IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 June 2023, accepted 2 July 2023, date of publication 5 July 2023, date of current version 12 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3292338

RESEARCH ARTICLE

An Improved Sand Cat Swarm Operation and Its Application in Engineering

YINGDA HU, RUIPING XIONG, JING LI, CHENGSHENG ZHOU, AND QIYUAN WU
School of Mechanical Engineering, Sichuan University, Chengdu 610065, China
Corresponding author: Ruiping Xiong (xiongruiping214@outlook.com)

ABSTRACT Meta-heuristic algorithms have emerged as a popular approach for solving engineering optimization problems. In this paper, an Improved Sand Cat Swarm Operation (ISCSO) is proposed and applied to optimize double-layer spraying path parameters. The Sand Cat Swarm Operation (SCSO) has some limitations, such as poor initial population quality, slow convergence speed, and a tendency to fall into local optima. To overcome these limitations, three improvement strategies are introduced in ISCSO. Firstly, the SPM chaotic mapping is used to enhance the initial population quality. Secondly, a nonlinear cycle adjustment strategy is introduced to balance global exploration and local exploitation, thereby accelerating the convergence speed. Finally, integrating the Immune Algorithm (IA) enables ISCSO to avoid falling into local optima, resulting in improved solution accuracy. Furthermore, we extended our experiments to include 21 low-dimensional functions and 15 test functions of LSOPs, where ISCSO was compared with seven other popular algorithms. The experimental results highlight the promising performance of ISCSO in solving different types of functions, achieving both higher solution accuracy and faster convergence speed. In particular, the effectiveness of ISCSO has been demonstrated through experiments aimed at optimizing the parameters of the double-layer spraying path. The results of these experiments further highlight the utility of ISCSO in tackling challenging optimization problems.

INDEX TERMS Sand cat swarm optimization, chaotic mapping, immune algorithm, nonlinear periodic adjustment, spray parameter optimization.

I. INTRODUCTION

Optimization problems are widely present in various fields of science and engineering technology, making it a popular research topic [1]. Meta-heuristic algorithms have emerged as effective problem-solving approaches for optimization. These algorithms can be classified into four major categories: evolution-based algorithms, group intelligence-based algorithms, human-based algorithms, and physical and chemical methods-based algorithms.

Evolution-based algorithm simulate the natural law of survival of the fittest in evolution. Through evolution, the overall progress of the population is achieved, and the optimal solution is eventually attained. Examples of such algorithms include Genetic Algorithm (GA) [2] and Differential Evolution (DE) [3]. As well as other proposed algorithms like Evolutionary Strategy (ES) [4], Evolutionary Programming (EP) [5], Gene Expression Programming (GEP) [6], Genetic Programming (GP) [7], Covariance Matrix Adaptation Evolution Strategy(CMA-ES) [8], Biogeography Based Optimization(BBO) [9], among others.

Population intelligence-based optimization algorithms simulate the cooperative behavior among members of the population to determine the overall best solution. For instance, Particle Swarm Optimization (PSO) [10] simulates the foraging behavior of birds, Ant Colony Optimization (ACO) [11] simulates the behavior of ant colonies, Cuckoo Search(CS) [12] simulates the egg-laying behavior of cuckoos, Firefly Algorithm (FA) [13] simulates the courtship behavior of fireflies, Gray Wolf Optimization (GWO) [14] and Whale Optimization Algorithm(WOA) [15] simulate the foraging behavior of animals, Bacterial Foraging(BF) [16]

The associate editor coordinating the review of this manuscript and approving it for publication was Xiwang Dong.

simulates the behavior of bacteria foraging, Harris Hawks optimization (HHO) [17] simulates the hunting behavior of Harris Hawks, and so on.

Human-based algorithms simulate human behavior in teaching, social, learning, emotional, and managerial behaviors to solve optimization problems. Examples include Teaching-Learning-Based Optimization (TLBO) [18], Harmony Search (HS) [19], Soccer League Competition (SLC) [20], Election Algorithm (EA) [21], Ideology Algorithm (IA) [22], Group Search Optimizer (GSO) [23], Imperialist Competitive Algorithm (ICA) [24], Social-Based Algorithm (SBA) [25].

Physical and chemical methods-based algorithms mimic the laws of physics and chemistry to solve optimization problems. For instance, Simulated Annealing (SA) [26] simulates the process of annealing of matter, Gravitational Local Search Algorithm(GLSA) [27] simulates Newton's law of gravity, Big-Bang Big-Crunch (BBBC) [28] based on the Big Bang and contraction theory, Charged System Search (CSS) [29] based on Coulomb's Law and Newton's Law, Artificial Chemical Reaction Optimization Algorithm (ACROA) [30] based on chemical reaction principles, Galaxy-based Search Algorithm (GbSA) [31] simulates the spiral motion of galaxies, Ray Optimization (RO) [32] simulates the law of refraction of light, and Water Cycle Algorithm (WCA) [33] simulates the water cycle process, among others.

In conclusion, metaheuristic algorithms have proven to be effective problem-solving approaches for optimization in various fields. The different categories of algorithms provide powerful tools for solving complex engineering optimization problems [34].

The Sand Cat Swarm Optimization(SCSO) [35] algorithm, proposed by Amir Seyyedabbasi and Farzad Kiani in 2022, simulates the hunting behavior of sand cats. Sand cats are known for their ability to detect low-frequency noise, which helps them locate prey both above and below the ground. SCSO is characterized by its simplicity, low parameter settings, and high performance in finding the best solution.

SCSO has been increasingly applied in engineering optimization, with researchers developing various approaches to enhance its performance. For example, Li et al. [36] introduced an elite collaboration and stochastic variation mechanism to improve the convergence accuracy of SCSO. Iraji et al. [37] used chaotic sequences to diversify the initial population of SCSO and combined it with a pattern search algorithm to evaluate the minimum safety factor of slopes. Jovanovic et al. [38] incorporated the Artificial Bee Colony (ABC) [39] algorithm to enhance SCSO's ability to escape local optima and used it to optimize an intrusion detection system. Wu et al. [40] added a wandering strategy to improve SCSO's ability to jump out of local optima. Seyyedabbasi et al. [41] successfully applied SCSO to solve the inverse kinematics of a robotic arm. Kiani et al. [42] improved SCSO's search capability by randomly selecting a new position between the optimal candidate solution and the current position. Seyyedabbasi et al. [43] combined SCSO with reinforcement learning techniques to further enhance the algorithm's performance. The above improvements to the algorithm have improved the performance of the algorithm, but there are still some shortcomings. For instance, the initialized populations lack sufficient diversity, the conversion settings of global exploration and local search are unclear, and the algorithm can easily converge to a local optimum. In the experimental section of this paper, the ISCSO is compared with two other existing variants SCSO.

In the field of spraying, there are also many algorithms used to optimize the spray trajectory. Biyu et al. [44] proposed an improved Particle Swarm Optimization algorithm based on edge composite operator for optimal combination of spray trajectories of torch with improved efficiency. Wang et al. [45] improved the traditional RRT algorithm and applied it to the path planning of autonomous spraying robots in orchards with good results. Fu et al. [46] developed a new method for complex surface segmentation in spray trajectory planning for two robots using genetic algorithms.

Despite its successes, SCSO does have some limitations. Firstly, the initial population in SCSO is created randomly, resulting in unstable quality. This affects the effectiveness of the global exploration phase. Secondly, SCSO balances local exploitation with global exploration by using linearly decreasing sensitivity, resulting in slow convergence in the initial phase before quickly achieving the local optimum. Finally, as the population converges towards the ideal individual, population diversity decreases in later iterations, making it easier for SCSO to get stuck in local optima.

To address these issues, we propose an Improved Sand Cat Swarm Optimization (ISCSO), in which three strategies are used to improve its comprehensive performance on diverse problems. The specific contributions of this paper are as follows:

1. SPM chaotic mapping is introduced to increase the diversity of the ISCSO's initial population. This overcomes the problem of poor initial population quality in SCSO.
2. Adding a nonlinear periodic adjustment mechanism to balance the algorithm's local exploitation and global search capabilities.
3. We calculated and ranked the incentive degree among individuals. The ability of ISCSO to escape local optima was greatly improved by treating the top 20% of individuals with cloning, mutation and cloning suppression.

This essay is divided into the following sections: Section II introduces SCSO. Section III presents the improvement strategy and algorithm steps of ISCSO. Section IV is the experimental results and analysis of test functions. Section V applies the ISCSO algorithm to the optimize of the double-layer spraying path parameters, and the results are compared and analyzed. Section VI concludes this paper.

II. SAND CAT SWARM OPTIMIZATION

The two primary activities of sand cats are simulated by the Sand Cat Swarm Optimization (SCSO): searching and attacking. In the wild, sand cats are solitary creatures. In order to emphasize the idea of population intelligence, the proposed algorithm assumes that sand cats live in colonies. The SCSO algorithm employs sand cats to represent various problem variables.

In the SCSO algorithm, the initialized sand cat population is first randomly generated. Assuming that the sand cat population size is N, and the search space is d-dimensional, the location of the i-th sand cat is labeled as $X_i = \{x_1, x_2, x_3, \cdots x_d\}$, $i = 1, 2, 3 \cdots N$. The location of the prey corresponds to the global optimal solution of the problem. Then, a global search is started and the sand cats use equation (1) to update the location.

$$\overrightarrow{X}(t+1) = \overrightarrow{r} \cdot \left(\overrightarrow{X}_b(t) - rand \cdot \overrightarrow{X}_c(t)\right) \tag{1}$$

where t denotes the current number of iterations, $\overrightarrow{X}_b(t)$ is the optimal individual position, $\overrightarrow{X}_c(t)$ is the current individual position of the sand cat. rand is a random number within [0, 1], and $\overrightarrow{r}$ simulates the sensitivity range of the sand cat, which is described as follows:

$$\overrightarrow{r} = \overrightarrow{r_G} \times rand \tag{2}$$

where $\overrightarrow{r_G}$ simulate of the sand cat's sensitivity, whose value declines linearly from 2 to 0 repetitions. The equation for $\overrightarrow{r_G}$ is described as:

$$\overrightarrow{r_G} = S_M - \left(\frac{S_M \times t}{t_{max}}\right) \tag{3}$$

where S_M simulates the auditory characteristics of the sand cat with a value of 2. t_{max} is the maximum number of iterations.

During the local development phase, SCSO updates each sand cat's position as follows:

$$\overrightarrow{X_{md}} = \mid rand \cdot \overrightarrow{X_b}(t) - \overrightarrow{X_c}(t) \mid \tag{4}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_b}(t) - \overrightarrow{r} \cdot \overrightarrow{X_{md}} \cdot cos(\theta) \tag{5}$$

where θ is a random angle between $[0, 360]$, obtained randomly using a roulette wheel. $\overrightarrow{X_{md}}$ denotes a random position.

The SCSO method switches between a local search and a global search depending on the value of the parameter $\overrightarrow{R}$. The formula for calculating $\overrightarrow{R}$ is described as follows:

$$\overrightarrow{R} = 2 \times \overrightarrow{r_G} \times rand - \overrightarrow{r_G} \tag{6}$$

From equation (6), it can be seen that $\overrightarrow{R}$ is a random value in the interval $[-\overrightarrow{r_G}, \overrightarrow{r_G}]$ and decreases gradually with the linear decrease of $\overrightarrow{r_G}$. When the value of $\overrightarrow{R}$ is in $[-1, 1]$, the sand cat individual updates its position by Eq. (5), otherwise it updates its position by Eq. (1). the latest individual position update of the SCSO algorithm is as follows:

$$\overrightarrow{X}(t+1) = \begin{cases} \overrightarrow{r} \cdot \left(\overrightarrow{X_b}(t) - rand \cdot \overrightarrow{X}_c(t)\right) & |\overrightarrow{R}| > 1 \\ \overrightarrow{X_b}(t) - \overrightarrow{r} \cdot \overrightarrow{X_{md}} \cdot \cos(\theta) & |\overrightarrow{R}| \leqslant 1 \end{cases} \tag{7}$$

In conclusion, the SCSO algorithm benefits from a straightforward structure, simple implementation, and minimal parameters. However, the initial population it generates lacks diversity. During the process of global exploration and local exploitation, SCSO is susceptible to falling into local optima and yielding low precision results.

III. IMPROVED SAND CAT SWARM OPTIMIZATION

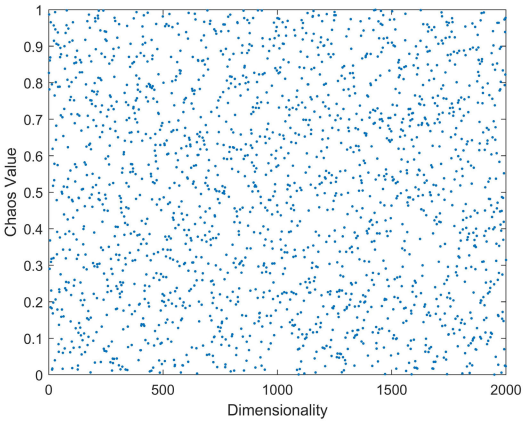
A. POPULATION INITIALIZATION BASED ON CHAOTIC MAPPING

Deng et al. [47] proposed that the quality of the initial population has a significant impact on both the accuracy and convergence speed of meta-heuristic algorithms. A diverse initial population can greatly enhance the performance of the algorithm. However, SCSO employs a random technique to generate the initial population, resulting in an uneven distribution and low diversity. Additionally, it is essential to distribute the population evenly across the search space since predicting the location of the global best solution is challenging.

