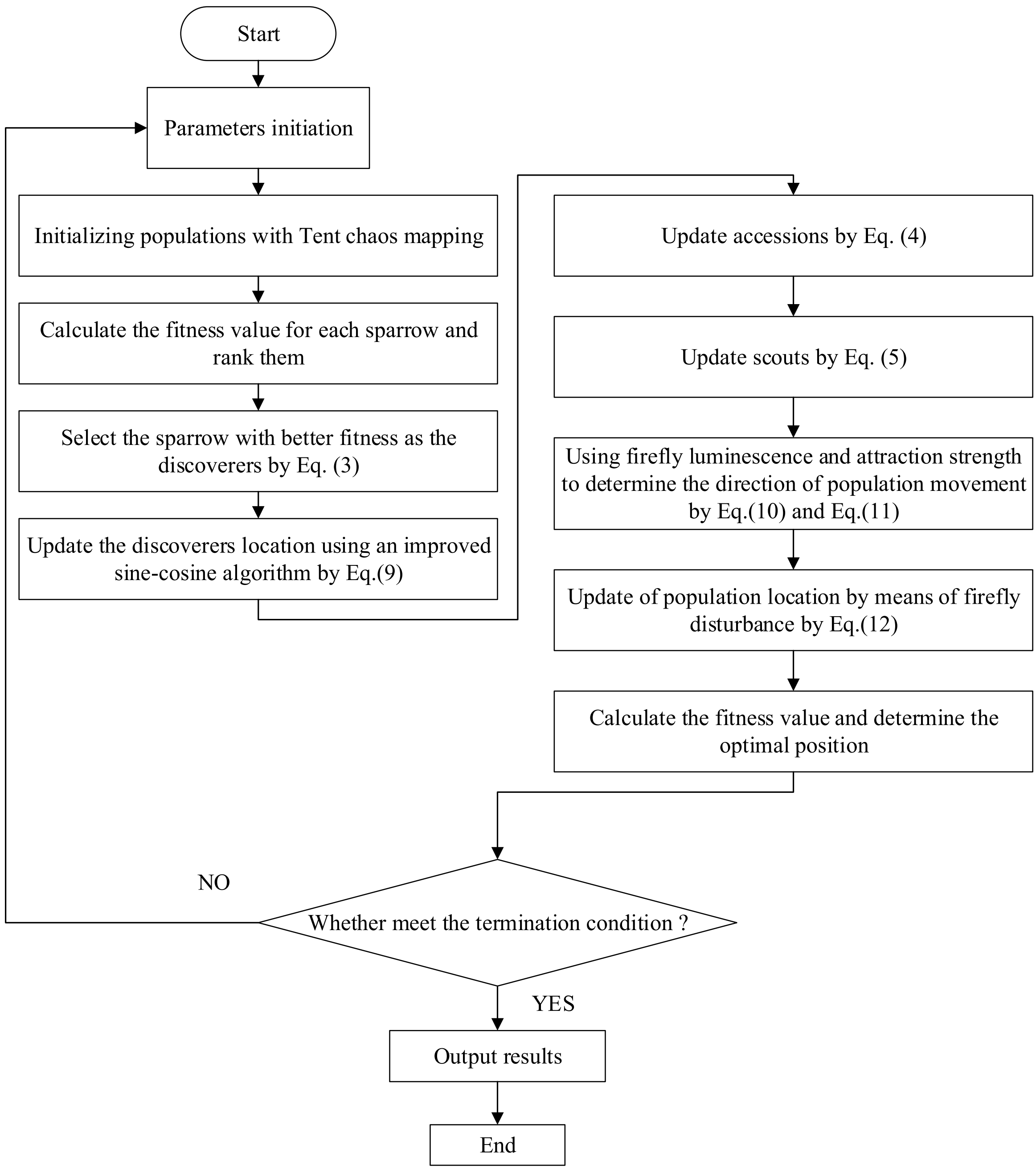

Figure 3. SFSSA flow chart.

4. Simulation experiments and results analysis

4.1. Experimental environment

The simulation experiment environment is 11thGen Intel(R) Core(TM) i5-11400H@2.70GHz, 16 GB memory, Windows 11 operating system MATLAB R2020b simulation experiment operating platform.

4.2. Test object and parameter setting

For the verification of the effectiveness of SFSSA, WOA, PSO, ChOA, SSA, LSSA and RWSSA are selected and compared with it in this paper. To ensure the fairness of the validation, the population size N of the algorithm is set to be 30 and the maximum number of iterations is set to be 500. Each experimental parameter is shown in Table 1.

Table 1. Experimental parameters of the test algorithm.

Algorithm name	Parameters setting
WOA	$b = 1, a_{max} = 2, a_{min} = 0$
PSO	$w = 0.8, c_1 = 1.49445, c_2 = 1.49445$
ChOA	$m = chaos\ (3, 1, 1)$
SSA	$PD = 0.2, SD = 0.1, ST = 0.8$
LSSA	PD = 0.2, SD = 0.1, ST = 0.8
RWSSA	PD = 0.2, SD = 0.1, ST = 0.8
SFSSA	$PD = 0.2, SD = 0.1, ST = 0.8$

Table 2. Unimodal test functions (dim = 30).

Function	Dimension	Variable range
$F_1(x) = \sum_{i=1}^{n} x_i^2$	30	[-100, 100]
$F_2(\mathrm{x}) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	[-10, 10]
$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30	[-100, 100]
$F_4 = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	[-100, 100]
$F_5(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2 \right]$	30	[-30, 30]
$F_6(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2$	30	[-100, 100]
$F_7(x) = \sum_{i=1}^{n} i x_i^4 + \mathrm{random}[0,1)$	30	[-1.28, 1.28]

4.3. Thirteen benchmark test functions

This paper selects 13 standard representative benchmark functions for testing [38], including 7 unimodal test functions. Unimodal test functions usually only have one global optimal solution, thus the comparison is conducted in terms of the search ability and convergence speed, as shown in Table 2. There are 4 multimodal test functions with local optimal solutions in the function definition domain, which are easily falling into local optimum. Multimodal function testing can compare the algorithms in terms of the global search ability and local optimum ability without falling into local optimum, as shown in Table 3. There are 2 low-dimensional dimensional multimodal test functions. Swarm intelligence algorithms need to be constantly explored and developed, and low-dimensional

dimensional multimodal test functions are used to compare the ability of each algorithm in global exploration and local development, as shown in Table 4.

Table 3. Multimodal test functions (dim = 30).

Function	Dimension	Variable range
$f_8(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$	30	[-5.12, 5.12]
$f_9(x) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n}cos(2\pi x_i)\right) + 20 + e$	30	[-32, 32]
$f_{10}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n}u(x_i, 10,100,4), y_i = 1 + \frac{x_i+1}{4}, u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a < x_i < a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	30	[-50, 50]
$f_{11}(X) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n}u(x_i, 5,100,4)$	30	[-50, 50]

Table 4. Low-dimensional multimodal test functions (dim = 4).

Function	Dimension	Variable range
$F_{12}(X) = -\sum_{i=1}^{5}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]
$F_{13}(X) = -\sum_{i=1}^{7}[(X - a_i)(X - a_i)^T + c_i]^{-1}$	4	[0, 10]

4.4. Comparative analysis of algorithm results

WOA, PSO, ChOA, SSA, LSSA and RWSSA are selected to compare the effectiveness with SFSSA, run each algorithm 30 times respectively on 13 benchmark test functions, and the results are shown in Tables 5–7. From Tables 5–7, the following conclusions can be drawn. 1) For the high-dimensional unimodal test functions F1–F7, the SFSSA algorithm proposed in this paper significantly outperforms WOA, PSO, ChOA, SSA, LSSA and RWSSA among functions F1–F4, the SFSSA effect for searching optimal solutions reach 100%, while in test functions F5–F7, SFSSA does not obtain an optimal solution, but the test results are also better than other algorithms. 2) For the high-dimensional multimodal test functions F8–F11, SFSSA also outperforms the other algorithms. There is no discernible difference between SFSSA and other algorithms in function F8, with the exception of PSO and ChOA, all other algorithms achieving 100% of the optimal solution search rate. The good ability of SFSSA to search for optimal solutions is also demonstrated in functions F10 and F11. 3) For the low-dimensional fixed-dimensional multimodal test F12 and F13, SFSSA also outperforms the other algorithms in terms of the optimal value and standard deviation in F12 and F13. Combining the above results, it can be concluded that SFSSA is significantly better than the other algorithms in terms of optimality and stability as well as robustness.

Table 5. Results of unimodal test functions.

