

# **Multi-strategy boosted dung beetle algorithm and its application for bankruptcy prediction**

Dedai Wei<sup>a</sup>, Kaichen Ouyang<sup>b</sup>, \*, Zimo Wang<sup>c</sup>, Xinye Sha<sup>d</sup>, Yiran Xie<sup>e</sup>, Minyu Qiu<sup>f</sup>, Zongfan Yi<sup>g</sup>.

<sup>a</sup>College of Economics, Shenyang University, Shenyang 110000, China

[wdd31212@163.com](mailto:wdd31212@163.com)

<sup>b</sup>Department of Mathematics, University of Science and Technology of China, Hefei 230026, China

[oykc@mail.ustc.edu.cn](mailto:oykc@mail.ustc.edu.cn)

<sup>c</sup>Department of Philosophy, Nankai University, Tianjin 300381, China

[miemie999@sina.com](mailto:miemie999@sina.com)

<sup>d</sup>Graduate School of Arts and Sciences, Columbia University, NY 10027, America

[xs2399@columbia.edu](mailto:xs2399@columbia.edu)

<sup>e</sup>School of Humanities and Social Sciences, University of Science and Technology of China, Hefei 230026, China

[yr\\_x@mail.ustc.edu.cn](mailto:yr_x@mail.ustc.edu.cn)

<sup>f</sup>Information Engineering College, China Jiliang University, Hangzhou 310018, China

[vpn2733@autuni.ac.nz](mailto:vpn2733@autuni.ac.nz)

<sup>g</sup>Faculty of Business Administration, The Chinese University of Hong Kong, Shatin 999077, Hong Kong SAR, China

[yizongfan@163.com](mailto:yizongfan@163.com)

\*Corresponding author: Kaichen Ouyang([oykc@mail.ustc.edu.cn](mailto:oykc@mail.ustc.edu.cn))

**Abstract:** Predicting bankruptcy is a vital tool in the finance industry, helping businesses make informed decisions. However, many existing models suffer from low accuracy due to non-uniform parameter selection. This paper introduces an innovative bankruptcy prediction model utilizing the Kernel Extreme Learning Machine (KELM). We propose an enhanced variant of the Dung Beetle Optimization (DBO) algorithm, named MSDBO, to optimize and balance the penalty and kernel parameters within KELM. The enhanced MSDBO algorithm incorporates three key strategies: (1) an adaptive dung beetle demarcation strategy that improves search efficiency in different stages, (2) an optimal boundary control strategy that dynamically adjusts the search space boundaries, and (3) a foraging enhancement strategy that avoids falling into local optima traps. The performance of this optimizer is validated through extensive experiments and subsequently tested in a bankruptcy prediction module. To further highlight the advantages of MSDBO, comparative analyses are conducted with the original DBO, other DBO variants, and a range of sophisticated algorithms. Experimental outcomes demonstrate that MSDBO achieves an exceptional balance between exploration and exploitation. Additionally, we apply MSDBO to 3 engineering design and implemented the KELM model, utilizing the MSDBO-optimized KELM classifier for bankruptcy prediction. The efficacy of the MSDBO-KELM model is thoroughly assessed using a financial dataset, in comparison to KELM-based models employing various competitive optimization algorithms. Overall research findings indicate that our newly introduced MSDBO-KELM model outperforms other models in terms of classification accuracy, Matthews correlation coefficient,

sensitivity, and specificity. As a more innovative and effective prediction model, MSDBO-KELM shows promise as a dependable tool for financial risk assessment, achieving superior performance in predicting bankruptcy.