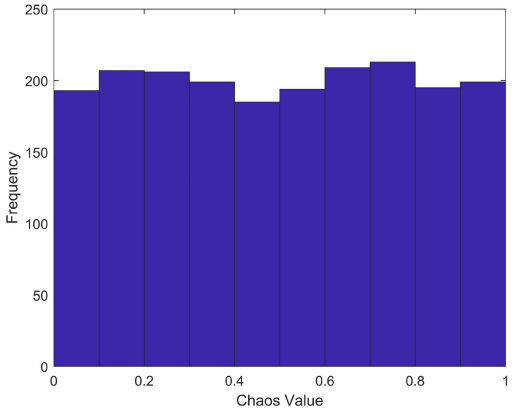
Therefore, ISCSO adopts the SPM chaotic mapping [48] which has better ergodicity and faster iteration speed. SPM chaotic mapping is a novel mapping method proposed by Ban et al. in 2020. This mapping method extends the range of chaotic mappings and guarantees the ergodicity of the results. This mapping greatly improves the efficiency of chaotic sequence generation without compromising security. The functional representation of SPM chaotic mapping is given by equation (8).

$$y(t+1) \begin{cases} mod\left(\frac{y(t)}{\eta} + \mu sin(\pi y(t) + r, 1)\right) \\ \quad 0 \leqslant y(t) < \eta \\ mod\left(\frac{y(t)/\eta}{0.5 - \eta} + \mu sin(\pi y(t) + r, 1)\right) \\ \quad \eta \leqslant y(t) < 0.5 \\ mod\left(\frac{1 - y(t)/\eta}{0.5 - \eta} + \mu sin(\pi(1 - y(t))) + r, 1\right) \\ \quad 0.5 \leqslant y(t) < 1 - \eta \\ mod\left(\frac{1 - y(t)}{\eta} + \mu sin(\pi(1 - y(t))) + r, 1\right) \\ \quad 1 - \eta \leqslant y(t) < 1 \end{cases} \tag{8}$$

Figure 1(a) shows the distribution of SPM chaos mapping in 2000 dimensions, and Figure 1(b) shows the statistical histogram of this distribution. It can be seen that the SPM mapping has a very good chaotic effect.



(a) **SPM chaotic mapping distribution at 2000 dimensions.**



(b) **Statistical histogram of the SPM chaos mapping distribution at 2000 dimensions.**

FIGURE 1. **SPM chaotic mapping distribution statistics.**

The initial population is obtained by mapping the SPM chaotic sequence into the solution space. A dimensional solution of the population of people is shown by Eq (9).

$$X_{id} = X_{min} + y(t)\,(X_{maxd} - X_{mind}) \tag{9}$$

where, X_{maxd} and X_{mind} are the upper and lower bounds of the search in the d-th dimension. The initial population of ISCSO is obtained according to the above method.

Based on the above introduction, the ISCSO's population initialization process is detailed as Algorithm 1.

Algorithm 1 Initializing the population

Input: μ =0.3,η =0.4, X_{maxd}, X_{mind}
01: Generate chaotic series **y(t)** according to **Eq.(8)**;
02: **For** i=1 to N
03: Calculate $\mathbf{X_i}$ according to **Eq.(9)**;
04: **End For**
Output: X

B. NONLINEAR CONVERGENCE FACTOR

The transition of SCSO between global exploration and local exploitation is mostly influenced by changes in the parameter $\overrightarrow{R}$. A larger $\overrightarrow{R}$ prevents becoming stuck in a local optimum and has stronger global search capabilities. A smaller $\overrightarrow{R}$ speeds up the algorithm's convergence and provides a stronger local exploitation capacity.

The parameter $\overrightarrow{R}$ in SCSO has a range of $[-\overrightarrow{r_G}, \overrightarrow{r_G}]$ and is linearly reduced during iterations from 2 to 0. However, the linearly decreasing strategy provides the algorithm a superior global exploration capability in the early stages of SCSO but a slower convergence speed. Although the speed of convergence is accelerated in the later period, it still frequently falls into the local optimum. Therefore, we expect that the parameter $\overrightarrow{R}$ will maintain a high value early on and gradually decline as rounds increase. When the SCSO reaches a certain number of iterations, the parameter $\overrightarrow{R}$ is then rapidly reduced to a smaller value. As a result, the following equation (10) introduces a nonlinear convergence factor.

$$\overrightarrow{r_G} = 2 - 2\left(\frac{e^{t/t_{max}} - 1}{e} - a\right)^{\mu} \tag{10}$$

where μ and a are constant coefficients, which take values greater than 0.

C. FUSION IMMUNIZATION ALGORITHM

In the late stage of the SCSO search, to avoid falling into the local optimum, this paper improves SCSO by fusing the Immune Algorithm [49]. In the Immun Algorithm, the better individuals are immunized according to the incentive degree of antibodies. This can ensure the diversity of the population and avoid falling into local optimum.

To determine whether the algorithm is trapped in a local optimum, this paper introduces the optimal fitness change rate $RC(t)$. The judgment formula is as follows.

$$RC(t) = \left|\frac{f(X_b(t)) - f(X_b(t-n))}{f(X_b(t))}\right| \le \Delta, t > n \tag{11}$$

where, $f\,(X_b\,(t))$ is the optimal individual fitness at the t-th iteration. If the optimal fitness change rate $RC(t)$ is less than a threshold Δ in n consecutive generations, it indicates that the population has fallen into a local optimum. Then, the immunization operation is performed on the population.

$ED(X_i)$ is the individual incentive degree of the population. The population's individual incentive degree is first calculated before the immunization procedure is carried out. The formula for the computation is as follows.

$$ED(X_i) = \alpha \cdot fit(X_i) + \beta \cdot ND(X_i) \tag{12}$$

where, α, β is the weighting factor, and $ND(X_i)$ is the current individual concentration. It is calculated as follows.

$$ND(X_i) = \frac{1}{N}\sum_{j=1}^{N} S(X_i, X_j) \tag{13}$$

$$S(X_i, X_j) = \begin{cases} 1 & aff\,(X_i, X_j) < \delta_s \\ 0 & aff\,(X_i, X_j) \ge \delta_s \end{cases} \tag{14}$$

where N is the population size, $S(X_i, X_j)$denotes the similarity between individuals of the population. $aff(X_i, X_j)$ denotes the Euclidean distance between individual i and individual j. δ_s denotes the similarity threshold between population individuals.

The population individuals are graded based on the level of incentive. Then, the individuals in the top 20% of the incentive degree ranking are cloned and mutated. The mathematical expression of the above operation is as follows.

$$T_c(X_i) = clone(X_i) \tag{15}$$

$$T_g(X_{m,i,j}) = \begin{cases} X_{m,i,j} + (r - 0.5) \cdot \lambda & rand < p_m \\ X_{m,i,j} & others \end{cases} \tag{16}$$

where $T_c(X_i)$ denotes the set consisting of m clones identical to X_i. $X_{m,i,j}$ denotes the j-th dimension of the m-th clone of an individual X_i. $T_g(X_{m,i,j})$ is the set of clones of X_i after mutation. λ is the range of the defined neighborhood. r is the random number between [0, 1]. p_m is the probability of mutation.

The mutant individuals are then put under clonal suppression. Clonal suppression refers to the elimination of low affinity individuals and the retention of high affinity individuals. In order to create a new population, the individuals who underwent the immunization operation are finally combined with those who did not.

Based on the above introduction, the immunization operations process is detailed as algorithm 2.

Algorithm 2 Immunization operations

Input: $\mathbf{X}$,$f(X)$, α, β, δ_s, λ, p_m, m
For i=1 to N
  Calculate $\boldsymbol{ND(X_i)}$ based on **Eq.(13)**and **Eq.(14);**
  Calculate$\boldsymbol{ED(X_i)}$ based on **Eq.(12);**
End For
Sort $\boldsymbol{ND(X_i)}$; $ND(X_1) < ND(X_2) < \ldots < ND(X_d)$;
Select the top 20% of individuals according to the ranking of $\boldsymbol{ND(X_i)}$;
For i=1 to N/5 **Do**
  Cloning m individuals $\boldsymbol{X_i}$ based on **Eq.(15);**
  For g=1 to m
    Generate new individuals$\boldsymbol{T_g(X_i)}$ based on **Eq.(16);**
  End For
  Sort $\mathbf{T_g(X_j)}$ satisfying;
    $f(T_1(X_i)) < f(T_2(X_i)) < \ldots < f(T_g(X_i))$
  If $f(T_1(X_i)) < f(X_i)$
    $\boldsymbol{X_i = T_1(X_i)}$
  End If
End For
Output:X

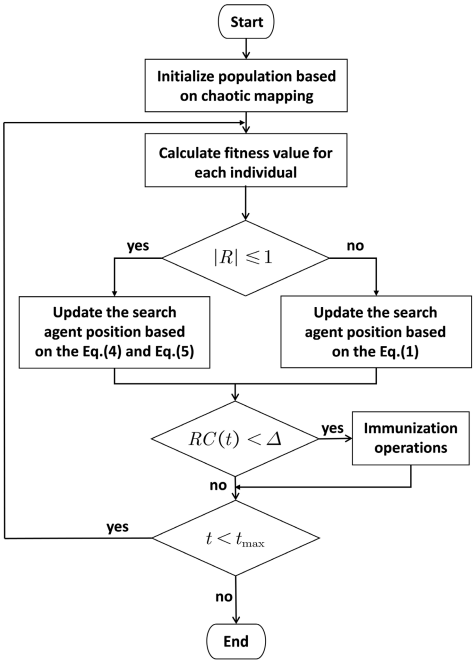


FIGURE 2. Flowchart of ISCSO.

D. FRAMEWORK OF ISCSO

By incorporating the aforementioned components, the pseudocode of ISCSO is shown in Algorithm 3, the complete fowchart of ISCSO is described as Fig. 1.

E. COMPUTATIONAL COMPLEXITY

In this paper, the ISCSO algorithm is composed of four main components: population initialization, calculation of the nonlinear convergence factor, updating population locations, and immunization operations. The computational complexity of the algorithm can be analyzed as follows, given a population size N and problem dimension D.

The population initialization has a computational complexity of $O(ND)$. The calculation of the nonlinear convergence factor has a complexity of $O(1)$. Updating individual positions has a computational complexity of $O(N)$. In the immunization operation phase, the incentive degree of individuals in the population is calculated and ranked, with a computational complexity of $O(2Nlog(N))$. he top 20% of individuals are then cloned and mutated, with a computational complexity of $O(2Nlog(10))$ When the number of clones $m = 10$.

Therefore, if the algorithm is stopped after T generations, the overall computational complexity of ISCSO can be expressed as $O(ND + T(O(1) + O(N)) + O(2Nlog(N)) + O(2Nlog(10)))$. Note that, different operators have different time usages.

IV. EXPERIMENTAL RESULTS AND COMPARISONS

To verify the performance of ISCSO, we conducted experiments using benchmarking functions. We selected

Algorithm 3 ISCSO pseudocode

/* Initialization */
Initialize population size N,**Δ**
Initialize the populations $\mathbf{X_i}$ according to **Algorithm 1;**
Evaluate $\mathbf{X_i}$, $\overrightarrow{\mathbf{X_b}}$=$\mathbf{X_i}$;
/* Main Loop */
Evaluate $\mathbf{X_i}$, Update $\overrightarrow{\mathbf{X_b}}$;
For t=1 to t_{max} **Do**
  Calculate $\overrightarrow{\mathbf{r_G}}$ based on **Eq.(10);**
  Calculate $\overrightarrow{\mathbf{r}}$, $\overrightarrow{\mathbf{R}}$ based on **Eq.(2)** and **Eq.(6);**
  For i = 1 to N **Do**
    Get a random angle based on Roulette Wheel Selection($0° < \theta < 360°$);
    If abs $\left(\overrightarrow{\mathbf{R}}\right) \leq 1$
      Update $\mathbf{X_i}$ based on **Eq.(4)** and **Eq.(5);**
    else
      Update $\mathbf{X_i}$ based on **Eq.(1);**
    End If
  End For
  Evaluate $\mathbf{X_i}$;
  Calculate **RC**(**t**) based on **Eq.(11);**
  If RC(t) =≤ Δ
    Perform immunization operations according to **Algorithm 2;**
  End If
  t=t+1;
End For
Output results.

21 low-dimensional functions that are commonly used by researchers, as well as 15 test functions of Large-Scale Optimization Problems (LSOPs) that are internationally recognized. This allowed us to test the ability of ISCSO to solve LSOPs problems.

A. EXPERIMENTAL SETUP

1) PEER ALGORITHMS AND BENCHMARK FUNCTIONS

For comparative analysis, we selected seven state-of-the-art algorithms, including RLLPSO [50], TAPSO [51], MAPSO [52], XPSO [53], WOA, GWO, SCSO, ISCSO1 [54] and SESCSO [36]. as peer algorithms in this article. The parameter settings of all peer algorithms are summarized in Table 1. It is worth noting that the parameter settings of the selected peer algorithm are the same as those in the reference paper.

Details of 21 low dimensional functions can be found in Table 2. In the 21 functions, F1-F7 are unimodal functions. The unimodal functions can test the local convergence ability of the algorithm. F8-F13 are multimodal functions. Multimodal function can test the ability of the algorithm to jump out of the local optimum. F14-F21 for the fixed-dimension multimodal function. The fixed-dimensional multimodal function can test the solution accuracy of the algorithm.

TABLE 1. Basic information of the seven algorithms.

Algorithm	Year	Parameters Settings
RLLPSO	2022	$\emptyset = 0.4, \alpha = 0.4, \gamma = 0.8, \epsilon = 0.9$
TAPSO	2020	$\omega = 0.7298, p_c = 0.5, p_{m=}0.02, M = N/4$
MAPSO	2020	$IG_{max} = 2, SG_{max} = 2, F = 1.0, CR = 0.5$
XPSO	2020	$\eta = 0.2, stag_{max} = 5, p = 0.5$
WOA	2016	/
GWO	2014	/
SCSO	2022	/
ISCSO1	2023	/
SESCSO	2022	/

/ There have no parameters that need to be set.

The details of the 15 test functions of LSOPs are shown in Table 3. Their dimensions are all set to 200,500,1000 dimensions.

After repeated experiments, the optimal parameters in ISCSO were:$\mu = 0.9, a = 0.1$ in the convergence factor; $\Delta = 0.00001$ in the optimal fitness change rate threshold, $\alpha = 1$ and $\beta = 1$ in the weight coefficient, $\delta_s = 0.1$ in the similarity threshold, and $p_m = 0.7$ in the variation coefficient in the immunization operation.

In the experiments, the population size of all algorithms was set to D=60 and the number of iterations was Maxiter=500. The maximum fitness evaluation was set as the population size multiplied by the maximum number of iterations. MaxFES=D× Maxiter. All algorithms were individually run 50 times for each test function. All of the algorithms are coded in MATLAB2020 and run on a PC with an AMD Ryzen 9 5900HS CPU@3.30 GHz/16 GB RAM. (Note that only a single processor is used.)

