A Hybrid Sparrow Search Algorithm Based on Constructing Similarity

LIU JIANHUA AND WANG ZHIHENG
School of Computing, Xi'an University of Posts and Telecommunications, Xi'an, Shaanxi 710100, China
Corresponding author: Wang Zhiheng (wzh3147255630@163.com)

ABSTRACT Sparrow search algorithm (SSA) is easy to fall into local convergence and convergence stagnation. In order to solve these problems, this paper introduced Circle chaos map into the original SSA to improve its global search ability at the beginning of iteration. Meanwhile, it introduced T-distribution variation to affect the sparrow population position update rules in different iteration periods. Finally, we constructed the “similarity function” to measure the “dispersion” of the sparrow population, and formulated the search rules of the sparrow population under different “dispersion”. In order to test the specific optimization performance of the proposed algorithm, the test results of 54 test functions are compared with those of 9 other algorithms which are widely used, and then the test results are analyzed using non-parametric tests in statistics. At the same time, this paper introduces this algorithm into three concrete engineering test problems for testing. The results of these tests all prove that the proposed algorithm has stronger global optimization ability and higher convergence precision compared with other algorithms.

INDEX TERMS Sparrow search algorithm, T-distribution variation, circle chaotic map, intrusion detection, similarity function, adaptive algorithms.

I. INTRODUCTION

In recent years, with the wide application of swarm intelligence optimization algorithms in machine learning, a large number of bionic intelligent algorithms have been proposed and studied [1], [2]. For example, ant colony optimization (ACO) [3], manta ray foraging optimization (MRFO) [4], glowworm swarm optimization (GSO) [5], gray wolf optimization (GWO) [6], artificial ecosystem-based optimization (AEQ) [7], shark smell optimization (SSO) [8], flamingo search algorithm (FSA) [9], and sparrow search algorithm (SSA). Among them, SSA was proposed in literature [10] in 2020. This is a new swarm intelligence optimization algorithm. SSA solves specific optimization problems by simulating the behavior of sparrows when foraging in nature. Compared with other intelligent optimization algorithms, SSA has the characteristics of high search accuracy, fast convergence speed, good stability and strong robustness [11]. However, SSA is extremely prone to local convergence and convergence stagnation in the later period of convergence [12]. These problems will directly affect the optimization effect of SSA, resulting in the failure to find the global optimal solution.

The problem of SSA is also the concern of swarm intelligence optimization technology. Researchers try to influence the decision in the process of swarm optimization through mathematical techniques. Literature [13] proposed an enhanced whale optimization algorithm (EWOA), which embedding fractional-order chaotic mapping in the search process of EWOA to improve the search accuracy of EWOA. Literature [14] proposed a sinusoidal chaotic gravity search algorithm (SCGSA) as a further step for GSA to get rid of its locally optimal stagnation. Literature [15] proposed an evolutionary programming (EP) using mutations based on the T-probability distribution (TEP). T-probability distributions can be related to Gaussian and Cauchy probability distributions. Its variance can be changed by adjusting n degrees of freedom. Literature [16] proposed an adaptive inverse inertial non-particle swarm optimization algorithm based on simulated annealing to solve the problem that the inverse particle swarm optimization algorithm is easy to fall into local optimum.

On the improvement of the sparrow search algorithm, some of the latest studies include: Literature [17] proposes a lens-learning sparrow search algorithm (LLSSA) to improve the new sparrow search algorithm random and easy to fall into local optimal defects. In the finder stage, the algorithm introduces the reverse learning strategy based on
the lens principle to improve the search range of individual sparrows, and then proposes the variable spiral search strategy to make the follower’s search more detailed and flexible. Finally, the simulated annealing algorithm is used to determine the optimal solution. Literature [12] proposes an improved sparrow search algorithm (ISSA) sparrows, this method USES the center of gravity reverse learning mechanism to initialization of population, the population has better spatial distribution, and in the position of the discoverer update part introduces learning coefficient, improve the global search ability of the algorithm. At the same time, mutation operator is used to improve the sparrow position update formula, so as to avoid the local convergence problem.

Inspired by the literature mentioned above, this paper proposes an adaptive sparrow search algorithm (CSSA) based on improved Circle chaos mapping, T-distribution variation and similarity to solve the key problem of local convergence of group optimization algorithm. Currently the sparrow research mainly is to solve the problem of local convergence algorithm, the solution of these problems lies in on the sparrow population of all the sparrow diffusion treatment, but no specific judgment on whether you need a sparrow diffusion, in order to solve this problem and combine the characteristics of the sparrow algorithm itself, this paper constructs a function to measure the clustering and scattered between individuals in a population. That is, the “Similarity function” and then select the optimal strategy through the similarity value. The over-aggregated sparrows were treated with diffusion. The improved Circle chaos map proposed in this paper was used to deal with the position of the sparrows with the general aggregation degree, and the T-distribution variation was used to deal with the position of the remaining sparrows, while the original position remained unchanged. This method is mainly used to solve the problem that the sparrow algorithm is easy to fall into local convergence, which greatly improves the convergence speed of the algorithm and effectively avoids the problem of convergence stagnation.

The main structure of this paper is as follows: In Section II, we describe the basic principles of the original SSA algorithm. In Section III, an improved Circle chaos map and T-distribution variation are proposed, and how these two technologies are applied to the SSA algorithm is deduced and constructed. Finally, the “similarity function” is constructed to judge the “clustering and dispersion” of individual sparrows. The pseudocode, algorithm flow and computational complexity of CSSA are also discussed. In Section IV, CSSA and four optimization algorithms are tested and compared. Nine benchmark functions with different characteristics are selected for the test function, and the results are discussed. In Section V, we compared CSSA with the ten latest optimization algorithms, and CEC-2015 and CEC-2017 were selected as the test functions. Finally, Wilcoxon Sign-Rank test was conducted on the test results. In Section VI and Section VII, CSSA is introduced into two specific engineering problems for testing, and the test results of other algorithms are compared. Finally, in Section VIII, we combine CSSA algorithm with SVM model to construct a CSSA-SVM system, which is used as a network intrusion detection system. Compared with other detection systems, this detection system has better detection accuracy. The above test results all show that CSSA and SSA can better solve the local convergence problem, so as to find the global optimal solution.

II. SPARROW SEARCH ALGORITHM

SSA can be simply abstracted into a finder-address-early-warning model. In a D-dimensional search space, if there are N sparrows, the position of the first sparrow in the D-dimensional search space is $X_i = [x_{i1}, \ldots, x_{id}, \ldots, x_{iD}]$, where $i = 1, 2, \ldots, n$, $x_{id}$ represents the position of the $ith$ sparrow in the $dth$ dimension. Finders generally account for 10%-30% of the population. The location update formula is shown in (1).

$$
X_{id}^{t+1} = \begin{cases} 
  X_{id}^t \times \exp \left( \frac{-i}{\alpha \times T} \right), & R_2 < ST \\
  X_{id}^t + Q \times L, & R_2 \geq ST 
\end{cases}
$$

(1)

where, $t$ represents the current iteration number; $T$ is the maximum number of iterations; Is the uniform random number between 0 and 1; $Q$ is a random number following the standard normal distribution; $L$ represents a matrix of size $[0,1]$ and $ST \in [0.5,1]$ represent the warning value and the safety value respectively. When $R_2 < ST$, the finder can be widely searched; When $R_2 < ST$, the warning sparrow spots a predator, the finder quickly moves toward safety.

