scientific reports

Check for updates

OPEN

Multi-strategy enhanced sand cat swarm optimization algorithm and its engineering applications

Meijin Lin✉, Zibin Dai, Zhirong Qiu, Hao Chen & Haokun Lin

To address the problem that the sand cat swarm optimization (SCSO) algorithm experiences a decline in convergence speed and a tendency to fall into local optima during the iteration process, this paper proposes a multi-strategy enhanced sand cat swarm optimization (MESCSO) algorithm to improve its ability to escape local optima and enhance convergence efficiency. Firstly, an improved sine mapping combined with random opposition-based learning (ISMROBL) is employed during population initialization to enhance the uniformity and diversity of initial solutions. Secondly, a nonlinear decreasing parameter is introduced to dynamically balance global exploration and local exploitation. Thirdly, generalized quadratic interpolation (GQI) is incorporated to strengthen global search capability, while the improved mean differential mutation (IMDM) strategy enhances local exploitation. Finally, accelerated opposition-based learning (AOBL) is applied to refine individual positions and improve the algorithm's ability to escape local optima. Experimental results on 23 standard benchmark functions and the CEC2014 benchmark functions show that MESCSO achieves superior performance compared to nine algorithms. In addition, MESCSO is tested on five constrained engineering design problems. The results demonstrate that, compared with SCSO, MESCSO yields improvements of 0.53%, 1.47%, 0.03%, 0.41%, and 0.31%, respectively, thereby confirming its effectiveness and applicability.

Keywords Sand cat swarm optimization algorithm, Generalized quadratic interpolation, Accelerated opposition-based learning, Engineering applications

With the rapid advancement of science and technology, numerous complex optimization problems have arisen in engineering domains. These problems are often characterized by high dimensionality, nonlinearity, multimodality, or discreteness. Traditional optimization methods, however, are typically limited by problem structure and scale, resulting in suboptimal performance in practical scenarios. Specifically, conventional approaches often involve high computational complexity and slow convergence, making it difficult to efficiently locate global optima within a reasonable timeframe when dealing with large-scale or highly complex problems[1,2]. Furthermore, in practice, constructing accurate mathematical models is often challenging and the optimization process is frequently subject to various constraints. These challenges further expose the limitations of traditional techniques and hinder their ability to deliver fast and high-precision solutions.

Metaheuristic algorithms (MAs) are a class of optimization methods inspired by natural phenomena, biological swarm behavior, or human social activities. These algorithms typically exhibit strong global search capabilities and do not rely on the specific mathematical characteristics or precise models of the problem. As a result, they are well-suited for solving high-dimensional, nonlinear, multimodal, discrete, and complex optimization problems with constraints. Owing to their flexibility and robustness, MAs are particularly effective in addressing complex scenarios commonly encountered in practical engineering problems.

MAs can generally be categorized into four groups based on their inspiration sources or operating mechanisms: evolutionary-based algorithms, swarm-based algorithms, physical-based algorithms, and human-based algorithms.

Evolutionary-based algorithms are inspired by biological evolution theory. These algorithms simulate natural selection, reproduction, and mutation processes to perform optimization. The most representative example is the Genetic Algorithm (GA)[3], which is grounded in Darwin's theory of evolution and applies selection, crossover, and mutation to simulate natural evolution. Related algorithms include Evolutionary Strategy (ES)[4], Genetic Programming (GP)[5], Differential Evolution (DE)[6], and Evolutionary Programming (EP)[7].

School of Mechatronic Engineering and Automation, Foshan University, Foshan 528200, China. ✉email: linmeijin@fosu.edu.cn

Swarm-based algorithms simulate collective behaviors and cooperative mechanisms observed in biological swarms. Particle Swarm Optimization (PSO)[8] simulates the foraging behavior of birds. Sparrow Search Algorithm (SSA)[9] simulates the foraging and vigilance of sparrow populations. Dragonfly Algorithm (DA)[10] simulates dragonfly predation and migration. And the Firefly Algorithm (FA)[11] is based on the mutual attraction via light signals among fireflies. Other notable swarm-based algorithms include Sled Dog Optimizer (SDO)[12], Snow Geese Algorithm (SGA)[13], Red-billed Blue Magpie Optimizer (RBMO)[14], and Blood-Sucking Leech Optimizer (BSLO)[15].

Physical-based algorithms are inspired by fundamental principles in physics, such as mechanics, thermodynamics, and electromagnetism. These methods simulate the physical properties and dynamic behaviors of particles or substances to address optimization problems. Simulated Annealing (SA)[16] is a classic example that emulates the annealing process in metallurgy, where temperature decreases gradually to achieve system stability. Similar algorithms include Atom Search Optimization (ASO)[17], Gravitational Search Algorithm (GSA)[18], Chernobyl Disaster Optimizer (CDO)[19], Energy Valley Optimizer (EVO)[20], Nuclear Reaction Optimization (NRO)[21], and Electromagnetic Field Optimization (EFO)[22].

Human-based algorithms are inspired by interactions, decision-making, and social phenomena in human societies. These algorithms simulate dynamic processes in collective human behavior. Brain Storm Optimization (BSO)[23] simulates the brainstorming process. Queuing Search Algorithm (QSA)[24] draws inspiration from human queuing behavior. And Teaching-Learning-Based Optimization (TLBO)[25] simulates the interactions between teachers and learners in a classroom. Similar algorithms include Mountaineering Team-Based Optimization (MTBO)[26], Poor And Rich Optimization (PRO)[27], Gaining Sharing Knowledge Based Algorithm (GSK)[28], Gold Rush Optimizer (GRO)[29], and Cognitive Learning Optimizer (CLO)[30].

In recent years, due to the growing complexity and diversity of real-world optimization problems, the field of metaheuristic optimization has undergone significant advancements. Numerous novel algorithms have been proposed to improve global search capability, convergence speed, and solution robustness. The Frigatebird Optimizer (FBO)[31] simulates the food-snatching and predatory behaviors of frigatebirds. It operates in two stages: the first emulates the disturbance behavior used to steal food from other seabirds, enabling global exploration; the second simulates diving attacks to guide individuals in converging toward the optimal solution, thereby facilitating local exploitation. The Gyro Fireworks Algorithm (GFA)[32] is inspired by the firing process of a gyro-style firework and adopts a multi-stage search strategy. During early iterations, the algorithm assumes a fully fueled firework, enabling extensive global exploration via Lévy flight and spiral motion. As the "fuel" depletes in later iterations, the search narrows with spiral convergence toward optimal solutions, thereby enhancing local refinement. The Fishing Cat Optimizer (FCO)[33] divides the search process into four stages, reflecting the ambush and hunting strategies of fishing cats. In the early phase, it simulates waiting and sensing behaviors to perform alternating expansive and contractive global exploration. In the later phase, it simulates diving and capturing behaviors to conduct a similarly alternating local exploitation strategy. Through this four-phase dynamic mechanism, FCO[33] achieves an effective balance between exploration and exploitation, thereby enhancing search efficiency and the capability to avoid local optima.

Although MAs have demonstrated broad applicability in solving complex non-linear and multimodal problems, their optimization performance varies significantly across different application scenarios. According to the "No Free Lunch" (NFL) theorem[34], no single algorithm can achieve optimal performance on all types of problems. As a result, continuously exploring and incorporating more adaptive improvement strategies has become essential for enhancing the generality and practical value of metaheuristic algorithms. This research direction has gradually emerged as a core trend in the development and advancement of metaheuristics in recent years[35].

The Sand Cat Swarm Optimization algorithm (SCSO)[36] is a novel swarm-based metaheuristic algorithm proposed by Amir Seyyedabbasi et al. in 2022. The SCSO algorithm simulates the hunting behavior of sand cats in the wild, especially their keen perception of low-frequency sounds and their ability to lock onto prey, and establishes an efficient balance between global exploration and local exploitation by simulating the two phases of detecting and attacking prey by sand cats, so as to achieve an efficient optimization solution. Peng et al. proposed a Multi-strategy Integrated Sand Cat Swarm Optimization Algorithm (MSCSO)[37], which integrates a good point set strategy, dynamic nonlinear adjustment of the search range, and an alert mechanism from the Sparrow Search Algorithm to significantly enhance global search ability. Li et al. proposed an improved version of SCSO, called CWXSCSO[38], which employs a new dynamic exponential factor, an elite decentralization strategy, and crossbar strategy to enhance the global optimization capability. Cai et al. proposed an Improved Sand Cat Swarm Optimization Based on Lens Opposition-based Learning and Sparrow Search Algorithm (LSSCSO)[39] by combining dynamic spiral search with lens contrast learning to broaden the search range and improve population diversity. Zhang et al. proposed an Improved Multi-Strategy Sand Cat Optimization Algorithm (IMSCSO)[40], which incorporates fitness-distance balancing based on roulette selection, population perturbation, and best worst mutation mechanism to significantly enhance search performance and effectively avoid premature convergence. These improved SCSO algorithms are summarized in Table 1.

Although numerous MAs have been widely applied to complex optimization problems, their performance still varies significantly across different tasks, indicating that their adaptability and generalization capabilities require further improvement. In recent years, strategy fusion and mechanism coordination have emerged as key research directions for enhancing algorithmic performance[35]. To this end, this paper proposes a Multi-Strategy Enhanced Sand Cat Swarm Optimization Algorithm (MESCSO) aimed at comprehensively improving optimization effectiveness.

The enhancement strategies in MESCSO algorithm focus on five main aspects. Firstly, to improve the quality of the initial population, an improved sine mapping is combined with a random opposition-based learning mechanism during initialization. This strategy enhances population diversity and broadens the search space

Algorithm	Advantages	Year
MSCSO[37]	Improved global search ability	2024
CWXSCSO[38]	Enhanced global optimization capability	2024
LSSCSO[39]	Broadened search range and diversity	2024
IMSCSO[40]	Improved search performance, avoids premature convergence	2024

Table 1. Summary of improved SCSO algorithms.

for potential optimal solutions. Secondly, a new nonlinear decreasing parameter is designed to dynamically regulate algorithmic parameters, improving adaptability and balance during the search process and enabling better coordination between exploration and exploitation at different stages. Thirdly, a generalized quadratic interpolation (GQI) strategy is introduced during the exploitation phase, allowing individuals to escape local regions and approach the global optimum more efficiently. Fourthly, an improved mean differential mutation (IMDM) strategy is employed in the exploration phase, which leverages average differences among individuals to enhance global search capability in high-dimensional complex spaces and prevent premature convergence. Finally, an accelerated opposition-based learning (AOBL) mechanism is probabilistically applied after each iteration based on a random threshold to refine population positions. Combined with a greedy selection strategy, it preserves individuals with better fitness, thereby improving both population diversity and overall search performance.

Through the integration of the five aforementioned strategies, the global exploration ability of the MESCSO algorithm is significantly enhanced, and its convergence performance is notably improved. To comprehensively evaluate its optimization capability, extensive experiments were conducted on 23 standard benchmark functions and CEC2014 benchmark functions. The experimental analysis includes data statistics, convergence curves, box plot, and the Wilcoxon rank sum test. Furthermore, to verify the practical applicability of the proposed algorithm in engineering optimization, five representative engineering problems were selected for testing. The experimental results demonstrate that MESCSO algorithm performs well in solving complex optimization problems.

The main contributions of this paper are as follows:

1. The ISMROBL strategy is employed to enhance the population diversity of sand cats, thereby improving the algorithm's global optimization capability.
2. An IMDM strategy is adopted during the exploration phase to expand the search space and further strengthen global exploration.
3. In the exploitation phase, a GQI strategy is incorporated to help sand cat individuals escape local optima and locate better solutions, enhancing the algorithm's search performance.
4. An AOBL mechanism is introduced after each iteration to dynamically correct individual positions according to the probabilities. When trapped in local optima, the MESCSO algorithm applies opposition-based random reinitialization to improve its ability to escape.
5. The proposed MESCSO algorithm is evaluated and compared with the other nine algorithms, demonstrating superior optimization performance.

The remainder of this paper is organized as follows: "The sand cat swarm optimization algorithm (SCSO)" section reviews the fundamental principles of the SCSO algorithm and "The multi-strategy enhanced sand cat swarm optimization algorithm (MESCSO)" section presents the improvement strategies of the MESCSO algorithm. "Experimental results and analysis" section reports the experimental results on 23 standard benchmark functions and CEC2014 benchmark functions. "Structural engineering optimization problems section" evaluates the performance of MESCSO algorithm on five constrained engineering design problems. Finally, "Conclusions" section concludes this paper.

The sand cat swarm optimization algorithm (SCSO)

Initialize population

In the initialization phase of the SCSO algorithm, the problem dimension d and the number of sand cat individuals m must first be defined. The position of each sand cat is represented as $X = (x_1, x_2, \ldots, x_d)$, where each component x_i must be located between the lower and upper boundaries. By generating independent random values for each dimension within the search space, an initial solution matrix of size $m \times d$ is formed, as shown in Eq. (1). Each individual in this initial population is then evaluated using the fitness function, and the results are ranked to identify and record the position of the best-performing sand cat. In subsequent generations, the remaining sand cats iteratively approach this optimal position or explore its vicinity, aiming to improve the overall solution quality. If the optimal solution found in the current generation outperforms that of the previous generation, it is stored as the new global best. Otherwise, the global best remains unchanged.

$$X = (ub - lb) \times rand(0, 1) + lb \tag{1}$$

where *ub* and *lb* represent the upper and lower limits of the search area, and *rand*(0, 1) is a random number between 0 and 1.

Exploration phase

During the exploration phase, sand cats rely on their high sensitivity to low-frequency sound waves to detect the location of prey. Their sensitivity to low-frequency signals is defined within a specific range denoted as r_G. To quantify this auditory sensitivity, a parameter S_M is induced to represent the maximum detectable range of the sound, and its value is set to 2, which indicates that the sensitivity gradually decays from 2 kHz to 0. As the number of iterations t increases, the sensitivity is linearly scaled down according to Eq. (2).

$$r_G = S_M - \left(\frac{S_M \times t}{T}\right) \tag{2}$$

where t represents the current iteration number, and T denotes the maximum number of iterations. Based on this, a global control factor R is defined to balance the processes of exploration and exploitation.

$$R = 2 \times r_G \times rand(0,1) - r_G \tag{3}$$

The actual radius of sensitivity for each sand cat r is determined by the following equation.

$$r = r_G \times rand(0,1) \tag{4}$$

Subsequently, each sand cat updates its position $X(t+1)$ for the next generation by comparing its current position $X_c(t)$ with the current best candidate position $X_{bc}(t)$. The position update is defined as:

$$X(t+1) = r \times (X_{bc}(t) - rand(0,1) \times X_c(t)) \tag{5}$$

Exploitation phase

In the exploitation phase, it is assumed that the sand cat's hunting range is represented as a circular area. Each sand cat randomly generates an attack angle θ using a roulette wheel selection mechanism. The value of θ is randomly selected within the range of 0° to 360°. This angle determines the direction in which the sand cat moves within the circular region. To describe the distance between the sand cat and its prey, Eq. (6) is defined as follows:

$$X_{rand} = |rand(0,1) \times X_b(t) - X_c(t)| \tag{6}$$

where X_{rand} represents the distance between the current best position $X_b(t)$ and the current position $X_c(t)$ of each sand cat in generation t. Then, based on the auditory sensitivity radius r and the randomly generated angle θ, the updated position of the sand cat in generation $t+1$, denoted as $X(t+1)$, is obtained as follows:

$$X(t+1) = X_b(t) - r \times X_{rand} \times \cos(\theta) \tag{7}$$

By introducing random angles and a cosine factor, sand cats are enabled to move and hunt in various directions. As the iteration progresses, the sand cats continuously reduce their search radius and attempt to lock onto prey more precisely. This mechanism enhances the diversity of the algorithm and effectively suppresses premature convergence.

Bridging phase

The SCSO algorithm achieves seamless switching between prey-searching and prey-attacking modes through a global control factor R, as defined in Eq. (3). When $|R| \leq 1$, the sand cat performs a prey-attacking operation (exploitation phase). Otherwise, it remains in the prey-searching operation (exploration phase). The specific position update rule is given in Eq. (8).

$$X(t+1) = \begin{cases} r \times (X_{bc}(t) - rand(0,1) \times X_c(t)), & |R| > 1 \\ X_b(t) - r \times X_{rand} \times \cos(\theta), & |R| \leq 1 \end{cases} \tag{8}$$

Through the above mechanism, the SCSO algorithm can dynamically adjust its search strategy at different stages and maintain a relatively fast convergence rate. The corresponding pseudo-code is presented in Algorithm 1.

```
Initialize the population and other parameters like r, r_G, and R
Calculate the fitness function based on the objective function
while t ≤ maximum iteration do
    for each search agent do
        Get a new angle value θ obtained by Roulette Wheel Selection (0° ≤ θ ≤ 360°)
        if |R| ≤ 1 then
            Update the search agent position based on Equation (7)
        else
            Update the search agent position based on Equation (5)
        end if
        Check whether the search agent is beyond the upper/lower boundaries
    end for
    t = t + 1
end while
```

Algorithm 1. Pseudo-Code of SCSO algorithm

The multi-strategy enhanced sand cat swarm optimization algorithm (MESCSO)

ISMROBL population diversity

Since the SCSO algorithm adopts a purely random strategy during the population initialization phase, it may result in uneven distribution of individuals, thereby reducing the diversity and quality of the initial population and further affecting the convergence performance of the algorithm. The initial position of each individual in the population plays a crucial role in determining the optimization effectiveness of swarm intelligence algorithms. Compared to traditional random generation based solely on probability, chaotic mapping is characterized by randomness, nonrepetitiveness, and unpredictability, which can help ensure better uniformity in population distribution. Based on this, MESCSO algorithm incorporates chaotic mapping during population initialization to enhance the diversity of potential solutions. In contrast to conventional random initialization methods, chaotic mappings introduce nonlinear dynamic behavior, which strengthens the exploration ability and robustness of swarm intelligence algorithms, while effectively avoiding premature convergence. Commonly used chaotic mappings include logistic mapping, tent mapping, cubic mapping, and sine mapping.

Sine mapping is widely used in chaotic mappings due to its simple structure and high computational efficiency, but its uneven probability density distribution limits its performance to some extent. In order to overcome this shortcoming, this paper improves the Sine mapping to enhance its performance in terms of spatial traversal, population diversity, and uniformity of phase plane distribution. The improved Sine chaos mapping is shown in Eq. (9).

$$\begin{cases} a_{i+1} = \sin(k_1 \pi a_i) \cdot \cos(k_2 \pi b_i) \\ b_{i+1} = \sin(k_3 \pi b_i) + \mu \cdot a_i \cdot (1 - b_i) \\ y_{i+1} = (a_{i+1} + b_{i+1}) \bmod 1 \end{cases} \tag{9}$$

In the Eq. (9), the initial values of a_i and b_i are randomly selected from the interval (0,1). The control parameters are set as follows: $k_1 = 5$, $k_2 = 3$ and $k_3 = 4$, where $\mu = 0.5$ denotes the nonlinear coupling strength coefficient. The variable y_{i+1} represents the normalized chaotic sequence generated by the chaotic mapping, and the operator "mod" denotes the modulo operation. To map the chaotic sequence to the actual solution space, it is necessary to perform scaling and translation so that the sequence falls within the defined problem domain [*lb*, *ub*]. The transformation equation is given as follows:

$$X_{sine} = (ub - lb) \cdot y_{i+1} + lb \tag{10}$$

where X_{sine} is a candidate solution generated using the improved sine mapping.

The core idea of Opposition-Based Learning (OBL)[41] is to compare the current solution with its corresponding opposite solution and select the better one to guide the next generation of evolution, thereby reducing the probability of falling into local optima. The basic formulation of OBL can be expressed as:

$$X_{OBL} = ub + lb - X \tag{11}$$

However, the original OBL strategy lacks sufficient randomness, which may lead to performance degradation in complex or high-dimensional search spaces. In such cases, the opposition solutions may suffer from limited diversity due to their uniform distribution. To address this issue, this paper introduces a random disturbance term to enhance the diversity and exploration ability of opposition solutions, resulting in the formulation of the Random Opposition-Based Learning (ROBL) strategy[42], as described below:

$$X_{ROBL} = rand(0, 1) \cdot (ub + lb) - X_{sine} \tag{12}$$

where X_{ROBL} is a candidate solution generated using the ROBL strategy.

The ISMROBL population diversity strategy is as follows: first, the improved sine chaotic mapping strategy is employed to initialize the population. Then, the ROBL strategy is applied to generate opposition individuals. By comparing the fitness values of the original individuals and their opposites, the top N individuals with the best fitness are selected to form the final initial population. This initialization strategy enables the algorithm to begin the evolutionary process with individuals of higher fitness, thereby accelerating convergence.