Function	Algorithm	Best	Worst	Avg	Std
F1	WOA	1.7749E-87	5.4505E-71	2.8436E-72	1.1136E-71
	PSO	2.7128E+02	1.8840E+03	1.0465E+03	4.6428E+02
	ChOA	2.0480E-09	8.5882E-05	6.4504E-06	1.5884E-05
	SSA	7.6758E-121	4.5686E-30	1.7278E-31	8.3764E-31
	LSSA	2.3267E-234	1.1261E-33	4.2660E-35	2.0531E-34
	RWSSA	0	1.1697E-30	4.0817E-32	2.1344E-31
	SFSSA	**0**	**0**	**0**	**0**
F2	WOA	2.6978E-59	6.3675E-51	4.1251E-52	1.2152E-51
	PSO	8.1241E+00	9.2827E+01	2.4546E+01	1.5211E+01
	ChOA	9.8120E-08	1.2895E-05	3.2730E-05	4.0441E-05
	SSA	2.0318E-63	1.1192E-28	5.2543E-30	2.1742E-29
	LSSA	1.2111E-186	1.6368E-29	5.5224E-31	2.9873E-30
	RWSSA	1.2478E-69	1.0912E-31	5.3244E-33	2.0656E-32
	SFSSA	**0**	**0**	**0**	**0**
F3	WOA	3.6622E+04	6.1040E+04	3.7680E+04	7.6248E+03
	PSO	9.5324E+02	1.0325E+04	3.3717E+03	2.1707E+03
	ChOA	4.5825E-01	2.0687E+03	1.7769E+02	4.2436E+02
	SSA	9.6040E-87	4.916E-13	3.4494E-14	1.2218E-13
	LSSA	1.0007E-313	2.4867E-10	8.2998E-12	4.5399E-11
	RWSSA	1.6146E-317	9.0559E-15	3.0439E-16	1.6529E-15
	SFSSA	**0**	**0**	**0**	**0**
F4	WOA	1.9574E+00	6.9312E+01	4.0668E+01	2.1106E+01
	PSO	1.2821E+01	2.5360E+01	1.8390E+01	3.8558E+00
	ChOA	2.8057E-02	1.6593E+00	2.7966E-01	3.5611E-01
	SSA	1.8578E-98	1.5608E-06	5.7496E-08	2.8502E-07
	LSSA	1.8901E-89	2.7429E-08	1.6842E-09	5.7065E-09
	RWSSA	1.5218E-61	7.5268E-07	2.5687E-08	1.3732E-07
	SFSSA	**0**	**0**	**0**	**0**
F5	WOA	2.6921E+01	2.8772E+01	2.8044E+01	5.3089E-01
	PSO	5.5753E+03	1.8946E+05	6.9931E+04	4.9913E+04
	ChOA	2.8085E+01	2.8980E+01	2.8846E+01	2.0503E-01
	SSA	7.1503E-08	3.3728E-03	9.0752E-04	1.0188E-03
	LSSA	1.0111E-07	1.1453E-02	1.1441E-03	2.5165E-03
	RWSSA	2.5429E-08	1.0926E-02	1.2258E-03	2.6036E-03
	SFSSA	**1.2671E-09**	**1.4275E-05**	**2.5416E-06**	**3.642E-06**
F6	WOA	9.2325E-02	9.4311E-01	3.0824E-01	1.8077E-01
	PSO	2.5333E+02	2.2256E+03	1.1057E+03	5.2198E+02
	ChOA	2.2836E+00	4.4825E+00	3.5936E+00	4.2266E-01
	SSA	3.7036E-08	5.7814E-05	6.1101E-06	1.1347E-05
	LSSA	1.3873E-10	2.4378E-04	1.6613E-05	4.6093E-05
	RWSSA	2.1385E-10	4.302E-05	6.0216E-06	1.0220E-05
	SFSSA	**3.2503E-11**	**1.4584E-08**	**9.4398E-10**	**2.6456E-09**
F7	WOA	6.1961E-05	2.2059E-02	4.2440E-03	5.0519E-03
	PSO	2.1333E-01	1.1259E+01	1.6896E+00	2.4348E+00
	ChOA	1.1742E-04	9.3132E-03	2.0413E-03	1.8312E-03
	SSA	2.283E-05	1.0886E-03	4.1807E-04	2.6245E-04
	LSSA	3.2584E-05	1.8505E-03	4.5828E-04	4.3209E-04
	RWSSA	2.2467E-05	4.5372E-03	5.9288E-04	8.3595E-04
	SFSSA	**5.0165E-06**	**9.2023E-04**	**2.3859E-04**	**2.3131E-04**

Table 6. The caption of the table.

Function	Algorithm	Best	Worst	Avg	Std
F8	WOA	**0**	**0**	**0**	**0**
	PSO	8.2984E+01	1.7219E+02	1.2930E+02	2.4192E+01
	ChOA	**4.6670E-02**	**3.6937E+01**	**1.2391E+01**	**1.1471E+01**
	SSA	**0**	**0**	**0**	**0**
	LSSA	**0**	**0**	**0**	**0**
	RWSSA	**0**	**0**	**0**	**0**
	SFSSA	**0**	**0**	**0**	**0**
F9	WOA	8.8818E-16	7.9936E-15	4.204E-15	2.4567E-15
	PSO	4.9078E+00	1.2835E+01	9.3752E+00	1.6889E+00
	ChOA	1.9959E+01	1.9964E+01	1.9962E+01	1.2380E-03
	SSA	8.8818E-16	7.9936E-15	1.9540E-15	2.1173E-15
	LSSA	8.8818E-16	7.9936E-15	2.1908E-15	1.9755E-15
	RWSSA	8.8818E-16	7.9936E-15	1.5987E-15	1.7203E-15
	SFSSA	**8.8818E-16**	**8.8818E-16**	**8.8818E-16**	**0**
F10	WOA	4.3889E-03	7.1205E-02	2.3959E-02	1.5600E-02
	PSO	4.9164E+00	2.6997E+01	1.3402E+01	5.5304E+00
	ChOA	2.6292E-01	9.7268E-01	5.1688E-01	2.0685E-01
	SSA	1.0417E-11	1.4721E-05	1.0489E-06	2.7427E-06
	LSSA	8.5662E-12	3.1387E-05	1.5144E-06	5.7386E-06
	RWSSA	1.6231E-11	7.9965E-06	5.7095E-07	1.4951E-06
	SFSSA	**7.7712E-12**	**4.607E-08**	**5.9028E-09**	**1.1061E-08**
F11	WOA	2.1753E-01	1.2340E+00	6.2339E-01	2.6953E-01
	PSO	2.9173E+01	1.3342E+04	1.1008E+03	2.7063E+03
	ChOA	2.3524E+00	2.9948E+00	2.7364E+00	1.3110E-01
	SSA	1.8261E-08	4.5469E-05	6.5337E-06	1.0626E-05
	LSSA	4.6774E-11	1.0495E-04	1.1207E-05	2.5139E-05
	RWSSA	2.7357E-09	5.3919E-05	5.7949E-06	1.2426E-05
	SFSSA	**6.7648E-11**	**5.7616E-07**	**7.7631E-08**	**1.3335E-07**

Table 7. The caption of the table.

Function	Algorithm	Best	Worst	Avg	Std
F12	WOA	-1.0151E+01	-5.0538E+00	-7.1782E+00	2.4924E+00
	PSO	-1.0153E+01	-2.6305E+00	-7.1423E+00	3.3781E+00
	ChOA	-5.0191E+00	-4.9820E-01	-2.0675E+00	2.0898E+00
	SSA	-1.0153E+01	-5.0552E+00	-6.9433E+00	2.4862E+00
	LSSA	-1.0153E+01	-5.0552E+00	-7.0944E+00	2.5402E+00
	RWSSA	-1.0153E+01	-5.0552E+00	-6.7545E+00	2.4443E+00
	SFSSA	**-1.0153E+01**	**-6.3614E+00**	**-9.8484E+00**	**8.8653E-01**
F13	WOA	-1.0402E+01	-2.7659E+00	-9.5333E+00	2.2781E+00
	PSO	-1.0402E+01	-5.1288E+00	-1.0227E+01	9.6291E-01
	ChOA	-1.0402E+01	-1.0402E+01	-1.0402E+01	1.1544E-04
	SSA	-1.0402E+01	-5.0877E+00	-9.5171E+00	2.0147E+00
	LSSA	-1.0402E+01	-1.0402E+01	-1.0402E+01	3.9712E-09
	RWSSA	-1.0402E+01	-5.0877E+00	-1.0225E+01	9.7043E-01
	SFSSA	**-1.0402E+01**	**-1.0402E+01**	**-1.0402E+01**	**4.3584E-11**

4.5. Algorithm convergence curve analysis

To show more intuitively the convergence accuracy and optimization search capability of the algorithm, the convergence curves of these 13 functions based on the number of iterations and adaptation values are presented in Figures 4–6. As shown in Figures 4–6, the convergence curves show that SFSSA outperforms other algorithms in unimodal, multimodal, and fixed-dimensional multimodal conditions. For example, in the unimodal test functions F1–F4, SFSSA has a great improvement in the early convergence speed compared with the other six algorithms, with the fastest convergence speed and the best convergence accuracy. In the unimodal test functions F5–F7, the convergence speed of SFSSA is not as fast as the F1–F4 test functions, but the convergence effect is better than the other six algorithms. In the multimodal test function F8, the convergence accuracy of the other four algorithms, except PSO and ChOA, is also good, but the convergence speed of SFSSA is faster and better in comparison. In comparison to other algorithms, the test results of SFSSA have reasonably stable search ability in multimodal test function F9, proving that this approach is superior to others, as shown in Figure 5. In the multimodal test functions F10 and F11, the SFSSA algorithm significantly outperforms the other algorithms in terms of optimal value, mean value, and standard deviation. In the fixed-dimensional multimodal test functions F12 and F13, it can be seen from the convergence plots that SFSSA converges faster compared to the other algorithms, indicating that SFSSA outperforms the other algorithms in terms of local exploitation and global search ability.

In summary, the aforementioned results aim to demonstrate that SFSSA has better global search ability and does not easily fall into local optimum, and that under the same conditions, the global search and local exploitation ability of SFSSA has relative advantages compared with the other six algorithms described in the paper.