**Key words:** Bankruptcy prediction; Kernel extreme learning machine; Dung beetle optimization; Global optimization.

## 1. Introduction

As the financial crisis has swept across the globe, corporate bankruptcy prediction has gained significant attention from financial institutions ([Jace, Koumanakos, & Tsagkanos, 2022](#)). It is crucial for companies to establish a credible early alerting framework and develop an effective tool to predict corporate bankruptcy ([Lohmann & Möllenhoff, 2023](#)), which can help make accurate decisions and predict the potential risk of corporate bankruptcy ([L. Li & Faff, 2019](#)) in advance. Bankruptcy prediction typically involves a binary classification problem that needs to be addressed rationally. Classification models ([Son, Hyun, Phan, & Hwang, 2019](#)) produce two types of outputs, i.e., type 1 represents an insolvent company, otherwise type 0. Inputs to classifiers ([Barboza, Kimura, & Altman, 2017](#)) are usually financial statistical ratios originating from reliable financial statements of actual firms. Up to now, numerous classifiers built upon various domain knowledge have been introduced for predicting bankruptcy ([Z. Chen, Chen, & Shi, 2020](#)). Generally, the proposals are categorized into statistical methods or artificial intelligence (AI) methods. Bankruptcy prediction as a binary classification problem has been extensively studied and many new classification models utilizing different tools ([Antunes, Ribeiro, & Pereira, 2017](#)) have been developed. Most of the bankruptcy prediction tools are either built upon traditional and plain statistical methods ([H. Chen, Ahmadianfar, Liang, & Heidari, 2024](#)) or on complicated AI techniques. AI models typically outperform analytical methods in most performance metrics due to their robust capability to handle nonlinear data. Simultaneously, the rapid advancement of Big Data and AI techniques has significantly influenced the evolution of bankruptcy prediction. ([Veganzones & Séverin, 2018](#)).

A large number of typical statistical methods constructed for predicting bankruptcy ([Manthoulis, Doumpos, Zopounidis, & Galarotis, 2020](#)) utilize simple univariate analysis, multivariate discriminant analysis, logistic regression, and factor analysis ([Citterio, 2024](#)). Lately, artificial intelligence methods have received increasing interest in the area of failure prediction ([T.-K. Chen, Liao, Chen, Kang, & Lin, 2023](#)). We can see that ELM, a single hidden layer feed-forward neural network algorithm ([H. Chen, et al., 2024](#)) with quick learning abilities, has exceptional generalization performance, and can quickly achieve multi-class classification ([Lohmann & Möllenhoff, 2023](#)). Also, ELM is a representative learning model for neural networks ([De Falco, Calabro, & Pragliola, 2024](#)), named after Single Hidden Layer Feedforward Neural Network (SLFN) ([Guliyev & Ismailov, 2018](#)). The hidden bias and input weights in this method are arbitrarily generated and the output weights are mathematically computed by Moore-Penrose (MP) ([Kim, 2021](#)) generalized inverse. Generally, universal approximation reflects a neural network's ability to approximate functions. Some researchers demonstrated the approximation ability of multilayer feed-forward networks ([Gao, et al., 2024](#)), i.e., that a very large number of bounded continuous activation functions and continuous mappings can be metrically estimated by a neural network. A drawback with ELM is that randomly assigned node parameters usually lead to variations in classification accuracy from run to run ([Abu Al-Haija, Altamimi, & AlWadi, 2024](#)). Additionally, there is no theory specifying the exact range for the quantity of hidden neurons. Generally, ELMs

require numerous hidden nodes to achieve satisfactory and robust performance ([Pan, Su, Liang, Liang, & Yang, 2024](#)), especially when the number of input variables is high. ELMs often require a large number of hidden neurons ([Anandhakumar, Sakthivel Murugan, & Kumaresan, 2024](#)), which leads to a substantial amount of computation to determine the weight matrix due to the large number of hidden neurons. Even with reduced input dimensions, cross-validation (CV) ([Kovačić, Radočaj, & Jurišić, 2024](#)) remains time-consuming when determining the number of hidden neurons over a very large range.

The kernel extreme learning machine ([Huang, Zhou, Ding, & Zhang, 2012](#)) (KELM) can handle the problem by computing the kernel array instead of the mapping matrix ([Zheng, Chen, Wang, Wang, & Qin, 2022](#)). For KELM, it is unnecessary to specify the quantity of hidden nodes (the user is not required to know the hidden layer feature mapping instead of its corresponding kernel function) ([Sun, et al., 2024](#)). In addition, KELM has stronger nonlinear capture capability through the powerful mapping ability of the kernel functions compared to ELM ([Majumder, Dash, & Bisoi, 2018](#)) and has been utilized in numerous pattern recognition applications ([Maria Jesi & Antony Asir Daniel, 2024](#)). Recently, in combination with spectroscopy, KELM has been successfully used in various prediction tasks. We have utilized KELM for predicting bankruptcy ([Zhou, Wei, Jie, & Zhang, 2024](#)) with the same real dataset. However, it is important to note that two important parameters of KELM with an RBF kernel ([Y. Zhang, Ma, Wang, Li, & Guo, 2023](#)) are the kernel penalty parameter  $C$  and the bandwidth  $\gamma$  ([Z. Zhang, et al., 2024](#)).  $C$  regulates the balance between model complexity and minimization of fitting error ([Xue & Shen, 2023](#)), while  $\gamma$  specifies a nonlinear transformation from the input space to a high-dimensional feature space ([H. Chen, et al., 2024](#)). A few research works have demonstrated that  $C$  and  $\gamma$  significantly impact the performance of KELM, similar to their effect in SVM ([Peng & Unluer, 2024](#)). Therefore, it is important to set these two key parameters correctly before applying them to real-world problems ([Quan, Liang, Wang, Li, & Chang, 2024](#)). These parameters are typically obtained using grid search methods, but their primary drawback is that they tend to get trapped in local optima.