Nonlinear decreasing parameter

In the SCSO algorithm, the convergence factor r_G is gradually reduced from 2 to 0 in a linearly decreasing manner. However, this approach starts to gradually shift to local search only in the middle of the iteration, which results in low search efficiency and limited optimization effect. To address this limitation, this paper proposes a nonlinear decreasing parameter to enhance the exploration ability of the algorithm in the global search space. Compared with the SCSO algorithm, the improved algorithm maintains a stable exploration intensity at the beginning of iterations to avoid missing potential critical regions. And it accelerates the convergence in the middle and late stages to significantly improve the local exploitation efficiency. This nonlinear decreasing parameter helps to better balance the global exploration and local exploitation during the whole iteration process, and its mathematical expression is as follows:

$$r_G = 2 \times \left(\frac{1}{1 + e^{12(t/T - 0.5)}} \right)^{(1.1 + 0.8(t/T)^3)} \tag{13}$$

GQI strategy

Polynomial interpolation is a method that estimates a local minimum of the objective function by constructing a polynomial based on the function values at several specific points. However, traditional quadratic interpolation methods typically require three selected points, and their accuracy and performance are highly dependent on the distribution of these points. If the selected points are poorly distributed, the interpolation accuracy may degrade significantly, and the algorithm may waste computational resources in ineffective regions of the search space.

To address these shortcomings, this paper introduces a Generalized Quadratic Interpolation (GQI) strategy[43], which constructs a quadratic model based on any three distinct points. This enables more accurate estimation of the local minimum of the objective function. The GQI method provides more explicit guidance for local search, thereby improving search efficiency and solution quality. The formulation of GQI for obtaining the minimum of the interpolation function under all conditions is given as follows:

$$\begin{cases} x^* = \dfrac{(x_j^2 - x_k^2)f(x_i) + (x_k^2 - x_i^2)f(x_j) + (x_i^2 - x_j^2)f(x_k)}{2((x_j - x_k)f(x_i) + (x_k - x_i)f(x_j) + (x_i - x_j)f(x_k))} & \text{if } x_j < x_i < x_k \text{ or } x_k < x_i < x_j \\ \left.\begin{array}{l} x' = \dfrac{(x_j^2 - x_k^2)f(x_i) + (x_k^2 - x_i^2)f(x_j) + (x_i^2 - x_j^2)f(x_k)}{2((x_j - x_k)f(x_i) + (x_k - x_i)f(x_j) + (x_i - x_j)f(x_k))} \\ x^* = x' \end{array}\right\} & \text{if } x' < x_j \\ \left.\begin{array}{l} x'k = 3x_i - 2x_j \\ x^* = \dfrac{(x_j^2 - {x_k}^2)f(x_i) + ({x_k}^2 - x_i^2)f(x_j) + (x_i^2 - x_j^2)f(x_k)}{2((x_j - x_k)f(x_i) + (x_k - x_i)f(x_j) + (x_i - x_j)f(x_k))} \end{array}\right\} & \text{else} \\ \left.\begin{array}{l} x' = \dfrac{(x_j^2 - x_k^2)f(x_i) + (x_k^2 - x_i^2)f(x_j) + (x_i^2 - x_j^2)f(x_k)}{2((x_j - x_k)f(x_i) + (x_k - x_i)f(x_j) + (x_i - x_j)f(x_k))} \\ x^* = x' \end{array}\right\} & \text{if } x' x_j \\ \left.\begin{array}{l} x'_k = 3x_i - 2x_j \\ x^* = \dfrac{(x_j^2 - {x_k}^2)f(x_i) + ({x_k}^2 - x_i^2)f(x_j) + (x_i^2 - x_j^2)f(x_k)}{2((x_j - x_k)f(x_i) + (x_k - x_i)f(x_j) + (x_i - x_j)f(x_k))} \end{array}\right\} & \text{else} \\ \left.\begin{array}{l} x'_j = 2x_i - x_k \\ x^* = \dfrac{({x_j}^2 - x_k^2)f(x_i) + (x_k^2 - x_i^2)f(x_j) + (x_i^2 - {x_j}^2)f(x_k)}{2((x'_j - x_k)f(x_i) + (x_k - x_i)f(x_j) + (x_i - x'_j)f(x_k))} \end{array}\right\} & \text{if } x_i < x_k < x_j \text{ or } x_j < x_k < x_i \end{cases} \tag{14}$$

where x_i, x_j, and x_k represent any three selected points. x', x'_j, and x'_k denote the updated positions of these points obtained through interpolation. $f(x_i)$, $f(x_j)$, and $f(x_k)$ are the fitness values corresponding to the three selected points. x^* denotes the optimal position obtained using the GQI strategy.

In MESCSO algorithm, the GQI strategy is employed to update the sand cat population in order to enhance local exploitation. Set a random number p between 0 and 1. When $p > 0.5$, the exploitation phase follows the formulas in the original SCSO algorithm. When $p < 0.5$, the GQI strategy is applied. The position update based on the GQI strategy is given as follows:

$$X(t+1) = X^*(t) + w_1 \cdot \left(X_b(t) - \text{round}\,(1 + rand) \cdot \frac{ub - lb}{ub_m - lb_m} \cdot X_{im}(t) \right) \tag{15}$$

$$X^*(t) = GQI\left(X_b(t), X_{r1}(t), X_{r2}(t), f(X_b(t)), f(X_{r1}(t)), f(X_{r2}(t))\right) \tag{16}$$

$$w_1 = 3\left(1 - \frac{t-1}{T}\right) n \tag{17}$$

where n follows a standard normal distribution, and m is a randomly selected integer within the interval [1, d]. Equation (16) represents the GQI function, which is used to determine the minimum of the interpolated function constructed by $(X_b(t),\ f(X_b(t))$, $(X_{r1}(t),\ f(X_{r1}(t))$, and $(X_{r2}(t),\ f(X_{r2}(t))$. Here, $X_{r1}(t)$ and $X_{r2}(t)$ are the positions of two different individuals randomly selected from the current population, and they are distinct from the position $X_b(t)$.

IMDM strategy

To address the issues of premature convergence and insufficient population diversity in SCSO algorithms, this paper builds upon the Mean Differential Mutation (MDM) strategy proposed by Wang et al.[44] and introduces further improvements. The MDM strategy is inspired by the concepts of DE and Mean Particle Swarm Optimization (MeanPSO)[45], and has demonstrated promising results in practical applications. However, MDM is susceptible to limitations due to the directionality of initial differences and the use of mean values, which may hinder the algorithm's ability to escape from local optima.

To further improve population diversity and global search performance, this paper proposes an improved version of MDM, referred to as IMDM. Based on the original difference vector guided by the mean, IMDM enhances the incorporation of global population information, allowing the search direction to be adaptively adjusted according to the current population state. In addition, a self-adaptive control factor is introduced into the update and mean-guided difference process. IMDM can dynamically regulate the search strength and direction based on the current population state, thereby reducing the risk of premature convergence to local optima.

In MESCSO algorithm, the update of each sand cat is guided not only by its own historical best position $Y(t)$ and the global best position $P_g(t)$, but also by the population mean position $P_m(t)$. Compared with the original mechanism, the IMDM strategy significantly improves the algorithm's ability to escape local optima. Moreover, it better leverages both individual and population information, thus effectively enhancing population diversity at both the individual and group levels. The mathematical formulation of the IMDM strategy during the exploration phase of sand cats is expressed as follows:

$$\hat{X}_{ij}(t) = w_2 X_{ij}(t) + c_1 r_1 \left(Y_{ij}(t) - X_{ij}(t)\right) + c_2 r_2 \left(P_{mj}(t) - X_{ij}(t)\right) \tag{18}$$

$$P_m(t) = \frac{1}{N}\sum_{i=1}^{N} P_i(t) \tag{19}$$

$$w_2 = w_{min} + \frac{(w_{max} - w_{min})}{1 + e^{-2.9\left(\frac{t}{T} - 0.5\right)}} \tag{20}$$

$$X_{ij}(t+1) = \hat{X}_{ij}(t) + c_3 r_3 \left[\frac{Y_{ij}(t) - P_{gj}(t)}{2} - \hat{X}_{ij}(t)\right] \tag{21}$$

where r_1, r_2, and r_3 are random numbers within the range [0, 1]. c_1 and c_2 are two important control parameters called acceleration coefficients with a value of 1.4. c_3 is the mutation coefficient with a value of 0.55. w_{max} and w_{min} represent the maximum and minimum values of the inertia weight, respectively.

AOBL update mechanism

In swarm intelligence optimization algorithms, the search process typically consists of two stages: exploration and exploitation. The position update strategies employed in each stage differ in nature. To further enhance the global search capability and convergence efficiency of the algorithm, this paper introduces an Accelerated Opposition-Based Learning (AOBL) mechanism[46] following the basic position update. AOBL is used to refine the current candidate solutions. This mechanism provides greater flexibility in the early stages of the algorithm, thereby enhancing global exploration capability and preventing premature convergence. In the later stages, it gradually reduces exploration and increases exploitation ability, improving the overall optimization performance and achieving a dynamic balance between exploration and exploitation.

Specifically, after completing the standard position update in the exploration or exploitation phase, a random number $rand \in [0, 1]$ is generated. If $rand < 0.5$, the AOBL mechanism is triggered. The corresponding opposite position is then calculated and used to replace the current updated position. This mechanism performs the position correction according to the following way:

$$X_{AOBL} = e_{ac} \cdot (r_4 \cdot ub + r_5 \cdot lb) - X \tag{22}$$

where r_4 and r_5 are random numbers within the range [0, 1]. X_{AOBL} is a candidate solution generated using the AOBL mechanism.

To achieve a dynamic balance between exploration and exploitation, an acceleration coefficient e_{ac} is introduced. This coefficient gradually decreases with the number of generations, thereby regulating the influence of the AOBL mechanism. The corresponding calculation formula is as follows:

$$e_{ac} = e_{max} - \frac{t}{T} \times (e_{max} - e_{min}) \tag{23}$$

where e_{max} and e_{min} represent the maximum and minimum values of the acceleration coefficient, respectively. In this paper, the values are set as $e_{max} = 1$ and $e_{min} = 1e\text{-}5$.

Finally, to ensure solution quality, a greedy selection strategy is employed. The updated position X_{AOBL}, obtained through AOBL mechanism, is compared with the original updated position X based on fitness. If the fitness of X_{AOBL} is better, it is adopted as the final updated position. Otherwise, the original position X is retained. This mechanism effectively enhances the algorithm's ability to escape local optima while ensuring the steady improvement of the optimization process.

MESCSO algorithm flowchart and pseudocode

The pseudocode of the MESCSO algorithm is presented in Algorithm 2, and its overall flow chart is illustrated in Fig. 1.

```
Initialize the population and other parameters like r, r_G, and R
Calculate the fitness function based on the objective function
while t < maximum iteration do
    for each search agent do
        Get a new angle value θ obtained by Roulette Wheel Selection (0° ≤ θ ≤ 360°)
        if abs(R) ≤ 1 then
            Update the search agent position based on Equation (20)
        else
            if p > 0.5 then
                Update the search agent position based on Equation (7)
            else
                Update the search agent position based on Equation (16)
            end if
        end if
    end for
    Check that the location of the search agent is beyond the upper and lower boundaries
    if rand < 0.5 then
        Carry out the AOBL mechanism according to Equation (22)
        Check that the location of the search agent is beyond the upper and lower boundaries
    end if
    t = t + 1
end while
```

Algorithm 2. Pseudo-Code of MESCSO algorithm

Time complexity analysis

The time complexity of the proposed MESCSO algorithm is primarily influenced by the population size (N), the dimensionality of the optimization problem (dim), the maximum number of iterations (T), and the cost of evaluating the objective function (C). The overall computational complexity comprises several components, including the initialization phase and the iterative update process, which incorporates multiple enhancement strategies. The specific time complexity analysis is as follows:

1. The initialization parameter time is $O(1)$.
2. The initial population position time is $O(N \times dim)$.
3. The calculation time cost of ISMROBL strategy $O(N \times C)$.
4. The time cost for the sand cat to prey is $O(T \times N \times dim)$.
5. The cost time of the calculation function includes the calculation time cost of the algorithm itself $O(T \times N \times C)$, the calculation time cost of GQI strategy $O(T \times N \times C)$.
6. A probabilistically triggered AOBL strategy mechanism. Since AOBL is executed with a probability of 0.5 per iteration, its expected computational cost is $O(0.5 \times T \times N \times dim)$ for position generation and $O(0.5 \times T \times N \times C)$ for the calculation time cost.

Combining all components, the total time complexity of the MESCSO algorithm is expressed as:

$$O(\text{MESCSO}) = O(1 + N \times dim + N \times C + 1.5 \times T \times N \times dim + 2.5 \times T \times N \times C) \tag{24}$$

Although MESCSO introduces a moderate computational overhead as a result of incorporating multiple enhancement strategies, this increase is justified by the significant improvements in optimization performance, as demonstrated through comparative experiments with SCSO presented in Section "Experimental Results and Analysis".

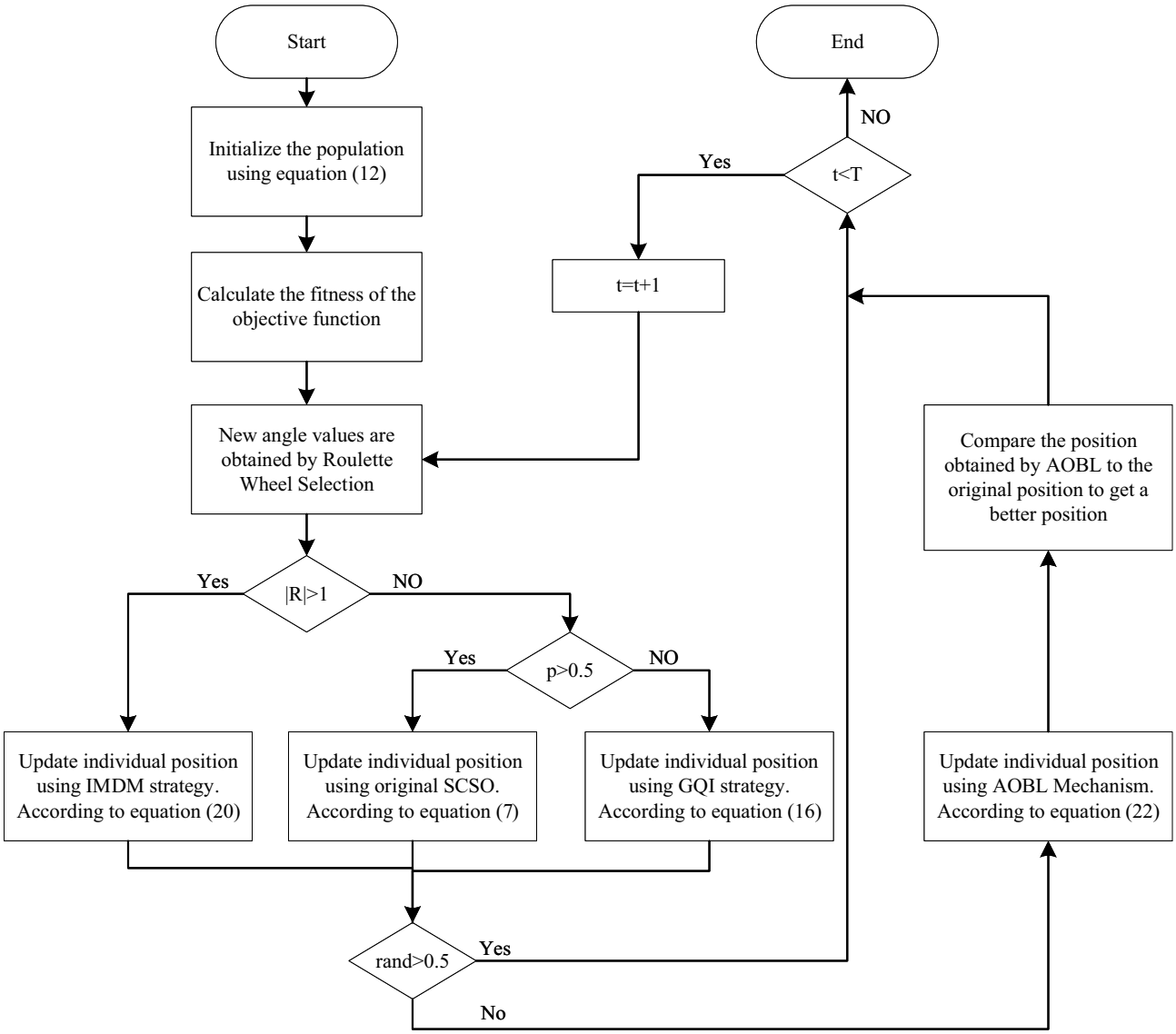

Fig. 1. Flow chart of the MESCSO algorithm.

Experimental results and analysis

All experiments in this paper were conducted on a computer equipped with an AMD 7940H processor, 16 GB memory, and a 64-bit Windows 11 operating system, using MATLAB 2023b.

To evaluate the overall performance of the MESCSO algorithm, 23 standard benchmark functions and CEC2014 benchmark functions were selected. These benchmark functions cover a variety of optimization problems, including unimodal, multi-modal, and high-dimensional noisy types, which provide a more comprehensive assessment of the algorithm's adaptability and performance across diverse problem scenarios.

To better demonstrate the optimization capability of MESCSO, it was compared with several well-known metaheuristic algorithms, including the Sand Cat Swarm Optimization (SCSO)[36], Arithmetic Optimization Algorithm (AOA)[47], Subtraction-Average-Based Optimizer (SABO)[48], Whale Optimization Algorithm (WOA)[49], Beluga Whale Optimization (BWO)[50], Dung Beetle Optimizer (DBO)[51], Harris Hawks Optimization (HHO)[52], Kepler Optimization Algorithm (KOA)[53], and Sine Cosine Algorithm (SCA)[54]. The parameter settings for these algorithms are listed in Table 2.

Experiments on the 23 standard benchmark functions

As shown in Table 3, a total of 23 standard benchmark functions were selected in this paper, including 7 unimodal functions, 6 multimodal functions, and 10 fixed-dimension multimodal functions. In the definitions, F represents the function itself, *dim* denotes the dimensionality of the function, Range indicates the search space, and F_{min} refers to the known global optimum value. For the experimental setup, the population size N was set to 30, the problem dimensions *dim* was set to 30 and 500, and the maximum number of iterations T was 500. To comprehensively evaluate the performance of the algorithms, MESCSO algorithm and the nine

Algorithm	Parameters	Value
MESCSO	S_M	2
	Roulette Wheel selection	[0, 360]
SCSO	S_M	2
	Roulette Wheel selection	[0, 360]
AOA	μ	0.499
SABO	α	5
	ν	$\{1, 2\}$
DBO	k	0.1
	λ	0.1
	b	0.3
	S	0.5
SCA	α	2
HHO	E_0	[-1, 1]
	J	[0, 2]
	T	3
KOA	u_0	0.1
	γ	15
BWO	W_f	[0.1, 0.05]
WOA	a	[2, 0]
	b	0.75
	l	[−1, 1]

Table 2. Parameter settings for the comparative algorithms.

other comparison algorithms were independently executed 30 times on each function. The average fitness and standard deviation of each algorithm were recorded.

Experimental results and analysis of 23 standard benchmark functions

Table 4 presents the statistical results of 10 algorithms tested on the 23 standard benchmark functions. To clearly display the statistics, the best average fitness value and standard deviation are bolded. For functions F1-F4 at dimensions 30 and 500, the MESCSO algorithm achieved the theoretical optimum in all cases. Among the other algorithms, only the AOA reached the theoretical optimum on the 30-dimensional function F2. For functions F5-F6 and F12-F13, the performance of MESCSO algorithm was second only to BWO algorithm and outperformed the other comparison algorithms. In the function F7, MESCSO algorithm achieved the best average fitness. For the function F8, MESCSO algorithm was outperformed by HHO and BWO algorithm, but still outperformed the rest. MESCSO algorithm also reached the theoretical optimum on functions F9-F11, along with SCSO, BWO, SABO, and HHO algorithm. The functions F14-F23 are relatively simple, making it easier to find good solutions. Compared with other algorithms, MESCSO algorithm found better fitness values on functions F14-F23.

However, the statistical results in Table 4 alone are not sufficient to fully demonstrate the optimization performance of the MESCSO algorithm across all 23 standard benchmark functions. To more intuitively compare the optimization capability of MESCSO algorithm, Figs. 2, 3 and 4 illustrate the convergence curves of MESCSO algorithm and the other nine comparison algorithms on the 23 standard benchmark functions.