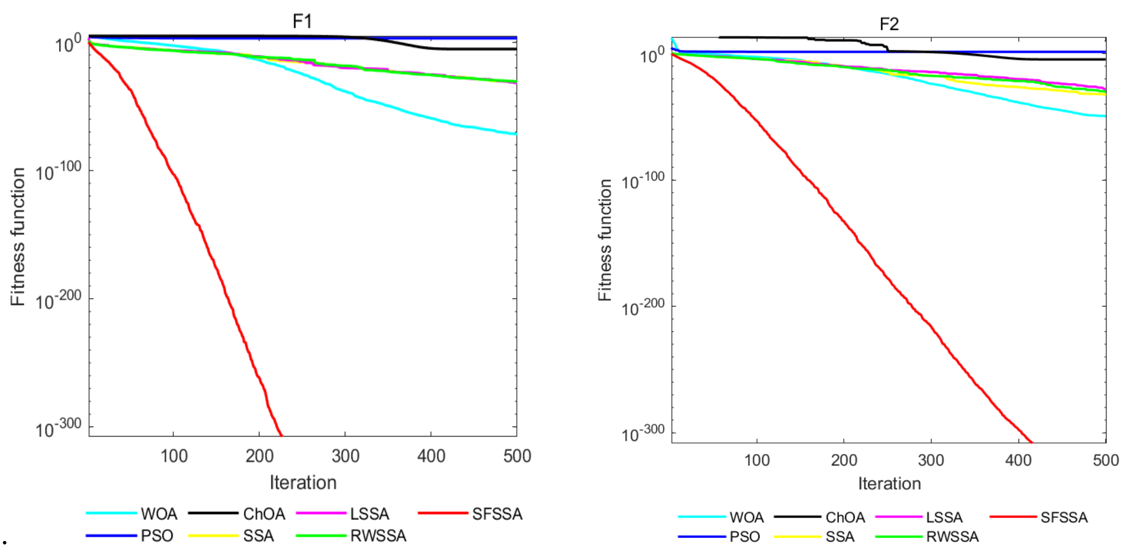

Figure 4. Convergence curves of the seven unimodal test functions.

Continued on next page

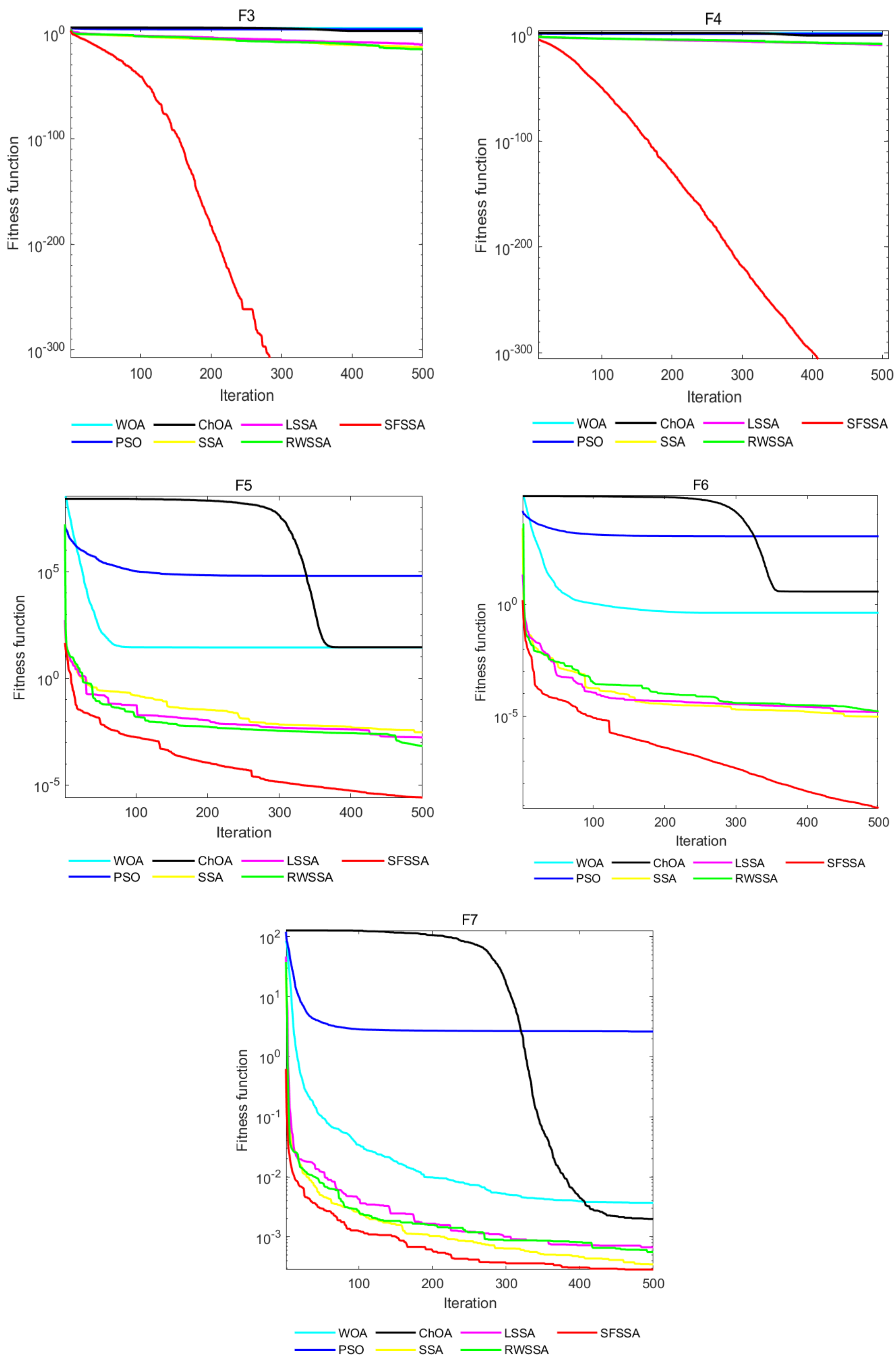

Figure 4. Convergence curves of the seven unimodal test functions.

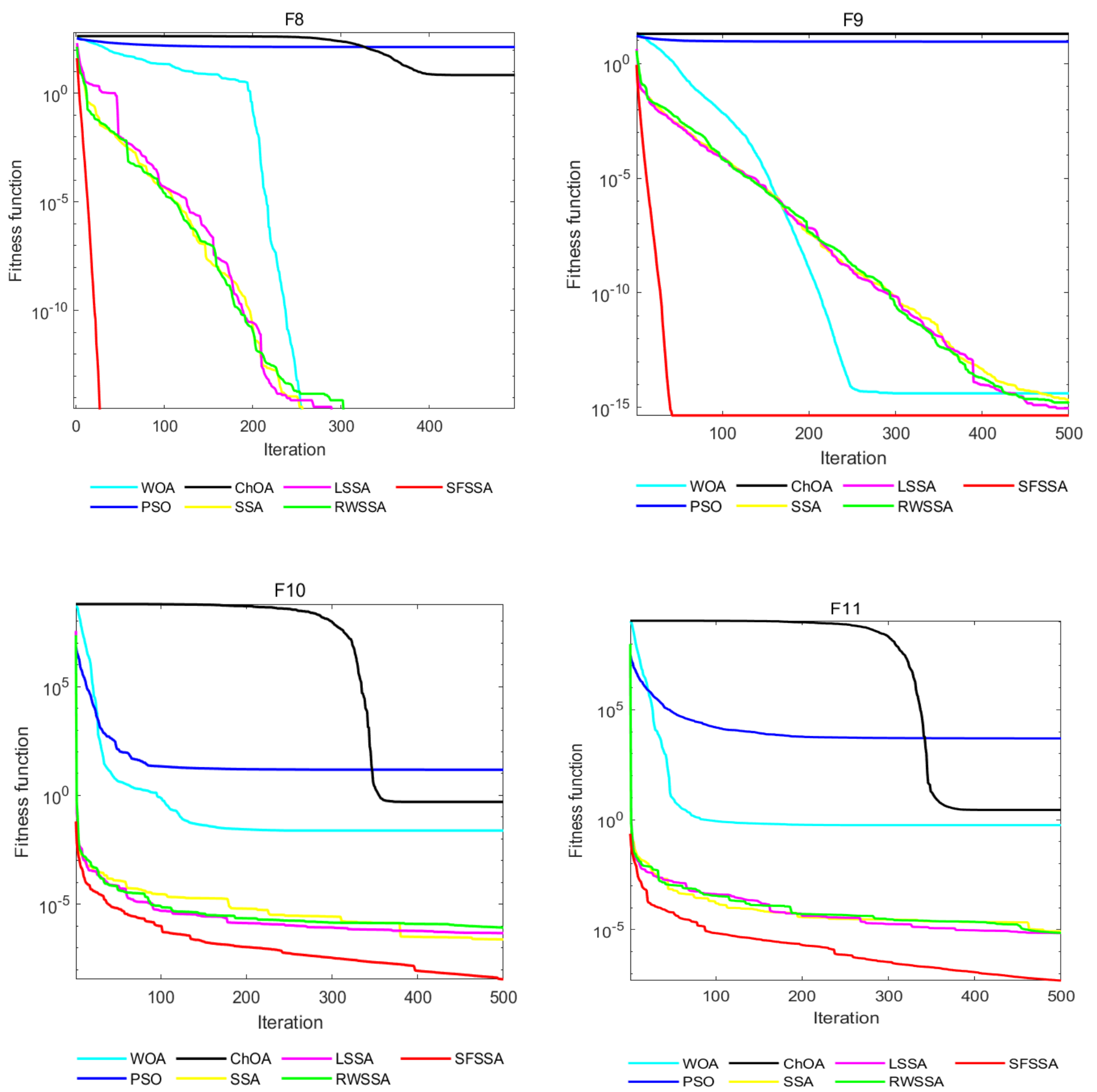

Figure 5. Convergence curves of the four multimodal test functions.