Equation (2), as shown at the bottom of the next page, is the update strategy of entrants. All sparrows in this population that are not finders are entrants. Where: $xw_{id}$ represents the sparrow’s worst position in the $d$ dimension when the population is iterated in the $tth$ iteration; $xb_{id}^{t+1}$ represents the optimal position of sparrow in $d$ dimension in the $(t + 1)$ th iteration of the population.

$$
X_{id}^{t+1} = \begin{cases} 
  xb_{id}^{t} + \beta (x_{id}^{t} - xb_{id}^{t}), & f_i \neq f_g \\
  X_{id}^{t} + K \left( x_{id}^{t} - xw_{id}^{t} \right), & f_i = f_g 
\end{cases}
$$

(3)

Equation (3) is the updating formula of the position of the early warning. In a sparrow population, 20% of the sparrows are generally the early warning sparrows, $\beta$ representing the step size control parameters and obeying the normal distribution random numbers with a mean of 0 and a variance of 1. $K$ is a random number between -1 to 1, and it is the step size control parameter; $e$ is a minimal constant; $f_j$ is the fitness value of the $ith$ sparrow, $f_g$ and $f_w$ are the optimal and the worst fitness values of the current sparrow population, respectively. When $f_i \neq f_g$, indicates that the sparrow is at the edge of the population, vulnerable to predator attack. When $f_i = f_g$, indicates that the sparrow will get close to other sparrows in time.
III. ADAPTIVE HYBRID SPARROW ALGORITHM BASED ON THE CONSTRUCTION OF “CONVERGENCE AND DIVERGENCE” SIMILARITY

A. IMPROVED CIRCLE CHAOS MAPPING

At present, there have been a large number of studies on the application of chaotic mapping to the optimization of swarm intelligence algorithms [18]. Circle map is an excellent mapping function with good order, existence and uniqueness [19]. In this paper, the problems existing in the chaotic mapping of Circle are mainly improved, and a new improved Circle mapping is constructed to enhance its randomness, so that it can overcome the mapping inequality problems existing in the chaotic mapping of Circle. The original formula of chaotic mapping of Circle is:

\[ C_{i+1} = \text{mod}(C_i + a - \left(\frac{b}{2\pi}\right) \sin(2\pi C_i), 1), \]  

where, \( a \) and \( b \) are both mapping parameters, generally \( a = 0.2 \) and \( b = 0.5 \) respectively. The distribution diagrams of sequences generated in 5000 iterations in different intervals are shown in Fig. 1a:

It can be clearly seen from Fig. 1a that the values formed by the original Circle map mostly concentrate between 0.2 and 0.6, and their probabilities vary at different values. In order to enhance its randomness, the original chaotic mapping equation of Circle is improved in this paper, and a new mapping function is constructed as follows:

\[ C_{i+1} = \text{mod}(C_i + a - \left(\frac{b}{4\pi}\right) \sin(2\pi C_i) - \left(\frac{b}{4\pi}\right) \cos(2\pi C_i) + \text{random}(0, 1), 1) \]  

where, \( a \) and \( b \) are parameters, generally \( a = 0.2 \), \( b = 0.5 \); \( \text{random}(0,1) \) is a random number between 0 and 1. The interval distribution diagram of the improved Circle chaotic map is shown in Fig. 1b. Compared with the original Circle chaotic map, the distribution is more uniform and the value is more random. It can be clearly seen from the scatter diagram (see Fig. 2a) and time sequence diagram (see Fig. 2b) of the map distribution.

The Circle chaotic map constructed above is introduced into the sparrow algorithm to ensure the uniformity and randomness of the sparrow population distribution. Then, the initial value of the ith dimension of the sparrow is shown in (6):

\[ x_{ij} = lb + (ub - lb) \times C_i \]  

In (6), \( lb \) is the lower limit of the search space and \( ub \) is the upper limit of the search space. Then the position of the sparrow can be obtained by taking the sequence of length \( D \) (\( D \) is the dimension of the goal problem).

B. T-DISTRIBUTION VARIATION

Here, this paper introduces the T-distribution in statistics [20]. T-distribution is a special distribution function, which contains parameter degree of freedom \( n \). The smaller \( n \) is, the flatter the curve is. Its equation is as follows:

\[ f(x, n) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi} \Gamma\left(\frac{n}{2}\right)} \left(1 + \frac{x^2}{n}\right)^{-\frac{n+1}{2}}, -\infty < x < +\infty \]  

At that time, the T-distribution curve is the Cauchy distribution curve, that is \( t(n = 1) = C(0,1), C(0,1) \) is the Cauchy distribution as shown in (8):

\[ f(x) = \frac{1}{\pi(1+x^2)}, -\infty < x < +\infty \]  

The larger \( n \) is, the closer the curve is to the normal distribution curve. In the case of \( n \rightarrow \infty \), the T-distribution curve is approximately a gaussian distribution curve, this is \( t(n \rightarrow \infty) \rightarrow N(0,1) \), where \( N(0,1) \) is a Gaussian distribution, as shown in (9):

\[ f(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right), -\infty < x < +\infty \]
In other words, with the change of \( n \) value, the distribution curve of T-distribution will gradually approach to Cauchy distribution or Gaussian distribution, as shown in Fig. 3. It can be seen that the Cauchy mutation is more likely than the Gaussian mutation to produce the next generation points far away from the parent. According to the unique properties of T-distribution, this paper proposes the following adaptive sparrow update strategy based on the number of iterations and the fitness value of each sparrow:

In this paper, the T-distribution idea is integrated into the finder update strategy of the sparrow algorithm, as shown in the following equation:

\[
x_{id}^{t+1} = \begin{cases} 
    x_{id} \cdot \exp\left(\frac{-i}{\alpha \times T}\right), & R_2 < ST \\
    x_{id} + t(t) \times L, & R_2 \geq ST 
\end{cases} 
\]

(10)

In the reconstructed finder update (10), denotation obeys the T-distribution with \( T \) of freedom. When the warning value is greater than the safety value, the population is more inclined to search in the form of Cauchy distribution probability at the early stage of the search, so as to ensure that the finder, as the sparrow with the better position in the whole population, increases the local search ability at the later stage of the iteration, and increases the global search ability at the early stage of the iteration.

At the same time, this paper improves the renewal strategy of participants. As shown in (11), if the sparrow’s fitness \( f_i \) is greater than the average fitness \( f_{avg} \), it means that the sparrow needs to fly to other locations to search for energy. If the sparrow’s fitness \( f_i \) is less than or equal to the average fitness \( f_{avg} \), then the sparrow will randomly search for a position near the current optimal position for foraging. At the same time, this paper introduces the T-distribution and the improved Circle mapping, namely \( C_{t+1} \) in (11), so as to increase the randomness of its value, where, rand \([-1,1]\) represents the random value \(-1 \) or \(1\), and \( t(t) \) represents the T-distribution with the degree of freedom of \( t \).

After a round of position updating of the whole sparrow population, T-distribution variation was carried out for some sparsors with better fitness based on sparrow \( x_{id}^{old} \) with better fitness. The variation formula was as follows: (12).

\[
x_{id}^{new} = x_{id}^{old} + t(n) \times x_{id}^{old} = x_{id}^{old} (1 + t(n)) 
\]

(12)

where, \( n \) is the current number of iterations, and \( t(n) \) represents a T-distribution with \( n \) degrees of freedom. Therefore, in the initial stage of optimizing the algorithm, the above formula is also inclined to Cauchy distribution variation, and its variation is more likely to produce the next generation far away from the parent. When the algorithm enters the later stage of optimization, the formula will be inclined to Gaussian distribution variation, and the algorithm will have good local variation performance.