As shown in the figures, MESCSO algorithm exhibits strong convergence ability on functions F1-F4 and F9-F11, quickly reaching the theoretical optimum. Although the SCSO, BWO, SABO, and HHO algorithms also found the optimal solutions on functions F9- F11, MESCSO algorithm achieved the fastest convergence rate. For functions F5-F6 and F12-F13, MESCSO algorithm converges slightly slower, but outperforms HHO algorithm and ranks just below BWO algorithm in terms of accuracy. In the case of functions F14-F23, all algorithms are able to obtain relatively good fitness values, indicating their general effectiveness in solving simpler problems. However, the MESCSO algorithm consistently achieves better fitness values compared to the others. Considering both the statistical results and the convergence curves, MESCSO algorithm demonstrates superior and more stable performance overall.

Convergence behavior analysis

To investigate differences in convergence behavior across algorithms and assess their impact on solution performance, a series of experiments were carried out to examine the convergence characteristics of MESCSO and SCSO. Several representative functions from the 23 standard benchmark functions were selected, each with a dimensionality of 30.

As shown in Fig. 5, the convergence results are presented in five columns, each representing a distinct analytical aspect. The first column visualizes the structure of the objective function's search space, offering insight into the function's inherent complexity and associated global optimization challenges. The second column displays the

Type	F	dim	Range	Fmin
Unimodal benchmark functions	$F_1(x) = \sum_{i=1}^{n} x_i^2$	30/500	[−100, 100]	0
	$F_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30/500	[−10, 10]	0
	$F_3(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{i} x_j \right)^2$	30/500	[−100, 100]	0
	$F_4(x) = \max_i \{ \lvert x_i \rvert, 1 \le i \le n \}$	30/500	[−100, 100]	0
	$F_5(x) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$	30/500	[−30, 30]	0
	$F_6(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$	30/500	[−100, 100]	0
	$F_7(x) = \sum_{i=1}^{n} i \cdot x_i^4 + \text{random}[0, 1)$	30/500	[−1.28, 1.28]	0
Multimodal benchmark functions	$F_8(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert} \right)$	30/500	[−500, 500]	−418.9829 × dim
	$F_9(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10 \right]$	30/500	[−5.12, 5.12]	0
	$F_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum x_i^2} \right) - \exp\left(\frac{1}{n}\sum \cos(2\pi x_i) \right) + 20 + e$	30/500	[−32, 32]	0
	$F_{11}(x) = \frac{1}{4000}\sum x_i^2 - \prod \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	30/500	[−600, 600]	0
	$F_{12}(x) = \frac{\pi}{n}\left[10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 (1 + 10\sin^2(\pi y_{i+1})) + (y_n - 1)^2 \right]$ $+ \sum_{i=1}^{n} u(x_i, 10, 100, 4), \quad y_i = 1 + \frac{x_i + 1}{4},$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \le x_i \le a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	30/500	[−50, 50]	0
	$F_{13}(x) = 0.1\left\{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] \right.$ $\left. + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \right\} + \sum u(x_i, 5, 100, 4)$	30/500	[−50, 50]	0
Fixed-dimension multimodal benchmark functions	$F_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$	2	[−65.536, 65.536]	1
	$F_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_1 (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	[−5, 5]	0.00030
	$F_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	2	[−5, 5]	−1.0316
	$F_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi} \right)\cos x_1 + 10$	2	[−5, 10] × [0, 15]	0.398
	$F_{18}(x) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2) \right]$ $\times \left[30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2) \right]$	2	[−2, 2]	3
	$F_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2 \right)$	3	[0, 1]	−3.86
	$F_{20}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2 \right)$	6	[0, 1]	−3.32
	$F_{21}(x) = -\sum_{i=1}^{5} \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	−10.1532
	$F_{22}(x) = -\sum_{i=1}^{7} \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	−10.4028
	$F_{23}(x) = -\sum_{i=1}^{10} \left[(x - a_i)(x - a_i)^T + c_i \right]^{-1}$	4	[0, 10]	−10.5363

Table 3. Details of the 23 standard benchmark functions.

historical search trajectories of the population throughout the search space. MESCSO demonstrates a wide and evenly distributed pattern, highlighting its robust global exploration capacity.

The third column tracks the evolution of the population's average fitness across iterations. MESCSO shows a more rapid and stable decline, converging earlier and more reliably. This indicates that MESCSO effectively exploits promising regions once identified. In contrast, SCSO shows an initial decline followed by oscillations and occasional rebounds, indicating susceptibility to local optima and unstable convergence. Notably, for function

F	*dim*	Metric	MESCSO	SCSO	AOA	SABO	DBO	SCA	HHO	KOA	BWO	WOA
F1	30	mean	**0**	7.32E-114	7.08E-34	4.15E-196	4.64E-105	1.01E+01	5.51E-90	1.37E+04	1.33E-253	3.78E-74
		std	**0**	3.65E-113	3.87E-33	0	2.53E-104	1.71E+01	2.74E-89	3.45E+03	0	1.47E-73
	500	mean	**0**	5.52E-100	6.52E-01	3.11E-199	1.85E-113	1.95E+05	5.77E-97	4.27E+05	2.90E-250	1.36E-69
		std	**0**	1.77E-99	4.00E-02	0	7.67E-113	6.66E+04	1.93E-96	6.27E+04	0	4.98E-69
F2	30	mean	**0**	4.32E-60	0	1.92E-110	4.60E-59	9.76E-03	9.46E-52	6.50E+22	8.40E-131	1.76E-49
		std	**0**	2.03E-59	0	6.95E-110	2.48E-58	9.89E-03	2.14E-51	2.98E+03	3.28E-130	7.83E-49
	500	mean	**0**	5.07E-52	1.01E-03	1.36E+01	3.09E-60	1.15E+02	3.93E-49	1.11E+128	1.84E-126	9.39E-48
		std	**0**	1.47E-51	1.64E-03	3.10E+01	1.56E-59	5.48E+01	1.73E-48	6.10E+128	9.57E-126	4.83E-47
F3	30	mean	**0**	2.83E-98	7.12E-03	2.23E-36	3.31E-49	9.11E+03	7.76E-75	4.71E+04	8.66E-244	4.27E+04
		std	**0**	1.54E-97	1.54E-02	1.22E-35	1.82E-48	5.03E+03	4.20E-74	1.45E+04	0	1.77E+04
	500	mean	**0**	2.02E-84	3.04E+01	1.12E-03	2.62E-07	6.84E+06	1.17E-29	1.17E+07	5.53E-229	3.14E+07
		std	**0**	1.06E-83	1.61E+01	6.10E-03	1.43E-06	1.48E+06	6.42E-29	3.41E+06	0	1.24E+07
F4	30	mean	**0**	1.31E-50	2.81E-02	3.53E-77	9.10E-49	3.77E+01	8.63E-50	5.33E+01	1.39E-126	5.59E+01
		std	**0**	5.31E-50	1.88E-02	6.38E-77	4.98E-48	9.67E+00	3.11E-49	5.82E+00	3.30E-126	2.80E+01
	500	mean	**0**	8.08E-45	1.82E-01	2.27E-71	3.91E-32	9.91E+01	1.81E-49	8.62E+01	2.75E-114	8.36E+01
		std	**0**	2.86E-44	2.04E-02	3.19E-71	2.14E-31	2.66E-01	6.86E-49	5.16E+00	1.13E-113	1.81E+01
F5	30	mean	8.91E-04	2.82E+01	2.84E+01	2.83E+01	2.58E+01	2.23E+04	1.01E-02	1.75E+07	**1.30E-05**	2.80E+01
		std	1.16E-03	6.98E-01	3.72E-01	4.66E-01	1.75E-01	5.08E+04	1.15E-02	1.15E+07	**3.48E-05**	4.64E-01
	500	mean	4.61E+02	4.98E+02	4.99E+02	4.98E+02	4.98E+02	1.85E+09	2.14E-01	8.40E+08	**2.60E-04**	4.96E+02
		std	1.25E+02	9.20E-02	1.07E-01	6.36E-02	2.12E-01	4.34E+08	2.84E-01	2.11E+08	**8.37E-04**	3.72E-01
F6	30	mean	3.44E-08	1.89E+00	3.21E+00	2.53E+00	7.28E-03	1.67E+01	1.36E-04	1.34E+04	**2.43E-14**	4.59E-01
		std	2.37E-08	5.42E-01	3.22E-01	4.37E-01	3.51E-02	3.58E+01	2.09E-04	3.02E+03	**3.53E-14**	2.83E-01
	500	mean	2.26E-05	1.06E+02	1.16E+02	1.11E+02	9.02E+01	1.89E+05	1.56E-03	4.26E+05	**1.33E-12**	3.30E+01
		std	3.79E-05	3.59E+00	9.12E-01	1.48E+00	3.43E+00	7.98E+04	2.00E-03	4.95E+04	**2.83E-12**	7.87E+00
F7	30	mean	**4.41E-05**	2.17E-04	8.16E-05	8.66E-05	1.19E-03	1.47E+01	1.48E-04	7.42E+00	1.45E-04	4.14E-03
		std	**3.76E-05**	3.17E-04	1.07E-04	6.62E-05	8.34E-04	2.36E-01	1.33E-04	3.45E+00	1.03E-04	4.19E-03
	500	mean	**8.28E-05**	2.24E-04	9.13E-05	1.10E-04	1.72E-03	1.48E+04	1.87E-04	6.90E+03	1.12E-04	5.49E-03
		std	8.65E-05	2.37E-04	**5.46E-05**	9.58E-05	1.57E-03	3.68E+03	1.86E-04	2.13E+03	1.13E-04	6.75E-03
F8	30	mean	-1.26E+04	-6.64E+03	-5.29E+03	-3.08E+03	-8.82E+03	-3.72E+03	-1.26E+04	-3.77E+03	**-1.26E+04**	-9.36E+03
		std	1.64E+00	7.54E+02	4.39E+02	3.66E+02	1.96E+03	2.52E+02	9.62E-01	3.78E+02	**1.21E-06**	1.45E+03
	500	mean	-2.09E+05	-5.99E+04	-2.31E+04	-1.21E+04	-1.35E+05	-1.55E+04	-2.09E+05	-4.40E+04	**-2.09E+05**	-1.61E+05
		std	1.84E+02	4.38E+03	1.57E+03	9.42E+02	3.83E+04	1.27E+03	2.59E+03	1.22E+04	**4.82E-07**	2.58E+04
F9	30	mean	**0**	**0**	**0**	**0**	5.67E-01	3.98E+01	**0**	2.84E+02	**0**	3.38E-02
		std	**0**	**0**	**0**	**0**	2.46E+01	3.20E+01	**0**	2.26E+01	**0**	1.85E-01
	500	mean	**0**	**0**	8.18E-06	**0**	**0**	1.14E+03	**0**	6.04E+03	**0**	3.03E-14
		std	**0**	**0**	7.75E-06	**0**	**0**	4.99E+02	**0**	2.03E+02	**0**	1.66E-13
F10	30	mean	**8.88E-16**	**8.88E-16**	**8.88E-16**	4.00E-15	9.63E-16	1.47E+01	**8.89E-16**	1.88E+01	**8.88E-16**	2.93E-15
		std	**0**	**0**	**0**	**0**	6.49E-15	8.38E+00	**0**	1.01E+00	**0**	1.90E-15
	500	mean	**8.88E-16**	**8.88E-16**	7.96E-03	4.00E-15	**8.88E-16**	1.91E+01	**8.88E-16**	1.99E+01	**8.88E-16**	4.94E-15
		std	**0**	**0**	8.30E-04	**0**	**0**	3.47E+00	**0**	1.85E-01	**0**	3.22E-15
F11	30	mean	**0**	**0**	2.17E-01	**0**	**0**	1.02E+00	**0**	1.40E+02	**0**	1.97E-02
		std	**0**	**0**	1.23E-01	**0**	**0**	3.97E-01	**0**	3.60E+01	**0**	6.40E-02
	500	mean	**0**	**0**	9.90E+03	**0**	**0**	1.80E+03	**0**	3.83E+03	**0**	**0**
		std	**0**	**0**	2.47E+03	**0**	**0**	7.34E+02	**0**	7.05E+02	**0**	**0**
F12	30	mean	1.37E-09	1.25E-01	5.15E-01	2.13E-01	2.37E-04	2.58E+04	1.10E-05	1.46E+07	**3.92E-14**	2.49E-02
		std	9.95E-10	6.20E-02	4.84E-02	1.09E-01	8.22E-04	1.06E+05	1.80E-05	9.51E+06	**7.49E-14**	2.62E-02
	500	mean	2.20E-08	7.93E-01	1.08E+00	9.47E-01	6.11E-01	5.80E+09	2.52E-06	1.16E+09	**6.86E-16**	1.07E-01
		std	2.79E-08	5.97E-02	9.44E-03	2.34E-02	4.16E-02	1.60E+09	5.49E-06	4.79E+08	**9.82E-16**	5.08E-02
F13	30	mean	2.75E-08	2.38E+00	2.85E+00	2.43E+00	6.98E-01	1.16E+05	7.57E-05	4.21E+07	**3.95E-13**	5.93E-01
		std	2.71E-08	4.50E-01	1.05E-01	6.50E-01	5.98E-01	3.17E+05	1.08E-04	2.27E+07	**1.06E-12**	2.48E-01
	500	mean	1.53E-06	4.98E+01	5.02E+01	4.99E+01	4.94E+01	9.31E+09	5.16E-04	2.61E+09	**4.03E-13**	1.97E+01
		std	1.27E-06	8.56E-02	3.99E-02	1.15E-02	1.53E-01	1.79E+09	7.74E-04	7.45E+08	**8.04E-13**	6.13E+00
F14	2	mean	**9.98E-01**	4.33E+00	1.06E+01	3.59E+00	1.36E+00	1.60E+00	1.13E+00	5.56E+00	1.03E+00	2.93E+00
		std	**8.75E-16**	4.03E+00	3.07E+00	2.70E-00	7.59E-01	9.23E-01	3.44E-01	3.45E+00	1.81E-01	2.91E+00
F15	4	mean	**3.49E-04**	4.73E-04	2.38E-02	1.54E-03	7.88E-04	1.04E-03	3.77E-04	8.21E-03	4.00E-04	7.60E-04
		std	1.67E-04	3.23E-04	3.19E-02	3.80E-03	3.23E-04	3.52E-04	1.98E-04	5.81E-03	**1.54E-04**	5.43E-04

Continued

F	*dim*	Metric	MESCSO	SCSO	AOA	SABO	DBO	SCA	HHO	KOA	BWO	WOA
F16	2	mean	**-1.03E+00**	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-1.03E+00	-9.74E-01	-1.03E+00	-1.03E+00
		std	**6.12E-16**	2.92E-10	1.23E-07	1.60E-02	8.29E-12	3.98E-05	4.41E-09	8.94E-02	6.61E-04	1.74E-09
F17	2	mean	**3.98E-01**	3.98E-01	4.11E-01	4.87E-01	3.98E-01	4.00E-01	3.98E-01	5.38E-01	3.99E-01	3.98E-01
		std	**2.65E-10**	1.51E-07	1.11E-02	2.08E-01	2.53E-02	2.73E-03	1.87E-05	1.78E-01	1.23E-03	5.08E-05
F18	2	mean	**3.00E+00**	3.00E+00	1.02E+01	4.76E+00	3.00E+00	3.00E+00	3.00E+00	4.72E+00	3.94E+00	3.00E+00
		std	**1.21E-08**	5.13E-06	1.21E+01	3.52E+00	3.03E-07	1.24E-04	1.50E-07	2.40E+00	9.02E-01	8.83E-05
F19	3	mean	**-3.86E+00**	-3.86E+00	-3.85E+00	-3.62E+00	-3.86E+00	-3.85E+00	-3.86E+00	-3.84E+00	-3.86E+00	-3.86E+00
		std	**7.38E-06**	3.43E-03	3.92E-03	1.77E-01	2.73E-03	2.56E-03	2.22E-02	2.54E-03	2.54E-03	9.68E-03
F20	6	mean	**-3.27E+00**	-3.16E+00	-3.04E+00	-3.25E+00	-3.21E+00	-2.88E+00	-3.06E+00	-3.01E+00	-3.27E+00	-3.22E+00
		std	**4.58E-02**	1.87E-01	9.09E-02	1.12E-01	7.66E-02	3.25E-01	1.47E-01	1.18E-01	5.99E-02	1.14E-01
F21	4	mean	**-1.02E+01**	-5.26E+00	-3.57E+00	-5.01E+00	-7.11E+00	-2.43E+00	-5.37E+00	-2.32E+00	-1.01E+01	-6.93E+00
		std	**8.74E-08**	1.53E+00	1.77E+00	4.60E-01	2.52E+00	1.75E+00	1.23E+00	1.57E+00	1.17E-02	2.95E+00
F22	4	mean	**-1.04E+01**	-6.06E+00	-4.46E+00	-5.12E+00	-7.93E+00	-2.89E+00	-5.42E+00	-2.56E+00	-1.04E+01	-6.81E+00
		std	**1.78E-07**	2.23E+00	1.82E+00	1.21E+00	2.69E+00	1.69E+00	1.28E+00	1.68E+00	5.49E-03	2.80E+00
F23	4	mean	**-1.05E+01**	-5.99E+00	-3.91E+00	-5.24E+00	-8.32E+00	-3.87E+00	-5.12E+00	-3.00E+01	-1.05E+01	-6.62E+00
		std	**1.11E-07**	2.08E+00	1.75E+00	1.07E+00	2.77E+00	1.65E+00	1.12E+00	1.51E+00	6.27E-03	3.43E+00

Table 4. Statistical results of the 23 standard benchmark functions.

F8, both algorithms exhibit significant fluctuations in average fitness, which may facilitate the exploration of potential optima in highly complex landscapes.

The fourth column illustrates the evolution of individual trajectories in the population's first dimension. SCSO individuals show slight early-stage fluctuations but quickly converge, implying premature convergence to local optima. MESCSO, on the other hand, maintains a broader trajectory range with more pronounced jumps, indicating enhanced capability to escape local traps and explore a wider solution space. The final column presents the convergence curves of the algorithms. MESCSO demonstrates superior overall convergence accuracy, with a more stable convergence trajectory. These results confirm that MESCSO exhibits significantly better optimization performance compared to the original SCSO algorithm.

Analysis of the Wilcoxon rank sum test results

The Wilcoxon rank sum test is a nonparametric statistical test commonly used to evaluate the statistical differences between improved algorithms and baseline algorithms. Although Table 4 provides the average fitness and standard deviation for each algorithm, it does not offer a sufficiently rigorous basis for determining the relative superiority of MESCSO algorithm over other optimization algorithms. Therefore, the Wilcoxon rank sum test is employed for further verification and evaluation. In this experiment, the significance level is set at 5%. If its value is less than 5%, it indicates that the performance difference between MESCSO and the comparison algorithm on a given benchmark function is statistically significant. Table 5 presents the Wilcoxon rank sum test results of MESCSO versus the other nine algorithms across the 23 standard benchmark functions. In the table, the symbols "+", "-", and "=" indicate that MESCSO algorithm performs better than, worse than, or equal to the corresponding comparison algorithm, respectively, on a given function.

As shown in Table 5 , most of the test results are less than 5%, indicating that there are significant differences between MESCSO and the majority of the comparison algorithms. However, some of the results are greater than 5%, suggesting that the optimization performance of MESCSO on those functions is not significantly different from that of the other algorithms. For functions F9-F11, there are many results equal to 1, indicating that the algorithms generally found the same optimal values on these functions, resulting in no significant differences. On other functions, MESCSO outperforms most of the comparison algorithms. According to the Wilcoxon rank sum test, MESCSO consistently yields better results, except for the BWO algorithm, which shows slightly better performance on a few simple functions. Nevertheless, MESCSO algorithm outperforms BWO algorithm on more than half of the benchmark functions.

Overall, the Wilcoxon test results confirm that MESCSO outperforms most algorithms on the majority of benchmark functions, especially in comparison to the SCSO algorithm, demonstrating stronger convergence and optimization capabilities. This also validates the effectiveness of the improvements proposed in this paper, including ISMROBL, GQI, IMDM, and AOBL mechanisms. Compared with BWO algorithm, MESCSO algorithm performs better on more than half of the benchmark functions, further highlighting its superiority. In summary, the MESCSO algorithm demonstrates superior global search ability and stability across the 23 standard benchmark functions, with improvements that significantly enhance the performance of the SCSO algorithm.