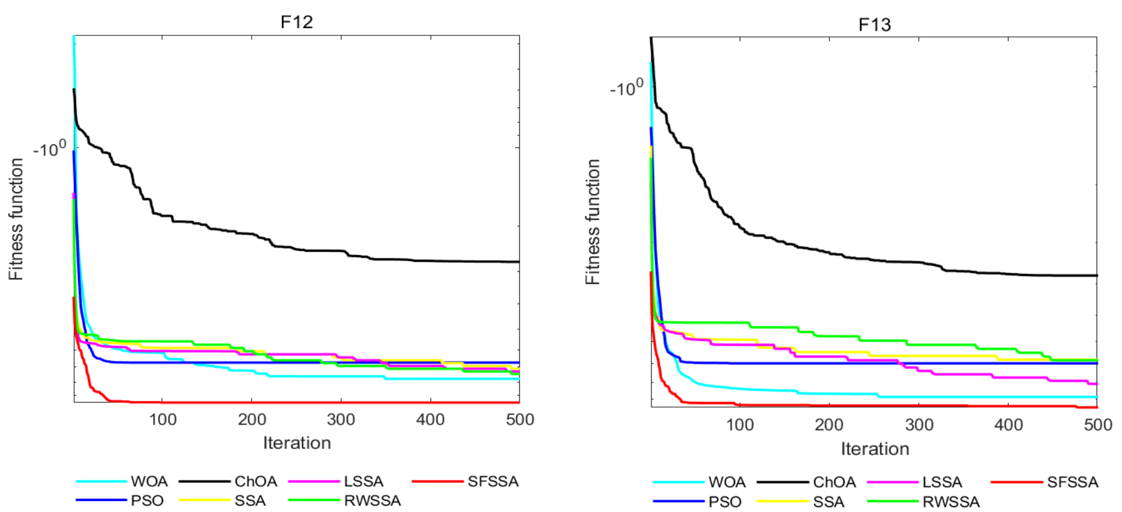

Figure 6. Convergence curves of the two low-dimensional multimodal test functions.

4.6. Wilcoxon rank sum test

To verify whether SFSSA outperforms the other four algorithms, this paper uses the Wilcoxon rank-sum test [39] to verify the significant difference between SFSSA and the other four algorithms. The significance level was set at p = 5%. When p < 5%, the original hypothesis is rejected, proving that there is a significant difference between the two algorithms; when p > 5%, the original hypothesis is accepted, proving that there is no significant difference between the two algorithms, i.e., the two algorithms are similar in terms of searching optimum. The results of the Wilcoxon rank-sum test are presented in Table 8. As shown in Table 8, the p-values of SFSSA are overwhelmingly less than 5%, which indicates that SFSSA outperforms the other six algorithms in the aspect of searching for optimal solutions.

Table 8. Results of Wilcoxon's rank-sum test.

Function	SFSSA/WOA		SFSSA/PSO		SFSSA/ ChOA		SFSSA/SSA		SFSSA/LSSA		SFSSA/RWSSA	
	P	S	P	S	P	S	P	S	P	S	P	S
F1	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+
F2	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+
F3	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+
F4	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+	1.2118E-12	+
F5	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+	1.5465E-09	+	6.5183E-09	+	8.8411E-07	+
F6	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+	3.1589E-10	+	9.7555E-10	+
F7	9.0632E-08	+	3.0199E-11	+	1.3111E-08	+	2.7548E-03	+	1.9883E-02	+	5.3221E-03	+
F8	N/A	=	1.2118e-12	+	1.2118e-12	=	N/A	=	N/A	=	N/A	=
F9	1.0873E-08	+	1.2118E-12	+	1.2118E-12	+	5.5398E-03	+	6.2958E-04	+	2.1498E-02	+
F10	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+	2.0023E-06	=	5.8587E-06	+	1.2493E-05	+
F11	3.0199E-11	+	3.0199E-11	+	3.0199E-11	+	1.1023E-08	+	1.4733E-07	+	3.3242E-06	+
F12	5.9673E-09	+	6.6065E-01	-	3.0199E-11	+	6.9724E-03	+	2.8129E-02	+	2.6077E-02	+
F13	2.3715E-10	+	6.6018E-01	-	3.0199E-11	+	5.5329E-08	+	1.698E-08	+	9.7917E-05	+

5. CEC 2017 function test

In this paper, we chose to evaluate 29 test functions on CEC 2017 to show more clearly the applicability and validity of the SFSSA method and to prevent ISSA from being meaningful only when the optimal value is 0. The number of test group participants was set to 30, the number of iterations was set to 500, the CEC 2017 dimension was set to 30, the SD was set to 0.6, and all other parameters were kept constant to ensure the fairness of the experiment. The best (Best), worst (Worst), mean (Ave), and standard deviation of each method were determined by running each of the above algorithms 30 times independently (Std). Conventionally, the CEC02 function was not employed because of high-dimension instability.

According to the data in Table 9, SFSSA performs better than the other algorithms in terms of the average solution accuracy across all 29 functions. In terms of average solution accuracy on 17 of the 29 functions in CEC2017, SFSSA outperforms the other six comparison methods. SFSSA optimizes effectively and can get close to each function's theoretically ideal value. Each SFSSA measure in F1, F4, F9, F11, F12, and F25 is the best among the methods. While WOA and PSO perform well in this algorithm test when the optimal solution is not 0, SFSSA performs better when trying to discover the ideal value.

Table 9. CEC2017 test function results.

Function	Algorithm	Best	Worst	Avg	Std
CEC01	WOA	2.7521E+09	1.2049E+10	5.3048E+09	2.0839E+09
	PSO	3.9511E+09	2.0882E+10	1.0629E+10	4.3592E+09
	ChOA	2.1379E+10	3.9268E+10	3.1125E+10	4.9822E+09
	SSA	1..5465E+10	3.9468E+10	2.5673E+10	6.4852E+09
	LSSA	1.3364E+10	3.4693E+10	2.6073E+10	4.9347E+09
	RWSSA	4.0229E+08	5.7645E+09	2.3873E+09	1.2613E+09
	SFSSA	**3.6436E+07**	**1.6832E+09**	**5.4266E+08**	**3.6960E+08**
CEC03	WOA	2.7108E+05	4.3703E+05	3.1316E+05	2.6333E+04
	PSO	**2.0200E+04**	1.9418E+05	7.9172E+04	3.8450E+03
	ChOA	8.7161E+04	1.9555E+05	1.3019E+05	2.6467E+03
	SSA	6.8990E+04	9.4072E+04	8.7805E+04	7.6863E+03
	LSSA	6.5817E+04	9.1420E+04	9.0454E+04	631659E+03
	RWSSA	5.5648E+04	9.3244E+04	8.3484E+04	9.0184E+03
	SFSSA	3.9320E+04	**5.7844E+04**	**4.8655E+04**	**4.0593E+03**
CEC04	WOA	7.6982E+02	1.9655E+03	1.3050E+03	2.8043E+02
	PSO	7.5968E+02	5.6531E+03	1.6659E+03	1.0134E+03
	ChOA	1.5527E+03	1.1421E+04	5.0594E+03	2.4137E+03
	SSA	2.8172E+03	1.0068E+04	6.2562E+03	1.7489E+03
	LSSA	2.5105E+03	1.2273E+04	6.6431E+03	2.3381E+02
	RWSSA	6.6658E+02	2.0496E+03	1.2087E+03	3.3348E+02
	SFSSA	**5.7392E+02**	**1.3180E+03**	**7.5047E+02**	**1.6351E+02**
CEC05	WOA	7.9113E+02	1.0112E+03	9.0714E+02	5.1951E+01
	PSO	**7.0261E+02**	8.9346E+02	7.8923E+02	5.0844E+02
	ChOA	7.6703E+02	8.8904E+02	8.3412E+02	2.9488E+01
	SSA	7.8028E+02	9.4760E+02	8.8796E+02	3.6099E+01
	LSSA	8.2192E+02	9.6674E+02	8.9608E+02	3.4698E+01
	RWSSA	7.7582E+02	9.0038E+02	8.2404E+02	**2.1849E+01**
	SFSSA	7.0501E+02	**8.7836E+02**	**7.8430E+02**	4.0051E+01
CEC06	WOA	6.5332E+02	6.8809E+02	6.7452E+02	8.2049E+00
	PSO	6.4466E+02	6.8553E+03	6.6481E+02	1.0671E+01
	ChOA	6.5959E+02	6.8790E+02	6.7538E+02	**6.5153E+00**
	SSA	6.7132E+02	7.1485E+02	6.8986E+02	1.0402E+01
	LSSA	6.7231E+02	7.0701E+02	6.9037E+02	9.4107E+00
	RWSSA	6.6035E+02	7.0122E+02	6.7772E+02	8.7272E+00
	SFSSA	**6.5116E+02**	**6.7730E+02**	**6.6455E+02**	7.3931E+00
CEC07	WOA	1.1495E+03	1.4800E+03	1.3355E+03	7.5504E+01
	PSO	**1.0120E+03**	1.4355E+03	**1.2220E+03**	1.0970E+02
	ChOA	1.1113E+03	**1.3620E+03**	1.2758E+03	4.5293E+01
	SSA	1.3921E+03	1.4634E+03	1.4303E+03	**1.9950E+01**
	LSSA	1.3979E+03	1.4844E+03	1.4283E+03	2.2257E+01
	RWSSA	1.2926E+03	1.3993E+03	1.3589E+03	2.5107E+01
	SFSSA	1.2405E+03	1.3649E+03	1.3210E+03	3.3245E+01