### C. SIMILARITY FUNCTION

This paper constructs a new similarity function based on cosine similarity and Euclidean similarity. In this paper, a certain point in the search space is selected as the reference point to calculate the similarity. That is, under the condition that the upper limit of the search space is \( ub \) and the lower limit is \( lb \), the coordinate of the reference point determined is \( M(ub, ub, \ldots, ub) \) and the coordinate of the edge point is \( B(lb, lb, \ldots, lb) \).

Euclidean similarity is derived from the distance formula between two points in Euclidean space, which measures the absolute distance between each point in multi-dimensional space [21]. Given that the position coordinate of the sparrow is \( X_i \), the Euclidean similarity of this point is as shown in (13).

\[
D_i(X_i, M) = \frac{\|X_i - X_j\|}{\|B - M\|} 
\]

(13)

Cosine similarity uses the cosine value of the angle between two vectors in the vector space as the size of the difference between two individuals [22], which pays more attention to the difference between two vectors in direction rather than in distance or length. The cosine similarity after normalization in the sparrow algorithm is shown in (14).

\[
C_i(X_i, M) = \frac{1}{2}(1 - \frac{\vec{X}_i \cdot \vec{M}}{\|\vec{X}_i\| \cdot \|\vec{M}\|}) 
\]

(14)

Then, the overall similarity calculation method of the \( i \)th sparrow is shown in (15) and (16).

\[
\rho(x_i) = \sum_{j=1}^{M} \begin{cases} 
    1, & M_1 \leq \frac{D_j}{D_i} \leq M_2 \text{ and } M_1 \leq \frac{C_i}{C_j} \leq M_2 \\
    0, & \text{otherwise} 
\end{cases} 
\]

\[d(x_i) = \frac{\rho(x_i)}{M}, \quad i = 1, 2, 3, \ldots, M \]

(15) \hspace{1cm} (16)

In (15), \( M_1 \) and \( M_2 \) are the parameters, and the values of 0.85 and 1.15 are the optimal after testing. \( \rho(x_i) \) is how many sparsors in the population are in a position similar to the \( i \)th sparrow. When the sparrow and a sparrow are similar in all dimensions, it is judged that they are similar, and the value is 1; Otherwise, it’s 0. \( d(x_i) \) is the similarity of the \( i \)th sparrow.
D. ALGORITHM FLOW
Aiming at the problem that the SSA is easy to fall into local convergence, this paper introduces the improved Circle chaos map and the improved T-distribution variation, and adds the similarity function constructed for the SSA to form the CSSA, which has a strong global optimization ability. The specific algorithm process is as follows:

Step1: Introduce initial parameters: population size \( N \), finder ratio \( FR \), early-warning ratio \( WR \), dimension \( M \) of objective function, maximum number of iterations \( T \), upper bound \( ub \) and lower bound \( lb \) of search range. The improved Circle chaotic mapping (5) is used to generate \( N \) \( m \)-dimensional vectors.

Step2: Initialize sparrow population through (6).

Step3: \( f_i \) of each sparrow and \( f_{avg} \) of the average fitness of the whole population were calculated. Sparrows with the top \( FR \) in the fitness ranking were regarded as finders, and the rest were considered as participants. Their distribution was updated according to (10) and (11), as shown at the bottom of the page, and \( FR+N \) sparrows were randomly selected to update the positions of early warning sparrows according to (3).

Step4: If \( f_i > f_{avg} \), then calculate the similarity value of each sparrow according to (13)-(16), and calculate the average similarity value of the whole sparrow population \( d_{avg} \); if \( d_i > d_{avg} \), introduce the chaotic mapping of improved Circle to diffuse it. If the similarity value of the ith sparrow is less than the average similarity value \( (d_i \leq d_{avg}) \), then the t-distribution variation is carried out for it according to (12).

Step5: If \( f_i \leq f_{avg} \), keep the sparrows in this part to enter the next iteration.

Step6: Boundary detection.

Step7: Whether the ending condition is met. No: go to Step3; Yes: output the optimal sparrow position and the optimal solution.

The algorithm flow chart is shown in Fig. 4.

E. ALGORITHM PSEUDO-CODE
This sub-section shows pseudo-code of CSSA.

F. COMPLEXITY ANALYSIS OF CSSA
This subsection analyzes the complexity of the CSSA. The time complexity and space complexity of CSSA are described separately below.

1) TIME COMPLEXITY
Initializing the population takes \( O(n \times d) \) time, where \( n \) is the population size and \( d \) is the dimension size. The \( O(Iter_{Max} \times n \times d) \) time required to find the fitness of each sparrow, where \( Iter_{Max} \) is the maximum number of iterations.

2) SPACE COMPLEXITY
The spatial complexity of the CSSA is \( O(n \times d) \), where \( n \) is the population size and \( d \) is the dimension size, that is, the maximum amount of space occupied when initializing the population.

IV. BENCHMARK FUNCTION TESTING
A. SELECTION OF BENCHMARK FUNCTIONS
In order to test the optimization ability of the improved algorithm in different functions, as well as the feasibility of the algorithm and the efficiency of optimization, this paper selects 9 different types of benchmark functions and carries out simulation on their selection of different dimensions to verify the optimization ability of the algorithm in low and high dimensional space, as shown in Table 1.

B. EXPERIMENTAL ENVIRONMENT
The experiments in this paper are both carried out on a computer configured as Intel(R) Core (TM) i7-10750H CPU @ 2.60GHz 2.59GHz, with a memory size of 16GB and an operating system of Window10. The code part of the experiment was written, run and tested by Python. The version of Python was 3.8.3, the development tool PyCharm version was 2020.3.3, and the mapping tool of the benchmark function was MATLAB, the version was 2018a.

C. IMPROVE THE SSA ALGORITHM TO TEST THE BENCHMARK FUNCTION
In order to verify the specific optimization performance of the CSSA constructed above, nine benchmark functions with different properties were selected in this section to test them in different dimensions (as shown in Table 1). At the same time, the popular PSO, GWO, WOA, and SSA before improvement were selected to optimize the nine benchmark functions in Table 1. The number of fixed iterations is 200, and the population is 50. In order to avoid the contingency of single optimization results, this paper carried out a total of ten tests, and took the optimal values of these test results, and calculated the average value (see AVG), standard deviation (see STD), and average running time (see Runtime). Parameter settings are shown in Table 2.

The test results are shown in Table 3: Through comparison, it can be found that the CSSA has better optimal value than the original SSA, GWO, PSO and WOA in terms of algorithm optimization performance, no matter in low dimensions (F1, F2, F4, F5, F8, and F9) or in high dimensions (F3, F6, and F7).
It performs well in both single-modal functions (F1 and F2), multi-modal functions (F3, F4, F5, F6, and F7), and fixed-dimensional functions (F8 and F9), among which the CSSA for F2, F3, and F4 functions is 2-3 orders of magnitude better than the original algorithm. On average, F1, F5, F6, and F7 will reach the optimal value 50-80 iterations ahead.