Experiments on the CEC2014 benchmark functions

To more comprehensively evaluate the optimization performance of the MESCSO algorithm, in addition to the 23 standard benchmark functions used in the preliminary experiments, this paper further employs the more challenging CEC2014 benchmark functions for performance validation. Table 6 lists the specific functions used in this experiment. In terms of experimental settings, the population size for each optimization algorithm was

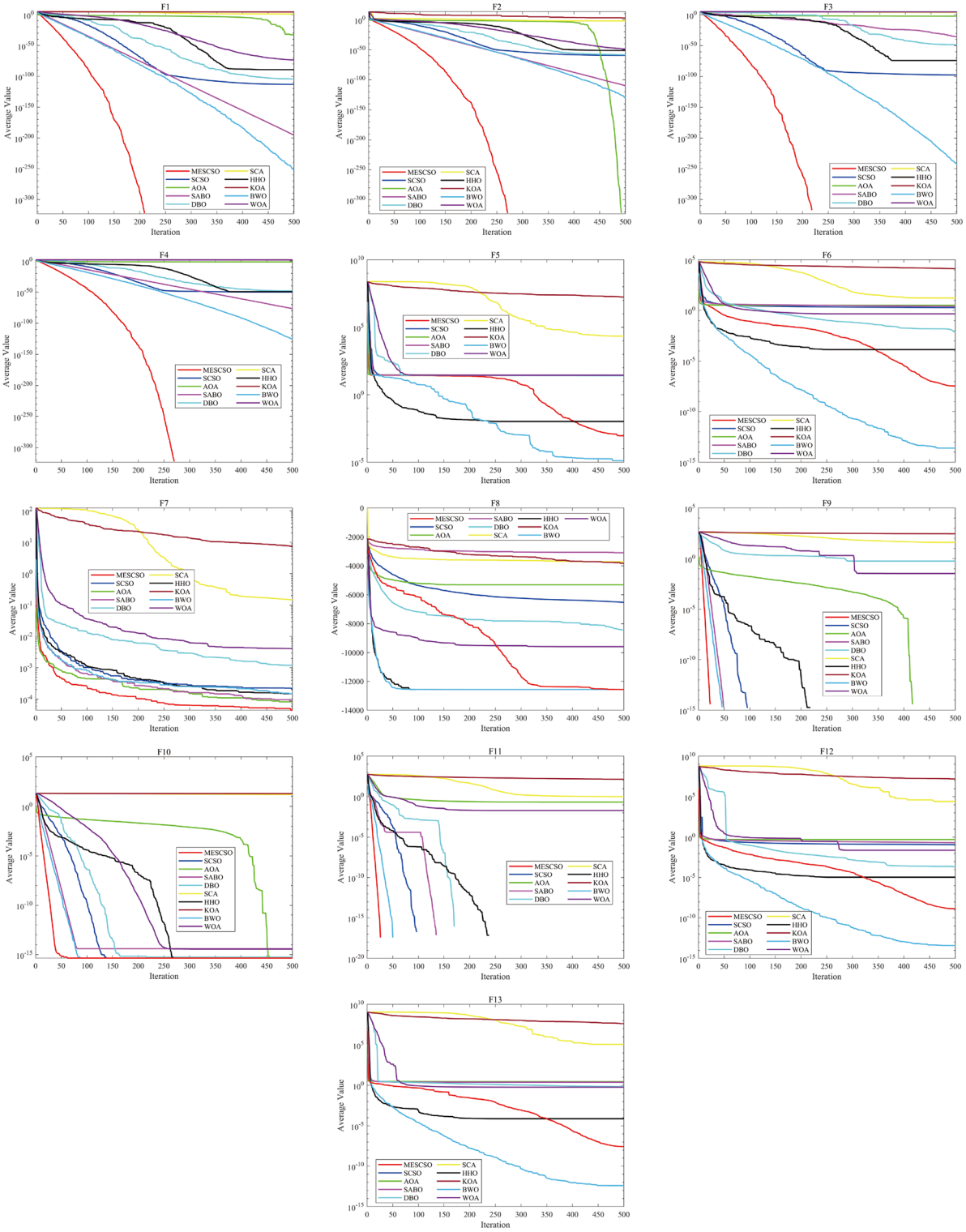

Fig. 2. Convergence curves of standard benchmark functions (F1–F13) of MESCSO algorithm with $dim = 30$·

set to $N = 30$, the maximum number of iterations was $T = 500$, and the dimensionality of the search space was fixed at $dim = 10$. To ensure statistical significance and result stability, each algorithm was independently executed 30 times. Finally, the average fitness and standard deviation were used as evaluation metrics to comprehensively assess the performance of each algorithm on different test functions.

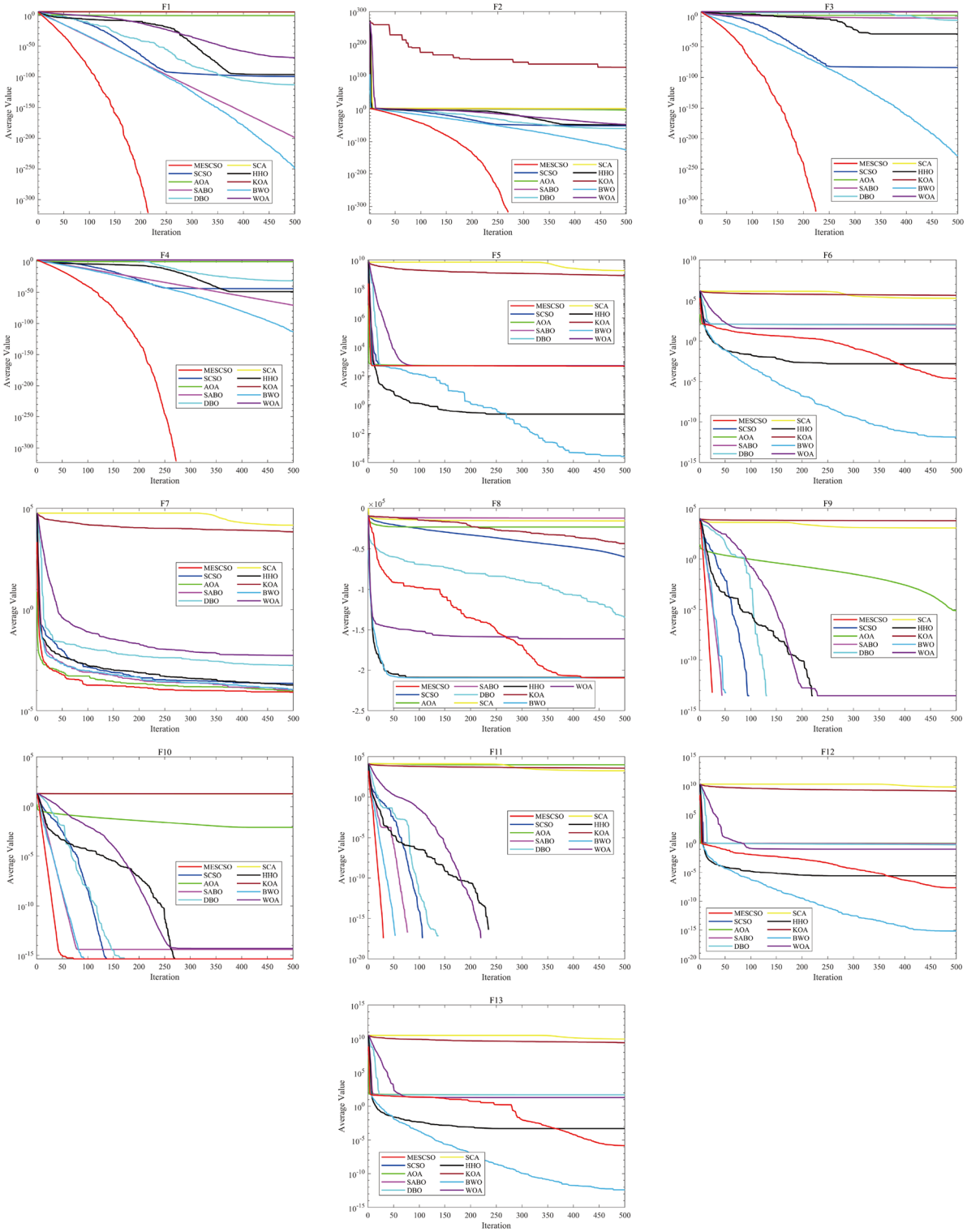

Fig. 3. Convergence curves of standard benchmark functions (F1–F13) of MESCSO algorithm with $dim = 500$·

Experimental results and analysis of CEC2014 benchmark functions

Table 7 summarizes the statistical results of 10 algorithms on the CEC2014 benchmark functions. To clearly present the statistical results, the best average fitness values and standard deviations are bolded. According to the table, the MESCSO algorithm demonstrates superior overall performance across the 30 CEC2014 benchmark functions, significantly outperforming the SCSO and other nine mainstream comparison algorithms. For the basic unimodal functions CEC1-CEC3, MESCSO algorithm not only achieves better average fitness values, but

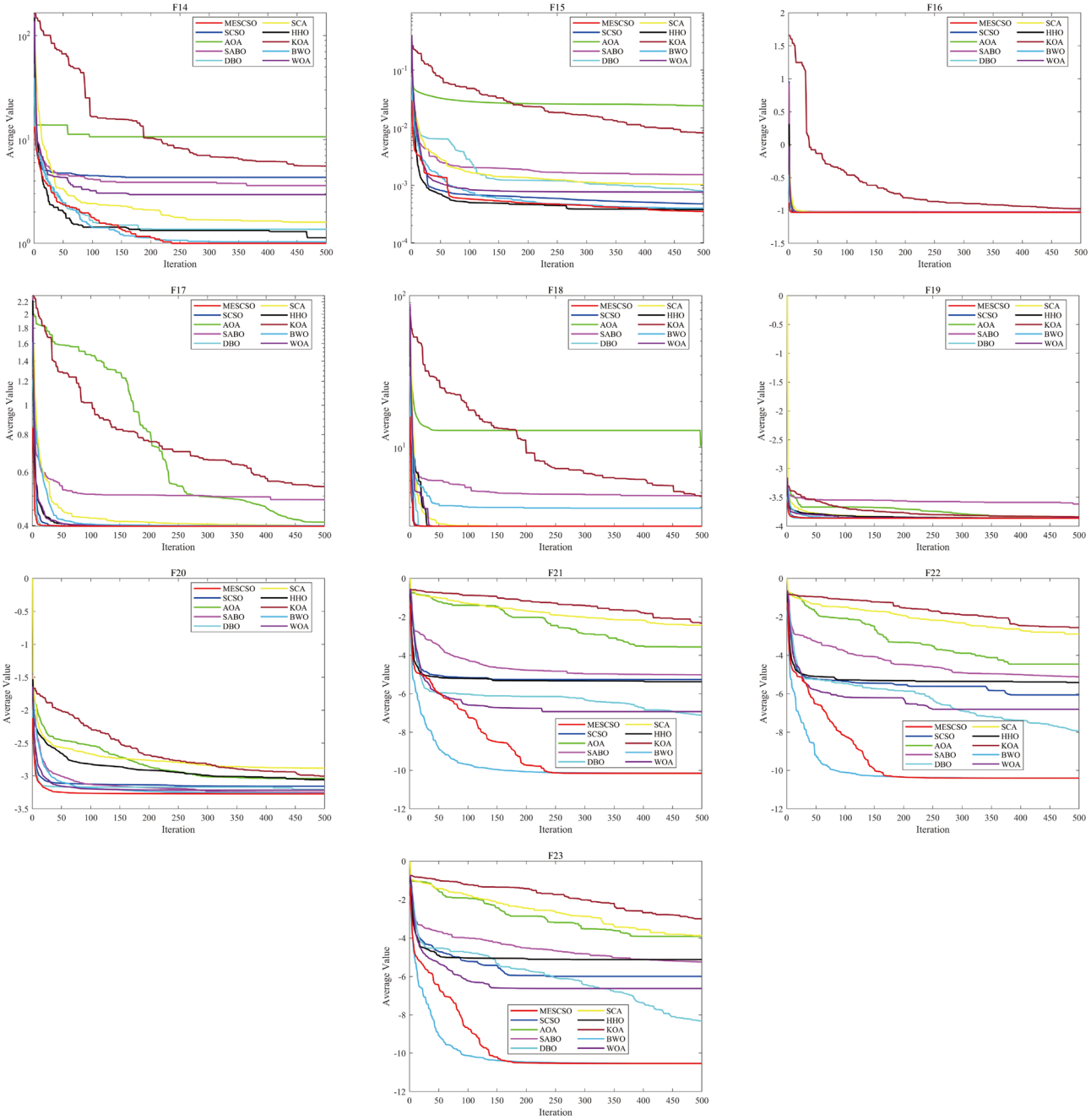

Fig. 4. Convergence curves of standard benchmark functions (F14–F23) of MESCSO algorithm.

also maintains smaller standard deviations in most cases, showing good convergence accuracy and stability. Although the standard deviation on function CEC2 was slightly higher than that of BWO algorithm, MESCSO algorithm still maintained strong optimization performance. For the simple multimodal functions CEC4-CEC8, MESCSO algorithm exhibited strong global search capability, achieving better average fitness on many functions. This indicates its ability to escape local optima across functions with many local extrema, despite relatively weaker stability in some cases. On the hybrid composition functions CEC9-CEC16, MESCSO continued to perform well, maintaining leading average fitness across all comparison algorithms. However, it showed larger standard deviations on some functions. For the more challenging composition functions CEC17-CEC30, MESCSO achieved superior performance on most functions, especially on CEC17, CEC21, CEC23, CEC26, and CEC29, where it not only obtained the best average fitness but also reached the smallest standard deviation, verifying its strong convergence ability and robustness under high-complexity conditions. Although the performance on function CEC18 was slightly lower than that of SABO and HHO algorithm, and it ranked behind SCA algorithm on function CEC22 and CEC24, and fell behind SCSO, HHO, and BWO algorithm on

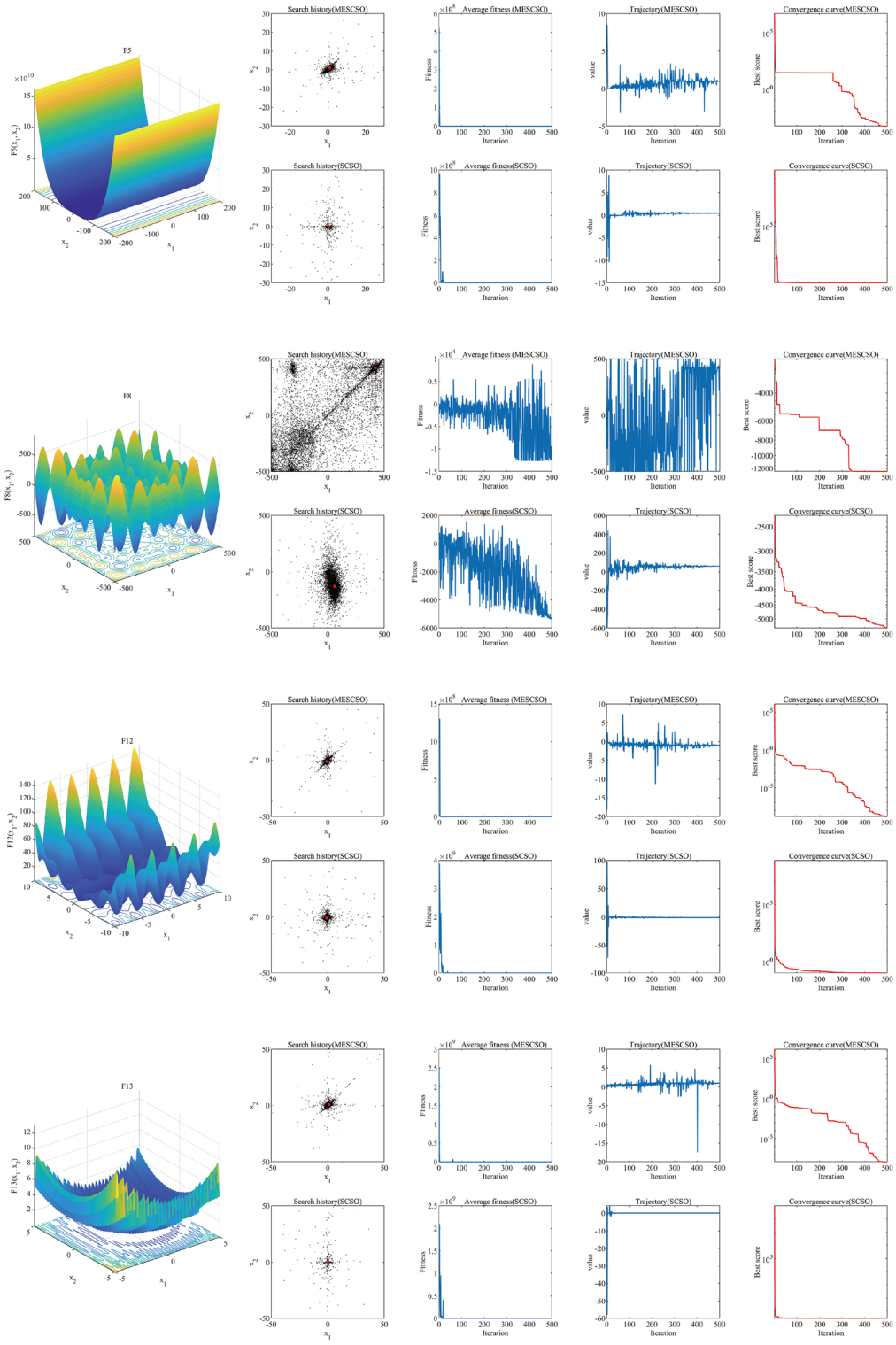

Fig. 5. Convergence behaviour of MESCSO and SCSO.

function CEC28, MESCSO algorithm still showed overall excellent performance. In function CEC30, MESCSO algorithm ranked just behind DBO algorithm, yet continued to exhibit strong competitiveness.

Figure 6 illustrates the convergence curves of MESCSO and nine comparison algorithms on the CEC2014 benchmark functions. As shown in the figure, MESCSO algorithm exhibits better convergence performance. On the unimodal functions CEC1-CEC3, MESCSO algorithm quickly approaches the global optimum and continues optimizing to reach relatively low fitness values. In contrast, the SCSO algorithm tends to fall into

F	*dim*	MESCSO VS. SCSO	MESCSO VS. AOA	MESCSO VS. SABO	MESCSO VS. DBO	MESCSO VS. SCA	MESCSO VS. HHO	MESCSO VS. KOA	MESCSO VS. BWO	MESCSO VS. WOA
F1	30	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
	500	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F2	30	1.21E-12	1	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
	500	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F3	30	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
	500	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F4	30	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
	500	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F5	30	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	5.86E-06	3.02E-11	2.61E-10	3.02E-11
	500	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	2.44E-09	3.02E-11	3.02E-11	3.02E-11
F6	30	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	8.99E-11	3.02E-11	3.02E-11	3.02E-11
	500	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	2.87E-10	3.02E-11	3.02E-11	3.02E-11
F7	30	5.56E-04	5.30E-01	3.85E-03	7.39E-11	3.02E-11	1.32E-04	3.02E-11	3.09E-06	2.87E-10
	500	4.23E-03	1.02E-01	2.34E-01	3.34E-11	3.02E-11	1.22E-02	3.02E-11	2.23E-01	2.87E-10
F8	30	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	4.71E-04	3.02E-11	3.02E-11	3.69E-11
	500	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	9.88E-03	3.02E-11	3.02E-11	2.37E-10
F9	30	1	1	1	1.61E-01	1.21E-12	1	1.21E-12	1	8.15E-02
	500	1	1.95E-09	1	1	1.21E-12	1	1.21E-12	1	3.34E-01
F10	30	1	1	1.69E-14	3.34E-01	1.21E-12	1	1.21E-12	1	6.63E-08
	500	1	1.21E-12	1.69E-14	1	1.21E-12	1	7.61E-13	1	1.39E-09
F11	30	1	1.21E-12	1	1	1.21E-12	1	1.21E-12	1	8.15E-02
	500	1	1.21E-12	1	1	1.21E-12	1	1.21E-12	1	1
F12	30	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.34E-11	3.02E-11	3.02E-11	3.02E-11
	500	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	2.78E-07	3.02E-11	3.02E-11	3.02E-11
F13	30	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
	500	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11	5.46E-09	3.02E-11	3.02E-11	3.02E-11
F14	2	2.62E-11	2.62E-11	2.62E-11	6.31E-05	2.62E-11	2.62E-11	2.62E-11	2.62E-11	2.62E-11
F15	4	9.59E-01	1.61E-10	1.70E-08	1.49E-06	2.15E-10	1.11E-04	3.02E-11	2.28E-05	7.77E-09
F16	2	5.57E-10	3.02E-11	3.02E-11	1.14E-11	3.02E-11	7.24E-02	3.02E-11	3.02E-11	2.88E-06
F17	2	3.47E-10	3.02E-11	3.02E-11	1.21E-12	3.02E-11	1.86E-03	3.02E-11	3.02E-11	1.41E-09
F18	2	1.78E-10	2.13E-05	3.02E-11	2.89E-11	3.02E-11	1.86E-01	3.02E-11	3.02E-11	3.02E-11
F19	3	1.37E-03	3.02E-11	3.02E-11	8.21E-07	3.02E-11	2.03E-09	3.02E-11	3.02E-11	1.96E-10
F20	6	3.09E-06	3.02E-11	1.08E-02	7.66E-02	9.92E-01	1.10E-08	9.92E-11	1.17E-02	8.66E-05
F21	4	3.02E-11	3.02E-11	3.02E-11	2.68E-02	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F22	4	3.02E-11	3.02E-11	3.02E-11	1.85E-01	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
F23	4	3.02E-11	3.02E-11	3.02E-11	2.70E-02	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
+/-/=		29/0/7	32/0/4	31/0/5	28/0/8	35/0/1	28/0/8	36/0/0	19/10/7	32/0/4

Table 5. Wilcoxon rank sum test results on 23 standard benchmark functions.

local optima during the optimization process and lacks an effective mechanism for escaping. The convergence curves of other comparison algorithms are also inferior to that of MESCSO algorithm.