Continued on next page

Function	Algorithm	Best	Worst	Avg	Std
CEC08	WOA	7.5696E+03	2.5407E+04	1.3260E+04	4.8184E+03
	PSO	**4.2952E+03**	9.8206E+03	6.7382E+03	1.2470E+03
	ChOA	5.6548E+03	1.2613E+04	9.1128E+03	1.6160E+03
	SSA	6.8511E+03	1.1510E+04	8.9890E+03	1.0278E+03
	LSSA	6.9385E+03	1.2024E+04	9.4181E+03	1.2029E+03
	RWSSA	5.6685E+03	8.0635E+03	6.9202E+03	6.2775E+02
	SFSSA	4.6829E+03	**5.9822E+03**	**5.5650E+03**	**2.2838E+02**
CEC09	WOA	7.4389E+03	1.8568E+04	1.2792E+04	3.2702E+03
	PSO	3.4986E+03	1.1308E+04	6.4921E+03	2.0013E+03
	ChOA	4.8964E+03	1.3170E+04	9.1332E+03	1.9134E+03
	SSA	7.1123E+03	1.2437E+04	9.1943E+03	1.1771E+03
	LSSA	7.4288E+03	1.1393E+04	9.0447E+03	1.1664E+03
	RWSSA	5.7353E+03	7.9246E+03	6.8449E+03	5.3322E+02
	SFSSA	**4.5091E+03**	**7.7324E+03**	**5.6211E+03**	**5.3093E+02**
CEC10	WOA	5.7680E+03	8.4292E+03	7.3146E+03	7.2165E+02
	PSO	5.3291E+03	8.0497E+03	6.5228E+03	6.2632E+02
	GWO	8.3581E+03	9.1007E+03	8.7795E+03	**1.9823E+02**
	SSA	7.4905E+03	9.1504E+03	8.1998E+03	4.9933E+02
	LSSA	5.9377E+03	9.8859E+03	8.1302E+03	9.1346E+02
	RWSSA	5.1390E+03	8.3466E+03	6.6828E+03	8.5331E+02
	SFSSA	**4.6344E+03**	**7.6951E+03**	**5.9251E+03**	8.3308E+02
CEC11	WOA	3.7837E+03	1.1634E+04	6.4783E+03	1.8295E+03
	PSO	1.4876E+03	1.8909E+04	2.6820E+03	3.1057E+03
	GWO	2.4231E+03	7.3086E+03	5.0805E+03	1.2590E+03
	SSA	5.7628E+03	1.8814E+04	1.2286E+04	3.1245E+03
	LSSA	3.8262E+03	2.3897E+04	1.1872E+04	4.5958E+03
	RWSSA	2.5603E+03	9.0534E+03	5.7003E+03	1.8128E+03
	SFSSA	**1.3531E+03**	**2.8267E+03**	**1.7323E+03**	**3.6872E+02**
CEC12	WOA	1.0559E+08	9.0534E+08	3.9380E+08	2.0240E+08
	PSO	1.5360E+07	4.2519E+09	6.7337E+08	1.1052E+09
	GWO	1.9339E+09	1.2270E+10	6.8625E+09	2.5827E+09
	SSA	1.1498E+09	9.4254E+09	3.7250E+09	1.8872E+09
	LSSA	1.1466E+09	7.2733E+09	3.5069E+09	1.6856E+09
	RWSSA	3.7261E+07	8.1869E+08	2.7703E+08	1.9502E+08
	SFSSA	**6.6131E+06**	**2.1018E+08**	**9.0355E+07**	**5.6503E+07**
CEC13	WOA	1.2359E+06	2.5000E+07	8.9504E+06	6.8906E+06
	PSO	5.3765E+04	7.4736E+09	4.1462E+08	1.4797E+09
	GWO	8.6734E+07	1.6213E+10	4.9580E+09	4.6402E+09
	SSA	1.1694E+07	1.0677E+09	2.1979E+08	2.9453E+08
	LSSA	7.3845E+06	2.8034E+08	1.6581E+09	4.3841E+08
	RWSSA	4.2060E+04	1.4838E+05	4.0402E+05	**9.1262E+04**
	SFSSA	**2.5187E+04**	**6.7486E+05**	**1.4249E+05**	1.3530E+05
CEC14	WOA	5.0662E+04	**3.7829E+05**	**1.8059E+05**	**8.3126E+04**
	PSO	**1.5925E+03**	9.5824E+06	5.4620E+05	1.8268E+06
	GWO	7.8369E+04	8.4803E+06	1.5140E+06	2.0845E+06
	SSA	3.5322E+04	8.2217E+06	3.0115E+06	2.5046E+06
	LSSA	5.7902E+04	2.2248E+07	3.8198E+06	4.9857E+06
	RWSSA	1.1365E+04	7.3562E+06	1.4566E+06	1.5586E+06
	SFSSA	2.3121E+04	2.8287E+06	6.7370E+05	9.1138E+05

Continued on next page

Function	Algorithm	Best	Worst	Avg	Std
CEC15	WOA	1.3757E+05	6.7477E+06	2.4380E+06	1.8938E+06
	PSO	**3.2831E+03**	2.2326E+05	4.0684E+04	5.1644E+04
	GWO	2.2134E+06	9.2189E+07	3.6206E+07	2.9940E+07
	SSA	1.0356E+06	1.4523E+08	3.1418E+07	3.3443E+07
	LSSA	2.2843E+06	1.7469E+08	3.2834E+07	3.9751E+07
	RWSSA	8.9759E+03	2.0924E+05	4.0391E+04	4.6417E+04
	SFSSA	7.9633E+03	**1.0928E+05**	**3.4139E+04**	**2.3208E+04**
CEC16	WOA	3.1264E+03	5.0435E+03	4.0121E+03	4.5025E+02
	PSO	2.7716E+03	**4.4881E+03**	**3.6100E+03**	4.3774E+02
	GWO	3.4807E+03	5.1046E+03	4.2415E+03	**4.0586E+02**
	SSA	4.0091E+02	8.0371E+03	6.0267E+03	1.2376E+03
	LSSA	3.6845E+03	7.6680E+03	5.2503E+03	1.1298E+03
	RWSSA	3.3695E+03	6.7598E+03	4.7389E+03	8.0155E+02
	SFSSA	**2.6371E+03**	5.6123E+03	4.1412E+03	6.1095E+02
CEC17	WOA	2.2639E+03	3.7306E+03	2.9500E+03	4.0249E+02
	PSO	**1.9820E+03**	3.7713E+03	**2.6767E+03**	**3.5289E+02**
	GWO	2.4026E+03	**3.4461E+03**	2.8943E+03	2.2266E+02
	SSA	2.7532E+03	7.3902E+03	3.8777E+03	1.1258E+03
	LSSA	2.4803E+03	7.6671E+03	3.8078E+03	1.0714E+03
	RWSSA	2.5019E+03	3.7998E+03	3.0277E+03	3.3862E+02
	SFSSA	2.0810E+03	3.8917E+03	2.9427E+03	3.5424E+02
CEC18	WOA	7.0970E+06	3.7106E+07	1.8883E+07	6.2165E+06
	PSO	**2.0422E+04**	1.2149E+08	8.5440E+06	2.3813E+07
	GWO	1.9150E+06	2.1404E+07	6.3796E+06	4.8690E+06
	SSA	7.8234E+05	9.4066E+07	2.2694E+07	2.2292E+07
	LSSA	8.3272E+04	8.5018E+07	3.2863E+07	2.5715E+07
	RWSSA	4.4480E+05	4.1271E+07	7.4085E+06	9.5895E+06
	SFSSA	6.6567E+04	**1.9085E+07**	**1.3256E+06**	**3.5213E+06**
CEC19	WOA	2.8857E+06	6.4733E+07	1.6375E+07	1.6482E+07
	PSO	**2.5814E+03**	3.9357E+07	1.5107E+06	7.1607E+06
	GWO	1.8073E+07	9.1196E+08	3.2895E+08	2.7275E+08
	SSA	2.1711E+06	2.0490E+08	6.4781E+07	6.0504E+07
	LSSA	2.7389E+06	3.3330E+08	7.6435E+07	7.8595E+07
	RWSSA	5.5444E+05	2.5059E+07	7.3089E+06	6.3931E+06
	SFSSA	3.5845E+05	**3.4345E+06**	**1.3812E+06**	**6.7722E+05**
CEC20	WOA	2.4662E+03	**3.2179E+03**	2.9234E+03	2.5149E+02
	PSO	2.3785E+03	3.2240E+03	**2.8122E+03**	2.6080E+02
	GWO	2.9581E+03	3.5206E+03	3.2262E+03	**1.5488E+02**
	SSA	2.5138E+03	3.7448E+03	3.0875E+03	2.9184E+02
	LSSA	2.4958E+03	3.5353E+03	3.1315E+03	2.7335E+02
	RWSSA	2.6617E+03	3.4742E+03	3.0084E+03	2.3022E+02
	SFSSA	**2.3365E+03**	3.4670E+03	2.8643E+03	2.3474E+02
CEC21	WOA	2.5482E+03	2.7706E+03	2.6689E+03	5.8748E+01
	PSO	2.4558E+03	2.6587E+03	**2.5703E+03**	5.5015E+01
	GWO	2.5558E+03	**2.7106E+03**	2.6219E+03	**3.9311E+01**
	SSA	2.6194E+03	2.8718E+03	2.7307E+03	6.4110E+01
	LSSA	2.5721E+03	2.9118E+03	2.7242E+03	7.6103E+01
	RWSSA	2.5326E+03	2.8997E+03	2.6544E+03	7.5969E+01
	SFSSA	**2.4379E+03**	2.8096E+03	2.6090E+03	8.1962E+01