2) PERFORMANCE METRICS

The article utilizes four basic performance metrics, namely average value (Ave), standard deviation (Std), rank of average value (Rank), and two-tailed t-test results. It is important to note that freedom at a 0.05 level of significance is adopted in the t-test. The tables present the t-test results as "+," "−," and "=," which indicate that ISCSO is significantly better than, significantly worse than, or almost the same as the corresponding competitor algorithms, respectively.

B. COMPARISON ON 21 LOW-DIMENSIONAL FUNCTIONS

In this section, the results of 21 low-dimensional functions are presented and analyzed statistically. The best results for the average and standard deviation on each problem among all algorithms are highlighted in bold black.

1) UNIMODAL FUNCTIONS (F1-F7)

Table 4 provides the optimization results for seven unimodal functions. Based on the average value and standard deviation of the experimental results, it is observed that ISCSO performs the best on F1, F2, F3, F4, and F7. MAPSO shows the best performance on F5, while XPSO performs the best on F6.

TABLE 2. 21 low dimensional functions.

Function	Type	Dim	Range	f_{min}
$F1(x)=\sum_{i=1}^{n}x_i^2$	U	30	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	U	30	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}(\sum_{j=1}^{n}x_j)^2$	U	30	[-100,100]	0
$F_4(x)=\max_i\{\lvert x_i\rvert,1\le i\le n\}$	U	30	[-100,100]	0
$F_5(x)=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	U	30	[-30,30]	0
$F_6(x)=\sum_{i=1}^{n}(x_i+0.5)^2$	U	30	[-100,100]	0
$F_7(x)=\sum_{i=1}^{n}ix_i^4+rand[0,1)$	U	30	[-1.28,1.28]	0
$F_8(x)=\sum_{i=1}^{n}-x_i\sin(\sqrt{\lvert x_i\rvert})$	M	30	[-500,500]	$-418.9829\times dim$
$F_9(x)=[x_i^2-10\cos(2\pi x_i)+10]$	M	30	[-5.12,5.12]	0
$F_{10}(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	M	30	[-32,32]	0
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}(\cos(\frac{x_i}{\sqrt{i}}))+1$	M	30	[-600,600]	0
$F_{12}(x)=\frac{\pi}{n}\left\{10\sin(\pi y_1)+\sum_{i=1}^{n}(y_i-1)^2[1+10\sin^2(\pi_{i+1})]\right\}+\sum_{i=1}^{n}u(x_i,10,100,4)$ $y_i=1+\frac{x_i+1}{4},u(x_i,a,k,m)=\begin{cases}k(x_i-a)m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)m & x_i<-a\end{cases}$	M	30	[-50,50]	0
$F_{13}(x)=0.1\times\{10\sin^2 3\pi x_i+\sum_{i=1}^{n}(x_i-1)^2[1+\sin^2(3\pi x_i+1)]+(x_n-1)^2[1+\sin^2(3\pi x_n)]\}+\sum_{i=1}^{n}u(x^i,5,100,4)$	M	30	[-50,50]	0
$F_{14}(x)=(\frac{1}{500}+\sum_{j=1}^{25}(\frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}))^{-1}$	C	2	[-65,65]	1
$F_{15}(x)=\sum_{i=1}^{11}[a_i-\frac{x_i(b_i^2+b_ix_2)}{b_i^2+b_ix_3+x_4}]^2$	C	4	[-5,5]	0.00030
$F_{16}(x)=4x_i^2-2.1x_i^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4$	C	2	[-5,5]	-1.0316
$F_{17}(x)=(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6)^2+10(1-\frac{1}{8\pi})\cos x_1+10$	C	2	[-5,5]	0.398
$F_{18}(x)=[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)]\times[30$ $+(2x_1-3x_2)^2\times(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)]$	C	2	[-2,2]	3
$F_{19}(x)=-\sum_{i=1}^{4}(c_i\exp(-\sum_{j=1}^{3}a_{ij}(x_j-p_{ij})2))$	C	3	[1,3]	-3.86
$F_{20}(x)=-\sum_{i=1}^{4}(c_i\exp(-\sum_{j=1}^{6}a_{ij}(x_j-p_{ij})2))$	C	6	[0,1]	-3.32
$F_{21}(x)=-\sum_{i=1}^{5}[(X-a_i)(X-a_i)^T+c_i]-1$	C	4	[0,10]	-10.1532

Furthermore, based on the rank and t-test results, it is evident that ISCSO significantly outperforms the other optimization algorithms on F1, F2, F3, and F4. Additionally, ISCSO is not significantly inferior to other algorithms on F5 and F6. As unimodal functions have only one global optimal, the relevant algorithms' performance is evaluated during the global exploitation phase. The experimental results demonstrate that ISCSO is successful in the global exploitation phase.

2) MULTIMODAL FUNCTIONS (F8-F13)

The table in Figure 5 presents the optimization results of six multimodal functions. By examining the average value and standard deviation of each test function, it is evident that ISCSO exhibits exceptional performance in F9, F10, AND F11. In fact, ISCSO was able to identify the theoretical optimal value of both F9 AND F11. ON the other hand, WOA performed best in F8, While XPSO demonstrated superior performance in F12 AND F13. When considering the ranking results, it is clear that ISCSO outperforms the other comparison algorithms significantly or almost equivalently, with the exception of F12 AND F13. The multimodal function features a single global optimal solution and multiple local optimal solutions, thereby testing an algorithm's ability to escape from local optima. The experimental results reveal that ISCSO exhibits a stronger capacity to escape from local optima compared to other algorithms.

TABLE 3. 15 large-scale standard test functions.

Function	Dim	Range	f_{min}
$F1(x)=\sum_{i=1}^{n}x_i^2$	200/500/1000	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	200/500/1000	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}ix_i^2$	200/500/1000	[-10,10]	0
$F_4(x)=\sum_{i=1}^{n-1}\lvert x_i\rvert^{(i+1)}$	200/500/1000	[-1,1]	0
$F_5(x)=\sum_{i=1}^{n}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	200/500/1000	[-5,10]	0
$F_6(x)=\sum_{i=1}^{n}ix_i^4+rand[0,1)$	200/500/1000	[-1.28,1.28]	0
$F_7(x)=\frac{\pi}{n}\left\{10\sin(\pi y_1)+\sum_{i=1}^{n}(y_i-1)^2[1+10\sin^2(\pi_{i+1})]\right\}+\sum_{i=1}^{n}u(x_i,10,100,4)$ $y_i=1+\frac{x_i+1}{4},u(x_i,a,k,m)=\begin{cases}k(x_i-a)m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)m & x_i<-a\end{cases}$	200/500/1000	[-50,50]	0
$F_8(x)=[x_i^2-10\cos(2\pi x_i)+10]$	200/500/1000	[-5.12,5.12]	0
$F_9(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	200/500/1000	[-32,32]	0
$F_{10}(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}(\cos(\frac{x_i}{\sqrt{i}}))+1$	200/500/1000	[-600,600]	0
$F_{11}(x)=\sum_{i=1}^{n}\lvert x_i\cdot\sin(x_i)+0.1\cdot x_i\rvert$	200/500/1000	[-10,10]	0
$F_{12}(x)=\sum_{i=1}^{n-1}(x_n-1)^2[1+\sin^2(3\pi x_{i+1})]+\sin^2(3\pi x_1)+\lvert x_n-1\rvert[1+\sin^2(3\pi x_n)]\}$	200/500/1000	[-10,10]	0
$F_{13}(x)=0.1n-(0.1\sum_{i=1}^{n}\cos(5\pi x_i)-\sum_{i=1}^{n}x_i^2)$	200/500/1000	[-1,1]	0
$F_{14}(x)=0.1\cdot\sin^2(3\pi x_1)+\sum_{i=1}^{n-1}(x_i-1)^2[1+\sin^2(3\pi x_{i+1})]+(x_n-1)^2[1+\sin^2(2\pi x_n)]$	200/500/1000	[-5,5]	0
$F_{15}(x)=\sum_{i=1}^{n}((10^6)^{\frac{i-1}{d-1}}x_i^2)$	200/500/1000	[-100,100]	0

TABLE 4. Optimization results on the seven unimodal functions.

Algorithms	F1				F2				F3				F4			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	3.21E+01	1.90E+01	9	+	8.93E+00	1.82E+00	9	+	3.95E+02	1.58E+02	8	+	4.53E+00	9.98E-01	8	+
TAPSO	5.84E+02	2.26E+02	10	+	4.89E+01	8.68E+00	10	+	4.48E+03	2.50E+03	9	+	1.07E+01	1.84E+00	10	+
MAPSO	4.07E-03	3.94E-03	8	+	4.74E-03	4.35E-03	8	+	4.25E-03	3.82E-03	6	+	4.72E-03	3.87E-03	6	+
XPSO	1.82E-11	6.18E-11	6	+	5.15E-04	1.57E-03	7	+	3.10E+00	2.40E+00	7	+	3.03E-01	1.41E-01	7	+
WOA	1.50E-05	9.33E-05	7	+	1.16E-05	6.01E-05	6	+	3.03E+04	2.77E+04	10	+	9.89E+00	1.40E+01	9	+
GWO	9.80E-39	1.38E-38	5	+	7.15E-23	4.96E-23	5	+	7.15E-11	3.57E-10	5	+	3.61E-10	2.62E-10	5	+
SCSO	1.37E-113	7.86E-113	4	+	4.04E-57	1.47E-56	4	+	7.07E-101	4.94E-100	4	+	2.87E-47	1.12E-47	4	+
ISCSO1	1.18E-119	6.41E-119	3	+	4.53E-64	3.17E-63	3	+	3.07E-103	2.13E-102	3	+	1.10E-53	7.69E-52	3	+
SESCSO	1.48E-130	7.52E-130	2	+	4.71E-71	2.56E-70	2	+	7.55E-112	4.76E-111	2	+	1.37E-57	4.63E-57	2	+
ISCSO	**5.29E-177**	**0.00E+00**	1		**1.34E-91**	**7.05E-91**	1		**3.68E-161**	**2.55E-160**	1		**1.92E-83**	**7.72E-83**	1	
Algorithms	F5				F6				F7			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	1.48E+03	1.24E+03	9	+	3.03E+01	1.23E+01	9	+	2.46E-02	1.13E-02	9	+
TAPSO	3.90E+05	2.33E+05	10	+	5.57E+02	1.94E+02	10	+	3.88E+01	2.08E+01	10	+
MAPSO	**3.64E-03**	**3.17E-03**	1	-	9.15E-02	2.32E-01	3	=	4.11E-03	3.07E-03	7	+
XPSO	3.14E+01	1.69E+01	7	+	**1.24E-07**	**3.99E-07**	1	-	2.14E-02	6.81E-03	8	+
WOA	5.45E+01	1.33E+02	8	+	2.18E+00	1.02E+00	8	+	1.52E-03	2.06E-03	6	+
GWO	2.77E+01	8.52E-01	6	=	3.19E-01	2.48E-01	4	+	1.53E-04	1.34E-04	4	+
SCSO	2.76E+01	9.19E-01	5	=	1.41E+00	5.70E-01	7	+	1.53E-04	1.90E-04	5	+
ISCSO1	2.75E+01	9.28E-01	3	=	1.03E+00	3.99E-01	5	+	1.23E-04	1.33E-04	3	+
SESCSO	2.76E+01	1.00E+00	4	=	1.23E+00	5.52E-01	6	+	1.08E-04	1.48E-04	2	+
ISCSO	2.70E+01	4.40E-01	2		1.03E-02	2.22E-02	2		**3.11E-05**	**3.12E-05**	1	

3) FIXED-DIMENSION MULTIMODAL FUNCTIONS (F14-F21)

Table 6 shows the optimization results on eight fixed-dimension multimodal functions. From the average value of each tested function, ISCSO demonstrates the best performance on F14, F15, F16, F17, F18, F19, and F20.

Additionally, ISCSO obtains the theoretical optimal solution of the function on F16, F17, F18, F19, AND F20. GWO Performs best on F21. The standard deviation of ISCSO is either equal to 0 or close to 0 on EACH TEST FUNCTION, indicating its high stability. Fixed-dimensional multi-peaked functions can test an algorithm's

TABLE 5. Optimization results on the six multimodal functions.

Algorithms	F8				F9				F10				F11			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	-3.04E+03	5.65E+02	10	+	1.62E+02	3.57E+01	9	+	3.98E+00	7.04E-01	9	+	1.36E+00	1.95E-01	9	+
TAPSO	-3.07E+03	3.97E+02	9	+	2.69E+02	3.04E+01	10	+	9.95E+00	9.52E-01	10	+	1.14E+01	1.98E+00	10	+
MAPSO	-3.18E+03	1.27E+03	7	+	4.26E-03	4.13E-03	6	+	5.25E-03	5.38E-03	7	+	5.05E-03	6.11E-03	6	+
XPSO	-3.43E+03	7.99E+02	6	+	4.15E+01	1.43E+01	8	+	1.88E-02	1.30E-01	8	+	8.17E-03	8.97E-03	7	+
WOA	**-9.40E+03**	1.59E+03	1	-	8.22E+00	2.39E+01	7	+	5.20E-05	1.69E-04	6	+	2.28E-02	8.23E-02	8	+
GWO	-3.11E+03	**3.08E+02**	8	+	0.00E+00	0.00E+00	1	=	6.71E-15	1.71E-15	5	+	9.04E-04	3.17E-03	5	+
SCSO	-7.05E+03	6.81E+02	4	=	0.00E+00	0.00E+00	1	=	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=
ISCSO1	-7.19E+03	7.08E+02	3	=	0.00E+00	0.00E+00	1	=	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=
SESCSO	-6.97E+03	7.13E+02	5	=	0.00E+00	0.00E+00	1	=	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=
ISCSO	-7.27E+03	7.96E+02	2		**0.00E+00**	**0.00E+00**	1		**8.88E-16**	**0.00E+00**	1		**0.00E+00**	**0.00E+00**	1	
Algorithms	F12				F13											
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test								
RLLPSO	7.80E-01	3.68E-01	9	+	3.91E+00	2.87E+00	9	+								
TAPSO	1.11E+01	4.01E+00	10	+	3.81E+03	7.13E+03	10	+								
MAPSO	4.11E-03	**2.99E-03**	2	=	4.29E-03	4.09E-03	2	-								
XPSO	**2.07E-03**	1.45E-02	1	-	**1.08E-03**	**3.86E-03**	1	-								
WOA	3.86E-01	1.38E+00	8	+	1.41E+00	7.58E-01	4	=								
GWO	3.15E-02	1.35E-02	4	+	2.46E-01	1.67E-01	3	-								
SCSO	4.78E-02	2.19E-02	5	+	2.22E+00	4.32E-01	6	=								
ISCSO1	9.26E-02	6.42E-02	7	+	2.42E+00	5.40E-01	8	=								
SESCSO	8.11E-02	4.46E-02	6	+	2.37E+00	4.64E-01	7	=								
ISCSO	4.44E-03	6.16E-03	3		1.72E+00	9.13E-01	5									

TABLE 6. Optimization results on the eight fixed-dimension multimodal function.