TABLE 1. Selection of benchmark functions.

| FUNC | Benchmark function | Function formula | The search space | Dimension | Best | Features |
|------|--------------------|------------------|-----------------|-----------|------|----------|
| F1   | Sphere             | $f(x) = \sum_{i=1}^{D} x_i^2$ | [-100,100] | 20 | 0 | Unimodal |
| F2   | Rosenbrock         | $f(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$ | [-2.86, 2.86] | 2 | 0 | Unimodal |
| F3   | Ackley             | $f(x) = -20 \exp(-0.2\sqrt{\frac{1}{D} \sum_{i=1}^{D} x_i^2} - \exp(\frac{1}{D} \sum_{i=1}^{D} \cos(2\pi x_i)) + 1)$ | [-32.9], [32.9] | 20 | 0 | Multimodal |
| F4   | Schwefel           | $f(x) = 418.9829D - \sum_{i=1}^{D} x_i \sin\sqrt{|x_i|}$ | [-500,500] | 2 | 0 | Multimodal |
| F5   | Schaffer           | $f(x) = 0.5 + \frac{(\sin(\sum_{i=1}^{D} x_i^2)^{\gamma} - 0.5)}{(1 + 0.001(\sum_{i=1}^{D} x_i)^{2})^\gamma}$ | [-10,10] | 5 | 0 | Multimodal |
| F6   | Rastrigin          | $f(x) = \sum_{i=1}^{D} [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | [-5.12, 5.12] | 20 | 0 | Multimodal |
| F7   | Griewank           | $f(x) = 1 + \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos(x_i/\sqrt{K})$ | [-600, 600] | 20 | 0 | Multimodal |
| F8   | Three-Hump         | $f(x) = 2x_i^2 - 1.055x_i^4 + x_i^6/6 + x_i + x_i^2$ | [-5, 5] | 2 | 0 | Fixed dimension |
| F9   | Cross-in-Tray      | $f(x) = -0.0001 \left[\sin(x_i)\sin(x_i)\exp\left(100 - \sqrt{x_i^2 + x_i^2/\pi} + 1\right)\right]^\gamma$ | [-10, 10] | 2 | -2.06261 | Fixed dimension |
of the original algorithm, which greatly improves the convergence speed and accuracy of the algorithm. Fig. 5 shows a line chart comparing the change trend of the optimal value of the five algorithms in the process of searching for the optimal function from F1 to F9 by randomly selecting one test result out of ten tests conducted by the five algorithms.

V. TEST OF CEC-2015 AND CEC-2017 TEST FUNCTIONS

In this section, I will test and compare CSSA with ten popular swarm intelligence algorithms at the present stage. The test function set I choose is the CEC-2015 data set and the CEC-2017 data set. These ten algorithms are as follows: Particle swarm Optimization (PSO) [23], Whale Optimization Algorithm (WOA) [24], Grey Wolf Optimizer (GWO), Tunicate Swarm Algorithm (TSA) [25], Butterfly Optimization Algorithm (BOA) [26], Satin bowerbird Optimization (SBO) [27], Pigeon Inspired Optimization (PIO) [28], Improved Sparrow Search Algorithm (ISSA), Lens Learning Sparrow Search Algorithm (LLSSA), and Sparrow search algorithm (SSA).

We conducted non-parametric Test on the Test results, and the Wilcoxon Sign-Rank Test was selected in this paper. The results are presented in subsection C.

A. EVALUATION OF IEEE CEC-2015 TEST FUNCTIONS

Table 4 is the CEC-2015 benchmark function set. Parameter selection of each algorithm in our test is shown in Table 2, the maximum number of iterations is 1000, the dimension size is shown in Table 4, and the population number is set to 50. The test results are shown in Table 6. As can be seen from Table 6, when testing 9 benchmark functions such as CEC-3, CEC-5, CEC-6, CEC-8, CEC-9, CEC-11, CEC-12, CEC-13, and CEC-15, the test results of CSSA are better than those of the other algorithms. PSO optimizes best when testing functions CEC-3, CEC-4, CEC-7, and CEC-10. Combined with these test results, CSSA is undoubtedly the best optimization algorithm.

B. EVALUATION OF IEEE CEC-2017 TEST FUNCTIONS

Table 5 shows the benchmark functions for CEC-2017. Parameter selection of each algorithm in our test is shown in Table 2, the maximum number of iterations is 1000, the dimension size is shown in Table 5, and the population number is set to 50. The test results are shown in Tables 7 and 8. We can see that the test results of 15 of the 30 test functions of CSSA are better than those of the other four optimization algorithms. These functions are as follows: CEC-1, CEC-2, CEC-5, CEC-6, CEC-7, CEC-9, CEC-12, CEC-13, CEC-15, CEC-18, CEC-19, CEC-24, CEC-25, CEC-27, and CEC-30. PSO has 10 test functions that perform better than the other algorithms, these functions are: CEC-10, CEC-14, CEC-16, CEC-17, CEC-20, CEC-21, CEC-23, CEC-26, CEC-27, and CEC-29. Combined with these test results, CSSA is undoubtedly the best optimization algorithm.

C. ANALYSIS OF STATISTICAL SIGNIFICANCE

Wilcoxon Signed-Rank Test (WSRT), as a nonparametric test, can effectively assess statistical significance difference between two optimizers. The statistical results of WSRT on 45 benchmark functions in 30 runs are presented in Tables 9 to 12, where $T+$ and $T−$ are calculated and their p-values can be obtained. “≈” indicates the case in which there is no significance difference between CSSA and its competitor, “+” indicates that CSSA performs worse than the comparison algorithm at the 95% significance level ($a = 0.05$), and “−” indicates that CSSA performs better than the comparison algorithm. We can clearly see from the table that the optimization effect of CSSA is obviously better than other

---

**TABLE 2. Parameter settings.**

| Algorithm | Parameters | Values |
|-----------|------------|--------|
| PSO       | Inertia coefficient $w$ | 0.9    |
|           | Parameter $c_1$ | 2      |
|           | Parameter $c_2$ | 2      |
| WOA       | Control parameter $a_1$ | [2, 0] |
|           | Control parameter $a_2$ | [-1, -2] |
| GWO       | Control parameter $a$ | [2, 0] |
| TSA       | Parameter $F_{win}$ | 1      |
|           | Parameter $F_{max}$ | 4      |
| BOA       | Conversion probability $p$ | 0.8    |
|           | The power $a$ | [0.1, 0.3] |
|           | Feeling factor $c$ | 0.01   |
| SBO       | Greatest step size $a$ | 0.94   |
|           | Mutation probability $p$ | 0.04   |
|           | Scaling factor $z$ | 0.02   |
| PIO       | Landmark operator $N_{i1}$ | Round ($Max_{opt}$ × 0.7) |
|           | Compass operator $N_{i2}$ | Max$_{opt}$ − $N_{i1}$ |
| ISSA      | The number of the producers | 70%    |
|           | The sparrow is aware of the danger of gravity | 20%    |
|           | The early warning value | 0.6    |
| LLSSA     | The number of the producers | 70%    |
|           | The sparrow is aware of the danger of gravity | 20%    |
|           | The early warning value | 0.6    |
|           | $L$ | Random [-1, 1] |
|           | Coefficient of change $K$ | 5      |
| SSA       | The number of the producers | 70%    |
|           | The sparrow is aware of the danger of gravity | 20%    |
|           | The early warning value | 0.6    |
| CSSA      | The number of the producers | 70%    |
|           | The sparrow is aware of the danger of gravity | 20%    |
|           | The early warning value | 0.6    |
|           | $M_1$ | 0.85    |
|           | $M_ε$ | 0.15    |
TABLE 3. Comparison of optimization results of SSA, CSSA, GWO, PSO, and WOA.