Continued on next page

Function	Algorithm	Best	Worst	Avg	Std
CEC22	WOA	**3.0037E+03**	1.0764E+04	9.0500E+03	1.3383E+03
	PSO	3.3426E+03	9.4255E+03	7.3984E+03	1.6419E+03
	GWO	9.1774E+03	1.0459E+04	9.9565E+03	**3.1975E+02**
	SSA	6.5866E+03	1.1366E+04	9.5136E+03	9.2080E+02
	LSSA	7.4379E+03	1.0950E+04	9.4340E+03	7.9417E+02
	RWSSA	6.6923E+03	9.6738E+03	8.1096E+03	7.4020E+02
	SFSSA	3.9904E+03	**8.4394E+03**	**7.3827E+03**	8.4227E+02
CEC23	WOA	**2.9939E+03**	3.3839E+03	3.1794E+03	1.0291E+02
	PSO	3.0358E+03	3.8355E+03	3.3797E+03	1.7538E+02
	GWO	3.0399E+03	**3.2762E+03**	**3.1471E+03**	**6.3925E+01**
	SSA	3.2044E+03	3.7693E+03	3.4968E+03	1.3681E+02
	LSSA	3.1946E+03	3.8172E+03	3.4933E+03	1.6578E+02
	RWSSA	3.0303E+03	3.5910E+03	3.2611E+03	1.2676E+02
	SFSSA	3.0625E+03	3.6956E+03	3.3363E+03	1.5722E+02
CEC24	WOA	**3.0561E+03**	**3.5102E+03**	**3.2468E+03**	1.0762E+02
	PSO	3.2471E+03	4.0426E+03	3.6049E+03	2.1511E+02
	GWO	3.2508E+03	3.3864E+03	3.3208E+03	**3.6121E+01**
	SSA	3.2254E+03	3.8271E+03	3.5845E+03	1.4337E+02
	LSSA	3.3988E+03	3.8229E+03	3.6026E+03	1.2300E+02
	RWSSA	3.1119E+03	3.8120E+03	3.3956E+03	1.8922E+02
	SFSSA	3.1794E+03	3.8721E+03	3.4689E+03	1.5825E+02
CEC25	WOA	3.0859E+03	3.4702E+03	3.2783E+03	8.4256E+01
	PSO	3.1733E+03	3.6108E+03	3.3353E+03	1.3725E+02
	GWO	3.4469E+03	5.2923E+03	4.6194E+03	4.1625E+02
	SSA	3.5785E+03	4.6485E+03	3.8684E+03	2.0939E+02
	LSSA	3.3542E+03	4.4255E+03	3.8720E+03	2.8369E+02
	RWSSA	3.0112E+03	3.2405E+03	3.1236E+03	5.7441E+01
	SFSSA	**2.9810E+03**	**3.0912E+03**	**3.0276E+03**	**2.7508E+01**
CEC26	WOA	7.1084E+03	1.0489E+04	8.2705E+03	8.3083E+02
	PSO	4.5568E+03	1.1258E+04	7.9573E+03	1.3161E+03
	GWO	6.6851E+03	**8.6786E+03**	**7.3898E+03**	**4.5071E+02**
	SSA	8.6313E+03	1.3566E+04	1.1127E+04	1.1011E+03
	LSSA	8.2141E+03	1.4402E+04	1.1398E+04	1.5443E+03
	RWSSA	**4.0501E+03**	1.1356E+04	9.1046E+03	1.5077E+03
	SFSSA	7.0392E+03	1.1301E+04	9.1244E+03	1.0243E+03
CEC27	WOA	**3.3013E+03**	**3.6408E+03**	**3.4241E+03**	**9.6885E+01**
	PSO	3.3092E+03	4.1896E+03	3.6326E+03	2.4144E+02
	GWO	3.5890E+03	4.2465E+03	3.8196E+03	1.5833E+02
	SSA	3.4560E+03	5.7074E+03	4.2390E+03	4.9111E+02
	LSSA	3.3879E+03	5.4298E+03	4.2099E+03	4.9514E+02
	RWSSA	3.4139E+03	4.2164E+03	3.7464E+03	1.9621E+02
	SFSSA	3.3365E+03	4.6083E+03	3.7666E+03	3.2408E+02
CEC28	WOA	3.5712E+03	**4.3275E+03**	3.8268E+03	1.8867E+02
	PSO	3.5717E+03	4.6722E+03	3.9758E+03	3.0743E+02
	GWO	3.9231E+03	6.3763E+03	5.1959E+03	5.9414E+02
	SSA	4.6268E+03	6.2573E+03	5.1811E+03	3.7857E+02
	LSSA	4.3060E+03	7.2419E+03	5.4124E+03	6.1770E+02
	RWSSA	3.4884E+03	4.3626E+03	3.7544E+03	**1.7331E+02**
	SFSSA	**3.3171E+03**	5.0410E+03	**3.7163E+03**	4.9261E+02

Continued on next page

Function	Algorithm	Best	Worst	Avg	Std
	WOA	**4.1174E+03**	**5.5934E+03**	**4.8218E+03**	3.1612E+02
	PSO	4.2572E+03	7.5296E+03	5.3141E+03	6.8721E+02
	GWO	4.5146E+03	5.7236E+03	5.0086E+03	**2.7854E+02**
CEC29	SSA	5.1787E+03	1.0407E+04	7.1945E+03	1.3815E+03
	LSSA	5.4806E+03	1.2326E+04	7.6982E+03	1.7367E+03
	RWSSA	4.8077E+03	8.4112E+03	6.1623E+03	8.8124E+02
	SFSSA	4.6946E+03	7.4922E+03	5.7552E+03	6.3070E+02

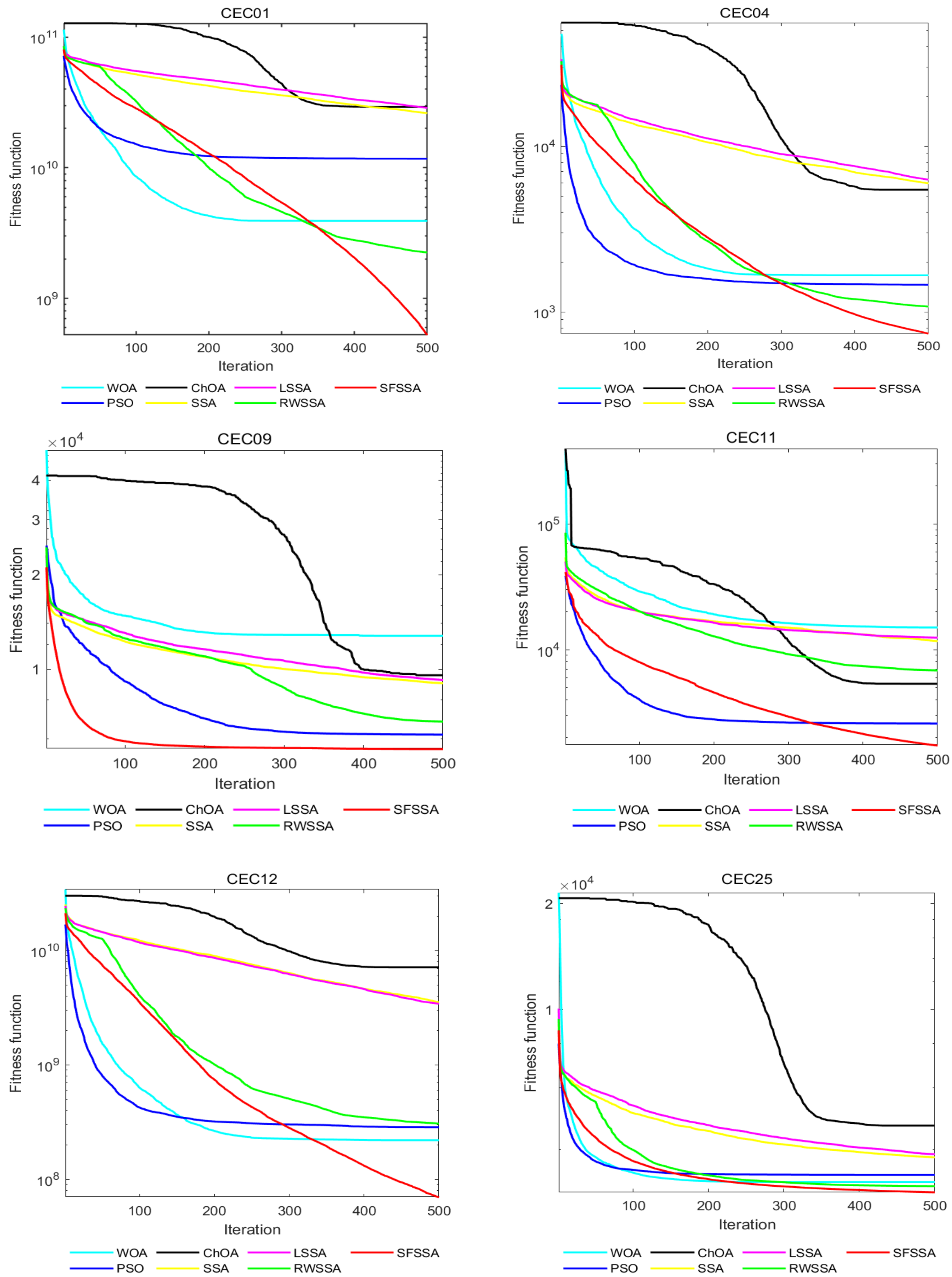

Figure 7. CEC2017 Function average convergence curve.

The convergence curves for SFSSA and the aforementioned comparative method are presented in order to show the convergence and stability of the revised approach. The Figure 7 shows that SFSSA has strong search ability, converges quickly in the early stage, and maintains the advantage of continuous search in the later stage, demonstrating that Tent chaos improves population diversity and expands the search space. SFSSA uses the improved positive cosine algorithm to make the algorithm compensate for the problem that the original algorithm falls into local optimum and improves the convergence accuracy of the algorithm; firefly is used to prove that Tent chaos improves population diversity and increases the search space. In a nutshell, SFSSA outperforms other algorithms in terms of optimization performance, has strong generality and efficiency, and can handle some complex optimization tasks.

6. Application of SFSSA in emergency location problems

6.1. Background analysis and parameter setting

When facing sudden natural disasters, how to protect people's living materials becomes a critical issue, and the government needs to make emergency plans immediately. Generally, temporary relief materials reserve centers to place relief materials from all over the country are open to solving the problem of transporting relief materials to each demand point in a reasonable way.

City A in a disaster situation is taken as a case study. There are 8 alternative locations of eligible emergency material storage centers in city A, suitable ones should be selected from these 8 alternative centers as emergency material distribution centers, serving 20 demand points in the city, to minimize the total construction cost. The corresponding location coordinates of 8 alternative centers, 20 demand points, the fixed construction cost, and the quantity of material demand for each demand point are shown in Tables 10 and 11. The assumptions made in this paper are listed as follows.