Algorithms	F14				F15				F16				F17			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	3.48E+00	1.79E+00	7	+	4.15E-03	7.12E-03	8	+	-1.03E+00	3.04E-08	8	=	3.98E-01	5.55E-09	7	=
TAPSO	3.33E+00	2.07E+00	6	+	1.00E-02	8.02E-03	10	+	-1.01E+00	3.86E-02	10	+	4.25E-01	4.49E-02	10	+
MAPSO	3.16E+00	3.20E+00	5	+	4.15E-04	3.00E-04	2	=	-1.03E+00	1.52E-10	7	=	3.98E-01	2.20E-08	5	=
XPSO	2.66E+00	1.92E+00	2	+	1.20E-03	3.92E-03	7	+	-1.03E+00	0.00E+00	1	=	3.98E-01	0.00E+00	1	=
WOA	4.21E+00	4.24E+00	9	+	8.57E-04	5.60E-04	6	+	-1.03E+00	1.50E-05	9	=	4.05E-01	1.68E-02	9	+
GWO	3.97E+00	2.95E+00	8	+	4.46E-03	7.96E-03	9	+	-1.03E+00	8.43E-09	6	=	3.98E-01	1.15E-06	8	=
SCSO	3.08E+00	3.23E+00	4	+	4.49E-04	3.38E-04	5	=	-1.03E+00	2.31E-10	5	=	3.98E-01	2.57E-08	6	=
ISCSO1	4.21E+00	4.24E+00	9	+	4.46E-04	3.21E-04	4	=	-1.03E+00	1.94E-11	3	=	3.98E-01	1.11E-09	3	=
SESCSO	2.78E+00	3.21E+00	3	+	4.43E-04	3.16E-04	3	=	-1.03E+00	1.01E-10	4	=	3.98E-01	8.26E-09	4	=
ISCSO	**1.27E+00**	**1.41E+00**	1		**3.49E-04**	**1.35E-04**	1		**-1.03E+00**	**0.00E+00**	1		**3.98E-01**	**0.00E+00**	1	
Algorithms	F18				F19				F20				F21			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	3.00E+00	1.76E-07	1	=	-3.86E+00	8.19E-07	1	=	-3.31E+00	3.57E-02	1	=	-7.30E+00	3.27E+00	4	=
TAPSO	7.00E+00	8.42E+00	10	+	-3.76E+00	1.07E-01	10	+	-2.46E+00	5.91E-01	10	+	-3.34E+00	2.33E+00	10	+
MAPSO	3.00E+00	2.02E-06	1	=	-3.86E+00	3.55E-03	1	=	-3.22E+00	1.06E-01	7	+	-6.23E+00	2.23E+00	6	=
XPSO	3.00E+00	2.66E-15	1	=	-3.86E+00	0.00E+00	1	=	-3.23E+00	6.06E-02	5	+	-7.70E+00	3.22E+00	3	=
WOA	3.55E+00	3.78E+00	9	+	-3.81E+00	1.16E-01	9	+	-3.11E+00	1.76E-01	9	+	-7.83E+00	2.49E+00	2	=
GWO	3.00E+00	2.12E-05	1	=	-3.86E+00	2.53E-03	1	=	-3.25E+00	9.56E-02	3	+	**-9.84E+00**	**1.22E+00**	1	-
SCSO	3.00E+00	2.06E-06	1	=	-3.86E+00	2.35E-03	1	=	-3.21E+00	1.60E-01	8	+	-5.67E+00	1.66E+00	9	+
ISCSO1	3.00E+00	2.97E-06	1	=	-3.86E+00	2.40E-03	1	=	-3.23E+00	1.12E-01	6	+	-5.77E+00	1.77E+00	7	=
SESCSO	3.00E+00	3.48E-06	1	=	-3.86E+00	3.01E-03	1	=	-3.25E+00	8.16E-02	4	+	-6.38E+00	2.23E+00	5	=
ISCSO	**3.00E+00**	**2.18E-06**	1		**-3.86E+00**	**2.83E-07**	1		**-3.31E+00**	**3.58E-02**	1		-5.67E+00	1.65E+00	8	

TABLE 7. T-test results between iscso and other peer algorithms on the 21 low dimensional functions.

Algorithm	#+	#=	#-	CP
RLLPSO	15	6	0	15
TAPSO	21	0	0	21
MAPSO	11	8	2	9
XPSO	13	5	3	10
WOA	17	3	1	16
GWO	13	6	2	11
SCSO	10	11	0	10
ISCSO1	9	12	0	9
SESCSO	9	12	0	9

TABLE 8. Friedman-test of mean values on the 21 low dimensional functions.

Overall Rank	Algorithm	Ranking
1	ISCSO	1.76
2	SESCSO	3.14
3	ISCSO1	3.71
4	SCSO	4.29
5	XPSO	4.52
6	GWO	4.62
7	MAPSO	4.90
8	WOA	7.14
9	RLLPSO	7.29
10	TAPSO	9.71

local exploration ability. The experimental results show that ISCSO has good local exploration ability compared with other comparison algorithms.

Overall, considering the performance of all algorithms on the 21 low-dimensional test functions, although the probabilistic immunity operation of ISCSO makes the number of iterations significantly less than that of the other peer algorithms with the same MAXFES, ISCSO achieves the best results and performs best on 75% of the tested functions. Fig.2 depicts the convergence curves for some functions, providing a more intuitive view of ISCSO'S performance.

4) T-TEST RESULTS

The results of the t-test between ISCSO and other peer algorithms are among the 21 low-dimensional functions are given in Table 7. In the table, "#+", "#=", AND "#-" indicate that the test results of ISCSO are significantly better than, almost the same as, and significantly worse than other peer Algorithms, respectively. The comprehensive performance (CP) is equal to "#+" minus "#−".

As can be seen in Table 7, ISCSO significantly outperforms the other peer algorithms in most of the tested functions.

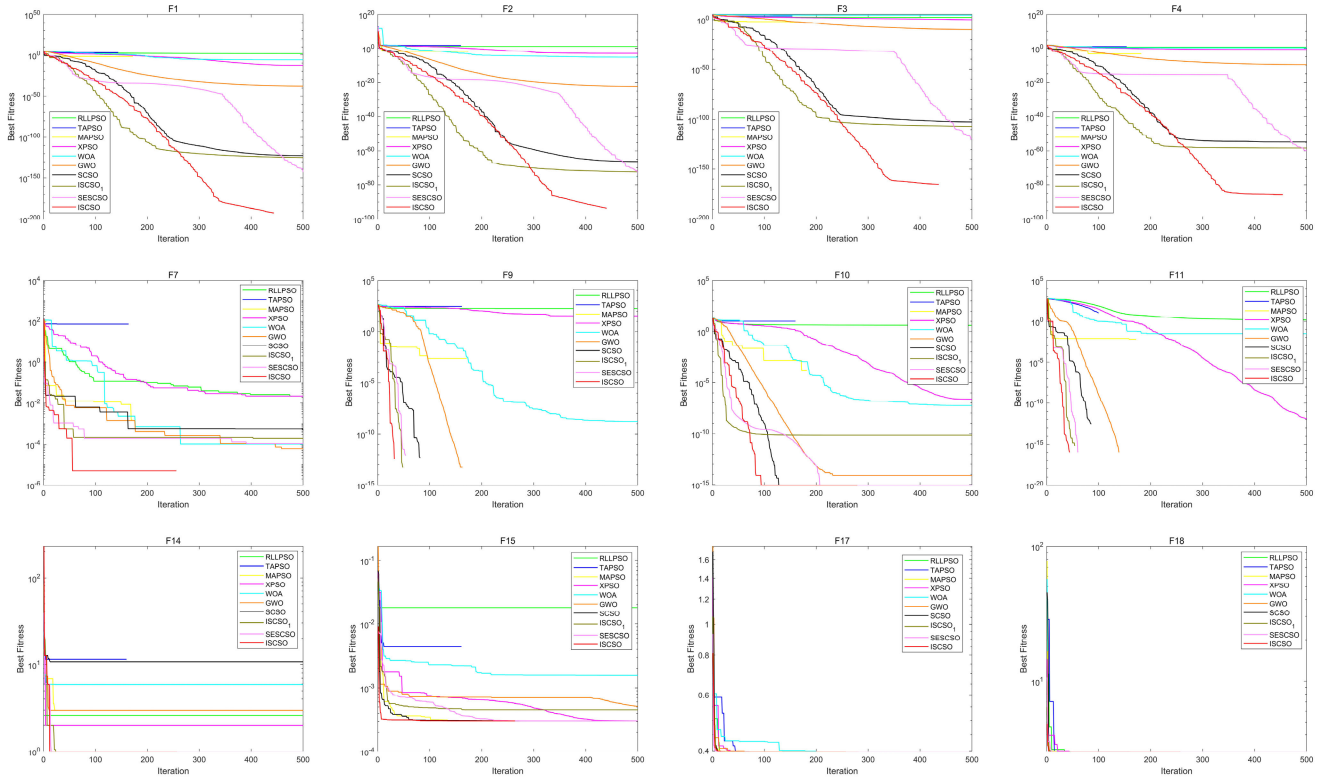


FIGURE 3. The convergence curves of some of the 21 low-dimensional tested functions.

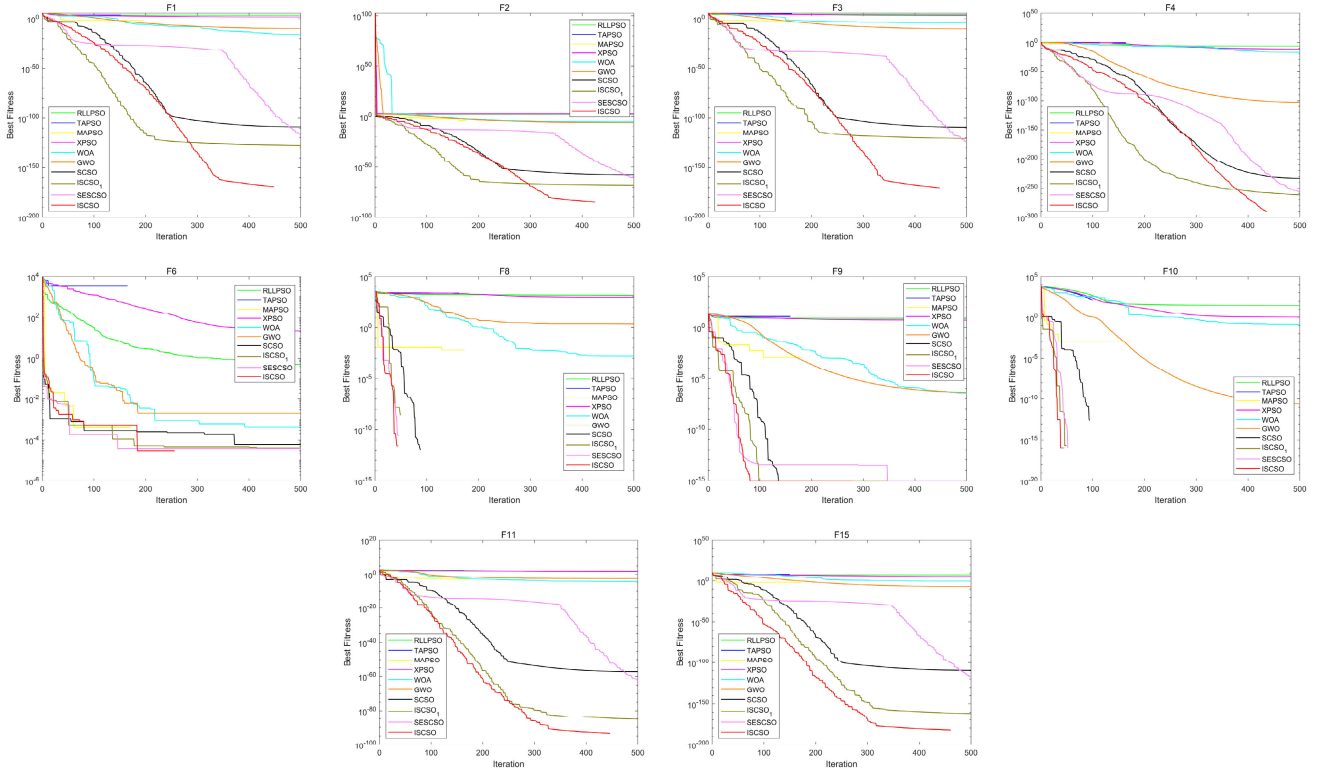


FIGURE 4. The schematic diagram of the model.

TABLE 9. Optimization results on 15 large-scale standard test functions (*D=200*).