| FUNC | Attribute | SSA         | CSSA        | GWO         | PSO         | WOA         |
|------|-----------|-------------|-------------|-------------|-------------|-------------|
| F1   | AVG       | 5.62E-79    | 9.32E-78    | 1.01E-15    | 3.00E+00    | 1.07E-09    |
|      | STD       | 9.32E-78    | 1.370s      | 3.82E-14    | 8.75E+00    | 5.24E-09    |
|      | Runtime   | 0.493s      | 1.370s      | 1.588s      | 1.024s      | 0.970s      |
| F2   | AVG       | 3.60E-03    | 9.01E-03    | 6.14E-04    | 6.01E-05    | 3.25E-02    |
|      | STD       | 9.01E-03    | 0.927s      | 0.390s      | 0.236s      | 0.171s      |
|      | Runtime   | 0.296 s     | 0.927s      | 0.390s      | 0.236s      | 0.171s      |
| F3   | AVG       | 1.32E-08    | 7.69E-11    | 5.29E-08    | 3.17E+00    | 5.78E-05    |
|      | STD       | 2.81E-07    | 3.24E-11    | 8.93E-07    | 2.85E+00    | 3.73E-05    |
|      | Runtime   | 0.425s      | 1.483s      | 2.491s      | 1.272s      | 1.462s      |
| F4   | AVG       | 4.06E-02    | 3.78E-04    | 1.32E-02    | 1.45E-03    | 9.36E-02    |
|      | STD       | 8.34E-02    | 7.18E-04    | 4.96E-02    | 5.76E-03    | 1.01E-01    |
|      | Runtime   | 0.166 s     | 0.447s      | 0.208 s     | 0.195 s     | 0.142 s     |
| F5   | AVG       | 0.00E+00    | 0.00E+00    | 2.12E-03    | 2.21E-03    | 2.31E-03    |
|      | STD       | 0.00E+00    | 0.823s      | 2.92E-03    | 2.77E-03    | 2.84E-03    |
|      | Runtime   | 0.277 s     | 0.823s      | 0.512 s     | 0.362s      | 0.312 s     |
| F6   | AVG       | 0.00E+00    | 0.00E+00    | 2.47E-08    | 1.17E+02    | 3.74E-03    |
|      | STD       | 0.00E+00    | 0.00E+00    | 7.20E-08    | 1.21E+02    | 8.45E-03    |
|      | Runtime   | 0.322 s     | 1.135s      | 1.627 s     | 0.853 s     | 1.001 s     |
| F7   | AVG       | 0.00E+00    | 0.00E+00    | 1.04E-14    | 6.22E-01    | 7.24E-10    |
|      | STD       | 0.00E+00    | 0.00E+00    | 5.68E-13    | 8.29E-01    | 1.53E-09    |
|      | Runtime   | 0.882 s     | 2.128s      | 2.474 s     | 1.154s      | 1.351s      |
| F8   | AVG       | 2.14E-238   | 1.06E-301   | 6.91E-11    | 1.90E-06    | 1.04E-83    |
|      | STD       | 0.00E+00    | 0.00E+00    | 5.41E-09    | 8.51E-06    | 6.00E-75    |
|      | Runtime   | 0.167 s     | 0.660s      | 0.202s      | 0.173s      | 0.125s      |
| F9   | AVG       | -2.06E+00   | -2.06E+00   | -2.06E+00   | -2.06E+00   | -2.06E+00   |
|      | STD       | 9.36E-11    | 1.26E-07    | 5.67E-07    | 7.48E-06    | 4.91E-03    |
|      | Runtime   | 0.156 s     | 0.608s      | 0.201 s     | 0.176 s     | 0.131 s     |

TABLE 4. IEEE CEC-2015 benchmark test functions.

| No.  | Functions                                      | Related basic functions                                      | Dim | f_{max} |
|------|------------------------------------------------|-------------------------------------------------------------|-----|---------|
| CEC-1| Rotated Bent Cigar Function                     | Bent Cigar Function                                           | 30  | 100     |
| CEC-2| Rotated Discus Function                         | Discus Function                                               | 30  | 200     |
| CEC-3| Shifted and Rotated Weienerstrass Function      | Weienerstrass Function                                        | 30  | 300     |
| CEC-4| Shifted and Rotated Schwefel’s Function         | Schwefel’s Function                                           | 30  | 400     |
| CEC-5| Shifted and Rotated Katsuura Function           | Katsuura Function                                             | 30  | 500     |
| CEC-6| Shifted and Rotated HappyCat Function           | HappyCat Function                                             | 30  | 600     |
| CEC-7| Shifted and Rotated HGBat Function              | HGBat Function                                                | 30  | 700     |
| CEC-8| Shifted and Rotated Expanded Griewank’s plus    | Griewank’s Function; Rosenbrook’s Function                    | 30  | 800     |
|      | Rosenbrook’s Function                           |                                                             |     |
| CEC-9| Shifted and Rotated Expanded Scaffer’s F6       | Expanded Scaffer’s F6 Function                                 | 30  | 900     |
| CEC-10| Hybrid Function 1 (N = 3)                       | Schwefel’s Function; Rastrigin’s Function; High Conditioned   | 30  | 1000    |
|      |                                                | Elliptic Function                                            |
| CEC-11| Hybrid Function 2 (N = 4)                       | Griewank’s Function; Weienerstrass Function; Rosenbrook’s     | 30  | 1100    |
|      |                                                | Function; Scaffer’s F6 Function                               |
| CEC-12| Hybrid Function 3 (N = 5)                       | Katsuura Function; HappyCat Function; Schwefel’s              | 30  | 1200    |
|      |                                                | plus Rosenbrook’s Function; Ackley’s Function                 |
| CEC-13| Composition Function 1 (N = 5)                  | Rosenbrook’s Function; Bent Cigar Function; Discuss Function;| 30  | 1300    |
|      |                                                | High Conditioned Elliptic Function; High Conditioned Elliptic |
| CEC-14| Composition Function 2 (N = 3)                  | Schwefel’s Function; Rastrigin’s Function; High Conditioned   | 30  | 1400    |
|      |                                                | Elliptic Function                                            |
| CEC-15| Composition Function 3 (N = 5)                  | HGBat Function; Rastrigin’s Function; Schwefel’s; Weienerstrass Function; High Conditioned Elliptic Function |

algorithms in the whole test function set. Compared with SSA, ISSA, and LLSSA, the optimization result of CSSA is obviously a global optimization algorithm. These results also prove that CSSA can better solve the local convergence problem.

VI. SPEED REDUCER DESIGN PROBLEM

The main purpose of this engineering design problem is to minimize the weight of the reducer [29], as shown in Fig. 6. When dealing with this problem, the following requirements need to be met [30]: bending stress of the gear teeth, surface stress, transverse deflections of the shafts, stresses in the shafts.

The design problem of this project has seven design variables: face width (b), module of teeth (m), number of teeth in the pinion (p), length of the first shaft between bearings (l1), length of the second shaft between bearings (l2), diameter of first (d1) shafts, and diameter of second shafts (d2). The
specific mathematical formula for this engineering design problem is as follows:

Consider:

\[ \overrightarrow{z} = [z_1 \ z_2 \ z_3 \ z_4 \ z_5 \ z_6 \ z_7] \]
\[ = [b \ m \ p \ l_1 \ l_2 \ d_1 \ d_2]. \]

Minimize:

\[ f(\overrightarrow{z}) = 0.7854z_1z_2^2(3.3333z_3^2 + 14.9334z_3 - 43.0934) \]
\[ -1.508z_1(z_2^2 + z_7^2) + 7.4777(z_6^3 + z_3^3) \]
\[ +0.7854(z_4z_2^2 + z_5z_7^2) \]