On simple multimodal functions, MESCSO demonstrates strong global search ability, with convergence curves that typically descend more rapidly. This allows the algorithm to enter promising regions early in the search process and continuously approach the global optimum through iterative evolution. This performance can be attributed to the introduction of the ISMROBL, GQI, IMDM, and AOBL mechanisms, which enhance the algorithm's ability to escape from local optima in complex search spaces.

From the convergence curves on functions CEC4-CEC17, MESCSO algorithm is able to find better solutions on many functions and achieves faster convergence. Although many algorithms struggle to converge effectively on the complex, high-dimensional functions CEC17-CEC30 due to premature convergence, MESCSO algorithm still locates better positions on function CEC17, CEC20, CEC21, CEC25, CEC27, and CEC29. This leads to better convergence toward the global optimum and further demonstrates MESCSO's strong search dynamics and its ability to escape local traps and approach global solutions in complex optimization scenarios.

Box plot results and analysis

The box plot is a commonly used statistical visualization method that displays the distribution characteristics of a dataset through five key statistics: the minimum, first quartile, median, third quartile, and maximum. The upper and lower boundaries of the box represent third quartile and first quartile, respectively, with the horizontal

Type	Function	Name	Range	F_{min}
Unimodal functions	CEC 1	Rotated High Conditioned Elliptic Function	[−100, 100]	100
	CEC 2	Rotated Bent Cigar Function	[−100, 100]	200
	CEC 3	Rotated Discus Function	[−100, 100]	300
Simple Multimodal Functions	CEC 4	Shifted and Rotated Rosenbrock's Function	[−100, 100]	400
	CEC 5	Shifted and Rotated Ackley's Function	[−100, 100]	500
	CEC 6	Shifted and Rotated Weierstrass Function	[−100, 100]	600
	CEC 7	Shifted and Rotated Griewank's Function	[−100, 100]	700
	CEC 8	Shifted Rastrigin's Function	[−100, 100]	800
	CEC 9	Shifted and Rotated Rastrigin's Function	[−100, 100]	900
	CEC 10	Shifted Schwefel's Function	[−100, 100]	1000
	CEC 11	Shifted and Rotated Schwefel's Function	[−100, 100]	1100
	CEC 12	Shifted and Rotated Katsuura Function	[−100, 100]	1200
	CEC 13	Shifted and Rotated HappyCat Function	[−100, 100]	1300
	CEC 14	Shifted and Rotated HGBat Function	[−100, 100]	1400
	CEC 15	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	[−100, 100]	1500
	CEC 16	Shifted and Rotated Expanded Scaffer's F6 Function	[−100, 100]	1600
Hybrid Function 1	CEC 17	Hybrid Function 1 ($N = 3$)	[−100, 100]	1700
	CEC 18	Hybrid Function 2 ($N = 3$)	[-100, 100]	1800
	CEC 19	Hybrid Function 3 ($N = 4$)	[-100, 100]	1900
	CEC 20	Hybrid Function 4 ($N = 4$)	[-100, 100]	2000
	CEC 21	Hybrid Function 5 ($N = 5$)	[-100, 100]	2100
	CEC 22	Hybrid Function 6 ($N = 5$)	[-100, 100]	2200
Composition Functions	CEC 23	Composition Function 1 ($N = 5$)	[-100, 100]	2300
	CEC 24	Composition Function 2 ($N = 3$)	[-100, 100]	2400
	CEC 25	Composition Function 3 ($N = 3$)	[-100, 100]	2500
	CEC 26	Composition Function 4 ($N = 5$)	[-100, 100]	2600
	CEC 27	Composition Function 5 ($N = 5$)	[-100, 100]	2700
	CEC 28	Composition Function 6 ($N = 5$)	[-100, 100]	2800
	CEC 29	Composition Function 7 ($N = 3$)	[-100, 100]	2900
	CEC 30	Composition Function 8 ($N = 3$)	[-100, 100]	3000

Table 6. Details of the CEC2014 benchmark functions.

line inside the box indicating the median. The whiskers extend to the minimum and maximum values within the data range. In addition, outliers are marked in the plot, providing an intuitive reflection of data dispersion, symmetry, and stability.

Figure 7 illustrates the box plots of the fitness values obtained by MESCSO and nine comparison algorithms after 30 independent runs on the CEC2014 benchmark functions. Through visual comparison, it is evident that MESCSO algorithm demonstrates smaller interquartile ranges and lower median lines on most test functions, indicating more concentrated results, lower variability, and significantly better stability. For instance, on functions such as CEC1, CEC3, CEC4, CEC5, CEC7, CEC17, CEC21, and CEC29, MESCSO not only achieves the lowest median values but also exhibits significantly narrower extreme value ranges compared to other algorithms. This demonstrates the MESCSO algorithm's ability to consistently deliver high-quality solutions with stable performance across unimodal, multimodal, and complex hybrid functions. In contrast, the SCSO and other algorithms exhibit wider boxes and longer whiskers on multiple test functions, suggesting greater variability and instability across runs, possibly due to premature convergence or stagnation. In particular, on functions such as CEC2, CEC10, CEC11, CEC12, CEC28, and CEC30, many comparison algorithms exhibit serious outliers and long tails. Nevertheless, MESCSO still maintains a relatively concentrated trend.

Overall, the analysis of the box plots further verifies the comprehensive advantages of the MESCSO algorithm in terms of fitness value distribution, convergence stability, and optimal solution acquisition capability, demonstrating its strong reliability and practical applicability across multiple independent runs.

Analysis of the Wilcoxon rank sum test results

Based on the above analysis, the MESCSO algorithm achieved promising results on the CEC2014 benchmark functions. Table 8 presents the Wilcoxon rank sum test results comparing MESCSO with nine other optimization algorithms across the CEC2014 benchmark functions. The significance level was set to 5%. The bottom of the table provides a summary of the "+/ /=" counts, indicating the number of functions where MESCSO performed significantly better, worse, or equivalent than each comparison algorithm.

From the overall statistical results, MESCSO algorithm exhibited statistically significant performance differences compared to most comparison algorithms, with the majority of its value less than 5%. Specifically,

CEC	Metric	MESCSO	SCSO	AOA	SABO	DBO	SCA	HHO	KOA	BWO	WOA
CEC 1	mean	**2.14E+06**	1.11E+07	5.42E+07	1.47E+07	7.26E+06	1.31E+07	1.66E+07	4.93E+07	7.23E+07	1.33E+07
	std	**1.29E+06**	6.41E+06	7.28E+06	8.40E+06	1.57E+07	5.92E+06	9.80E+06	2.64E+07	3.21E+07	8.77E+06
CEC 2	mean	**2.25E+03**	2.74E+08	6.33E+09	1.41E+08	8.56E+03	9.41E+08	6.54E+05	2.07E+09	4.11E+09	4.49E+07
	std	**3.51E+03**	4.23E+08	2.51E+09	2.97E+08	1.29E+04	4.41E+08	3.44E+05	8.92E+08	7.43E+08	5.30E+07
CEC 3	mean	**4.21E+03**	6.94E+03	2.31E+04	7.34E+03	1.17E+04	1.18E+04	6.65E+03	5.34E+04	1.27E+04	5.77E+04
	std	2.26E+03	3.68E+03	6.78E+03	3.48E+03	1.16E+04	7.65E+03	2.47E+03	3.95E+04	**2.01E+03**	2.16E+04
CEC 4	mean	**4.23E+02**	4.34E+02	1.63E+03	4.67E+02	4.50E+02	4.78E+02	4.53E+02	6.87E+02	1.27E+03	4.58E+02
	std	**1.53E+01**	2.45E+01	7.61E+02	4.88E+01	5.48E+01	2.53E+01	3.68E+01	1.52E+02	3.09E+02	4.04E+01
CEC 5	mean	**5.20E+02**	5.20E+02	5.20E+02	5.20E+02	5.20E+02	5.20E+02	5.20E+02	5.21E+02	5.20E+02	5.20E+02
	std	6.79E-02	8.92E-02	**6.33E-02**	8.03E-02	1.09E-01	1.18E-01	1.16E-01	1.40E-01	9.93E-02	1.34E-01
CEC 6	mean	**6.05E+02**	6.06E+02	6.10E+02	6.07E+02	6.06E+02	6.08E+02	6.08E+02	6.11E+02	6.10E+02	6.09E+02
	std	1.82E+00	1.56E+00	9.99E-01	1.92E+00	1.68E+00	1.02E+00	1.31E+00	1.04E+00	**6.06E-01**	1.56E+00
CEC 7	mean	**7.00E+02**	7.04E+02	8.27E+02	7.04E+02	7.01E+02	7.14E+02	7.01E+02	7.37E+02	7.35E+02	7.02E+02
	std	**1.35E-01**	6.23E+00	4.63E+01	3.37E+00	4.84E-01	4.81E+00	2.87E-01	1.49E+01	7.97E+00	8.43E-01
CEC 8	mean	**8.28E+02**	8.36E+02	8.53E+02	8.58E+02	8.31E+02	8.46E+02	8.30E+02	8.73E+02	8.71E+02	8.44E+02
	std	**6.97E+00**	1.28E+01	1.40E+01	9.26E+00	1.10E+01	7.39E+00	9.16E+00	1.29E+01	7.18E+00	1.62E+01
CEC 9	mean	**9.35E+02**	9.36E+02	9.47E+02	9.54E+02	9.39E+02	9.53E+02	9.36E+02	9.79E+02	9.66E+02	9.55E+02
	std	8.80E+00	1.06E+01	**5.77E+00**	1.18E+01	1.29E+01	8.16E+00	9.79E+00	1.01E+01	6.65E+00	1.70E+01
CEC 10	mean	**1.56E+03**	1.69E+03	1.67E+03	2.55E+03	1.57E+03	2.10E+03	1.62E+03	2.71E+03	2.17E+03	1.70E+03
	std	2.63E+02	3.37E+02	2.45E+02	2.05E+02	3.07E+02	2.48E+02	2.41E+02	2.87E+02	**1.29E+02**	3.51E+02
CEC 11	mean	**2.00E+03**	2.08E+03	2.08E+03	2.65E+03	2.14E+03	2.54E+03	2.07E+03	3.17E+03	2.61E+03	2.33E+03
	std	2.52E+02	2.86E+02	2.79E+02	2.08E+02	3.09E+02	2.41E+02	2.67E+02	2.03E+02	**1.57E+02**	3.46E+02
CEC 12	mean	**1.20E+03**	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03	1.20E+03
	std	2.65E-01	3.25E-01	**2.62E-01**	3.00E-01	4.56E-01	3.05E-01	3.04E-01	6.93E-01	2.67E-01	3.77E-01
CEC 13	mean	**1.30E+03**	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03	1.30E+03
	std	1.73E-01	4.44E-01	6.09E-01	2.06E-01	1.35E-01	1.69E-01	1.84E-01	4.00E-01	3.29E-01	**1.15E-01**
CEC 14	mean	**1.40E+03**	1.40E+03	1.43E+03	1.40E+03	1.40E+03	1.40E+03	1.40E+03	1.41E+03	1.42E+03	1.44E+03
	std	**2.41E-01**	2.56E-01	9.13E+00	1.03E+00	2.83E-01	5.57E-01	3.31E-01	4.70E+00	4.01E+00	3.26E-01
CEC 15	mean	**1.50E+03**	1.51E+03	5.27E+03	1.51E+03	1.50E+03	1.51E+03	1.51E+03	2.43E+03	1.75E+03	1.51E+03
	std	**1.41E+00**	6.62E+00	7.02E+03	4.73E+00	2.51E+00	2.81E+00	2.50E+00	1.29E+03	2.26E+02	4.87E+00
CEC 16	mean	**1.60E+03**	1.60E+03	1.60E+03	1.60E+03	1.60E+03	1.60E+03	1.60E+03	1.60E+03	1.60E+03	1.60E+03
	std	3.67E-01	3.10E-01	2.78E-01	1.82E-01	3.89E-01	2.46E-01	2.83E-01	**1.30E-01**	1.75E-01	3.61E-01
CEC 17	mean	**6.35E+03**	8.14E+04	3.45E+05	7.24E+03	2.29E+04	5.54E+04	6.53E+04	1.36E+06	2.91E+05	3.25E+05
	std	**3.20E+03**	1.70E+05	1.77E+05	7.4E0+03	2.64E+04	7.18E+04	4.88E+04	1.18E+06	1.24E+05	4.39E+05
CEC 18	mean	1.10E+04	1.13E+04	1.39E+04	**9.21E+03**	1.12E+04	4.07E+04	1.02E+04	2.71E+06	2.43E+05	1.47E+04
	std	5.13E+03	7.64E+03	1.07E+04	**3.08E+03**	1.16E+04	3.13E+04	6.75E+03	2.75E+06	2.73E+05	1.38E+04
CEC 19	mean	**1.90E+03**	1.90E+03	1.94E+03	1.91E+03	1.90E+03	1.91E+03	1.90E+03	1.91E+03	1.91E+03	1.91E+03
	std	1.00E+00	**9.73E-01**	2.52E+01	2.39E+00	1.48E+00	1.15E+00	1.50E+00	3.67E+00	4.56E+00	2.08E+00
CEC 20	mean	**4.67E+03**	8.83E+03	1.01E+04	6.22E+03	8.43E+03	8.13E+03	7.83E+03	2.04E+05	1.70E+04	1.33E+04
	std	2.80E+03	3.82E+03	7.00E+03	**2.78E+03**	8.83E+03	4.54E+03	4.26E+03	3.71E+05	1.32E+04	7.44E+03
CEC 21	mean	**7.19E+03**	9.38E+03	7.41E+05	3.85E+04	8.65E+03	1.78E+04	8.36E+04	1.84E+05	7.38E+04	7.91E+05
	std	**4.99E+03**	5.72E+03	1.43E+06	8.41E+04	1.02E+04	1.13E+04	1.38E+05	1.55E+05	6.71E+04	1.50E+06
CEC 22	mean	2.29E+03	2.32E+03	2.36E+03	2.40E+03	2.32E+03	**2.28E+03**	2.35E+03	2.46E+03	2.41E+03	2.33E+03
	std	6.35E+01	5.89E+01	9.34E+01	5.77E+01	8.33E+01	**2.40E+01**	7.44E+01	1.04E+02	7.95E+01	1.02E+02
CEC 23	mean	**2.50E+03**	**2.50E+03**	**2.50E+03**	2.51E+03	2.64E+03	2.65E+03	**2.50E+03**	2.67E+03	**2.50E+03**	2.63E+03
	std	**0**	**0**	**0**	2.59E+01	1.52E+01	8.34E+00	**0**	1.80E+01	**0**	3.61E+01
CEC 24	mean	2.56E+03	2.58E+03	2.59E+03	2.60E+03	2.56E+03	**2.56E+03**	2.59E+03	2.60E+03	2.60E+03	2.59E+03
	std	3.10E+01	2.79E+01	1.92E+01	4.12E+01	2.68E+01	7.61E+00	1.66E+01	1.50E+01	**3.43E+00**	2.75E+01
CEC 25	mean	**2.69E+03**	2.70E+03	2.70E+03	2.70E+03	2.69E+03	2.70E+03	2.70E+03	2.71E+03	2.70E+03	2.70E+03
	std	1.36E+01	8.56E+00	4.00E+00	2.03E+00	1.51E+01	9.82E+00	3.18E+00	4.87E+00	7.75E-01	7.47E+00
CEC 26	mean	**2.70E+03**	2.70E+03	2.71E+03	2.70E+03	2.70E+03	2.70E+03	2.70E+03	2.70E+03	2.70E+03	2.70E+03
	std	**1.32E-01**	1.34E-01	2.51E+01	2.41E+00	2.05E-01	1.85E-01	2.18E-01	4.47E-01	6.73E-01	2.51E-01
CEC 27	mean	**2.79E+03**	2.89E+03	2.89E+03	3.16E+03	2.97E+03	3.04E+03	2.88E+03	3.16E+03	2.89E+03	3.17E+03
	std	1.42E+02	3.58E+01	**2.50E+01**	8.79E+01	1.86E+02	1.45E+02	5.83E+01	1.35E+02	2.88E+01	7.71E+01
CEC 28	mean	3.04E+03	**3.00E+03**	3.04E+03	3.54E+03	3.33E+03	3.29E+03	**3.00E+03**	3.42E+03	**3.00E+03**	3.41E+03
	std	7.38E+01	**0**	1.51E+02	2.93E+03	1.08E+02	6.04E+01	**0**	1.77E+02	**0**	1.82E+02

Continued

CEC	Metric	MESCSO	SCSO	AOA	SABO	DBO	SCA	HHO	KOA	BWO	WOA
CEC 29	mean	**6.24E+03**	1.20E+05	3.70E+05	1.39E+06	4.54E+05	1.86E+04	3.57E+05	8.89E+05	2.67E+04	1.19E+05
	std	**5.04E+03**	4.37E+05	1.13E+06	1.22E+06	1.04E+06	1.22E+04	9.56E+05	9.24E+05	2.79E+04	4.37E+05
CEC 30	mean	4.66E+03	4.97E+03	5.12E+04	6.02E+03	**4.54E+03**	5.19E+03	6.08E+03	1.92E+04	1.52E+04	5.93E+03
	std	8.55E+02	9.97E+02	7.18E+04	1.05E+03	1.59E+03	**6.40E+03**	1.34E+03	1.50E+04	6.67E+03	1.84E+03

Table 7. Statistical results of the CEC2014 benchmark functions.

in comparison with SABO, SCA, KOA, and BWO algorithm, MESCSO algorithm demonstrated notably better optimization performance on 27 functions, 28 functions, and 30 functions, highlighting its superior global optimization capability. Additionally, MESCSO algorithm outperformed AOA, HHO, and WOA algorithm on 19 functions, 24 functions, and 25 functions, respectively, indicating its adaptability across diverse types of optimization problems, including unimodal, multimodal, and hybrid composite functions, along with strong solution quality. Compared to the SCSO algorithm, MESCSO algorithm showed clear advantages in 14 functions, slight inferiority in only one, and comparable performance in the remaining 15, confirming that the proposed improvements significantly enhance both search accuracy and exploration capability.

In summary, the Wilcoxon rank sum test results clearly demonstrated that MESCSO outperforms several mainstream optimization algorithms on many benchmark functions with statistically significant advantages. These results further validate the effectiveness of the integrated strategies including ISMROBL, GQI, IMDM, and AOBL mechanism in enhancing convergence speed, improving global exploration ability, and suppressing premature convergence.

Structural engineering optimization problems

To further investigate the practical applicability of the MESCSO algorithm, this section evaluates its performance on five representative structural engineering optimization problems. These problems include pressure vessel design, welded beam design, multi-disc clutch braking design, tension/compression spring design, and cantilever beam design. For these five different constrained optimization problems, the population size N was set at 30 and the maximum number of iterations T was 500. For each problem, MESCSO is run 30 times independently. In this section, the best and statistical results of different algorithms are used for comparison. A simple and commonly used penalty method is applied to solve the constrained optimization problems. The penalized objective function f_p is expressed as follows:

$$f_p(y) = f(y) + \lambda \sum_{k=1}^{m} \delta_k \left[g_k(y)\right]^2 \tag{25}$$

where λ is a penalty factor which is set as 10^{20} in this paper. For each constraint g_k, if it is violated, then $\delta_k = 1$; otherwise, $\delta_k = 0$.

Pressure vessel design problem

The primary objective of the pressure vessel design problem is to minimize the total manufacturing cost of the vessel while satisfying multiple engineering constraints. A schematic diagram of the pressure vessel is shown in Fig. 8. This problem involves four decision variables: the shell thickness T_s, head thickness T_h, inner radius R, and vessel length L.