Introduced in 2022, the dung beetle optimization (DBO) algorithm offers an innovative approach to population intelligence optimization. It mainly emulates the movements, dances, foraging, theft, and mating rituals of dung beetles. Despite its demonstrated superiority, DBO has shortcomings, including a tendency to get trapped in local optima and limited global searching capability. As far as we know, the potential of DBO remains uncharted for fine-tuning the optimal parameters of KELM. Hence, to discover a more effective method for predicting bankruptcy and addressing bank default events and credit risk management, this paper proposes an improved DBO algorithm ([Wang, et al., 2023](#)), named Multiple-Strategies DBO (MSDBO) the improved algorithm parts including 3 strategies (adaptive ball-rolling dung beetle demarcation strategy, convergent boundary control strategy and foraging enhancement strategy).

Therefore, this study aims to investigate the improved DBO method's capability of dealing with the model selection task in engineering design problems and KELM for classification, and subsequently apply the resulting model MSDBO-KELM to accurately and efficiently forecast company bankruptcy. To verify its effectiveness, this study evaluates the performance of the novel IDBO-KELM in comparison to several established algorithms including QHDBO ([Zhu, et al., 2024](#)), NRBO ([Sowmya, Premkumar, & Jangir, 2024](#)), WOA ([Mirjalili & Lewis, 2016](#)), GWO ([Mirjalili, Mirjalili, & Lewis, 2014](#)), HHO ([Heidari, Faris, Aljarah, Mafarja, & Chen, 2019](#)), EVO ([Azizi, Aickelin, A. Khorshidi, & Baghalzadeh Shishehgarkhaneh, 2023](#)), PSO ([Kennedy & Eberhart,](#)

1995), CPO ([Abdel-Basset, Mohamed, & Abouhawwash, 2024](#)), and SVM ([Suthaharan, 2016](#)), which are known for their efficacy in machine learning. These methodologies are assessed based on training, validation, and testing accuracies ([N. Li, He, Ma, Wang, & Zhang, 2020](#)), Type I and Type II errors, as well as the area under the receiver operating characteristic curve (AUC) ([Liu, et al., 2023](#)). To ensure the robustness of the experimental outcomes, a cross-validation strategy ([Afzal, Nair, & Asharaf, 2021](#)) is employed, which includes both external 10-fold and internal 5-fold cross-validation. The findings indicate that our proposed MSDBO-KELM outperforms these conventional approaches, demonstrating superior performance on a real-life financial dataset. Furthermore, Optimization of engineering design problems has always been a significant challenge, especially in adjusting certain parameters during the design process, which many traditional algorithms cannot effectively address. In this paper, in addition to applying MSDBO to the KELM model, it is also utilized to optimize three classic engineering design problems: Speed Reducer Design Problem, Pressure Vessel Design Problem, and Step-Cone Pulley Problem. Comparative experiments with 11 other algorithms show that MSDBO demonstrates superior optimization performance, significantly outperforming the other algorithms.

The structure of this paper is as follows. Section 2 presents the original DBO and the three improved strategies. Convergence behavior analysis and exploration and exploitation analysis are presented in Section 3. Section 4 presents the quantitative analysis and statistical analysis built upon benchmark functions. Section 5 applies MSDBO to the 3 engineering design problems. Section 6 covers the implementation of the newly introduced methodology for predicting bankruptcy. Finally, Section 7 presents the conclusions.