1) The selected emergency material distribution center needs to meet the demand of the demand point.

2) The emergency material distribution center can only be selected among the alternative locations.

3) One demand point should only be served by one distribution center.

4) The fixed construction cost of each alternative center is given.

5) The emergency material distribution center has the maximum capacity constraint.

6) The transportation cost is proportional to the transportation volume.

The parameter symbols and their meanings are listed as follows:

M: number of alternative centers for emergency supplies.

N: number of demand points.

d_j: the amount of demand at each demand point j.

p: number of emergency material distribution centers to open.

cap_i: capacity of distribution center i.

x_{ij}: transportation volume from distribution center i to demand point j.

t_{ij}: transportation cost from distribution center i to demand point j.

y_i: 0-1 variable, y_i=1 indicates that distribution center i is selected, otherwise, y_i=0.

C_i: fixed cost of emergency material distribution center construction.

Based on the above assumptions, the objective is to achieve the minimum cost under the constraint of maximum capacity and satisfy contingency requirements. The mathematical model is formulated as follows:

$$minz = \sum_{i=1}^{M}\sum_{j=1}^{N} x_{ij}t_{ij} + \sum_{i=1}^{M} y_i\, C_i \tag{13}$$

Subject to:

$$\sum_{i=1} x_{ij} \geq \ d_j,\ j = 1,2,\cdots n \tag{14}$$

$$\sum_{i=1} y_i = p \tag{15}$$

$$\sum_{i=1} d_j y_i \leq \ cap_i, i = 1,2,\cdots m \tag{16}$$

$$x_{ij} \geq 0, i = 1,2,\cdots m; j = 1,2,\cdots n \tag{17}$$

$$y_i \geq 0, i = 1,2,\cdots m \tag{18}$$

Equation (13) represents the objective function, which is seeking the minimum fixed construction cost and total transportation cost of the emergency material distribution center. Equation (14) indicates that the selected emergency distribution center can meet demand. Equation (15) represents selecting p centers from M alternatives as emergency materials distribution centers. Equation (16) is to ensure that alternative centers do not exceed their capacity limit. Equations (17) and (18) indicate the range of values of the variables.

Table 10. Information on alternative centers.

Alternative center	Position coordinates (X, Y)	Construction cost (yuan)
1	(82,12)	35,037
2	(24,78)	38,357
3	(21,35)	37,170
4	(88,60)	38,198
5	(76,36)	33,320
6	(75,74)	32,723
7	(43,23)	36,293
8	(32,62)	38,021

6.2. Results analysis

The improved SFSSA is chosen to compare with the original SSA for the emergency siting problem in this study because to the comparative findings of the aforementioned methods. The aforementioned emergency material distribution center location model was resolved by multiple experimental simulations using MATLAB software to demonstrate the benefits of SFSSA over SSA. Table 12 displays the SASSA and SSA parameters. The results of the optimization comparison between SFSSA and SSA for the emergency distribution center location problem are displayed in Table 13.

Table 11. Information on demand points.

Demand point	Position coordinates (X,Y)	Material requirements (box)
1	(75, 25)	103
2	(68, 14)	123
3	(92, 31)	113
4	(28, 25)	142
5	(83, 78)	173
6	(12, 41)	176
7	(64, 54)	113
8	(43, 45)	181
9	(71, 82)	103
10	(73, 68)	165
11	(15, 23)	102
12	(51, 23)	141
13	(43, 55)	149
14	(82, 41)	181
15	(58, 18)	157
16	(39, 17)	176
17	(33, 83)	118
18	(52, 55)	191
19	(58, 35)	179
20	(21, 52)	137

Table 12. Parameter setting.

Parameters	SFSSA	SSA
Population number	5000	5000
iteration number	100	100
ST	0.8	0.8
PD	0.2	0.2
SD	0.1	0.1
ω_{max}	1.0	—
ω_{min}	0.4	—

Analysis drawn from comparison results in Table 13 is clear and understandable. Since the location results calculated by both algorithms are the same, so is the fixed construction cost, with a total of RMB 140,357. However, the demand points served by each distribution center are different, which leads to differences in transportation costs. The transportation cost calculated by SFSSA is RMB 47,712.0915, while that calculated by SSA is much larger, which is RMB 51,707.7721. It can be concluded that the emergency material transportation cost is less calculated by SFSSA than SSA, which proves that SFSSA has better feasibility than SSA.

Table 13. Comparison of location results.

Algorithm	No. of distribution center	Demand point	Fixed cost (yuan)	Transportation cost (yuan)
SFSSA	5	1, 2, 3, 7, 14, 15	140,357	47,712.0915
	6	5, 9, 10		
	7	4, 8, 11, 12, 16, 19		
	8	6, 13, 17, 18, 20		
SSA	5	1, 2, 3, 8, 14, 15	140,357	51,707.7721
	6	5, 9, 10		
	7	4, 7, 11, 12, 16, 19		
	8	6, 13, 17, 18, 20		

Figure 8 shows the routes for the optimal location scheme for SFSSA and SSA.

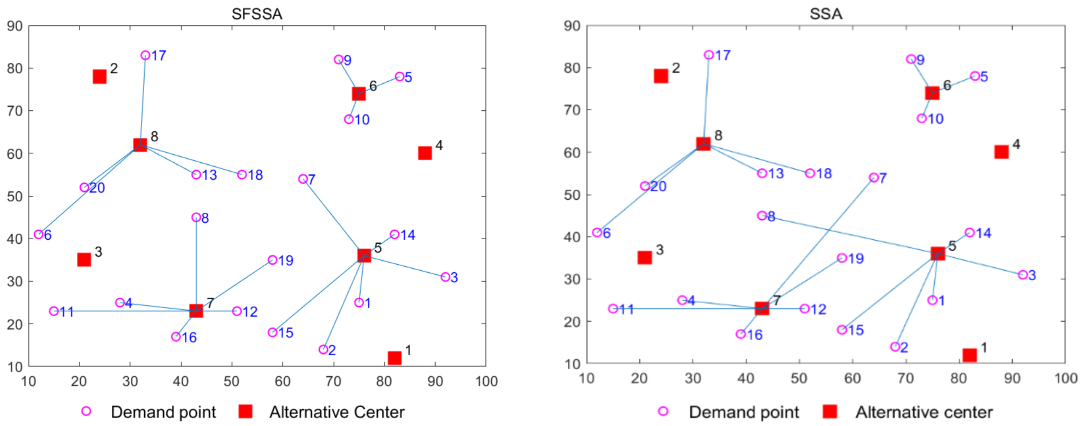

Figure 8. Route for SFSSA and SSA optimal location solution.

Figure 8 illustrates how distribution center No. 5's routes for distributing emergency supplies change between SFSSA and SSA. There are 1, 2, 3, 7, 14 and 15 demand points in the No. 5 distribution for SFSSA, whereas there are 1, 2, 3, 8, 14 and 15 demand points in the No. 5 distribution for SSA. The distinction is that Nos. 7 and 8 have different demand point distributions. The improved sine-cosine method used in this research helps the algorithm avoid running into the optimum solution problem up front, and firefly perturbation is used to boost the original algorithm's capacity to do optimal searches. Figure 5 shows how the algorithm's search capability has improved for the demand point locations of distribution centers 5 and 7 and 8. It is also clear from the figure that SFSSA's optimal site distribution solution is more effective than SSA's.

Figure 9 shows the convergence plots of SFSSA and SSA in the optimal location schemes.

SFSSA and SSA optimal values of 1.8807e+05 and 1.9206e+05, respectively, were obtained from the optimal outcomes of alternative center siting ran on MATLAB, demonstrating that the SFSSA approach surpasses the SSA algorithm. Tent chaos perturbation improves the algorithm's global search ability, as does enhancing the positive cosine in the discoverer position update, which improves the algorithm's search accuracy and speeds up the algorithm's convergence. The convergence graph in Figure 9 clearly shows that the SFSSA method converges quicker and has higher convergence accuracy than the SSA approach. As a consequence, the SFSSA strategy proposed in this study is more practical and successful than the SSA approach in the application of the emergency distribution center location problem.

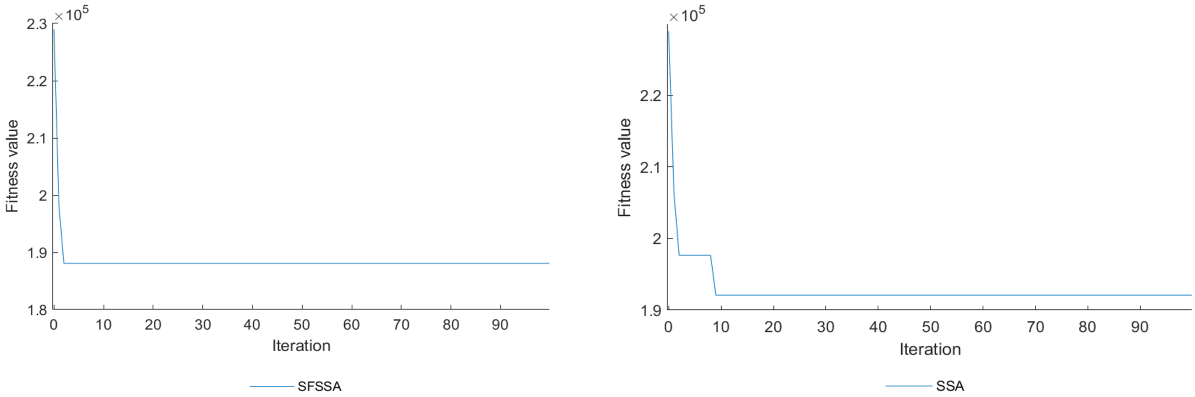

Figure 9. SFSSA and SSA convergence curves.