The mathematical formulation of the pressure vessel design problem is as follows.

Consider:

$$\vec{y} = [y_1\ y_2\ y_3\ y_4] = [T_s\ T_h\ R\ L] \tag{26}$$

Objective function:

$$f(\vec{y}) = 0.6224\, y_1 y_3 y_4 + 1.7781\, y_2 y_3^2 + 3.1661\, y_1^2 y_4 + 19.84\, y_1^2 y_3 \tag{27}$$

Subject to:

$$g_1(\vec{y}) = -\, y_1 + 0.0193 y_3 \leq 0 \tag{28}$$

$$g_2(\vec{y}) = -\, y_2 + 0.00954 y_3 \leq 0 \tag{29}$$

$$g_3(\vec{y}) = -\, \pi y_3^2 y_4 + \frac{4}{3}\pi y_3^3 + 1{,}296{,}000 \leq 0 \tag{30}$$

$$g_4(\vec{y}) = y_4 - 240 \leq 0 \tag{31}$$

Variable Range:

$$0 \leq y_1, y_2 \leq 99, \quad 10 \leq y_3, y_4 \leq 200 \tag{32}$$

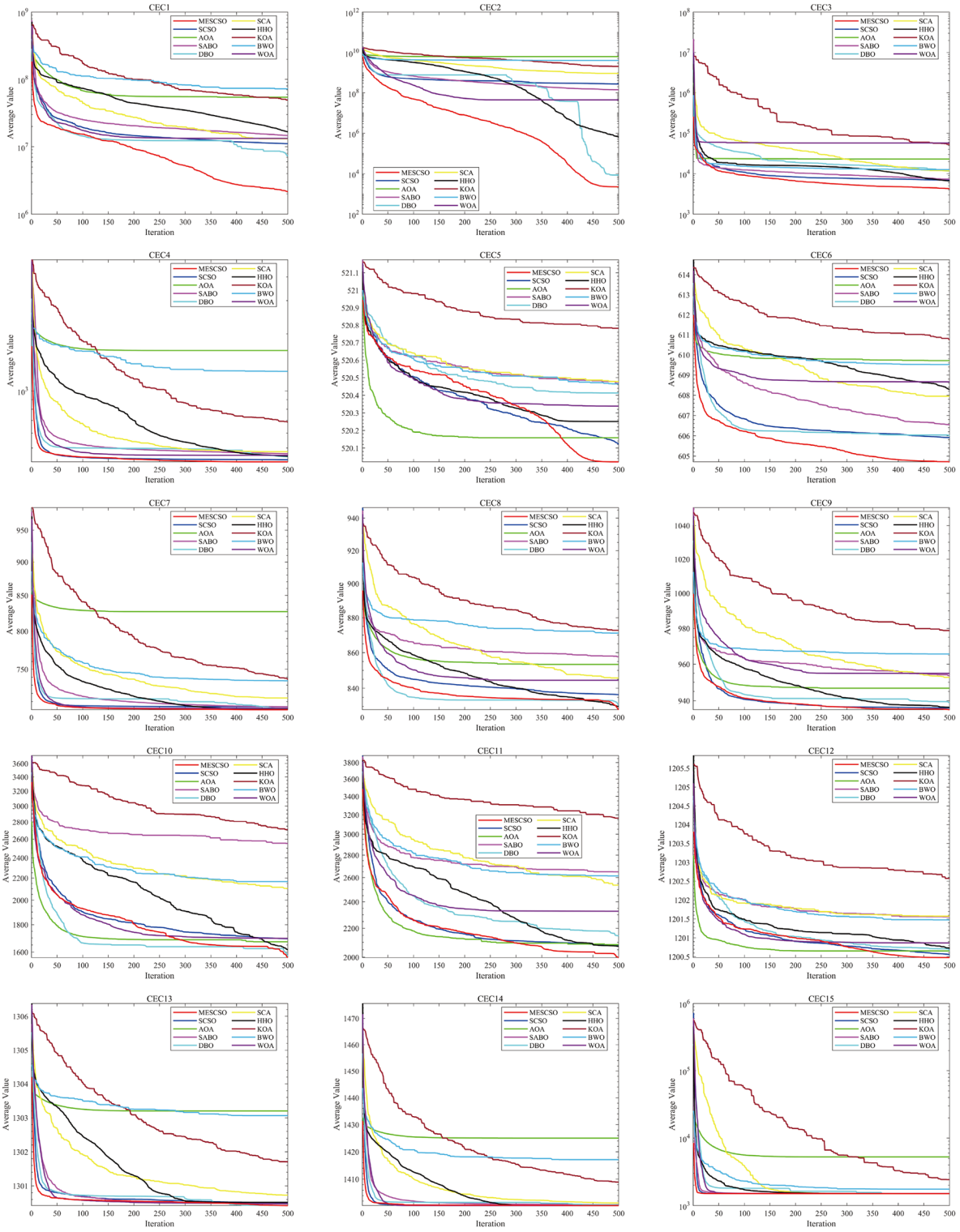

Fig. 6. Convergence curves of CEC2014 benchmark functions (F1–F13) of MESCSO algorithm with $dim = 10$·

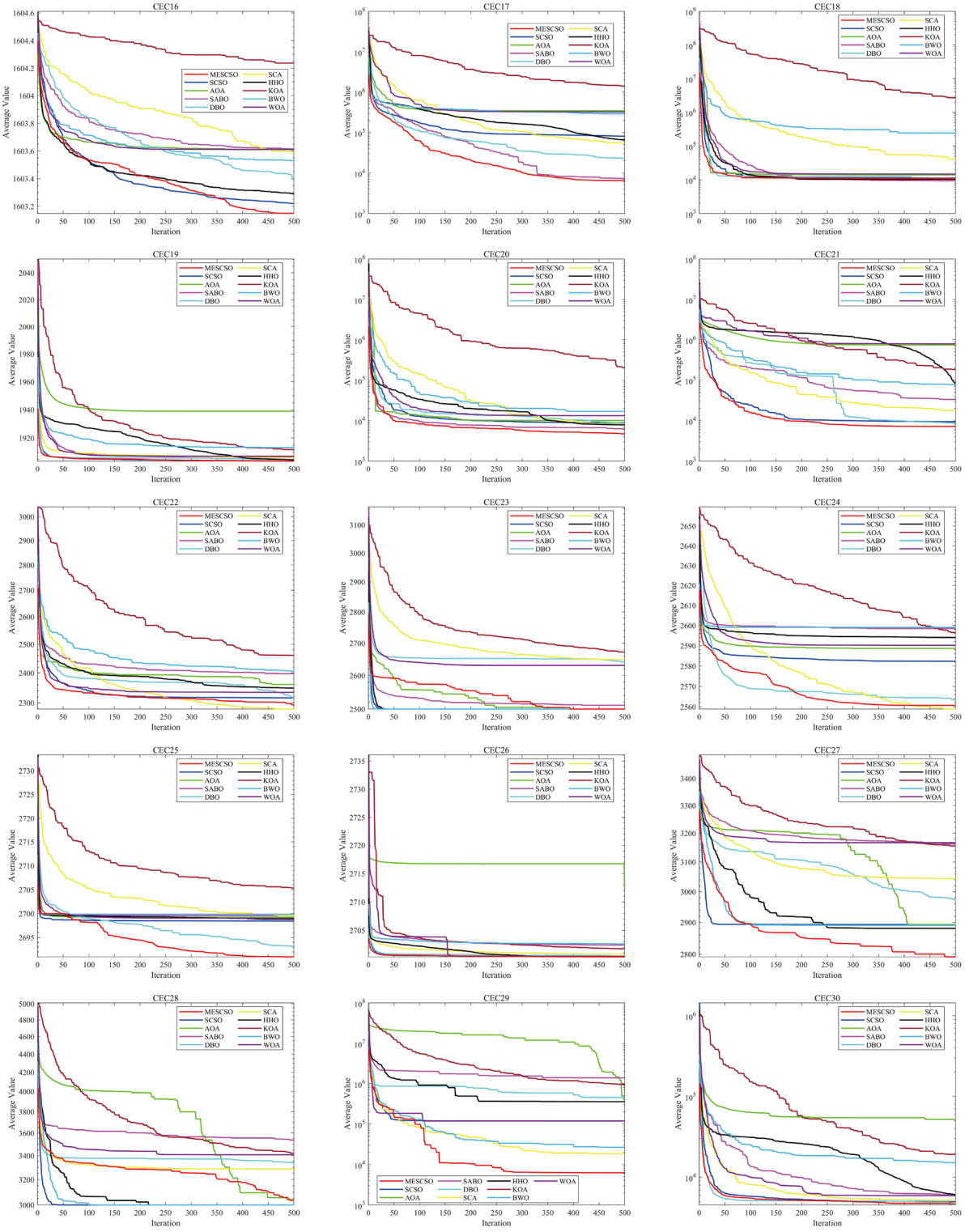

Fig. 6. (continued)

The experimental results for the pressure vessel design problem are presented in Table 9, demonstrating that the MESCSO algorithm performs effectively in solving engineering optimization problems. As shown in the table, the MESCSO algorithm obtained the following optimal solution: $T_s = 0.778267, T_h = 0.384764, R = 40.323219$, and $L = 199.950488$, with a minimum cost of 5885.90733. Among the other comparison algorithms, six produced cost values greater than 6000, while two yielded costs below 6000. Therefore, the MESCSO algorithm achieved the lowest cost among all methods, indicating its superior optimization performance for this problem.

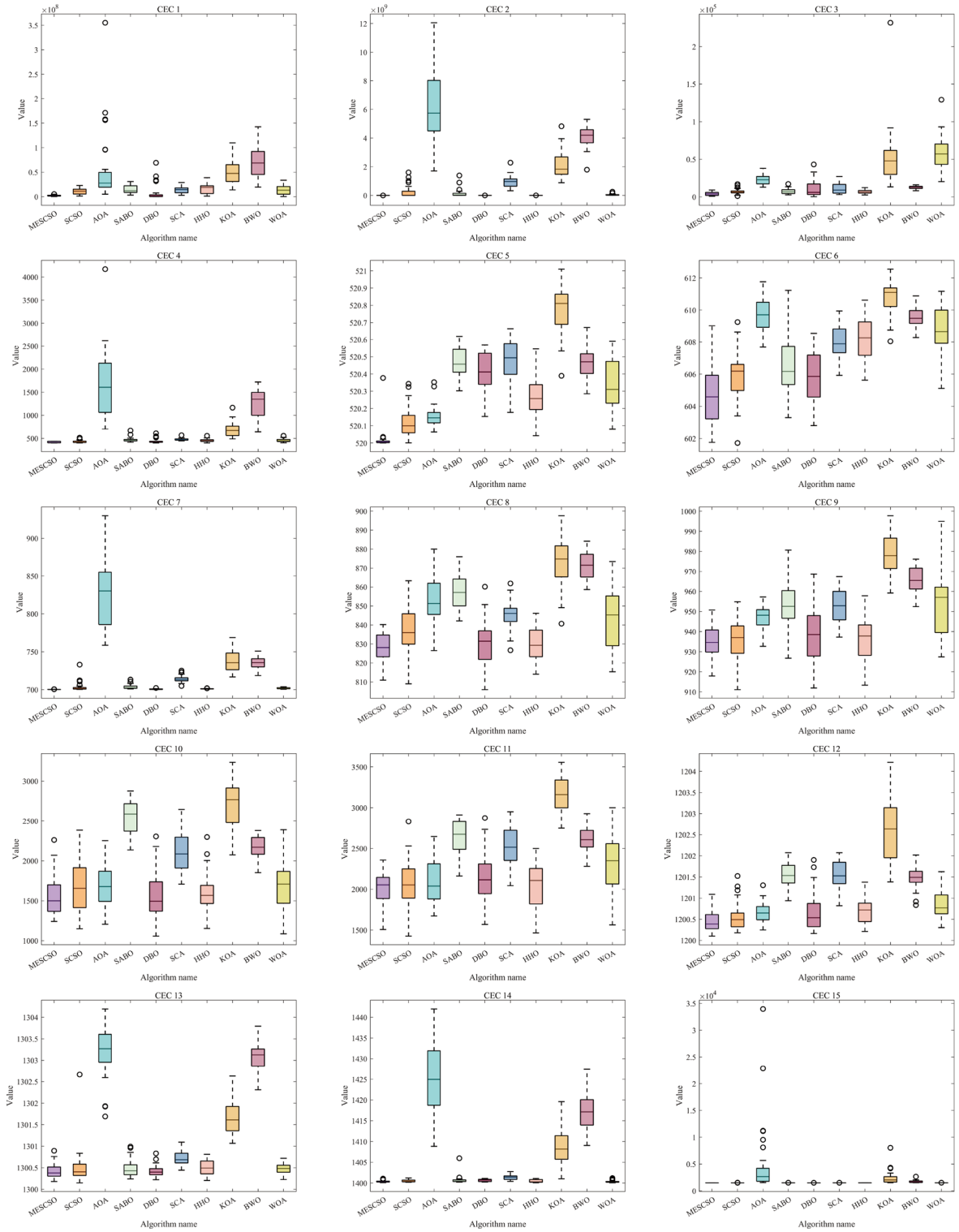

Fig. 7. Box plots of CEC2014 benchmark functions of MESCSO algorithm with $dim = 10$.

Welded beam design problem

The welded beam design problem is a commonly encountered and challenging issue in structural engineering. The primary objective of this problem is to minimize the fabrication cost of the welded beam while satisfying a series of structural and mechanical constraints. A schematic diagram of the welded beam is illustrated in Fig. 9. This problem involves four decision variables: weld width h, connecting beam length l, beam height t, and connecting beam thickness b.

The mathematical formulation of the welded beam design problem is as follows.

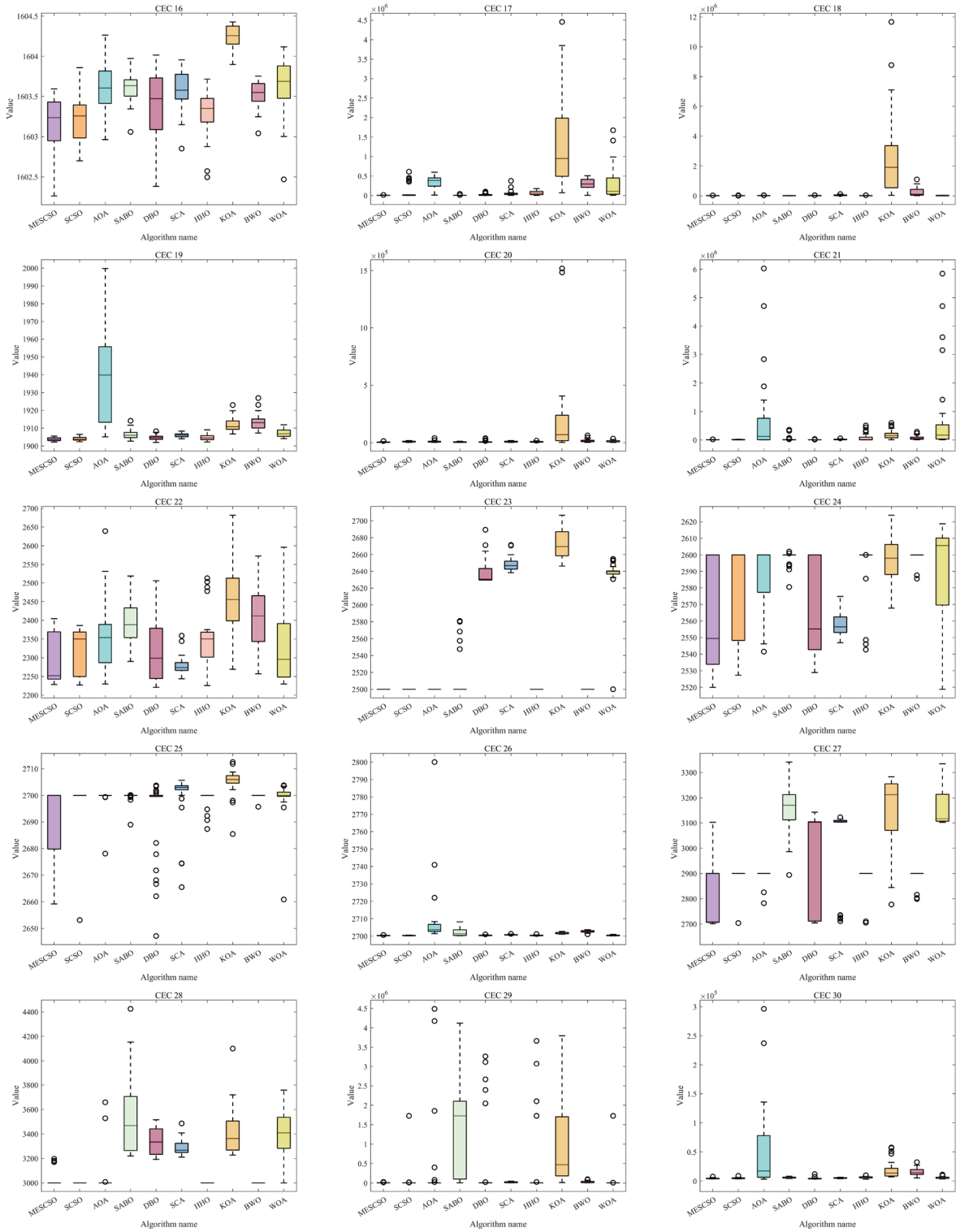

Fig. 7. (continued)

Consider:

$$\vec{y} = [y_1\ y_2\ y_3\ y_4] = [h\ l\ t\ b] \tag{33}$$

Objective function:

$$f(\vec{y}) = 1.10471 y_1^2 y_2 + 0.04811 y_3 y_4 (14.0 + y_2) \tag{34}$$

CEC	MESCSO VS. SCSO	MESCSO VS. AOA	MESCSO VS. SABO	MESCSO VS. DBO	MESCSO VS. SCA	MESCSO VS. HHO	MESCSO VS. KOA	MESCSO VS. BWO	MESCSO VS. WOA
CEC 1	3.08E-08	3.34E-11	9.92E-11	3.63E-01	8.99E-11	9.26E-09	3.02E-11	3.02E-11	1.70E-08
CEC 2	2.37E-10	3.02E-11	3.02E-11	4.71E-04	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.02E-11
CEC 3	1.17E-03	3.02E-11	3.18E-04	3.27E-02	1.64E-05	5.26E-11	3.02E-11	4.08E-11	3.02E-11
CEC 4	1.52E-03	3.02E-11	8.48E-09	9.12E-01	3.02E-11	5.61E-05	3.02E-11	3.02E-11	1.75E-05
CEC 5	6.01E-08	5.57E-10	3.69E-11	8.15E-11	4.50E-11	3.82E-10	3.02E-11	5.49E-11	1.96E-10
CEC 6	8.31E-03	6.70E-11	1.00E-03	9.47E-03	5.46E-09	3.50E-09	3.69E-11	5.49E-11	3.20E-09
CEC 7	8.99E-11	3.02E-11	3.02E-11	3.59E-05	3.02E-11	3.02E-11	3.02E-11	3.02E-11	3.34E-11
CEC 8	3.85E-03	2.03E-09	3.02E-11	3.71E-01	8.89E-10	4.29E-01	3.02E-11	3.02E-11	6.77E-05
CEC 9	6.31E-01	1.19E-06	7.69E-08	2.17E-01	1.20E-08	5.30E-01	3.02E-11	3.02E-11	4.74E-06
CEC10	1.41E-01	7.24E-02	4.08E-11	9.94E-01	7.77E-09	2.97E-01	3.34E-11	8.10E-10	6.57E-02
CEC11	4.20E-01	4.04E-01	1.09E-10	1.05E-01	3.20E-09	1.76E-01	3.02E-11	8.15E-11	3.56E-04
CEC12	4.64E-01	1.91E-02	4.08E-11	1.19E-01	4.98E-11	2.16E-03	3.02E-11	6.07E-11	5.97E-05
CEC13	7.17E-01	3.02E-11	2.71E-01	6.00E-01	1.47E-07	4.51E-02	3.02E-11	3.02E-11	4.36E-02
CEC14	2.46E-01	3.02E-11	6.38E-03	7.70E-04	9.76E-10	5.30E-01	3.34E-11	3.02E-11	2.17E-01
CEC15	9.47E-03	3.02E-11	1.09E-10	3.11E-01	5.49E-11	1.61E-10	3.02E-11	3.02E-11	1.69E-09
CEC16	7.96E-01	1.86E-06	6.01E-08	2.42E-02	1.39E-06	1.37E-01	3.02E-11	2.32E-06	4.12E-06
CEC17	3.85E-03	3.69E-11	6.95E-01	2.62E-03	1.96E-10	2.67E-09	3.02E-11	1.61E-10	1.09E-10
CEC18	6.41E-01	6.00E-01	2.06E-01	8.24E-02	7.69E-08	2.77E-01	3.02E-11	8.15E-11	6.73E-01
CEC19	6.63E-01	4.98E-11	3.26E-07	6.97E-03	1.01E-08	1.03E-02	3.02E-11	3.02E-11	2.03E-09
CEC20	2.13E-05	2.32E-06	2.07E-02	2.07E-02	7.70E-04	1.30E-03	1.86E-09	1.60E-07	1.61E-06
CEC21	1.67E-01	2.20E-07	1.76E-02	1.86E-01	1.39E-06	8.12E-04	1.46E-10	3.47E-10	3.82E-10
CEC22	1.71E-01	3.67E-03	1.61E-06	4.20E-01	3.63E-01	1.08E-02	2.60E-08	7.04E-07	8.24E-02
CEC23	1	1	1.42E-04	1.21E-12	1.21E-12	1	1.21E-12	1	4.57E-12
CEC24	7.22E-03	8.35E-03	3.79E-09	1.62E-01	2.34E-01	3.72E-05	6.33E-05	3.79E-04	5.58E-05
CEC25	2.31E-03	8.47E-03	4.25E-04	8.92E-02	5.02E-07	1.72E-02	4.86E-09	1.04E-03	7.74E-03
CEC26	8.19E-01	3.02E-11	2.03E-07	1.77E-03	8.99E-11	7.48E-02	3.02E-11	3.02E-11	1.37E-01
CEC 27	1.49E-04	5.35E-05	2.34E-10	2.58E-05	2.20E-09	2.69E-04	1.68E-09	7.35E-05	3.30E-11
CEC 28	1.10E-02	3.23E-01	6.47E-12	7.21E-12	6.48E-12	1.10E-02	6.46E-12	1.10E-02	1.87E-11
CEC 29	2.71E-01	3.36E-03	8.84E-10	6.31E-01	2.20E-07	5.01E-01	4.50E-11	7.69E-08	7.62E-01
CEC 30	1.02E-01	5.46E-09	9.53E-07	5.37E-02	1.52E-03	1.75E-05	4.08E-11	1.21E-10	1.37E-03
+/-/=	14/1/15	25/0/5	27/0/3	14/0/16	28/0/2	19/1/10	30/0/0	28/1/1	24/0/6

Table 8. Wilcoxon rank sum test results on CEC2014 benchmark functions.