Subject to:

\[ g_1(\overrightarrow{z}) = \frac{27}{z_1z_2z_3} - 1 \leq 0, \]
\[ g_2(\overrightarrow{z}) = \frac{397.5}{z_1z_2z_3} - 1 \leq 0, \]
\[ g_3(\overrightarrow{z}) = \frac{1.93z_3^3}{z_2z_6z_3} - 1 \leq 0, \]
\[ g_4(\overrightarrow{z}) = \frac{1.93z_3^3}{z_2z_4z_3} - 1 \leq 0, \]
\[ g_5(\overrightarrow{z}) = \frac{[745(z_4z_2z_3)]^2 + 16.9 \times 10^6}{110z_6} - 1 \leq 0, \]
\[ g_6(\overrightarrow{z}) = \frac{[745(z_5z_2z_3)]^2 + 157.5 \times 10^6}{85z_7} - 1 \leq 0, \]

\[ g_7(\overrightarrow{z}) = \frac{z_2z_3}{40} - 1 \leq 0, \]
\[ g_8(\overrightarrow{z}) = \frac{z_1}{12z_2} - 1 \leq 0, \]
\[ g_9(\overrightarrow{z}) = \frac{1.5z_6 + 1.9}{z_4} - 1 \leq 0, \]
\[ g_{10}(\overrightarrow{z}) = \frac{1.1z_7 + 1.9}{z_5} - 1 \leq 0, \]

where, 2.6 \leq z_1 \leq 3.6, 0.7 \leq z_2 \leq 0.8, 17 \leq z_3 \leq 28, 7.3 \leq z_4 \leq 8.3, 7.3 \leq z_5 \leq 8.3, 2.9 \leq z_6 \leq 3.9, 5.0 \leq z_7 \leq 5.5.

After testing, the test results are shown in Table 13 and Fig. 7. The optimal solution of SSA is [3.5459843, 0.70607238,17.0560762,7.51461454,8.18743412,3.48360511,
TABLE 5. IEEE CEC-2017 benchmark test functions.

| No. | Functions                          | Dim | \( f_{	ext{opt}} \) |
|-----|-----------------------------------|-----|---------------------|
| CEC-1 | Shifted and Rotated Bent Cigar Function | 30  | 100                 |
| CEC-2 | Shifted and Rotated Sum of Different Power Function | 30  | 200                 |
| CEC-3 | Shifted and Rotated Zakharov Function | 30  | 300                 |
| CEC-4 | Shifted and Rotated Rosenbrock’s Function | 30  | 400                 |
| CEC-5 | Shifted and Rotated Rastrigin’s Function | 30  | 500                 |
| CEC-6 | Shifted and Rotated Expanded Scatter’s Function | 30  | 600                 |
| CEC-7 | Shifted and Rotated Lunacek Bi_Rastrigin’s Function | 30  | 700                 |
| CEC-8 | Shifted and Rotated Non-Continuous Rastrigin’s Function | 30  | 800                 |
| CEC-9 | Shifted and Rotated Levy Function | 30  | 900                 |
| CEC-10 | Shifted and Rotated Schwefel’s Function | 30  | 1000                |
| CEC-11 | Hybrid Function 1 (N = 3) | 30  | 1100                |
| CEC-12 | Hybrid Function 2 (N = 3) | 30  | 1200                |
| CEC-13 | Hybrid Function 3 (N = 3) | 30  | 1300                |
| CEC-14 | Hybrid Function 4 (N = 4) | 30  | 1400                |
| CEC-15 | Hybrid Function 5 (N = 4) | 30  | 1500                |
| CEC-16 | Hybrid Function 6 (N = 4) | 30  | 1600                |
| CEC-17 | Hybrid Function 7 (N = 5) | 30  | 1700                |
| CEC-18 | Hybrid Function 8 (N = 5) | 30  | 1800                |
| CEC-19 | Hybrid Function 9 (N = 5) | 30  | 1900                |
| CEC-20 | Hybrid Function 10 (N = 6) | 30  | 2000                |
| CEC-21 | Composition Function 1 (N = 3) | 30  | 2100                |
| CEC-22 | Composition Function 2 (N = 3) | 30  | 2200                |
| CEC-23 | Composition Function 3 (N = 4) | 30  | 2300                |
| CEC-24 | Composition Function 4 (N = 4) | 30  | 2400                |
| CEC-25 | Composition Function 5 (N = 4) | 30  | 2500                |
| CEC-26 | Composition Function 6 (N = 5) | 30  | 2600                |
| CEC-27 | Composition Function 7 (N = 5) | 30  | 2700                |
| CEC-28 | Composition Function 8 (N = 5) | 30  | 2800                |
| CEC-29 | Composition Function 9 (N = 3) | 30  | 2900                |
| CEC-30 | Composition Function 10 (N = 3) | 30  | 3000                |

5.2899798], the corresponding optimal value is 3100.355814. At the same time, the optimal solution of CSSA is [3.5108267, 0.7014563, 1.2580983, 8.3310331, 2.2537360, 5.8581286]. The optimal value is 3100.355814. It can be clearly seen from the test results that CSSA’s optimization results are better than SSA when dealing with reducer design problems.

VII. WELDED BEAM DESIGN PROBLEM

The main purpose of this engineering design problem is to minimize the manufacturing cost of welded beams, as shown in Fig. 8. When dealing with this problem, the following requirements need to be met [16]: shear stress (\( \tau \)), bending stress (\( \theta \)) in the beam, buckling load (\( P_c \)) on the bar, end deflection (\( \delta \)) of the beam.

This project design problem has four design variables: thickness of weld (\( h \)), length of the clamped bar (\( l \)), height of the bar (\( t \)), thickness of the bar (\( b \)). The specific mathematical formula for this engineering design problem is as follows:

\[
\mathbf{z} = [z_1 z_2 z_3 z_4] = [h l t b]
\]
TABLE 7. AVG and STD deviation of best optimal solution for 30 independent runs on CEC-2017 benchmark test functions (CEC1-CEC15).

| FUNC | CEC1 | CEC2 | CEC3 | CEC4 | CEC5 | CEC6 | CEC7 | CEC8 | CEC9 | CEC10 | CEC11 | CEC12 | CEC13 | CEC14 | CEC15 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| PSO  | 4.05E+03 | 9.82E+03 | 2.47E+04 | 5.44E+02 | 5.68E+02 | 6.00E+02 | 8.51E+02 | 7.80E+02 | 7.09E+03 | 2.02E+03 | 1.23E+03 | 5.87E+04 | 1.58E+04 | 2.48E+04 | 1.48E+04 |
| WOA  | 4.14E+00 | 9.45E+00 | 2.18E+00 | 5.95E+02 | 7.71E+02 | 6.73E+02 | 1.25E+03 | 1.01E+03 | 9.46E+03 | 6.52E+03 | 2.78E+03 | 9.48E+07 | 2.30E+05 | 2.06E+06 | 1.01E+05 |
| GWO  | 3.83E+01 | 5.78E+00 | 8.05E+00 | 9.04E+00 | 8.86E+02 | 6.87E+02 | 1.32E+03 | 1.12E+03 | 9.46E+03 | 8.54E+03 | 6.27E+02 | 9.65E+09 | 4.95E+09 | 1.56E+09 | 1.01E+08 |
| TSA  | 4.96E+00 | 1.24E+00 | 9.02E+00 | 1.32E+00 | 1.32E+00 | 7.15E+00 | 5.57E+00 | 1.71E+01 | 9.18E+02 | 4.17E+02 | 1.43E+01 | 2.25E+09 | 2.67E+09 | 6.84E+05 | 3.98E+08 |
| BOA  | 2.37E+00 | 1.42E+00 | 5.25E+02 | 6.14E+02 | 2.82E+02 | 6.24E+01 | 7.15E+01 | 1.71E+01 | 1.01E+02 | 9.71E+00 | 1.11E+03 | 9.69E+05 | 2.09E+05 | 5.66E+04 | 4.10E+06 |

TABLE 8. AVG and STD deviation of best optimal solution for 30 independent runs on CEC-2017 benchmark test functions (CEC16-CEC30).