## 2. MSDBO



**Fig. 1.** The dung beetles

### 2.1 Conceptual overview of original DBO

The DBO algorithm is designed to discover high-quality solutions by mimicking behaviors of dung beetles. The five updating rules within the algorithms are inspired by their rolling, dancing, foraging, stealing and reproduction behaviors of the dung beetles, as illustrated in **Fig. 1**.

Based on the food-searching behaviors of dung beetles, the comprehensive evolutionary steps of DBO are outlined as follows:

Step 1: Population initialization

The DBO algorithm generates an initial population by randomization, where the position of each dung beetle corresponds to a solution of the optimization problem. For a d-dimensional

optimization problem, the position of the first dung beetle is denoted as  $x_1$ , and the dung beetle population size is denoted as  $N$ , as shown in Eq. (1).

$$X = \begin{bmatrix} x_{1,1} & \cdots & x_{1,d} \\ \vdots & \ddots & \vdots \\ x_{N,1} & \cdots & x_{N,d} \end{bmatrix} \quad (1)$$

### Step 2: Rolling behavior

Obstacle-free mode: In the absence of obstacles, the dung beetle navigates using the sun to keep the dung ball rolling in a straight line, where the intensity of the light affects the dung beetle's path, and the rolling ball position is updated according to Eq. (2) and Eq. (3).

$$x_i(t+1) = x_i(t) + \alpha * k * x_i(t-1) + b * \Delta x \quad (2)$$

$$\Delta x = |x_i(t) - X^W| \quad (3)$$

Obstacle mode: In case of an obstacle, dung beetles use dancing to reorient themselves to a new route, the position of the dancing behavior is updated in the way shown in Eq. (4).

$$x_i(t+1) = x_i(t) + \tan(\theta) |x_i(t) - x_i(t-1)| \quad (4)$$

### Step 3: Reproductive behavior

In the natural world, female dung beetles transport their dung balls to secure locations where they can lay their eggs, subsequently concealing them to create ideal environments for their offspring. Drawing inspiration from this behavior, the authors introduced a boundary selection strategy replicating the egg-laying sites of female dung beetles, as shown in Eq. (5), Eq. (6), and Eq. (7).

$$Lb^* = \max(X^* * (1 - R), Lb) \quad (5)$$

$$Ub^* = \min(X^* * (1 + R), Ub) \quad (6)$$

$$R = 1 - \frac{t}{T_{max}} \quad (7)$$

where,  $Lb^*, Ub^*$  denotes the lower and upper bounds of the spawning area;  $X^*$  denotes the current local optimal solution;  $T_{max}$  denotes the maximum number of iterations;  $Lb, Ub$  represents the lower and upper bounds of the optimization problem. The spawning area is dynamically adjusted with each iteration, resulting in the position of the oospheres also changing throughout the process, as shown in Eq. (8).

$$B_i(t+1) = X^* + b_1 * (B_i(t) - Lb^*) + b_2 * (B_i(t) - Ub^*) \quad (8)$$

where,  $B_i(t)$  represents the position information of the  $i^{th}$  oosphere in the first iteration;  $b_1$  and  $b_2$  are two independent random vectors of  $1 * d$ . The oocytes are strictly confined to the spawning area.

Step 4: Some adult dung beetles surface from the ground to search for food, while the optimal feeding areas for young ones are continuously updated, as shown in Eq. (9) and Eq. (10).

$$Lb^b = \max(X^b * (1 - R), Lb) \quad (9)$$

$$Ub^b = \min(X^b * (1 + R), Ub) \quad (10)$$

where,  $X^b$  represents the global optimal location,  $Lb^b, Ub^b$  respectively represents the lower and upper bounds of the best feeding area.

The position of the little dung beetle is updated according to Eq. (11).

$$x_i(t+1) = x_i(t) + C_1 * (x_i(t) - Lb^b) + C_2 * (x_i(t) - Ub^b) \quad (11)$$

where,  $C_1$  is an arbitrary number that follows normal distribution;  $C_2$  is a vector with every element randomly generated between 0 and 1.

Step 5: A few dung beetles steal dung balls from their peers, and their position is updated according to Eq. (12).

$$x_i(t+1) = X^b + S * g * \{ |x_i(t) - X^*| + |x_i(t) - X^b| \} \quad (12)$$

where,  $g$  is a random vector of size  $1 * d$  that follows a normal distribution, and  $S$  is a constant.