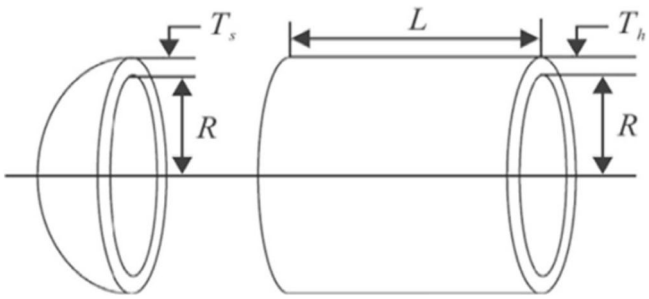

Fig. 8. Schematic diagram of the pressure vessel design.

Algorithm	y_1	y_2	y_3	y_4	Best cost
MESCSO	0.7782	0.3848	40.3232	199.9505	5885.9073
SCSO[36]	0.7798	0.9390	40.3864	199.2918	5917.46
CPSO[55]	0.8125	0.4375	42.0913	176.7465	6061.0777
HPSO[56]	0.8125	0.4345	42.0984	176.6366	6059.7143
GWO[57]	0.8125	0.4375	42.08918	176.7587	6059.5639
MVO[58]	0.8125	0.4375	42.09074	176.7387	6060.8066
HOA[59]	0.9166	0.4491	50.4672	96.5029	6344.6464
MSROA[60]	0.7734	0.3749	41.8366	180.1871	5807.8499
ISCA[61]	0.8125	0.4375	42.0984	176.6382	6059.7457

Table 9. Experimental results of the pressure vessel design.

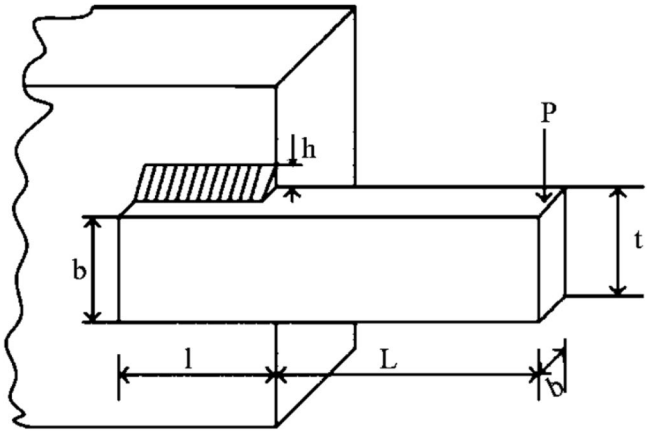

Fig. 9. Schematic diagram of the welded beam design.

Subject to:

$$g_1(\vec{y}) = \tau(\vec{y}) - \tau_{\max} \leqslant 0 \tag{35}$$

$$g_2(\vec{y}) = \sigma(\vec{y}) - \sigma_{\max} \leqslant 0 \tag{36}$$

$$g_3(\vec{y}) = \delta(\vec{y}) - \delta_{\max} \leqslant 0 \tag{37}$$

$$g_4(\vec{y}) = y_1 - y_4 \leqslant 0 \tag{38}$$

$$g_5(\vec{y}) = P - P_c(\vec{y}) \leqslant 0 \tag{39}$$

$$g_6(\vec{y}) = 0.125 - y_1 \leqslant 0 \tag{40}$$

$$g_7(\vec{y}) = 1.10471 y_1^2 + 0.04811 y_3 y_4 (14.0 + y_2) - 5 \leqslant 0 \tag{41}$$

where:

$$\tau(\vec{y}) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{y_2}{2R} + (\tau'')}, \quad \tau' = \frac{P}{\sqrt{2}y_1 y_2}, \quad \tau'' = \frac{MR}{J} \tag{42}$$

$$M = P\left(L + \frac{y_2}{2}\right), \quad R = \sqrt{\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2}, \quad \sigma(\vec{y}) = \frac{6PL}{y_4 y_3^2} \tag{43}$$

$$J = 2\left\{\sqrt{2}y_1 y_2\left[\frac{y_2^2}{4} + \left(\frac{y_1 + y_3}{2}\right)^2\right]\right\}, \quad \delta(\vec{y}) = \frac{6PL^3}{E y_4 y_3^2} \tag{44}$$

$$P_c(\vec{y}) = \frac{4.013E\sqrt{\frac{y_3^2 y_4^6}{36}}}{L^2}\left(1 - \frac{y_3}{2L}\sqrt{\frac{E}{4G}}\right) \tag{45}$$

$$P = 6000\,\text{lb}, \quad L = 14\,\text{in}, \quad \delta_{\max} = 0.25\,\text{in}, \quad E = 30 \times 10^6\,\text{psi}, \tag{46}$$

Algorithm	y_1	y_2	y_3	y_4	Best cost
MESCSO	0.198704	3.339804	9.192051	0.198833	1.670363
SCSO[36]	0.2057	3.2530	9.0366	0.2057	1.6952
HPSDE[62]	0.2057	3.4705	9.0366	0.2057	1.7249
GJO[63]	0.2056	3.4719	9.0392	0.2057	1.7252
GSA[18]	0.1821	3.8570	10	0.2057	1.8799
TSA[64]	0.2442	6.2230	8.2956	0.2057	2.3824
SHO[65]	0.2344	3.4597	7.3825	0.3681	2.4930
COA[66]	0.2054	3.2588	9.0367	0.205	1.6956
SMA[67]	0.2057	3.2529	9.0365	0.2057	1.6963
RO[68]	0.2037	3.5285	9.0042	0.2072	1.7353

Table 10. Experimental results of the welded beam design.

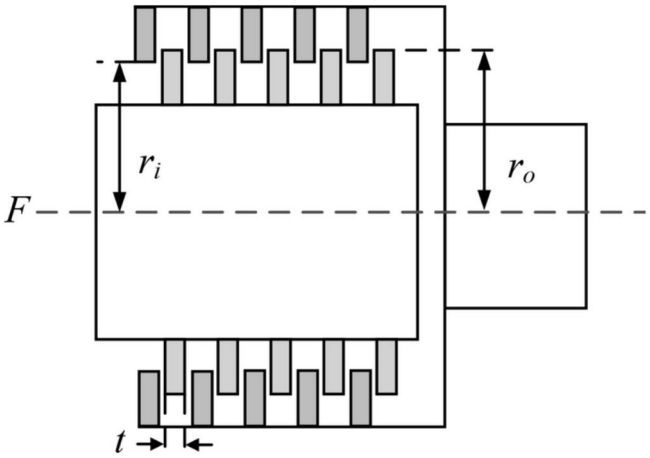

Fig. 10. Schematic diagram of the multi-disc clutch braking problem.

$$\tau_{\max} = 13{,}600\,\text{psi}, \quad \sigma_{\max} = 30{,}000\,\text{psi} \tag{47}$$

Variable Range:

$$0.1 \leq y_1, y_4 \leq 2, \quad 0.1 \leq y_2, y_3 \leq 10 \tag{48}$$

The results of the welded beam design problem are presented in Table 10. Using the MESCSO algorithm, the optimal values of the decision variables were obtained as follows: weld width $h = 0.198704$, connecting beam length $l = 3.339804$, beam heigh $t = 9.192051$, and connecting beam thickness $b = 0.198833$. Compared to other algorithms, MESCSO algorithm achieved the lowest weight, with a final weight value of 1.670363. In the welded beam design problem, many algorithms yielded minimum weights greater than 1.7, whereas MESCSO algorithm produced the smallest weight. This indicates that the MESCSO algorithm provides the best optimization performance.

Multi-disc clutch braking problem

The design optimization of a multi-disc clutch braking system is a representative constrained optimization problem in the field of automotive engineering. The primary objective is to minimize the system mass while ensuring braking performance, thereby enhancing overall vehicle energy efficiency and operational stability. This problem involves five key design variables: inner diameter r_i, outer diameter r_o, brake disc thickness t, driving force F, and surface friction coefficient z. The structural configuration of the system is illustrated in Fig. 10.

The mathematical formulation of the multi-disc clutch braking problem is as follows.

Consider:

$$\vec{y} = [y_1\ y_2\ y_3\ y_4\ y_5] = [r_i\ r_o\ t\ F\ z] \tag{49}$$

Objective function:

$$f(\vec{y}) = \pi(y_2^2 - y_1^2)y_3(y_5 + 1)\rho \quad (\rho = 0.0000078) \tag{50}$$

Algorithm	y_1	y_2	y_3	y_4	y_5	Best cost
MESCSO	69.999	90	1	735.016	2	0.23524
SCSO[36]	69.999	90	1	837.503	2	0.23531
DMO[69]	69.457	90	1	478.427	2	0.24081
LCA[70]	60	90	1	18.626	2	0.33081
WCA[71]	70	90	1	910	3	0.31365
AOA[47]	69.728	90	1	643.037	2	0.23803
LSHADE[72]	69.999	90	1	2.458	2	0.23525
MFO[73]	69.545	90	1	321.131	2	0.23991

Table 11. Experimental results of the welded beam design.

Fig. 11. Schematic diagram of the tension/compression spring design.

Subject to:

$$g_1(\vec{y}) = \Delta r + y_1 - y_2 \leq 0 \tag{51}$$

$$g_2(\vec{y}) = -l_{\max} + (y_5 + 1)(y_3 + \delta) \leq 0 \tag{52}$$

$$g_3(\vec{y}) = P_{rz} - P_{\max} \leq 0 \tag{53}$$

$$g_4(\vec{y}) = P_{rz} v_{sr} - P_{\max} v_{sr\,\max} \leq 0 \tag{54}$$

$$g_5(\vec{y}) = v_{sr} - v_{sr\,\max} \leq 0 \tag{55}$$

$$g_6(\vec{y}) = T - T_{\max} \leq 0 \tag{56}$$

$$g_7(\vec{y}) = sM_s - M_h \leq 0 \tag{57}$$

$$g_8(\vec{y}) = T \geq 0 \tag{58}$$

where:

$$M_h = \frac{2}{3}\mu y_4 y_5 \frac{y_2^3 - y_1^3}{y_2^2 - y_1^2}\ \mathrm{N \cdot mm}, \quad P_{rz} = \frac{y_4}{\pi(y_2^2 - y_1^2)}\ \mathrm{N/mm^2}, \tag{59}$$

$$v_{rz} = \frac{2\pi n(y_2^3 - y_1^3)}{90(y_2^2 y_1^2)}\ \mathrm{mm/s}, \quad T = \frac{I_z \pi n}{30(M_h + M_f)} \tag{60}$$

$$\Delta r = 20\ \mathrm{mm}, \quad I_z = 55\ \mathrm{Kg \cdot m^2}, \quad P_{\max} = 1, \quad T_{\max} = 15\ \mathrm{s}, \tag{61}$$

$$\mu = 0.5, \quad s = 1.5, \quad M_s = 40\,\mathrm{Nm}, \quad M_f = 3\,\mathrm{Nm}, \tag{62}$$

$$n = 250\,\mathrm{rpm}, \quad v_{sr\,\max} = 10\,\mathrm{m/s}, \quad l_{\max} = 30\,\mathrm{mm} \tag{63}$$

Variable Range:

$$\begin{gathered} 60 \leq y_1 \leq 80, \quad 90 \leq y_2 \leq 110, \quad 1 \leq y_3 \leq 3, \\ 0 \leq y_4 \leq 1000, \quad 2 \leq y_5 \leq 9 \end{gathered} \tag{64}$$

The experimental results for the multi-plate clutch braking problem are presented in Table 11. As shown in Table, the MESCSO algorithm yielded an inner diameter $r_i = 69.999$, an outer diameter $r_o = 90$, brake disc thickness $t = 1$, driving force $F = 735.016$, and surface friction coefficient $z = 2$. Compared to other algorithms, MESCSO algorithm achieved the lowest objective weight, with a final value of 0.235242. Among the seven algorithms tested for this problem, MESCSO demonstrated the best optimization performance.

Algorithm	y_1	y_2	y_3	Best cost
MESCSO	0.051713	0.357294	11.255436	0.012665
SCSO[36]	0.0500	0.3175	14.0200	0.01271702
HS[74]	0.051154	0.349871	12.076432	0.012671
GWO[57]	0.05169	0.356737	11.28885	0.012666
SSA[9]	0.051207	0.345215	12.00403	0.012676
BWO[75]	0.0500	0.3122	14.7963	0.0131095
DE[6]	0.051609	0.354714	11.41083	0.01267
GSA[18]	0.0606	0.2749	4.8674	0.0177629

Table 12. Experimental results of the welded beam design.

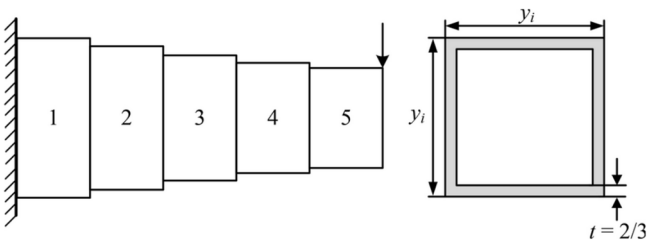

Fig. 12. Schematic diagram of the cantilever beam design.

Tension/compression spring design problem

The structural optimization of a tension/compression spring aims to minimize the mass of the spring while satisfying a set of mechanical performance and geometric constraints, thereby improving system efficiency and structural compactness. This problem involves three primary design variables: wire diameter d, average coil diameter D, and effective coil number N. A schematic diagram of the spring is illustrated in Fig. 11.

The mathematical formulation of the tension/compression spring design problem is as follows.

Consider:

$$\vec{y} = [y_1 \ y_2 \ y_3] = [d \ D \ N] \tag{65}$$

Objective function:

$$f(\vec{y}) = (y_3 + 2) y_2 y_1^2 \tag{66}$$

Subject to:

$$g_1(\vec{y}) = 1 - \frac{y_2^3 y_3}{71785 y_1^4} \leq 0 \tag{67}$$

$$g_2(\vec{y}) = \frac{4y_2^2 - y_1 y_2}{12566(y_2 y_1^3 - y_1^4)} + \frac{1}{5108 y_1^2} - 1 \leq 0 \tag{68}$$

$$g_3(\vec{y}) = 1 - \frac{140.45 y_1}{y_2^2 y_3} \leq 0 \tag{69}$$

$$g_4(\vec{y}) = \frac{y_1 + y_2}{1.5} - 1 \leq 0 \tag{70}$$

Variable Range:

$$0.05 \leq y_1 \leq 2.00, \quad 0.25 \leq y_2 \leq 1.30, \quad 2.00 \leq y_3 \leq 15.0 \tag{71}$$

The experimental results of various optimization algorithms for the tension/compression spring design problem are summarized in Table 12. As shown, the MESCSO algorithm achieved a well-balanced solution across all three design variables, with values of $d = 0.051713$, $D = 0.357294$, and $N = 11.255436$. The corresponding optimal objective function value was 0.012665, which was the smallest among all the algorithms, demonstrating the MESCSO algorithm's superior performance in this problem.

Algorithm	y_1	y_2	y_3	y_4	y_5	Best cost
MESCSO	6.036097	5.309212	4.478850	3.501063	2.148696	1.339972
SCSO[36]	6.06771	5.33599	4.46869	3.51773	2.08916	1.34416
ERHHO[76]	6.0509	5.2639	4.514	3.4605	2.1878	1.3402
SFO[77]	6.81960	5.63017	4.37540	7.97319	2.24601	1.68757
EDO[78]	5.4046	5.9187	5.1647	3.6009	2.4928	1.4091
AVOA[79]	6.0066	5.3214	4.4836	3.5096	2.1526	1.3400
LFD[80]	5.7495	6.3313	3.6469	4.3527	2.6045	1.4155

Table 13. Experimental results of the cantilever beam design.

Cantilever beam design problem

The primary objective of the cantilever beam design problem is to minimize the total weight of the beam by adjusting the dimensional parameters of its structural components while satisfying a set of design constraints. In this problem, the design variables typically consist of the widths or heights of five hollow square cross-section beam elements, with each variable representing the size of a specific beam segment. All beam segments share the same wall thickness. A schematic diagram of the cantilever beam model is shown in Fig. 12.

The mathematical formulation of the cantilever beam design problem is as follows.

Consider:

$$\vec{y} = [y_1 \ y_2 \ y_3 \ y_4 \ y_5] \tag{72}$$

Objective function:

$$f(\vec{y}) = 0.0624(y_1 + y_2 + y_3 + y_4 + y_5) \tag{73}$$

Subject to:

$$g(\vec{y}) = \frac{61}{x_1^3} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} - 1 \leq 0 \tag{74}$$

Variable Range:

$$0.01 \leq y_1, y_2, y_3, y_4, y_5 \leq 100 \tag{75}$$

The experimental results for the cantilever beam design problem are presented in Table 13. As shown in the table, the MESCSO algorithm produced optimal widths or heights for the five equal-thickness hollow square sections as 6.036097, 5.309212, 4.478850, 3.501063, and 2.148696, respectively, resulting in a minimum total weight of 1.339972. In this problem, the MESCSO algorithm demonstrated superior optimization performance compared to the other algorithms.

Conclusions

This paper proposes a Multi-Strategy Enhanced Sand Cat Swarm Optimization (MESCSO) algorithm, which effectively addresses several limitations of the original Sand Cat Swarm Optimization (SCSO) algorithm, including poor exploration capability in the later stages, a tendency to fall into local optima, and reduced convergence speed. Firstly, during population initialization, an improved sine mapping and random opposition-based learning (ISMROBL) is introduced to enhance the uniformity and diversity of initial solutions, preventing the population from prematurely converging to a local region in the early phase. Secondly, a nonlinear decreasing parameter is proposed to dynamically balance global exploration and local exploitation. In addition, the introduction of the generalized quadratic interpolation (GQI) strategy significantly strengthens local search capabilities, improving search accuracy around local optima and enhancing exploitation efficiency. The improved mean differential mutation (IMDM) strategy further improves global exploration ability and enhances the algorithm's capacity to escape local optima. Finally, to improve overall convergence stability, an accelerated opposition-based learning (AOBL) mechanism combined with a greedy selection strategy is probabilistically applied after each iteration based on a random threshold. When triggered, this mechanism adjusts individual positions to enhance the algorithm's ability to escape local optima and improve convergence performance. To comprehensively validate the performance of the MESCSO algorithm, extensive comparative experiments were conducted against nine mainstream metaheuristic algorithms on 23 standard benchmark functions with 30 and 500 dimensions and the 10-dimensional CEC2014 benchmark functions. Experimental results show that MESCSO demonstrates superior search accuracy and convergence speed across a wide range of test cases. Furthermore, five typical constrained structural engineering design problems were used to verify the MESCSO algorithm's practical adaptability and applicability. The results confirm that MESCSO consistently achieves better optimization outcomes than the compared algorithms.