| FUNC | CEC16 | CEC17 | CEC18 | CEC19 | CEC20 | CEC21 | CEC22 | CEC23 | CEC24 | CEC25 | CEC26 | CEC27 | CEC28 | CEC29 | CEC30 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| PSO  | 2.43E+01 | 1.96E+03 | 1.00E+06 | 1.06E+00 | 1.82E+00 | 2.25E+00 | 2.37E+00 | 4.56E+00 | 2.72E+03 | 2.74E+03 | 2.91E+03 | 4.24E+03 | 3.52E+03 | 3.21E+03 | 5.91E+03 |
| WOA  | 3.48E+00 | 2.63E+00 | 3.81E+00 | 5.28E+00 | 6.02E+00 | 6.02E+00 | 6.02E+00 | 6.02E+00 | 6.02E+00 | 6.02E+00 | 5.38E+00 | 5.38E+00 | 5.38E+00 | 5.38E+00 | 5.38E+00 |
| GWO  | 4.98E+00 | 3.18E+00 | 1.30E+00 | 1.33E+00 | 2.98E+00 | 8.08E+00 | 5.04E+00 | 3.83E+00 | 3.83E+00 | 3.83E+00 | 9.37E+00 | 4.78E+00 | 3.19E+00 | 3.19E+00 | 3.19E+00 |
| BOA  | 3.49E+00 | 3.40E+00 | 9.92E+00 | 1.74E+00 | 3.31E+00 | 1.14E+00 | 3.02E+00 | 3.02E+00 | 3.02E+00 | 3.02E+00 | 3.02E+00 | 3.02E+00 | 3.02E+00 | 3.02E+00 | 3.02E+00 |
| SSA  | 1.72E+00 | 4.59E+00 | 5.52E+00 | 6.21E+00 | 6.21E+00 | 6.21E+00 | 6.21E+00 | 6.21E+00 | 6.21E+00 | 6.21E+00 | 6.21E+00 | 6.21E+00 | 6.21E+00 | 6.21E+00 | 6.21E+00 |
| ILSA | 4.01E+00 | 5.40E+00 | 9.80E+00 | 1.25E+00 | 1.14E+00 | 1.64E+00 | 6.12E+00 | 5.47E+00 | 5.47E+00 | 5.47E+00 | 5.47E+00 | 5.47E+00 | 3.48E+00 | 3.48E+00 | 3.48E+00 |

TABLE 9. Statistical comparisons of CSSA vs. PSO, WOA, GWO, TSA, and BOA. (CEC-2015).

| FUNC | CEC16 | CEC17 | CEC18 | CEC19 | CEC20 | CEC21 | CEC22 | CEC23 | CEC24 | CEC25 | CEC26 | CEC27 | CEC28 | CEC29 | CEC30 |
|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| CSSA | 3.40E+03 | 9.88E+03 | 9.54E+03 | 3.51E+01 | 9.38E+00 | 2.38E+01 | 3.02E+00 | 5.93E+01 | 1.56E+02 | 4.56E+01 | 9.46E+03 | 2.36E+04 | 4.59E+03 | 1.17591 |
TABLE 10. Statistical comparisons of CSSA vs. SBO, TLBO, PIO, and SSA. (CEC-2015).

| FUNC | CSSA vs. SBO | CSSA vs. TLBO | CSSA vs. PIO | CSSA vs. SSA | p-value | T-value | Significance |
|------|--------------|---------------|--------------|--------------|---------|---------|-------------|
| CEC-1 | 2.055 × 10² | 413 52       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-2 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-3 | 1.00 × 10² | 0 465       | 1.00 × 10² | 0 465       | 1.00 × 10² | 0 465   | 1.00 × 10² | 0 465   |
| CEC-4 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-5 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-6 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-7 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-8 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-9 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |

TABLE 11. Statistical comparisons of CSSA vs. PSO, WOA, GWO, TSA, and BOA. (CEC-2017).

| FUNC | CSSA vs. PSO | CSSA vs. WOA | CSSA vs. GWO | CSSA vs. TSA | CSSA vs. BOA | p-value | T-value | Significance |
|------|--------------|--------------|--------------|--------------|--------------|---------|---------|-------------|
| CEC-1 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-2 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-3 | 1.00 × 10² | 0 465       | 1.00 × 10² | 0 465       | 1.00 × 10² | 0 465   | 1.00 × 10² | 0 465   |
| CEC-4 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-5 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-6 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-7 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-8 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-9 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |

TABLE 12. Statistical comparisons of CSSA vs. SBO, TLBO, PIO, and SSA. (CEC-2017).

| FUNC | CSSA vs. SBO | CSSA vs. TLBO | CSSA vs. PIO | CSSA vs. SSA | p-value | T-value | Significance |
|------|--------------|---------------|--------------|--------------|---------|---------|-------------|
| CEC-1 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-2 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-3 | 1.00 × 10² | 0 465       | 1.00 × 10² | 0 465       | 1.00 × 10² | 0 465   | 1.00 × 10² | 0 465   |
| CEC-4 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-5 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-6 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-7 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-8 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |
| CEC-9 | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465       | 1.731 × 10³ | 0 465   | 1.731 × 10³ | 0 465   |

\[
0.1 \leq z_3 \leq 10.0, 0.1 \leq z_4 \leq 2.0, \\
\tau (z) = \sqrt{(\tau')^2 + (\tau'')^2 + (l \tau' \tau'')/0.25(l^2 + h \tau')}, \\
\delta (z) = 65850000/(30 \times 10^6) \tau'^3.
\]
The comparison between the optimal solution and the optimal value of the reducer problem.

| Algorithm | SSA          | CSSA         |
|-----------|--------------|--------------|
| $b$       | 3.5459843    | 3.51058267   |
| $m$       | 0.70607238   | 0.70141566   |
| $p$       | 17.05650762  | 17.02162146  |
| $l_1$     | 7.51461454   | 7.48680426   |
| $l_2$     | 8.18743412   | 8.11370926   |
| $d_1$     | 3.48360511   | 3.35948839   |
| $d_2$     | 5.2899798    | 5.32394411   |
| $g_1$     | -0.10455608  | -0.08159821  |
| $g_2$     | -0.22710295  | -0.20566232  |
| $g_3$     | -0.5382335   | -0.46741284  |
| $g_4$     | -0.88768303  | -0.89252475  |
| $g_5$     | -0.11034235  | -0.00796445  |
| $g_6$     | -0.00181118  | -0.02078878  |
| $g_7$     | -0.69892178  | -0.70151921  |
| $g_8$     | -0.00440565  | -0.00099823  |
| $g_9$     | -0.58148952  | -0.58291699  |
| $g_{10}$  | -0.05179332  | -0.07314324  |
| $g_{11}$  | -0.05272165  | -0.04404661  |
| Optimal value | 3100.358143 | 3045.542957 |

The comparison between the optimal solution and the optimal value of the welded beam design problem.

| Algorithm | SSA          | CSSA         |
|-----------|--------------|--------------|
| $h$       | 0.13561309   | 0.17324276   |
| $l$       | 6.77516019   | 4.00413158   |
| $t$       | 9.74777439   | 9.793008     |
| $b$       | 0.18708584   | 0.18093193   |
| $g_1$     | -1840.089235 | -33.15998228 |
| $g_2$     | -1648.328237 | -954.216793  |
| $g_3$     | -0.23733175  | -0.23708155  |
| $g_4$     | -0.05147276  | -0.00768917  |
| $g_5$     | -16449.1839  | -148662.4156 |
| $g_6$     | -0.01061309  | 0.04824276   |
| $g_7$     | -3.17532837  | -3.46210299  |
| Optimal value | 1.96039428  | 1.66751421   |

After testing, the test results are shown in Table 14 and Fig. 9. The optimal solution of SSA is [0.13561309, 6.77516019, 9.74777439, 0.18708584], the corresponding optimal value is 1.96039428. At the same time, the optimal solution of CSSA is [0.17324276, 4.00413158, 9.793008, 0.18093193], the corresponding optimal value is 1.66751421. It can be clearly seen from the test results that CSSA is better than SSA when dealing with the design problems of welded beams.