Algorithms	F1				F2				F3				F4			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	4.67E+03	5.91E+02	9	+	1.72E+02	1.76E+01	8	+	1.96E+04	2.80E+03	9	+	1.22E-07	1.75E-07	8	+
TAPSO	8.81E+03	1.34E+03	10	+	4.76E+02	4.84E+01	10	+	1.11E+05	1.71E+04	10	+	6.13E-01	3.53E-01	10	+
MAPSO	4.95E-03	5.23E-03	7	+	4.49E-03	2.92E-03	7	+	3.02E-03	3.86E-03	6	+	4.39E-03	4.46E-03	9	+
XPSO	8.59E+01	2.95E+01	8	+	3.87E+02	3.95E+01	9	+	3.33E+03	1.80E+03	8	+	3.00E-13	5.02E-13	6	+
WOA	2.56E-04	4.87E-04	6	+	7.44E-04	1.12E-03	6	+	3.45E-03	8.83E-03	7	+	8.40E-13	1.61E-12	7	+
GWO	8.63E-11	2.40E-11	5	+	4.27E-07	6.86E-08	5	+	6.08E-11	3.77E-11	5	+	2.75E-93	8.26E-93	5	+
SCSO	3.26E-108	4.87E-108	4	+	2.05E-49	6.16E-49	4	+	5.45E-105	1.64E-104	4	+	7.22E-235	0.00E+00	3	+
ISCSO1	4.24E-110	1.25E-109	3	+	6.39E-57	1.48E-56	3	+	8.20E-109	1.93E-108	3	+	1.52E-220	0.00E+00	4	+
SESCSO	6.62E-120	1.76E-119	2	+	2.31E-61	4.81E-61	2	+	2.14E-116	5.85E-116	2	+	1.32E-238	0.00E+00	2	+
ISCSO	**2.03E-165**	**0.00E+00**	1		**8.21E-87**	**1.16E-86**	1		**1.22E-168**	**0.00E+00**	1		**1.78E-276**	**0.00E+00**	1	
Algorithms	**F5**				**F6**				**F7**				**F8**			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	5.90E+05	1.11E+05	9	+	8.29E-01	3.85E-01	8	+	7.58E+00	9.62E-01	9	+	1.41E+03	6.39E+01	9	+
TAPSO	4.33E+06	7.40E+05	10	+	4.68E+03	9.14E+02	10	+	5.94E+03	1.06E+04	10	+	2.34E+03	7.99E+01	10	+
MAPSO	**4.71E-03**	**3.99E-03**	1	-	3.81E-03	3.23E-03	7	+	**5.75E-03**	**5.23E-03**	1	-	6.78E-03	3.68E-03	5	+
XPSO	2.63E+04	8.87E+03	8	+	2.69E+01	8.66E+00	9	+	4.47E+00	8.81E-01	8	+	8.07E+02	7.85E+01	8	+
WOA	1.98E+02	3.35E-01	7	=	2.80E-03	4.03E-03	5	+	6.52E-01	2.31E-01	7	=	1.31E+00	2.65E+00	6	+
GWO	1.98E+02	6.29E-01	2	=	2.87E-03	1.68E-03	6	+	3.79E-01	4.29E-02	4	=	2.15E+00	2.11E+00	7	+
SCSO	1.98E+02	2.89E-01	4	=	2.40E-04	1.90E-04	4	+	3.59E-01	5.48E-02	3	=	0.00E+00	0.00E+00	1	=
ISCSO1	1.98E+02	7.38E-02	5	=	2.08E-04	2.43E-04	3	+	4.46E-01	7.17E-02	6	=	0.00E+00	0.00E+00	1	=
SESCSO	1.98E+02	9.91E-02	6	=	1.30E-04	2.12E-04	2	+	4.05E-01	5.70E-02	5	=	0.00E+00	0.00E+00	1	=
ISCSO	1.98E+02	6.71E-01	3		**8.18E-05**	**5.42E-05**	1		1.27E-01	1.35E-02	2		**0.00E+00**	**0.00E+00**	1	
Algorithms	**F9**				**F10**				**F11**				**F12**			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	8.60E+00	2.39E-01	9	+	3.08E+01	3.47E+00	9	+	1.09E+02	8.39E+00	9	+	1.05E-07	1.36E-07	2	-
TAPSO	1.23E+01	5.54E-01	10	+	1.06E+02	6.75E+00	10	+	2.37E+02	1.28E+01	10	+	4.03E-03	7.49E-03	10	+
MAPSO	2.88E-03	2.37E-03	7	+	5.74E-03	4.49E-03	6	+	1.81E-03	1.55E-03	6	+	1.53E-03	1.18E-03	9	+
XPSO	5.32E+00	3.55E-01	8	+	1.11E+00	1.10E-01	8	+	4.81E+01	7.57E+00	8	+	**6.99E-29**	**5.67E-29**	1	-
WOA	2.23E-03	5.07E-03	6	+	1.33E-02	2.68E-02	7	+	3.56E-04	7.17E-04	5	+	6.28E-04	1.40E-03	8	+
GWO	6.22E-07	1.32E-07	5	+	1.93E-03	5.79E-03	5	+	1.91E-03	6.25E-04	7	+	1.37E-05	1.22E-05	7	+
SCSO	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=	2.26E-58	3.43E-58	3	+	1.62E-06	1.45E-06	6	=
ISCSO1	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=	3.47E-58	9.98E-58	4	+	1.87E-07	1.68E-07	3	=
SESCSO	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=	5.36E-63	9.15E-63	2	+	5.47E-07	5.77E-07	4	=
ISCSO	**8.88E-16**	**0.00E+00**	1		**0.00E+00**	**0.00E+00**	1		**8.79E-86**	**2.61E-85**	1		1.32E-06	1.34E-06	5	
Algorithms	**F13**				**F14**				**F15**							
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test				
RLLPSO	-2.00E+02	0.00E+00	1	-	3.06E+02	2.28E+01	9	+	9.32E+07	1.93E+07	9	+				
TAPSO	-1.94E+02	2.25E+00	5	=	6.83E+02	1.07E+02	10	+	3.22E+08	8.36E+07	10	+				
MAPSO	**-2.00E+02**	**0.00E+00**	1	-	**5.60E-03**	**7.63E-03**	1	-	3.75E-03	2.48E-03	6	+				
XPSO	-2.00E+02	0.00E+00	1	-	1.36E+02	2.80E+01	3	=	6.24E+05	2.20E+05	8	+				
WOA	-2.00E+02	0.00E+00	1	-	5.77E+01	2.50E+01	2	-	4.86E+02	1.39E+03	7	+				
GWO	-7.87E+01	3.22E+00	10	+	1.42E+02	5.25E+00	4	=	1.07E-07	2.81E-08	5	+				
SCSO	-1.39E+02	5.77E+00	7	=	1.98E+02	4.43E-01	6	=	2.11E-103	4.31E-103	4	+				
ISCSO1	-1.30E+02	1.75E+01	9	=	1.98E+02	8.51E-01	7	=	2.12E-105	6.34E-105	3	+				
SESCSO	-1.35E+02	5.91E+00	8	=	1.99E+02	5.58E-01	8	=	1.84E-115	4.57E-115	2	+				
ISCSO	-1.69E+02	7.86E+00	6		1.98E+02	4.63E-01	5		**1.63E-160**	**4.87E-160**	1					

This indicates that ISCSO has good performance compared to other peer algorithms. Moreover, based on the CP values, ISCSO shows the most promising performance, followed by ISCSO1, SESCSO and MAPSO

5) FRIEDMAN-TEST RESULTS

In this part, a Friedman-test of Average values is used to compare the overall performance among all ten competitors. The results are listed in Table 8, in which each algorithm and its rankings are listed in ascending order (the lower the better).

From the results in Table 8, ISCSO shows the best overall performance. This is consistent with the results of the t-test. Therefore, we can further conclude that ISCSO has better performance in global exploration, local search and jumping out of local optimal solutions compared to other peer algorithms.

C. COMPARISON ON 15 TEST FUNCTIONS OF LSOPS

The large-scale optimization problem is famous because of his large dimensionality. As the dimensionality of the problem increases, the performance of the algorithm deteriorates dramatically. Despite this challenge, LSOPs have enormous potential for real-world applications, making it crucial to study the effectiveness of algorithms like ISCSO on these problems

To evaluate the performance of ISCSO and its counterparts, we selected 15 test functions of LSOPs outlined in Table 2. We also conducted tests on three dimensions D=200, D=500, and D=1000 to gain further insight into the algorithms' capabilities

1) RESULTS COMPARISON AND ANALYSIS

The experimental results of ISCSO and other peer algorithms on 15 large-scale (D=200,D=500,D=1000) test functions are given in Tables 9, 10 and 11. As shown in the table, ISCSO outperforms the other algorithms on most of the tested functions. In detail, ISCSO shows excellent performance in 3-D of F1,F2,F3,F4,F6,F8,F9,F10,F11,F15. It is worth mentioning that in F4, ISCSO not only finds the theoretical optimal solution of the function, but also overwhelmingly gets better than other algorithms. With the increase of dimensionality, the performance of ISCSO does not change much, which fully illustrates the strong robustness of ISCSO in solving large-scale problems. From the results of t-tests, ISCSO significantly outperforms other peer algorithms for most functions. Overall, the performance of ISCSO remains stable and significantly better than other peer algorithms as the dimensionality increases. Fig.3 depicts the convergence curves for some functions on D=200. The convergence curves of other dimensions are similar to it and therefore are not listed.

TABLE 10. Optimization results on 15 large-scale standard test functions (*D=500*).

Algorithms	F1				F2				F3				F4			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	2.59E+01	7.15E+00	9	+	9.88E+00	1.79E+00	9	+	2.96E+01	1.53E+01	9	+	2.61E-08	3.07E-08	8	+
TAPSO	5.05E+02	1.70E+02	10	+	4.59E+01	7.94E+00	10	+	1.42E+03	2.93E+02	10	+	1.11E-01	8.54E-02	10	+
MAPSO	9.73E-03	7.02E-03	8	+	3.94E-03	2.84E-03	8	+	4.35E-03	2.94E-03	8	+	3.92E-03	3.39E-03	9	+
XPSO	8.28E-13	2.13E-12	6	+	1.89E-04	3.61E-04	7	+	6.74E-12	9.27E-12	6	+	2.12E-23	3.77E-23	6	+
WOA	2.17E-06	6.51E-06	7	+	2.56E-05	7.56E-05	6	+	1.28E-07	2.35E-07	7	+	4.17E-20	1.12E-19	7	+
GWO	1.70E-38	2.14E-38	5	+	7.80E-23	4.96E-23	5	+	1.66E-39	3.91E-39	5	+	9.19E-137	2.14E-136	5	+
SCSO	2.72E-118	8.15E-118	4	+	1.84E-64	2.75E-64	3	+	1.19E-119	3.40E-119	4	+	5.58E-227	0.00E+00	4	+
ISCSO1	2.46E-123	7.37E-123	3	+	3.60E-64	7.18E-64	4	+	4.65E-123	1.39E-122	3	+	1.24E-248	0.00E+00	2	+
SESCSO	3.44E-134	9.77E-134	2	+	2.88E-70	8.26E-70	2	+	5.65E-135	1.70E-134	2	+	3.28E-241	0.00E+00	3	+
ISCSO	**1.20E-179**	**0.00E+00**	1		**9.59E-94**	**1.12E-93**	1		**1.76E-180**	**0.00E+00**	1		**2.81E-263**	**0.00E+00**	1	
Algorithms	**F5**				**F6**				**F7**				**F8**			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	8.01E+02	6.37E+02	9	+	2.06E-02	1.08E-02	8	+	7.10E-01	3.54E-01	9	+	1.66E+02	2.49E+01	9	+
TAPSO	1.97E+05	5.13E+04	10	+	5.46E+01	1.96E+01	10	+	1.01E+01	4.77E+00	10	+	2.76E+02	3.50E+01	10	+
MAPSO	**3.55E-03**	**2.29E-03**	1	-	3.31E-03	2.69E-03	7	+	5.45E-03	6.43E-03	3	=	3.52E-03	3.00E-03	6	+
XPSO	6.40E+01	3.33E+01	8	+	2.25E-02	9.79E-03	9	+	**2.02E-10**	**3.94E-10**	1	-	3.74E+01	1.06E+01	8	+
WOA	2.84E+01	4.60E-01	7	=	8.19E-04	9.38E-04	6	=	1.77E-01	1.40E-01	8	=	2.72E+01	4.95E+01	7	+
GWO	2.70E+01	5.46E-01	3	=	2.60E-04	2.99E-04	5	=	4.29E-02	8.46E-03	4	=	0.00E+00	0.00E+00	1	+
SCSO	2.76E+01	1.03E+00	4	=	5.54E-05	3.99E-05	2	=	5.25E-02	2.89E-02	5	=	0.00E+00	0.00E+00	1	=
ISCSO1	2.78E+01	1.18E+00	6	=	1.34E-04	1.54E-04	3	=	8.67E-02	3.89E-02	7	=	0.00E+00	0.00E+00	1	=
SESCSO	2.78E+01	8.80E-01	5	=	1.48E-04	1.32E-04	4	=	7.93E-02	3.07E-02	6	=	0.00E+00	0.00E+00	1	=
ISCSO	2.58E+01	5.15E-01	2		**3.78E-05**	**3.42E-05**	1		2.85E-03	4.56E-03	2		**0.00E+00**	**0.00E+00**	1	
Algorithms	**F9**				**F10**				**F11**				**F12**			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	3.79E+00	5.33E-01	9	+	1.46E+00	2.58E-01	9	+	2.73E+00	8.22E-01	9	+	5.48E-08	8.21E-08	2	-
TAPSO	1.01E+01	9.00E-01	10	+	1.15E+01	2.43E+00	10	+	2.72E+01	3.22E+00	10	+	1.72E-03	2.28E-03	9	+
MAPSO	4.68E-03	3.81E-03	8	+	3.66E-03	4.18E-03	6	+	6.13E-03	6.08E-03	6	+	3.05E-03	2.03E-03	10	+
XPSO	8.37E-07	1.21E-06	6	+	9.60E-03	9.68E-03	7	+	1.39E-02	1.81E-02	7	+	**7.00E-29**	**5.66E-29**	1	-
WOA	2.08E-05	5.88E-05	7	+	1.04E-02	3.13E-02	8	+	1.06E+00	3.09E+00	8	+	2.00E-04	2.82E-04	8	+
GWO	6.93E-15	1.63E-15	5	+	2.02E-03	6.07E-03	5	+	5.12E-06	7.85E-06	5	+	1.10E-05	1.09E-05	7	+
SCSO	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=	9.92E-65	2.33E-64	4	+	3.99E-07	3.86E-07	5	=
ISCSO1	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=	9.91E-66	2.87E-65	3	+	1.04E-07	1.35E-07	3	=
SESCSO	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=	7.49E-74	8.76E-74	2	+	5.04E-07	7.09E-07	6	=
ISCSO	**8.88E-16**	**0.00E+00**	1		**0.00E+00**	**0.00E+00**	1		**4.74E-93**	**8.79E-93**	1		2.08E-07	1.06E-07	4	
Algorithms	**F13**				**F14**				**F15**							
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test				
RLLPSO	-3.00E+01	0.00E+00	1	=	6.94E+00	2.82E+00	5	+	3.17E+05	1.60E+05	9	+				
TAPSO	-2.98E+01	3.63E-01	5	=	6.76E+01	1.14E+01	10	=	7.89E+07	1.31E+08	10	+				
MAPSO	**-3.00E+01**	**0.00E+00**	1	=	7.95E-03	5.97E-03	2	-	3.24E-03	3.04E-03	7	+				
XPSO	-3.00E+01	0.00E+00	1	=	**9.54E-07**	**2.83E-06**	1	-	1.89E-05	3.79E-05	6	+				
WOA	-3.00E+01	0.00E+00	1	=	5.83E+00	4.10E+00	4	-	2.88E-01	8.61E-01	8	+				
GWO	-2.67E+01	1.48E+00	9	=	1.94E+00	7.09E-01	3	-	4.21E-36	3.22E-36	5	+				
SCSO	-2.71E+01	1.59E+00	8	=	2.00E+01	6.30E+00	7	=	5.25E-116	1.55E-115	4	+				
ISCSO1	-2.81E+01	8.64E-01	7	=	2.47E+01	5.58E+00	9	=	4.97E-116	1.49E-115	3	+				
SESCSO	-2.61E+01	1.09E+00	10	=	2.39E+01	6.03E+00	8	=	8.69E-131	2.59E-130	2	+				
ISCSO	-2.91E+01	1.87E+00	6		1.70E+01	6.63E+00	6		**3.40E-179**	**0.00E+00**	1					

2) T-TEST RESULTS

The t-test results for ISCSO and other peer algorithms in the 3-D cases of 15 test functions of LSOPs have been analyzed and presented in Table 12. In the table, "#+", "#=", and "#-" indicate that the test results of ISCSO are significantly better than, almost the same as, and significantly worse than other peer functions, respectively. The comprehensive performance (CP) is equal to "#+" minus "#−".