## 2.2 MSDBO strategy

### 2.2.1. Adaptive dung beetle demarcation strategy

A fixed number of rolling dung beetles does not maximize their role in the process. In fact, during the initial phase of the entire optimization process, the global search is more demanding, and more rolling dung beetles are needed to compete for the best position, so that individuals can improve their own exploration ability. With the progress of iteration, a small number of rolling dung beetles can not only retain a certain jumping mechanism, but also will not frequently interfere with the development of optimal solutions. Therefore, this paper applies the adaptive demarcation strategy, i.e., dynamically changes the ratio of rolling dung beetles within the total number to maximize the role of rolling dung beetles. These are shown in Eq. (13) and Eq. (14) below.

$$N_r = \text{round}(N * p) \quad (13)$$

$$p = \left( p_{\max} - (p_{\max} - p_{\min}) * \frac{t}{T} \right) \quad (14)$$

where,  $p$  is the proportion of rolling dung beetles,  $p_{\min} = 0.1$ ,  $p_{\max} = 0.3$ , and  $\text{round}()$  is the rounding function.  $N_r$  is calculated by multiplying  $N$  by  $p$  and then rounding the result to the nearest integer. If  $N * p$  times results in a value exactly between two integers, like 2.5 or 3.5, it is typically rounded to the nearest even integer. This is commonly known as rounding to the nearest or bankers rounding.

### 2.2.2 Optimal boundary control strategy

Within iteration, certain individuals tend to surpass the search boundaries. Typically, the most prevalent approach is to allocate a fixed upper or lower bound value to an individual that crosses the boundary. However, this method does not effectively utilize relevant location information. In the whole search process of optimization algorithm, the whole population should be committed to finding new locations, and the current global optimal is the potential most suitable location point. Therefore, to fully utilize this position information, the concept of sub-optimal solution is proposed in this improvement, that is, the solution whose fitness in the current population is second only to that of the current global optimal individual. The location information from both the current optimal solution and the sub-optimal solution is used to find a new location. The updating equations are included below in Eq. (15) and Eq. (16).

$$x_i(t+1) = r_1 * x_{b1}^t + r_1 * (x_{b1}^t - x_{b2}^t), x < lb \quad (15)$$

$$x_i(t+1) = r_3 * x_{b2}^t + r_4 * (x_{b1}^t - x_{b2}^t), x > ub \quad (16)$$

where,  $x_{b1}^t$  is the global optimal solution of the current iteration, and  $x_{b2}^t$  is the sub-optimal solution of the current iteration population,  $r_1, r_2, r_3, r_4$  are random numbers.

Since the search space is accompanied by the uncertainty of the optimal position, the individual boundary processing should retain some randomness, so random numbers are introduced. The

treatment method with strong randomness gives the population the population to escape the local optimum with a certain probability.