7. Conclusions

This paper proposes a sparrow search algorithm based on sine cosine and firefly perturbation. In the proposed algorithm, the diversity of the original population is optimized through Tent mapping initializing. Then the sine cosine algorithm and random inertia weights are invoked to update the discoverer position to avoid the algorithm falling into the optimum, and the convergence accuracy of the algorithm is improved. Finally, the firefly perturbation strategy is introduced to bring all sparrows closer to the optimal position and improve the optimal solution of the algorithm.

To sum up, SFSSA is shown to be more practical and effective when compared with four algorithms, namely WOA, PSO, GWO, and SSA, on 13 benchmark test functions and the Wilcoxon rank-sum test. Additionally, the above algorithms are contrasted in the CEC 2017 test function to further confirm the optimization performance of the algorithm when the optimal solution is not 0. In order to further demonstrate the viability of the SFSSA algorithm on real-world issues, the article's conclusion involves developing a mathematical model for the placement of emergency material distribution centers. In order to address real-world issues, future studies may look into expanding SFSSA to additional domains.

Acknowledgments

This paper was supported by the Social Science Grand Research of the Hebei Education Department (ZD202105). The authors gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

Conflict of interest

The authors declare no conflicts of interest.

References

1. L. Brezočnik, I. Fister, V. Podgorelec, Swarm intelligence algorithms for feature selection: a review, *Appl. Sci.*, **8** (2018), 1521. https://doi.org/10.3390/app8091521
2. Y. L. Li, S. Q. Wang, Q. R. Chen, X. G. Wang, Comparative study of several new swarm intelligence optimization algorithms, *Comput. Eng. Appl.*, **56** (2020), 1–12. https://doi.org/10.3778/j.issn.1002-8331.2006-0291
3. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of the ICNN'95-international conference on neural networks*, **4** (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
4. R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in *Proceedings of the MHS'95 6th International Symposium on Micro Machine and Human Science*, IEEE, (1995), 39–43. https://doi.org/10.1109/MHS.1995.494215
5. M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, **26** (1996), 29–41. https://doi.org/10.1109/3477.484436
6. M. Dorigo, L. M. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Trans. Evol. Comput.*, **1** (1997), 53–66. https://doi.org/10.1109/4235.585892
7. X. L. Li, *A New Intelligent Optimization Method-Artificial Fish Swarm Algorithm*, Ph.D thesis, Zhejiang University, Hangzhou, 2003. https://doi.org/cnki.cdmd.1.2003.051212
8. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
9. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
10. X. S. Yang, A new metaheuristic bat-inspired algorithm, *Comput. Knowl. Technol.*, **284** (2010), 65–74. https://doi.org/arxiv-1004.4170
11. I. Fister, I. Fister Jr., X. S. Yang, J. Brest, A comprehensive review of firefly algorithms, *Swarm Evol. Comput.*, **13** (2013), 34–46. https://doi.org/10.1016/j.swevo.2013.06.001
12. M. Khishe, M. R. Mosavi, Chimp optimization algorithm, *Expert systems with applications*, **149** (2020), 113338. https://doi.org/10.1016/j.eswa.2020.113338
13. J. K. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, *Syst. Sci. Control Eng.*, **8** (2020), 22–34. https://doi.org/ 10.1080/21642583.2019.1708830
14. Q. C. Duan, M. X. Mao, P. Duan, B. Hu, An improved artificial fish swarm algorithm optimized by particle swarm optimization algorithm with extended memory, *Kybernetes*, **45** (2016). https://doi.org/10.1108/k-09-2014-0198
15. Z. L. Han, M. Chen, S. Y. Shao, Q. X. Wu, Improved artificial bee colony algorithm-based path planning of unmanned autonomous helicopter using multi-strategy evolutionary learning, *Aerosp. Sci. Technol.*, **122** (2022), 107374. https://doi.org/10.1016/j.ast.2022.107374
16. S. Dereli, A novel approach based on average swarm intelligence to improve the whale optimization algorithm, *Arabian J. Sci. Eng.*, **47** (2022), 1763–1776. https://doi.org/10.1007/s13369-021-06042-3
17. R. Chaudhary, H. Banati, Swarm bat algorithm with improved search (SBAIS), *Soft Comput.*, **23** (2019), 11461–11491. https://doi.org/10.1007/s00500-018-03688-4

18. J. R. Wu, Y. G. Wang, K. Burrang, Y. C. Tian, B. Lawson, Z. Ding, An improved firefly algorithm for global continuous optimization problems, *Expert Syst. Appl.*, **149** (2020), 113340. https://doi.org/10.1016/j.eswa.2020.113340
19. X. Lv, X. D. Mu, J. Zhang, Z. Wang, Chaos sparrow search optimization algorithm, *J. Beijing Univ. Aeronaut. Astronaut.*, **47** (2020), 1–10. https://doi.org/10.13700/j.bh.1001-5965.2020.0298
20. B. Ma, P. M. Lu, L. F. Zhang, Y. Liu, Q. Zhou, Y. Chen, et al., Enhanced sparrow search algorithm with mutation strategy for global optimization, *IEEE Access*, **9** (2021), 159218–159261. https://doi.org/10.1109/access.2021.3129255
21. J. Ma, Z. Y. Hao, W. J. Sun, Enhancing sparrow search algorithm via multi-strategies for continuous optimization problems, *Inf. Process. Manage.*, **59** (2022), 102854. https://doi.org/10.1016/j.ipm.2021.102854
22. C. Y. Wu, X. S. Fu, J. K. Pei, Z. G. Dong, A novel sparrow search algorithm for the traveling salesman problem, *IEEE Access*, **9** (2021), 153456–153471. https://doi.org/10.1109/access.2021.3128433
23. Z. Zhang, R. He, K. Yang, A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm, *Adv. Manuf.*, **10** (2022), 114–130. https://doi.org/10.1007/s40436-021-00366-x
24. Y. Yu, S. C. Gao, S. Chen, S. Cheng, Y. R. Wang, S. Y. Song, et al., CBSO: a memetic brain storm optimization with chaotic local search, *Memetic Comput.*, **10** (2018), 353–367. https://doi.org/10.1007/s12293-017-0247-0
25. E. V. Altay, B. Alatas, Bird swarm algorithms with chaotic mapping, *Artif. Intell. Rev.*, **53** (2020), 1373–1414. https://doi.org/10.1007/s10462-019-09704-9
26. L. F. Yue, R. N. Yang, Y. J. Zhang, Y. Yu, Z. X. Zhang, Tent chaos and simulated annealing improved moth-flame optimization algorithm, *J. Harbin Inst. Technol.*, **51** (2019), 146–154. https://doi.org/10.11918/j.issn.0367-6234.201811027
27. S. R. Chen, S. Y. Wang, An optimization method for an integrated energy system scheduling process based on NSGA-II improved by tent mapping chaotic algorithms, *Processes*, **8** (2020), 426. https://doi.org/10.3390/pr8040426
28. F. B. Demir, T. Tuncer, A. F. Kocamaz, A chaotic optimization method based on logistic-sine map for numerical function optimization, *Neural Comput. Appl.*, **32** (2020), 14227–14239. https://doi.org/10.1007/s00521-020-04815-9
29. Z. J. Teng, J. L. Lv, L. W. Guo, Y. Y. Xu, An improved hybrid grey wolf optimization algorithm based on Tent mapping, *J. Harbin Inst. Technol.*, **50** (2018), 40–49. https://doi.org/10.11918/j.issn.0367-6234.201806096
30. X. F. Zhang, F. Xiao, X. L. Tong, J. Yun, Y. Liu, Y. Sun, et al., Time optimal trajectory planing based on improved Sparrow Search Algorithm, *Front. Bioeng. Biotechnol.*, **10** (2022), 852408. https://doi.org/10.3389/fbioe.2022.852408
31. J. M. Liu, W, Zhang, S. S. Qiu, Method for improving randomness of Tent chaotic sequences, *Comput. Eng.*, **38** (2012), 10–12. https://doi.org/10.3969/j.issn.1000-3428.2012.06.003
32. D. Zhao, Y. He, Chaotic binary bat algorithm for analog test point selection, *Analog Integr. Circuits Signal Process.*, **84** (2015), 201–214. https://doi.org/10.1007/s10470-015-0548-5
33. S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, *Knowl.-Based Syst.*, **96** (2016), 120–133. https://doi.org/10.1016/j.knosys.2015.12.022

34. C. Gan, W. H. Cao, M. Wu, X. Chen, A new bat algorithm based on iterative local search and stochastic inertia weight, *Expert Syst. Appl.*, **104** (2018), 202–212. https://doi.org/10.1016/j.eswa.2018.03.015
35. A. H. Gandomi, X. S. Yang, S. Talatahari, A. H. Alavi, Firefly algorithm with chaos, *Commun. Nonlinear Sci. Numer. Simul.*, **18** (2013), 89–98. https://doi.org/ 10.1016/j.cnsns.2012.06.009
36. D. L. Zhang, H. W. Xia, C. X. Chao, G. C. Ma, C. H. Wang, Improved firefly algorithm and its convergence analysis, *Syst. Eng. Electron.*, **44** (2022), 1291–1300. https://doi.org/10.12305/j.issn.1001-506X.2022.04.27
37. A. Yelghi, C. Köse, A modified firefly algorithm for global minimum optimization, *Appl. Soft Comput.*, **62** (2018), 29–44. https://doi.org/10.1016/j.asoc.2017.10.032
38. W. Long, T. B. Wu, S. H. Cai, X. M. Liang, J. J. Jiao, M. Xu, A novel grey wolf optimizer algorithm with refraction learning, *IEEE Access*, **7** (2019), 57805–57819. https://doi.org/10.1109/access.2019.2910813
39. J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm Evol. Comput.*, **1** (2011), 3–18. https://doi.org/10.1016/j.swevo.2011.02.002