In summary, compared to the original SCSO and other related algorithms, MESCSO algorithm through the integration of ISMROBL, the nonlinear decreasing parameter, GQI, IMDM, and AOBL mechanism effectively overcomes the deficiencies of insufficient exploration, local stagnation, and slow convergence in the later

stages of SCSO algorithm. It significantly enhances global exploration, local exploitation precision, and overall algorithmic stability, thereby achieving a comprehensive performance improvement.

Despite the demonstrated advantages of MESCSO, there are still some limitations that need to be addressed in future research. Firstly, the integration of multiple enhancement strategies introduces additional computational overhead, potentially impacting the algorithm's efficiency in handling large-scale or real-time optimization problems. Secondly, despite the improved dynamic balance enabled by nonlinear parameter control, certain strategies (e.g., ISMROBL and AOBL) require manual parameter tuning, which may limit the algorithm's adaptability across diverse problem domains. To address these limitations, we will further optimize the algorithm structure and adaptive parameter adjustment strategy to continuously enhance the generalization performance and stability of MESCSO algorithm in complex optimization environments, and especially conduct in-depth research on the issues of algorithm stability that still need to be improved. In addition, we will expand the MESCSO algorithm to practical applications, including but not limited to UAV 3D path planning, text clustering analysis, feature selection optimization, parameter estimation, image segmentation, fault diagnosis and other typical application scenarios. At the same time, we will explore the deep integration of the MESCSO algorithm with advanced machine learning techniques to optimize and regulate the hyper-parameters of the machine learning model, which will further expand the application prospects and practical value of the algorithm.

Data availability

Data are contained within the article.

Received: 2 July 2025; Accepted: 22 August 2025
Published online: 02 October 2025

References

1. Long, W., Wu, T., Liang, X. & Xu, S. Solving high-dimensional global optimization problems using an improved sine cosine algorithm. *Expert Syst. Appl.* **123**, 108–126. https://doi.org/10.1016/j.eswa.2018.11.032 (2019).
2. Zhou, C., Zhao, H. & Xu, S. High-efficient sample point transform algorithm for large-scale complex optimization. *Comput. Methods Appl. Mech. Eng.* **432**, 117451. https://doi.org/10.1016/j.cma.2024.117451 (2024).
3. Holland, J. H. Genetic algorithms. *Sci. Am.* **267**, 66–73 (1992).
4. Beyer, H.-G. & Schwefel, H.-P. Evolution strategies—-A comprehensive introduction. *Nat. Comput.* **1**, 3–52. https://doi.org/10.1023/A:1015059928466 (2002).
5. Banzhaf, W., Koza, J., Ryan, C., Spector, L. & Jacob, C. Genetic programming. *IEEE Intell. Syst. Appl.* **15**, 74–84. https://doi.org/10.1109/5254.846288 (2000).
6. Storn, R. & Price, K. Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. *J. Global Optim.* **11**, 341–359. https://doi.org/10.1023/A:1008202821328 (1997).
7. Sinha, N., Chakrabarti, R. & Chattopadhyay, P. K. Evolutionary programming techniques for economic load dispatch. *IEEE Trans. Evol. Comput.* **7**, 83–94. https://doi.org/10.1109/TEVC.2002.806788 (2003).
8. Kennedy, J. & Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, vol. 4, 1942–1948 (IEEE, 1995).
9. Xue, J. & Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* **8**, 22–34. https://doi.org/10.1080/21642583.2019.1708830 (2020).
10. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **27**, 1053–1073. https://doi.org/10.1007/s00521-015-1920-1 (2016).
11. Fister, I., Fister, I. Jr., Yang, X.-S. & Brest, J. A comprehensive review of firefly algorithms. *Swarm Evol. Comput.* **13**, 34–46. https://doi.org/10.1016/j.swevo.2013.06.001 (2013).
12. Hu, G., Cheng, M., Houssein, E. H., Hussien, A. G. & Abualigah, L. Sdo: A novel sled dog-inspired optimizer for solving engineering problems. *Adv. Eng. Inform.* **62**, 102783. https://doi.org/10.1016/j.aei.2024.102783 (2024).
13. Tian, A.-Q., Liu, F.-F. & Lv, H.-X. Snow geese algorithm: A novel migration-inspired meta-heuristic algorithm for constrained engineering optimization problems. *Appl. Math. Model.* **126**, 327–347. https://doi.org/10.1016/j.apm.2023.10.045 (2024).
14. Fu, S. et al. Red-billed blue magpie optimizer: A novel metaheuristic algorithm for 2d/3d uav path planning and engineering design problems. *Artif. Intell. Rev.* **57**, 134. https://doi.org/10.1007/s10462-024-10716-3 (2024).
15. Bai, J. et al. Blood-sucking leech optimizer. *Adv. Eng. Softw.* **195**, 103696. https://doi.org/10.1016/j.advengsoft.2024.10369 (2024).
16. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671–680. https://doi.org/10.1126/science.220.4598.671 (1983).
17. Zhao, W., Wang, L. & Zhang, Z. A novel atom search optimization for dispersion coefficient estimation in groundwater. *Futur. Gener. Comput. Syst.* **91**, 601–610. https://doi.org/10.1016/j.future.2018.05.037 (2019).
18. Rashedi, E., Nezamabadi-Pour, H. & Saryazdi, S. Gsa: A gravitational search algorithm. *Inf. Sci.* **179**, 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004 (2009).
19. Shehadeh, H. A. Chernobyl disaster optimizer (cdo): A novel meta-heuristic method for global optimization. *Neural Comput. Appl.* **35**, 10733–10749. https://doi.org/10.1007/s00521-023-08261-1 (2023).
20. Azizi, M., Aickelin, U., Khorshidi, H. & Baghalzadeh Shishehgarkhaneh, M. Energy valley optimizer: A novel metaheuristic algorithm for global and engineering optimization. *Sci. Rep.* **13**, 226. https://doi.org/10.1038/s41598-022-27344-y (2023).
21. Wei, Z., Huang, C., Wang, X., Han, T. & Li, Y. Nuclear reaction optimization: A novel and powerful physics-based algorithm for global optimization. *IEEE Access* **7**, 66084–66109. https://doi.org/10.1109/ACCESS.2019.2918406 (2019).
22. Abedinpourshotorban, H., Shamsuddin, S. M., Beheshti, Z. & Jawawi, D. N. Electromagnetic field optimization: A physics-inspired metaheuristic optimization algorithm. *Swarm Evol. Comput.* **26**, 8–22. https://doi.org/10.1016/j.swevo.2015.07.002 (2016).
23. Shi, Y. Brain storm optimization algorithm. In *Advances in Swarm Intelligence: Second International Conference, ICSI 2011, Chongqing, China, June 12–15, 2011, Proceedings, Part I 2* 303–309. https://doi.org/10.1007/978-3-642-21515-5_36 (Springer, 2011).
24. Zhang, J., Xiao, M., Gao, L. & Pan, Q. Queuing search algorithm: A novel metaheuristic algorithm for solving engineering optimization problems. *Appl. Math. Model.* **63**, 464–490. https://doi.org/10.1016/j.apm.2018.06.036 (2018).
25. Rao, R. V., Savsani, V. J. & Vakharia, D. Teaching-learning-based optimization: An optimization method for continuous non-linear large scale problems. *Inf. Sci.* **183**, 1–15. https://doi.org/10.1016/j.ins.2011.08.006 (2012).
26. Faridmehr, I., Nehdi, M. L., Davoudkhani, I. F. & Poolad, A. Mountaineering team-based optimization: A novel human-based metaheuristic algorithm. *Mathematics* **11**, 1273. https://doi.org/10.3390/math11051273 (2023).
27. Moosavi, S. H. S. & Bardsiri, V. K. Poor and rich optimization algorithm: A new human-based and multi populations algorithm. *Eng. Appl. Artif. Intell.* **86**, 165–181. https://doi.org/10.1016/j.engappai.2019.08.025 (2019).

28. Mohamed, A. W., Hadi, A. A. & Mohamed, A. K. Gaining-sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm. *Int. J. Mach. Learn. Cybern.* **11**, 1501–1529. https://doi.org/10.1007/s13042-019-01053-x (2020).
29. Zolfi, K. Gold rush optimizer: A new population-based metaheuristic algorithm. *Oper. Res. Decis.* **33**, 113–150. https://doi.org/10.37190/ord230108 (2023).
30. Javed, S. T., Zafar, K. & Younas, I. Imitation-based cognitive learning optimizer for solving numerical and engineering optimization problems. *Cogn. Syst. Res.* **86**, 101237. https://doi.org/10.1016/j.cogsys.2024.101237 (2024).
31. Wang, X. Frigatebird optimizer: A novel metaheuristic algorithm. *Phys. Scr.* **99**, 125233. https://doi.org/10.1088/1402-4896/ad8e0e (2024).
32. Wang, X. Gyro fireworks algorithm: A new metaheuristic algorithm. *AIP Adv.* https://doi.org/10.1063/5.0213886 (2024).
33. Wang, X. Fishing cat optimizer: A novel metaheuristic technique. *Eng. Comput.* **42**, 780–833. https://doi.org/10.1108/EC-10-2024-0904 (2025).
34. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**, 67–82. https://doi.org/10.1109/4235.585893 (1997).
35. Velasco, L., Guerrero, H. & Hospitaler, A. A literature review and critical analysis of metaheuristics recently developed. *Arch. Comput. Methods Eng.* **31**, 125–146. https://doi.org/10.1007/s11831-023-09975-0 (2024).
36. Seyyedabbasi, A. & Kiani, F. Sand cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. *Eng. Comput.* **39**, 2627–2651. https://doi.org/10.1007/s00366-022-01604-x (2023).
37. Peng, H. et al. A modified sand cat swarm optimization algorithm based on multi-strategy fusion and its application in engineering problems. *Mathematics* **12**, 2153. https://doi.org/10.3390/math12142153 (2024).
38. Li, Y., Yu, Q. & Du, Z. Sand cat swarm optimization algorithm and its application integrating elite decentralization and crossbar strategy. *Sci. Rep.* **14**, 8927. https://doi.org/10.1038/s41598-024-59597-0 (2024).
39. Cai, Y., Guo, C. & Chen, X. An improved sand cat swarm optimization with lens opposition-based learning and sparrow search algorithm. *Sci. Rep.* **14**, 20690. https://doi.org/10.1038/s41598-024-71581-2 (2024).
40. Zhang, K., He, Y., Wang, Y. & Sun, C. Improved multi-strategy sand cat swarm optimization for solving global optimization. *Biomimetics* **9**, 280. https://doi.org/10.3390/biomimetics9050280 (2024).
41. Tizhoosh, H. R. Opposition-based learning: A new scheme for machine intelligence. In *International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06)*, vol. 1, 695–701. https://doi.org/10.1109/CIMCA.2005.1631345 (IEEE, 2005).
42. Mohapatra, S. & Mohapatra, P. Fast random opposition-based learning golden jackal optimization algorithm. *Knowl. Based Syst.* **275**, 110679. https://doi.org/10.1016/j.knosys.2023.110679 (2023).
43. Zhao, W. et al. Quadratic interpolation optimization (qio): A new optimization algorithm based on generalized quadratic interpolation and its applications to real-world engineering problems. *Comput. Methods Appl. Mech. Eng.* **417**, 116446. https://doi.org/10.1016/j.cma.2023.116446 (2023).
44. Lin, M., Wang, Z., Wang, F. & Chen, D. Improved simplified particle swarm optimization based on piecewise nonlinear acceleration coefficients and mean differential mutation strategy. *IEEE Access* **8**, 92842–92860. https://doi.org/10.1109/ACCESS.2020.2994984 (2020).
45. Deep, K. & Bansal, J. C. Mean particle swarm optimisation for function optimisation. *Int. J. Comput. Intell. Stud.* **1**, 72–92. https://doi.org/10.1504/IJCIStudies.2009.025339 (2009).
46. Yuzgec, U. Accelerated opposition learning based chaotic single candidate optimization algorithm: A new alternative to population-based heuristics. *Knowl. Based Syst.* **314**, 113169. https://doi.org/10.1016/j.knosys.2025.113169 (2025).
47. Abualigah, L., Diabat, A., Mirjalili, S., Abd Elaziz, M. & Gandomi, A. H. The arithmetic optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **376**, 113609. https://doi.org/10.1016/j.cma.2020.113609 (2021).
48. Trojovskỳ, P. & Dehghani, M. Subtraction-average-based optimizer: A new swarm-inspired metaheuristic algorithm for solving optimization problems. *Biomimetics* **8**, 149. https://doi.org/10.3390/biomimetics8020149 (2023).
49. Mirjalili, S. & Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **95**, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008 (2016).
50. Zhong, C., Li, G. & Meng, Z. Beluga whale optimization: A novel nature-inspired metaheuristic algorithm. *Knowl. Based Syst.* **251**, 109215. https://doi.org/10.1016/j.knosys.2022.109215 (2022).
51. Xue, J. & Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomput.* **79**, 7305–7336. https://doi.org/10.1007/s11227-022-04959-6 (2023).
52. Heidari, A. A. et al. Harris hawks optimization: Algorithm and applications. *Futur. Gener. Comput. Syst.* **97**, 849–872. https://doi.org/10.1016/j.future.2019.02.028 (2019).
53. Abdel-Basset, M., Mohamed, R., Azeem, S. A. A., Jameel, M. & Abouhawwash, M. Kepler optimization algorithm: A new metaheuristic algorithm inspired by Kepler's laws of planetary motion. *Knowl. Based Syst.* **268**, 110454. https://doi.org/10.1016/j.knosys.2023.110454 (2023).
54. Mirjalili, S. Sca: A sine cosine algorithm for solving optimization problems. *Knowl. Based Syst.* **96**, 120–133. https://doi.org/10.1016/j.knosys.2015.12.022 (2016).
55. He, Q. & Wang, L. An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Eng. Appl. Artif. Intell.* **20**, 89–99. https://doi.org/10.1016/j.engappai.2006.03.003 (2007).
56. He, Q. & Wang, L. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. *Appl. Math. Comput.* **186**, 1407–1422. https://doi.org/10.1016/j.amc.2006.07.134 (2007).
57. Mirjalili, S., Mirjalili, S. M. & Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **69**, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007 (2014).
58. Mirjalili, S., Mirjalili, S. M. & Hatamlou, A. Multi-verse optimizer: A nature-inspired algorithm for global optimization. *Neural Comput. Appl.* **27**, 495–513. https://doi.org/10.1007/s00521-015-1870-7 (2016).
59. MiarNaeimi, F., Azizyan, G. & Rashki, M. Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowl. Based Syst.* **213**, 106711. https://doi.org/10.1016/j.knosys.2020.106711 (2021).
60. Jia, H. et al. Multi-strategy remora optimization algorithm for solving multi-extremum problems. *J. Comput. Des. Eng.* **10**, 1315–1349. https://doi.org/10.1093/jcde/qwad04 (2023).
61. Gupta, S. & Deep, K. Improved sine cosine algorithm with crossover scheme for global optimization. *Knowl. Based Syst.* **165**, 374–406. https://doi.org/10.1016/j.knosys.2018.12.00 (2019).
62. Lin, M., Wang, Z. & Zheng, W. Hybrid particle swarm-differential evolution algorithm and its engineering applications. *Soft. Comput.* **27**, 16983–17010. https://doi.org/10.1007/s00500-023-09025-8 (2023).
63. Chopra, N. & Ansari, M. M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications. *Expert Syst. Appl.* **198**, 116924. https://doi.org/10.1016/j.eswa.2022.116924 (2022).
64. Babalik, A., Cinar, A. C. & Kiran, M. S. A modification of tree-seed algorithm using Deb's rules for constrained optimization. *Appl. Soft Comput.* **63**, 289–305. https://doi.org/10.1016/j.asoc.2017.10.013 (2018).
65. Dhiman, G. & Kumar, V. Spotted hyena optimizer: A novel bio-inspired based metaheuristic technique for engineering applications. *Adv. Eng. Softw.* **114**, 48–70. https://doi.org/10.1016/j.advengsoft.2017.05.014 (2017).
66. Pierezan, J. & Coelho, L. D. S. Coyote optimization algorithm: A new metaheuristic for global optimization problems. In *2018 IEEE Congress on Evolutionary Computation (CEC)* 1–8. https://doi.org/10.1109/CEC.2018.8477769 (IEEE, 2018).

67. Li, S., Chen, H., Wang, M., Heidari, A. A. & Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Futur. Gener. Comput. Syst.* **111**, 300–323. https://doi.org/10.1016/j.future.2020.03.055 (2020).
68. Kaveh, A. & Khayatazad, M. A new meta-heuristic method: Ray optimization. *Comput. Struct.* **112**, 283–294. https://doi.org/10.1016/j.compstruc.2012.09.00 (2012).
69. Agushaka, J. O., Ezugwu, A. E. & Abualigah, L. Dwarf mongoose optimization algorithm. *Comput. Methods Appl. Mech. Eng.* **391**, 114570. https://doi.org/10.1016/j.cma.2022.114570 (2022).
70. Houssein, E. H., Oliva, D., Samee, N. A., Mahmoud, N. F. & Emam, M. M. Liver cancer algorithm: A novel bio-inspired optimizer. *Comput. Biol. Med.* **165**, 107389. https://doi.org/10.1016/j.compbiomed.2023.107389 (2023).
71. Eskandar, H., Sadollah, A., Bahreininejad, A. & Hamdi, M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Comput. Struct.* **110**, 151–166. https://doi.org/10.1016/j.compstruc.2012.07.010 (2012).
72. Tanabe, R. & Fukunaga, A. S. Improving the search performance of shade using linear population size reduction. In *2014 IEEE Congress on Evolutionary Computation (CEC)* 1658–1665. https://doi.org/10.1109/CEC.2014.6900380 (IEEE, 2014).
73. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl. Based Syst.* **89**, 228–249. https://doi.org/10.1016/j.knosys.2015.07.006 (2015).
74. Lee, K. S. & Geem, Z. W. A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice. *Comput. Methods Appl. Mech. Eng.* **194**, 3902–3933. https://doi.org/10.1016/j.cma.2004.09.007 (2005).
75. Hayyolalam, V. & Kazem, A. A. P. Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **87**, 103249. https://doi.org/10.1016/j.engappai.2019.103249 (2020).
76. Song, M. et al. Modified Harris Hawks optimization algorithm with exploration factor and random walk strategy. *Comput. Intell. Neurosci.* **2022**, 4673665. https://doi.org/10.1155/2022/4673665 (2022).
77. Gomes, G. F., Cunha, S. S. & Ancelotti, A. C. A sunflower optimization (sfo) algorithm applied to damage identification on laminated composite plates. *Eng. Comput.* **35**, 619–626. https://doi.org/10.1007/s00366-018-0620-8 (2019).
78. Abdel-Basset, M., El-Shahat, D., Jameel, M. & Abouhawwash, M. Exponential distribution optimizer (edo): A novel math-inspired algorithm for global optimization and engineering problems. *Artif. Intell. Rev.* **56**, 9329–9400. https://doi.org/10.1007/s10462-023-10403-9 (2023).
79. Abdollahzadeh, B., Gharehchopogh, F. S. & Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **158**, 107408. https://doi.org/10.1016/j.cie.2021.107408 (2021).
80. Houssein, E. H., Saad, M. R., Hashim, F. A., Shaban, H. & Hassaballah, M. Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **94**, 103731. https://doi.org/10.1016/j.engappai.2020.103731 (2020).

Author contributions

Conceptualization, M.L. and Z.D.; methodology, M.L. and Z.D.; software, M.L. and Z.D.; validation, Z.D. and Z.Q.; formal analysis, H.C. and H.L.; investigation, Z.D., Z.Q., H.C. and H.L.; resources, M.L.; data curation, Z.D. and Z.Q.; writing original draft preparation, M.L. and Z.D.; writing review and editing, M.L. and Z.D.; project administration, M.L.; funding acquisition, M.L. All authors have read and agreed to the published version of the manuscript.

Funding

This research was funded by Projects of Guangdong Province Department of Education (No.2019KZDZX1034, 2021KCXTD027 and 2018KTSCX237) and the project of China Southern Power Grid Technology Corporation (No.GDKJXM20222596).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.