It shows that ISCSO outperforms other peer algorithms significantly in 3-D of the 15 test functions of LSOPs.

Moreover, the performance of ISCSO remains stable even with an increase in dimensionality, which further validates its reliability. Based on the CP value, ISCSO exhibits the best performance, followed by SCSO and SESCSO, which aligns with the results obtained from low dimensional function tests.

3) FRIEDMAN-TEST RESULTS

We conducted a Friedman test on all algorithms in 3-D and the results are presented in Table 13. The algorithms are listed in ascending order based on their ranking values.The results reveal that ISCSO achieved the most favorable outcomes in 3-D of the experiments. This is further corroborated by its overall ranking. SESCSO and ISCSO1 obtained the second and third highest performance rankings, respectively.

V. DOUBLE-LAYER SPRAYING PATH PARAMETER OPTIMIZATION

A. DOUBLE-LAYER SPRAYING PAINT FILM THICKNESS MODEL

This paper uses ISCSO to resolve the paint thickness uniformity problem in air spraying, further demonstrating the viability of the method.

The spray gun's paint distribution is better described in this study by an elliptical double β spraying distribution model. Figure 4 displays the model's schematic diagram.

The following equation (17) represents the total rate of spraying per unit of time at a point P in the plane.

$$D(x,y) = D_{max}\left(1-\frac{x^2}{a^2}\right)^{\beta_1-1}\left(1-\frac{y^2}{b^2\left(1-\frac{x^2}{a^2}\right)}\right)^{\beta_2-1} \tag{17}$$

where, $-a \le x \le a$, $-b\left(1-\frac{x^2}{a^2}\right)^{\frac{1}{2}} \le y \le b\left(1-\frac{x^2}{a^2}\right)^{\frac{1}{2}}$, D_{max} are the maximum layer thickness values in the spraying area per unit of time. β_1, β_2 are function distribution parameters. $D(x, y)$ is the accumulated thickness of the layer per unit of time at point P.

During the spraying process, the distance between the spray gun and the sprayed surface needs to be varied. The three-dimensional layer thickness model for different spraying distances h can be obtained according

TABLE 11. Optimization results on 15 large-scale standard test functions (*D=1000*).

Algorithms	F1				F2				F3				F4			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	2.62E+02	7.59E+01	9	+	2.30E+01	3.10E+00	9	+	3.54E+02	1.07E+02	9	+	7.01E-08	6.66E-08	8	+
TAPSO	1.25E+03	3.30E+02	10	+	1.00E+02	1.92E+01	10	+	4.67E+03	7.52E+02	10	+	2.37E-01	2.72E-01	10	+
MAPSO	5.18E-03	4.60E-03	8	+	3.10E-03	2.69E-03	7	+	3.08E-03	2.74E-03	7	+	3.24E-03	2.93E-03	9	+
XPSO	2.64E-05	3.77E-05	6	+	8.99E-02	6.95E-02	8	+	6.67E-02	1.94E-01	8	+	1.10E-18	1.08E-18	6	+
WOA	1.01E-04	2.94E-04	7	+	2.05E-06	4.43E-06	6	+	3.73E-06	8.05E-06	6	+	4.63E-18	1.04E-17	7	+
GWO	1.09E-27	1.35E-27	5	+	1.12E-16	4.73E-17	5	+	1.68E-28	1.05E-28	5	+	1.67E-125	4.89E-125	5	+
SCSO	3.37E-114	1.01E-113	4	+	6.23E-61	1.41E-60	3	+	3.45E-116	7.17E-116	3	+	7.08E-220	0.00E+00	4	+
ISCSO1	1.53E-114	4.60E-114	3	+	3.08E-58	9.23E-58	4	+	1.06E-112	3.16E-112	4	+	8.99E-234	0.00E+00	3	+
SESCSO	1.81E-126	5.32E-126	2	+	2.46E-68	5.50E-68	2	+	1.40E-124	4.19E-124	2	+	1.97E-241	0.00E+00	2	+
ISCSO	**8.82E-176**	**0.00E+00**	1		**1.97E-89**	**5.34E-89**	1		**4.69E-174**	**0.00E+00**	1		**1.17E-281**	**0.00E+00**	1	
Algorithms	**F5**				**F6**				**F7**				**F8**			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	1.34E+04	4.51E+03	9	+	4.58E-02	1.89E-02	8	+	1.79E+00	3.16E-01	9	+	3.03E+02	6.44E+01	9	+
TAPSO	4.78E+05	1.48E+05	10	+	1.39E+02	5.18E+01	10	+	1.71E+01	2.92E+00	10	+	4.79E+02	3.00E+01	10	+
MAPSO	**2.88E-03**	**3.20E-03**	1	-	3.93E-03	3.86E-03	7	+	**6.40E-03**	**5.41E-03**	1	-	3.52E-03	3.52E-03	6	+
XPSO	1.30E+02	4.30E+01	8	+	1.21E-01	5.07E-02	9	+	1.25E-02	3.73E-02	2	=	9.62E+01	2.41E+01	8	+
WOA	4.87E+01	1.55E-01	7	=	2.18E-03	2.80E-03	6	+	4.42E-01	6.91E-01	8	+	1.39E+00	4.16E+00	7	+
GWO	4.70E+01	7.60E-01	3	=	1.71E-04	2.06E-04	3	+	7.74E-02	3.01E-02	4	=	0.00E+00	0.00E+00	1	=
SCSO	4.73E+01	7.76E-01	4	=	2.08E-04	2.98E-04	4	+	9.50E-02	2.84E-02	5	=	0.00E+00	0.00E+00	1	=
ISCSO1	4.77E+01	8.85E-01	6	=	2.16E-04	2.71E-04	5	+	1.46E-01	2.94E-02	6	+	0.00E+00	0.00E+00	1	=
SESCSO	4.76E+01	1.00E+00	5	=	**8.08E-05**	**6.81E-05**	1	=	1.50E-01	6.02E-02	7	+	0.00E+00	0.00E+00	1	=
ISCSO	4.64E+01	6.82E-01	2		9.46E-05	7.21E-05	2		2.36E-02	7.95E-03	3		**0.00E+00**	**0.00E+00**	1	
Algorithms	**F9**				**F10**				**F11**				**F12**			
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test
RLLPSO	5.65E+00	5.97E-01	9	+	2.95E+00	7.59E-01	9	+	1.21E+01	1.18E+00	9	+	4.87E-08	5.04E-08	2	-
TAPSO	1.08E+01	6.61E-01	10	+	2.24E+01	3.05E+00	10	+	5.10E+01	5.33E+00	10	+	1.54E-03	1.56E-03	9	+
MAPSO	3.94E-03	2.45E-03	7	+	6.36E-03	3.11E-03	6	+	4.55E-03	4.12E-03	7	+	2.77E-03	2.65E-03	10	+
XPSO	1.41E+00	7.72E-01	8	+	8.17E-03	8.18E-03	7	+	3.38E-01	2.58E-01	8	+	**8.19E-29**	**5.24E-29**	1	-
WOA	2.31E-04	5.45E-04	6	+	9.94E-02	1.99E-01	8	+	1.77E-03	5.06E-03	6	+	1.09E-04	2.18E-04	8	+
GWO	5.10E-14	9.19E-15	5	+	7.84E-04	2.35E-03	5	+	5.38E-05	5.80E-05	5	+	1.37E-05	2.05E-05	7	+
SCSO	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=	1.78E-62	2.33E-62	3	+	2.26E-06	3.29E-06	6	=
ISCSO1	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=	3.67E-61	7.44E-61	4	+	3.41E-07	4.94E-07	3	=
SESCSO	8.88E-16	0.00E+00	1	=	0.00E+00	0.00E+00	1	=	2.34E-69	3.90E-69	2	+	1.56E-06	2.91E-06	5	=
ISCSO	**8.88E-16**	**0.00E+00**	1		**0.00E+00**	**0.00E+00**	1		**1.42E-90**	**3.71E-90**	1		8.99E-07	1.62E-06	4	
Algorithms	**F13**				**F14**				**F15**							
	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test	Ave	Std	Rank	t-test				
RLLPSO	-5.00E+01	0.00E+00	1	-	2.71E+01	7.92E+00	5	=	2.53E+06	1.32E+06	9	+				
TAPSO	-4.92E+01	8.57E-01	5	=	1.21E+02	2.01E+01	10	+	3.57E+07	1.25E+07	10	+				
MAPSO	**-5.00E+01**	**0.00E+00**	1	-	**2.83E-03**	**2.02E-03**	1	-	4.79E-03	3.82E-03	6	+				
XPSO	-5.00E+01	0.00E+00	1	-	2.00E-02	3.98E-02	2	-	1.36E+02	1.80E+02	8	+				
WOA	-5.00E+01	0.00E+00	1	-	1.28E+01	6.38E+00	4	=	2.46E-02	5.90E-02	7	+				
GWO	-3.38E+01	2.15E+00	10	+	1.13E+01	1.81E+00	3	=	2.29E-24	2.32E-24	5	+				
SCSO	-4.12E+01	1.60E+00	8	=	4.65E+01	8.56E-01	9	=	3.73E-113	7.45E-113	3	+				
ISCSO1	-4.46E+01	2.22E+00	7	=	4.59E+01	5.35E+00	7	=	2.86E-112	8.57E-112	4	+				
SESCSO	-3.99E+01	2.05E+00	9	=	4.64E+01	1.52E+00	8	=	8.23E-123	2.46E-122	2	+				
ISCSO	-4.59E+01	6.38E+00	6		4.58E+01	6.16E-01	6		**3.97E-167**	**0.00E+00**	1					

TABLE 12. T-test results between ISCSO and other peer algorithms on the large-scale standard test functions.

Algorithm	D=200			D=500			D=1000			CP
	#+	#=	#-	#+	#=	#-	#+	#=	#-	
RLLPSO	13	0	2	13	1	1	12	2	1	34
TAPSO	14	1	0	13	2	0	14	1	0	41
MAPSO	11	0	4	11	2	2	11	0	4	23
XPSO	12	1	2	11	1	3	11	1	3	26
WOA	11	2	2	10	4	1	12	2	1	29
GWO	12	3	0	10	4	1	11	4	0	32
ISCSO1	7	8	0	6	9	0	8	7	0	21
SESCSO	7	8	0	6	9	0	7	8	0	20
SCSO	7	8	0	6	9	0	7	8	0	20

TABLE 13. Friedman-test of mean values on 15 large-scale standard test functions.

Overall Rank	Algorithm	Rank	D=200		D=500		D=1000	
			Algorithm	Rank	Algorithm	Rank	Algorithm	Rank
1	ISCSO	2.07	ISCSO	2.07	ISCSO	2.00	ISCSO	2.13
2	SESCSO	3.40	SESCSO	3.20	SESCSO	3.67	SESCSO	3.33
3	ISCSO1	3.80	ISCSO1	3.73	ISCSO1	3.73	ISCSO1	3.93
4	SCSO	3.80	SCSO	3.67	SCSO	3.80	SCSO	3.93
5	GWO	4.88	GWO	5.47	GWO	4.80	GWO	4.37
6	MAPSO	5.62	MAPSO	5.27	MAPSO	6.00	MAPSO	5.60
7	XPSO	6.02	XPSO	6.73	XPSO	5.33	XPSO	6.00
8	WOA	6.22	WOA	5.80	WOA	6.60	WOA	6.27
9	RLLPSO	7.67	RLLPSO	7.80	RLLPSO	7.60	RLLPSO	7.60
10	TAPSO	9.62	TAPSO	9.67	TAPSO	9.60	TAPSO	9.60

to the projection method. The model equation is as follows.

$$D(x,y,h) = D_{max}\left(\frac{h_0}{h}\right)^2\left(1-\frac{x^2}{\left(a_0\cdot\frac{h}{h_0}\right)^2}\right)^{\beta_1-1} \times\left(1-\frac{y^2}{(b_0\cdot\frac{h}{h_0})^2(1-\frac{x^2}{(a_0\cdot\frac{h}{h_0})^2})}\right)^{\beta_2-1} \tag{18}$$

where h_0 is the height of the gun reference. a_0 and b_0 are the long and short axes of the spraying surface ellipse at this time.

The center point of the spray gun is moved along the y-axis. Assuming the travel speed of v, the total time elapsed for any point on the spray plane is shown in Equation (19).

$$t_a = 2(b_0\cdot\frac{h}{h_0})(1-\frac{x^2}{(a_0\cdot\frac{h}{h_0})^2})^{\frac{1}{2}}/v \tag{19}$$

Eq. (18) is then time-integrated to produce a model for the buildup of paint thickness on the plane.