VIII. APPLICATION TEST OF CSSA ALGORITHM IN NETWORK INTRUSION DETECTION

In order to verify the feasibility and improvement of the CSSA proposed in this paper in practical application, the Support Vector Machine (SVM) model commonly used in network intrusion detection [31]–[33] was selected in this paper. The CSSA can be used to solve the parameter optimization problem of the model, that is, to find the optimal solution of
TABLE 15. Comparison of SVM optimization results.

| Models     | Test set accuracy | DoS prediction accuracy | Probe prediction accuracy | R2L prediction accuracy | U2R prediction accuracy |
|------------|-------------------|-------------------------|--------------------------|-------------------------|-------------------------|
| GA-SVM     | 88.21%            | 86.25%                  | 47.36%                   | 43.75%                  | 94.28%                  |
| SSA-SVM    | 91.37%            | 93.91%                  | 75.20%                   | 91.48%                  | 89.13%                  |
| CSSA-SVM   | 94.26%            | 96.12%                  | 81.59%                   | 95.28%                  | 92.76%                  |

FIGURE 10. Distribution of network intrusion detection prediction results.

As can be seen from Table 15, compared with GA and SSA, the overall prediction accuracy of CSSA in the optimized SVM model is 3% higher than that of SSA and 6% higher than that of GA. In addition, for different attacks, the prediction accuracy of intrusion detection models built according to this algorithm has increased to different degrees. From the 10 randomly selected experimental results, the scatter diagram of the prediction results of the SVM model and the actual results is shown in Fig. 10.

IX. CONCLUSION

This article mainly aims at SSA easy to fall into local optimal problems, introduces the improved chaotic mapping Circle, as well as the t distribution variation, and puts forward a kind of function to measure similarity of sparrow cluster divergence to improve the original sparrow algorithm. Circle chaotic mapping can maximum limit increase the randomness of the sparrow population distribution, t distribution variation can satisfy different iterations the sparrow population for value to update the size of the size, the similarity function can be used to measure each iteration cycle the sparrow population aggregation and dispersion, thereby minimize CSSA in a local convergence problem. The algorithm is based on different population distribution of adaptive iteration times and adjust the sparrow position. In order to test the performance of the algorithm, this paper mainly carries out the following five experiments: In experiment 1, we mainly test and compare the existing good algorithm with the test function of the original SSA and CSSA in the optimization, the experimental results show that the CSSA can better solve the problem of local convergence of the original algorithm, at the same time greatly increases the convergence speed. In Experiment 2, we tested and compared CSSA with ten other latest optimization algorithms. CEC-2015 and CEC-2017 were selected as Test function sets, and Wilcoxon Sign-Rank Test was used. In experiment 3 and experiment 4, we used CSSA and SSA to optimize problem reducer design problem and welded beam design problem. The experimental results show that the test results of CSSA are better than that of SSA, because CSSA has a strong global optimization capability. In Experiment 5, we applied CSSA to the network intrusion detection model and constructed the CSSA-SVM model. The experimental results show that the CSSA-SVM model has higher prediction accuracy, which verifies the feasibility of the algorithm in practical application. These experimental results show that CSSA has been able to greatly solve the local convergence problem and is an optimization algorithm with local and global optimization ability.

Even though CSSA has a strong performance and performance in the optimization performance, it has a relatively obvious improvement on the original SSA algorithm, but there is also a certain direction of improvement, that is, the running cost of CSSA algorithm is also increased compared with SSA, which will be one of the directions for the next improvement. In addition to that, the application of CSSA in specific practical projects still needs further research and verification, which will also be the main direction of future improvement and development.
REFERENCES

[1] N. H. Khan, Y. Wang, D. Tian, R. Jamal, S. Iqbal, M. A. A. Saif, and M. Ebeed, “A novel modified lightning attachment procedure optimization technique for optimal allocation of the FACTS devices in power systems,” IEEE Access, vol. 9, pp. 47976–47997, 2021, doi: 10.1109/ACCESS.2021.3059201.

[2] M. H. Hassan, S. Kanel, S. Q. Salih, T. Khurshaid, and M. Ebeed, “Developing chaotic artificial ecosystem-based optimization algorithm for combined economic emission dispatch,” IEEE Access, vol. 9, pp. 51146–51165, 2021, doi: 10.1109/ACCESS.2021.3069914.

[3] M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” IEEE Trans. Syst. Man, Cybern., B, Cybern., vol. 26, no. 1, pp. 29–40, Dec. 1994.

[4] W. Zhao, Z. Zhang, and L. Wang, “Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications,” Eng. Appl. Artif. Intell., vol. 87, Jan. 2020, Art. no. 103300, doi: 10.1016/j.engappai.2019.103300.

[5] K. N. Krishnamanand and D. Ghose, “Glowlworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions,” Swarm Intell., vol. 3, no. 2, pp. 87–124, Jun. 2009, doi: 10.1007/s11721-008-0021-5.

[6] M. Liu, K. Luo, J. Zhang, and S. Chen, “A stock selection algorithm hybridizing grey wolf optimizer and support vector regression,” Expert Syst. Appl., vol. 179, Oct. 2021, Art. no. 115078, doi: 10.1016/j.eswa.2021.115078.

[7] W. Zhao, L. Wang, and Z. Zhang, “Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm,” Neurocomputing, vol. 323, no. 13, pp. 9383–9425, Jul. 2020, doi: 10.1007/978-3-030-36368-0_7.

[8] S. Kaur, L. K. Awasthi, A. L. Sangal, and G. Dhiman, “Tunicate swarm intelligence optimization algorithm,” IEEE Access, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2959260.

[9] G. Xiang, C. Xie, L. Zou, and M. Ebeed, “A novel modified lightning attachment procedure optimization technique to improve the neural network performance with aid of adaptive GA operators,” Expert Syst. Appl., vol. 12, no. 2, pp. 1559–1576, 2019, doi: 10.1007/s11721-008-0021-5.

[10] S. H. S. Moosavi and V. R. Badrishi, “Satin bowser def: Improved binary gray wolf optimizer and SVM for intrusion detection system in wireless sensor networks,” J. Ambient Intell. Hum. Comput., vol. 12, no. 2, pp. 1559–1576, Feb. 2021, doi: 10.1007/s12652-020-02228-z.

WANG ZHIHENG received the bachelor’s degree in engineering from Southwest Petroleum University, in 2019. He is currently pursuing the master’s degree in computer science with Xi’an University of Posts and Telecommunications. His current research interests include biologically inspired algorithms and deep learning.

LIU JIANHUA was born in September 1963. He received the degree from Beijing Institute of Posts and Telecommunications, in July 1986. He worked with the 10th Research Institute of the Ministry of Posts and Telecommunications, Xi’an, in 1986, transferred to the Institute of Communication Technology, Xi’an University of Posts and Telecommunications, in September 1994, and has since been involved in scientific research and teaching with the Department of Information and Control and the School of Computer Science. He was appointed as the Director of the Information Security Laboratory, in 2004, the Deputy Director of the Information and Control Department, in 2006, and the Vice President of the Computer Science College, in 2009. He is currently a Senior Engineer.