The cumulative model of paint thickness at each location on the plane when the gun is at any height is represented by equation (20).

From equation (20), it can be seen that when the spraying parameters $h_0, a_0, b_0, D_{max}, \beta_1, \beta_2$ are determined, the cumulative model of the plane paint thickness of the gun is also determined.

$$T(x,h,v) = \int_0^t {}_aD_{max}\left(\frac{h}{h_0}\right)^2\left(1-\frac{x^2}{\left(a_0\cdot\frac{h}{h_0}\right)^2}\right)^{\beta_1-1} \times\left(1-\frac{[(b_0\cdot\frac{h}{h_0})(1-\frac{x^2}{(a_0\cdot\frac{h}{h_0})^2})^{\frac{1}{2}}-vt]^2}{(b_0\cdot\frac{h}{h_0})(1-\frac{x^2}{(a_0\cdot\frac{h}{h_0})^2})}\right)dt \tag{20}$$

The Z-path is now the spraying path chosen by spraying robots the most frequently. In this paper, the research is carried out based on the Z-path.

Figure 5 (a) shows the schematic diagram for the planar double-layer spraying trajectory. In the first layer of spraying, two adjacent trajectory spray gun center distance d, spray gun height h_1, and spraying speed v_1. The second layer of the spraying gun axis is situated midway between the previous layer's two neighboring spraying trajectory. In the second layer of spraying, the height of the gun is h_2 and the speed of spraying is v_1.

As shown in Figure 5 (b), the coordinates of the points in the figure are $O(0,0), B(d-a_1,0), C(a_1,0), D(\frac{d}{2}+a_2,0), O_1(d,0)$. a_1 is the length of the long axis of the first spraying area. a_2 is the length of the long axis of the second spraying area.

$T_{1a}(x,h_1,v_1)$ stands for the first trace of the first coating layer's coating thickness, and the coating thickness of the second trace is obtained by translating $T_{1a}(x,h_1,v_1)$ in the x-direction d, denoted by $T_{1b}(x,h_1,v_1,d)$. The coating thickness of the second layer is then denoted as $T_2(x,h_2,v_2)$. Equation (21) is the distribution function of coating thickness for the double-layer spraying.

$$T(x,d,h_1,v_1,h_2,v_2)\begin{cases} T_{1a}(x,h_1,v_1) \\ \quad 0\leqslant x<\frac{d}{2}-a_2 \\ T_{1a}(x,h_1,v_1)+T_2(x,h_2,v_2) \\ \quad \frac{d}{2}-a_2\leqslant x<d-a_1 \\ T_{1a}(x,h_1,v_1)+T_2(x,h_2,v_2) \\ \quad +T_{1b}(x,h_1,v_1) \\ \quad d-a_1\leqslant x<a_1 \\ T_2(x,h_2,v_2)+T_{1b}(x,h_1,v_1) \\ \quad a_1\leqslant x<\frac{d}{2}+a_2 \\ T_{1b}(x,h_1,v_1) \\ \quad \frac{d}{2}+a_2\leqslant x\leq d \end{cases} \tag{21}$$

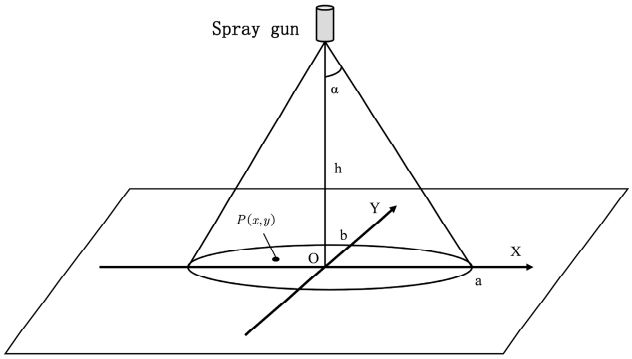
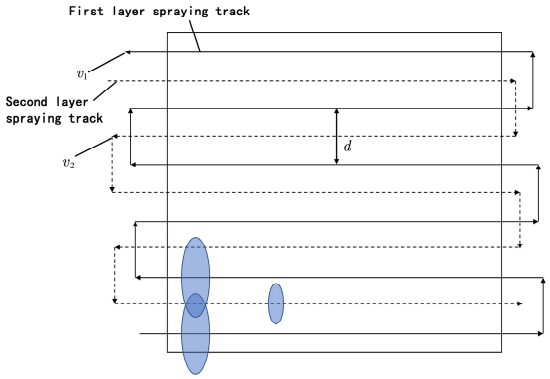
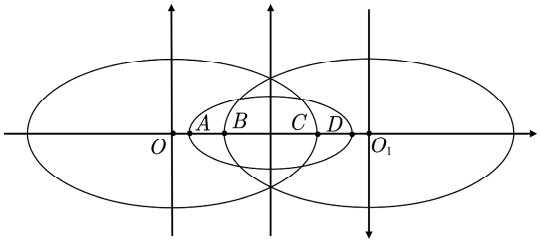


FIGURE 5. The schematic diagram of the model.



(a) Schematic diagram of the plane double-layer spraying process.



(b) Schematic diagram of the paint overlap area.

FIGURE 6. Planar double-layer spraying trajectory diagram.

B. PRAYING EXPERIMENT

Spraying experiments were conducted in this study with the relative position of the spray gun and workpiece kept unchanged. The purpose was to obtain the essential technical parameters of the coating thickness accumulation model as specified in subsection V-A.

A steel plate with a thickness of 1 mm was placed perpendicular to the spray gun axis, with the muzzle distance from the spray surface set at 500 mm. The spraying time lasted for 3 seconds and was timed using a stopwatch. Once the paint layer dried, a marking point was used as the origin, and the long axis direction of the ellipse area was represented by the X-axis, while the short axis direction was represented by the Y-axis. Then, as depicted in Figure 6, parallel lines

were marked along the X and Y axes at 20 mm intervals, and the junction of the two lines was taken as the measurement point for the coating thickness.

To measure the coating thickness, a TC580 coating thickness gauge (measurement range 0∼1700, resolution 0.1) was used. For each measurement point, the thickness was measured 5 times, and the average value was recorded. Finally, to determine the coating thickness per unit of spraying time, the measured coating thickness was divided by 3. Table 14 shows some of the experimental data, with only the measurement data on the X and Y axes included due to space limitations.

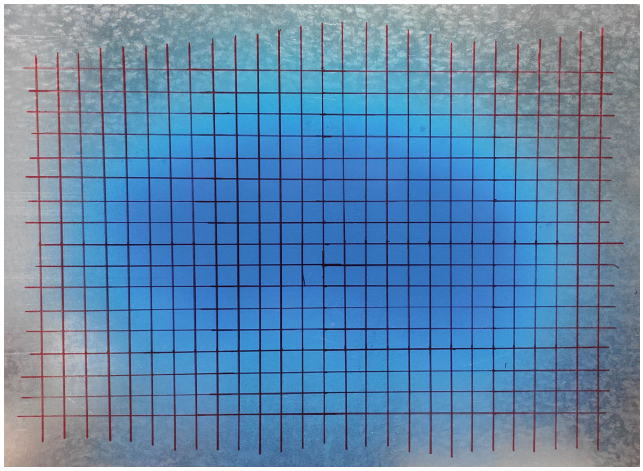


FIGURE 7. Planar spraying experimental diagram.

The least squares method is used to fit the experimentally collected measurement data to the paint film surface. Since the surface functions are known, the solution was directly performed using the MATLAB genetic algorithm toolbox. The final key parameters of the model obtained are: $\mathbf{a_0}$= **204.665**, $\mathbf{b_0}$= **117.776**,$\mathbf{D_{max}}$= **11.448**, $\boldsymbol{\beta_1}$= **2.763**, $\boldsymbol{\beta_2}$= **3.174**. Figure 7 shows the surface fitting model.

C. SIMULATION EXPERIMENT FOR OPTIMIZATION OF PARAMETERS OF DOUBLE-LAYER SPRAYING TRAJECTORY

1) ESTABLISHMENT OF THE OBJECTIVE FUNCTION

The objective of this paper is to increase the sprayed layer's uniformity. First, the variance between the film thickness and the desired film thickness is calculated for each point in the spraying area. A point is selected at $\mathbf{0.5}\boldsymbol{mm}$ intervals between 0 and $\boldsymbol{d}$. The variance calculation function is given by equation (22).

$$L = \frac{1}{n}\sum_{i=1}^{n}(T(x_i) - T_d)^2 \tag{22}$$

where $\boldsymbol{T(x_i)}$ is the thickness of the paint film at each point and $\boldsymbol{T_d}$ is the desired paint film thickness. $\boldsymbol{n}$ is the total number of points.

At the same time, to ensure that the coating will not be extremely thick or thin, the polar difference of the painting is constrained. Equation (23) is the polar difference calculation function.

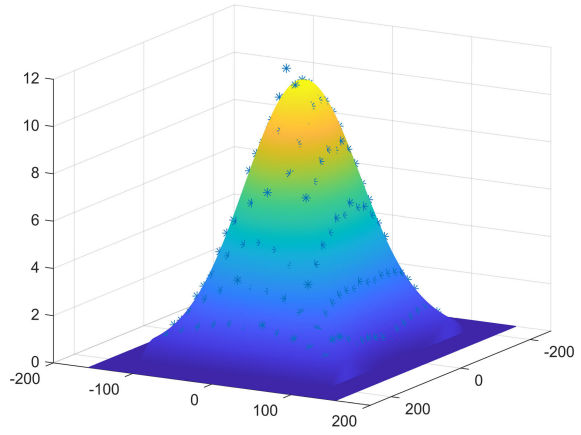


FIGURE 8. Experimental fitting plot.

$$E = T_{max} - T_{min} \tag{23}$$

where $\boldsymbol{T_{max}}$ is the maximum value of the layer thickness and $\boldsymbol{T_{min}}$ is the minimum value of the layer thickness. Equation (24) is the final objective function.

$$Minimize : 0.5 * L + E \tag{24}$$

Combining the painting experience and equipment performance, the desired film thickness is $\boldsymbol{T_d}$ $\mathcal{D}$ **80** $\boldsymbol{\mu m}$. Also constrain each target solution: $\boldsymbol{d}$ ∈ [**300**, **1000**], $\boldsymbol{h_1}$ ∈ [**300**, **1000**], $\boldsymbol{v_1}$ ∈ [**1**, **100**], $\boldsymbol{h_2}$ ∈ [**300**, **1000**], $\boldsymbol{v_2}$ ∈ [**1**, **100**].

2) OPTIMIZATION ALGORITHM SOLVING AND SIMULATION

The spraying trajectory parameters were optimized using the ten aforementioned algorithms. The specific algorithm parameters are selected as follows: The fitness function is Equation (24); the population size is 60, the maximum number of iterations is 100, and other optimization parameters are set as shown in subsection III-A

Finally, the convergence curves of the ten algorithms after algorithm execution are shown in Figure 8. The layer thickness distribution is shown in Fig. 9. The optimal results are shown in Table 15. Among them, the optimal solution obtained by the ISCSO algorithm is [620.066, 999.797, 17.200, 585.154, 79.550]. IT shows that, in the first layer of spraying, the center distance of two adjacent trajectories $d = 620.066mm$, gun height $h_1 = 999.797mm$, and spraying speed $v_1 = 17.200mm/s$. In the second layer of spraying, the gun height $h_2 = 585.154mm$, and spraying speed $v_2 = 79.55mm/s$

According to Fig. 8, ISCSO shows stronger convergence and better merit-seeking performance. From the specific convergence data, the range and variance of ISCSO are substantially improved compared to the original algorithm. This further demonstrates that ISCSO has a better result in finding the best performance. Based on the graphical results, this paper concludes that ISCSO is more efficient and superior to

TABLE 14. Some data of the spraying experiment.

Coordinates	Thickness (μm)	Coordinates	Thickness (μm)	Coordinates	Thickness (μm)	Coordinates	Thickness (μm)
(0,-100)	0.00	(0,40)	25.5	(-100,0)	20.4	(60,0)	28.5
(0,-80)	10.1	(0,60)	19.1	(-80,0)	24.6	(80,0)	22.3
(0,-60)	17.8	(0,80)	9.33	(-60,0)	25.6	(100,0)	21.4
(0,-40)	27.4	(0,100)	2.54	(-40,0)	28.4	(120,0)	15.4
(0,-20)	32.1	(-160,0)	5.61	(-20,0)	33.1	(140,0)	9.91
(0,0)	33.9	(-140,0)	11.1	(20,0)	33.6	(160,0)	5.21
(0,20)	31.6	(-120,0)	14.3	(40,0)	30.2		

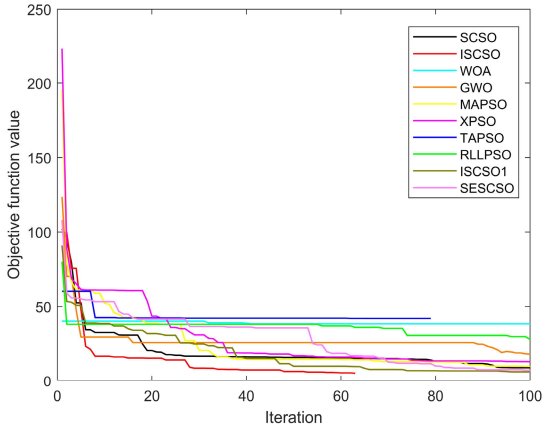


FIGURE 9. Convergence curve of optimization algorithm for planar double-layer spraying.

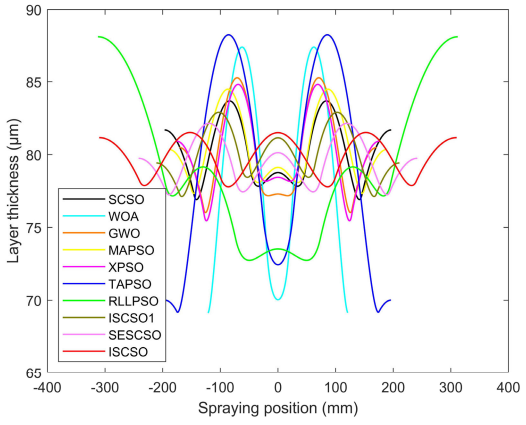


FIGURE 10. Thickness distribution of planar double-layer spraying.

the original algorithm in iteratively solving the double-layer spraying trajectory optimization problem.

3) COMPARISON OF OPTIMIZATION RESULTS OF PLANAR SINGLE-LAYER SPRAYING AND PLANAR DOUBLE-LAYER SPRAYING

To evaluate the efficacy of double-layer spraying, a comparative simulation experiment was conducted in this study. ISCSO was used to optimize the trajectory of

TABLE 15. Algorithm optimization of optimal results.

Algorithm	ISCSO	ISCSO1	SESCSO	SCSO	WOA
range	3.73	5.78	4.89	6.77	18.24
variance	1.53	2.38	2.45	4.33	40.34
Algorithm	**GWO**	**MAPSO**	**XPSO**	**TAPSO**	**RLLPSO**
range	10.64	7.15	9.36	19.05	15.90
variance	8.79	5.39	7.34	45.94	25.11

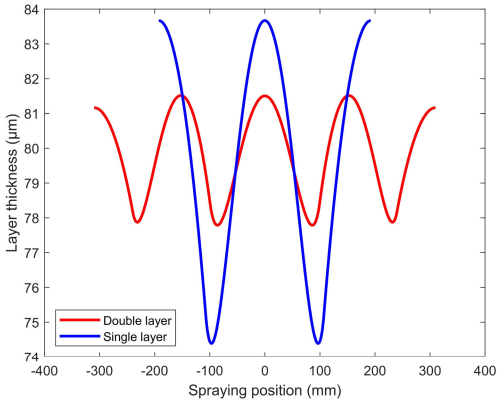


FIGURE 11. Single and double spray painting thickness comparison chart.

single-layer spraying, and the optimal solution was obtained, [385.270, 732.485, 22.778], with the overlap width of adjacent trajectories being d = 385.270mm, the spray gun height being h = 732.485mm, AND THE SPRAYING SPEED BEING v_1 = 22.778mm/S. The thickness distribution of the single-layer sprayed layer and the double-layer sprayed layer was compared and presented in Figure 10. The figure shows that the layer THICKNESS uniformity has significantly improved after optimization.

VI. DISCUSSION

In this paper, we introduce an Improved Sand Cat Swarm Optimization (ISCSO) algorithm to address the limitations of the original Sand Cat Swarm Optimization (SCSO) algorithm. While SCSO is a simple and easy-to-implement algorithm, it suffers from issues such as poor initial

population quality, slow convergence speed, and a tendency to fall into local optima. To overcome these problems, we propose three improvement strategies.

Firstly, we introduce the SPM chaotic mapping technique to increase the diversity of the ISCSO's initial population. This helps to improve the quality of the initial population and overcome the problem of poor initial population quality in SCSO. Secondly, we add a nonlinear periodic adjustment mechanism to balance the algorithm's local exploitation and global search capabilities. Finally, We calculated and ranked the incentive degree among individuals. The ability of ISCSO to escape local optima was greatly improved by treating the top 20% of individuals with cloning, mutation and cloning suppression.

To verify the performance of ISCSO, we conducted extensive experiments on 21 low-dimensional functions and 15 test functions of LSOPs, comparing it with nine other peer algorithms. The results show that ISCSO has excellent performance on unimodal functions, multimodal functions, and fixed-dimension multimodal functions. Moreover, it demonstrates strong adaptability for solving LSOPs problems.

To further validate the practicality of the ISCSO algorithm, we applied it to the field of spraying and optimized spraying parameters using ISCSO and each of the other seven peer algorithms. The experimental results demonstrate that ISCSO is highly practical and shows good promise for engineering optimization problems.

In the future, we will continue to investigate the real-world applications of ISCSO and further improve the performance of the algorithm.

REFERENCES

[1] A. H. Gandomi, K. Deb, R. C. Averill, S. Rahnamayan, and M. N. Omidvar, "Variable functioning and its application to large scale steel frame design optimization," *Struct. Multidisciplinary Optim.*, vol. 66, no. 1, pp. 1–10, Jan. 2023.

[2] J. H. Holland, "Genetic algorithms," *Sci. Amer.*, vol. 267, no. 1, pp. 66–73, 1992.

[3] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[4] G. Rudolph, "Evolutionary strategies," in *Handbook of Natural Computing*, G. Rozenberg, T. Bäck, and J. N. Kok, Eds. Berlin, Germany: Springer, 2012, pp. 673–698.

[5] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

[6] C. Ferreira, "Gene expression programming: A new adaptive algorithm for solving problems," 2001, *arXiv:cs/0102027*.

[7] R. Poli, "Genetic programming," in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, E. K. Burke and G. Kendall, Eds. Boston, MA, USA: Springer, 2005, pp. 64–127.

[8] N. Hansen and S. Kern, "Evaluating the CMA evolution strategy on multimodal test functions," in *Proc. Int. Conf. Parallel Problem Solving Nature*. Berlin, Germany: Springer, 2004, pp. 282–291.

[9] D. Simon, "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, vol. 12, no. 6, pp. 702–713, Dec. 2008.

[10] G. Venter and J. Sobieszczanski-Sobieski, "Particle swarm optimization," *AIAA J.*, vol. 41, no. 8, pp. 1583–1589, 2002.

[11] D. Marco and S. Thomas, "Ant colony optimization theory," in *Ant Colony Optimization*. Cambridge, MA, USA: MIT Press, 2004, pp. 52–121.

[12] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *World Congr. Nature Biol. Inspired Comput. (NaBIC)*. IEEE, 2009, pp. 210–214.

[13] X.-S. Yang, "Firefly algorithm, stochastic test functions and design optimisation," *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.

[14] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[15] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, Jan. 2016.

[16] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Syst. Mag.*, vol. 22, no. 3, pp. 52–67, Mar. 2002.

[17] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

[18] R. V. Rao, V. J. Savsani, and D. P. Vakharia, "Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems," *Comput.-Aided Des.*, vol. 43, no. 3, pp. 303–315, Mar. 2011.

[19] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: Harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, Feb. 2001.

[20] N. Moosavian and B. K. Roodsari, "Soccer league competition algorithm, a new method for solving systems of nonlinear equations," *Int. J. Intell. Sci.*, vol. 4, no. 1, pp. 7–16, 2014.

[21] H. Emami and F. Derakhshan, "Election algorithm: A new socio-politically inspired strategy," *AI Commun.*, vol. 28, no. 3, pp. 591–603, Jul. 2015.

[22] T. T. Huan, A. J. Kulkarni, J. Kanesan, C. J. Huang, and A. Abraham, "Ideology algorithm: A socio-inspired optimization methodology," *Neural Comput. Appl.*, vol. 28, no. S1, pp. 845–876, Dec. 2017.

[23] S. He, Q. H. Wu, and J. R. Saunders, "A novel group search optimizer inspired by animal behavioural ecology," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 1–6.

[24] E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2007, pp. 4661–4667.

[25] F. Ramezani and S. Lotfi, "Social-based algorithm (SBA)," *Appl. Soft Comput.*, vol. 13, no. 5, pp. 2837–2856, May 2013.

[26] S. Kirkpatrick and D. M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, Mar. 1983.

[27] B. Webster and P. J. Bernhard, "A local search optimization algorithm based on natural principles of gravitation," in *Proc. Int. Conf. Inf. Knowl. Eng.*, 2003.

[28] E. Ok and I. Eksin, "A new optimization method: Big bang-big crunch," *J. Adv. Eng. Softw.*, vol. 37, no. 2, pp. 106–111, 2006.

[29] A. Kaveh and S. Talatahari, "A novel heuristic optimization method: Charged system search," *Acta Mechanica*, vol. 213, nos. 3–4, pp. 267–289, Sep. 2010.

[30] B. Alatas, "ACROA: Artificial chemical reaction optimization algorithm for global optimization," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13170–13180, Sep. 2011.

[31] H. S. Hosseini, "Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation," *Int. J. Comput. Sci. Eng.*, vol. 6, no. 1/2, p. 132, 2011.

[32] A. Kaveh and M. Khayatazad, "A new meta-heuristic method: Ray optimization," *Comput. Struct.*, vols. 112–113, pp. 283–294, Dec. 2012.

[33] H. Eskandar, A. Sadollah, A. Bahreininejad, and M. Hamdi, "Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems," *Comput. Struct.*, vols. 110–111, pp. 151–166, Nov. 2012.

[34] S. Mahdavi, M. E. Shiri, and S. Rahnamayan, "Metaheuristics in large-scale global continues optimization: A survey," *Inf. Sci.*, vol. 295, pp. 407–428, Feb. 2015.

[35] A. Seyyedabbasi and F. Kiani, "Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems," *Eng. Comput.*, vol. 2022, pp. 1–16, Apr. 2022.

[36] Y. Li and G. Wang, "Sand cat swarm optimization based on stochastic variation with elite collaboration," *IEEE Access*, vol. 10, pp. 89989–90003, 2022.

[37] A. Iraji, J. Karimi, S. Keawsawasvong, and M. L. Nehdi, "Minimum safety factor evaluation of slopes using hybrid chaotic sand cat and pattern search approach," *Sustainability*, vol. 14, no. 13, p. 8097, Jul. 2022.

[38] D. Jovanovic, M. Marjanovic, M. Antonijevic, M. Zivkovic, N. Budimirovic, and N. Bacanin, "Feature selection by improved sand cat swarm optimizer for intrusion detection," in *Proc. Int. Conf. Artif. Intell. Everything (AIE)*, Aug. 2022, pp. 685–690.

[39] B. Basturk and D. Karaboga, "An artificial bee colony (ABC) algorithm for numeric function optimization," in *Proc. IEEE Swarm Intell. Symp.*, Indianapolis, IN, USA. May 2006, pp. 1–12.
[40] D. Wu, H. Rao, C. Wen, H. Jia, Q. Liu, and L. Abualigah, "Modified sand cat swarm optimization algorithm for solving constrained engineering optimization problems," *Mathematics*, vol. 10, no. 22, p. 4350, Nov. 2022.
[41] A. Seyyedabbasi, "Solve the inverse kinematics of robot arms using sand cat swarm optimization (SCSO) algorithm," in *Proc. Int. Conf. Theor. Appl. Comput. Sci. Eng. (ICTASCE)*, Sep. 2022, pp. 127–131.
[42] F. Kiani, F. A. Anka, and F. Erenel, "PSCSO: Enhanced sand cat swarm optimization inspired by the political system to solve complex problems," *Adv. Eng. Softw.*, vol. 178, Apr. 2023, Art. no. 103423.
[43] A. Seyyedabbasi, "A reinforcement learning-based metaheuristic algorithm for solving global optimization problems," *Adv. Eng. Softw.*, vol. 178, Apr. 2023, Art. no. 103411.
[44] G. Biyu, C. Wei, Z. Jingfeng, and W. Qingyu, "Optimization of auto lamp spraying trajectory based on improved PSO algorithm," in *Proc. China Automat. Congr.*, 2021, pp. 7487–7492.
[45] W. Wencheng, Z. Gege, C. Xinlin, and W. Weixian, "Research on path planning of orchard spraying robot based on improved RRT algorithm," in *Proc. 2nd Int. Conf. Big Data Artif. Intell.*, Apr. 2020, pp. 311–316.
[46] Z. Fu, B. Xiao, C. Wu, and J. Yang, "A genetic algorithm-based surface segmentation method for spray painting robotics," in *Proc. 29th Chin. Control Decis. Conf. (CCDC)*, May 2017, pp. 4049–4054.
[47] W. Deng, H. Zhao, Y. Song, and J. Xu, "An effective improved co-evolution ant colony optimisation algorithm with multi-strategies and its application," *Int. J. Bio-Inspired Comput.*, vol. 16, no. 3, p. 158, 2020.
[48] B. Duo-Han, L. Xin, and W. Xin-Yuan, "Efficient image encryption algorithm based on 1D chaotic map," *J. Comput. Sci.*, vol. 47, no. 4, pp. 278–284, 2020.
[49] L. Jiao and L. Wang, "A novel genetic algorithm based on immunity," *IEEE Trans. Syst., Man, Cybern., A, Syst. Hum.*, vol. 30, no. 5, pp. 552–561, Sep. 2000.
[50] F. Wang, X. Wang, and S. Sun, "A reinforcement learning level-based particle swarm optimization algorithm for large-scale optimization," *Inf. Sci.*, vol. 602, pp. 298–312, Jul. 2022.
[51] X. Xia, L. Gui, F. Yu, H. Wu, B. Wei, Y. Zhang, and Z. Zhan, "Triple archives particle swarm optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4862–4875, Dec. 2020.
[52] B. Wei, X. Xia, F. Yu, Y. Zhang, X. Xu, H. Wu, L. Gui, and G. He, "Multiple adaptive strategies based particle swarm optimization algorithm," *Swarm Evol. Comput.*, vol. 57, Sep. 2020, Art. no. 100731.
[53] X. Xia, L. Gui, G. He, B. Wei, Y. Zhang, F. Yu, H. Wu, and Z.-H. Zhan, "An expanded particle swarm optimization based on multi-exemplar and forgetting ability," *Inf. Sci.*, vol. 508, pp. 105–120, Jan. 2020.
[54] W. Lu, C. Shi, H. Fu, and Y. Xu, "A power transformer fault diagnosis method based on improved sand cat swarm optimization algorithm and bidirectional gated recurrent unit," *Electronics*, vol. 12, no. 3, p. 672, Jan. 2023.

YINGDA HU was born in Xingtai, in 1998. He is currently pursuing the master's degree with the School of Mechanical Engineering, Sichuan University, China. His research interests include control theory, robotics, and computer theory.

RUIPING XIONG was born in Ruijin, in 1970. He received the Ph.D. degree in mechanical manufacturing and automation from Sichuan University, in 2006. His main research interests include electromechanical hydraulic integrated control technology, robotic control technology, and intelligent control technology.

JING LI was born in Weiyang, in 1995. He is currently pursuing the master's degree with the School of Mechanical Engineering, Sichuan University, China. His research interests include artificial intelligence and machine learning.

CHENGSHENG ZHOU was born in Chongqing, in 1997. He is currently pursuing the master's degree with the School of Mechanical Engineering, Sichuan University, China. His research interests include robot kinematics and control technology.

QIYUAN WU was born in Ruijin, in 1999. He is currently pursuing the master's degree with the College of Mechanical Engineering, Sichuan University, China. His research interests include hydraulic technology and control theory.