### 2.2.3 Foraging enhancement strategy

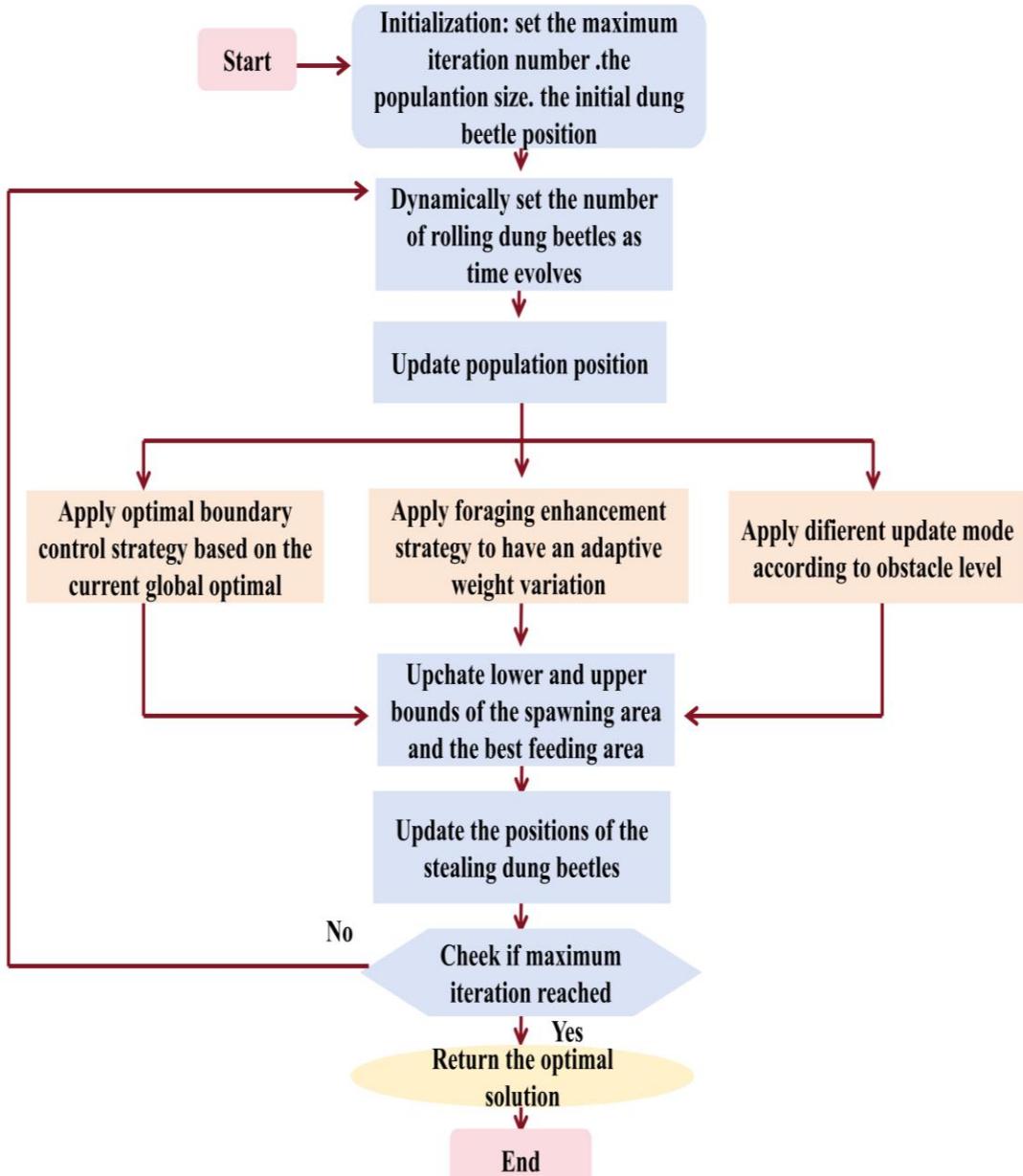
In addition, to prevent the population from clustering and trapping the DBO into the local optimal, it is essential to encourage the population to jump away from its current position. While the original DBO uses normally distributed perturbations to update locations, this method fails to consider that the capability needed for an individual to escape the current location varies at different stages. Therefore, an adaptive weight variation method is introduced to increase the probability of the algorithm escaping the local optimal when the random number exceeds 0.5, while maintaining the original update mode.

$$x_i(t+1) = r_5 * x_{b1}^t + w * (x_{b1}^t - x_i(t)) \quad (17)$$

where,  $w$  is defined in Eq. (18).  $r_5, r_6, r_7, r_8$  are random number,  $x_{b1}^t$  is the global optimal solution of the current iteration.

$$w = |(2 * r_6) - (1 * r_7 + r_8)| \quad (18)$$

## 2.3 Pseudo-code and the description of MSDBO framework



**Fig. 2.** The flow chart of MSDBO

**Fig. 2** shows the execution flow of MSDBO algorithm. First, the maximum number of iterations, population size and initial dung beetle location were initialized, and the number of rolling dung beetles was dynamically adjusted and the population location was updated. The algorithm is then updated using different strategies, including optimal boundary control strategies, foraging enhancement strategies, and different update modes applied according to obstacle levels. Update the upper and lower boundaries of the feeding area and update the location of the thieving beetles, then check if the maximum number of iterations has been reached. If it has not been reached, the iteration continues; If reached, the algorithm returns the optimal solution and the process ends.

And the details of pseudo-code are shown as follows: Firstly, initialization was carried out, parameters were set, initial position of dung beetle population was randomly generated, and optimal solution was set. In the iteration process, the algorithm selects two scenarios based on the random number generation results: the base scenario and the foraging enhancement scenario. If there are no

obstacles, the position is updated according to the specified equation, otherwise the update method of the base scenario is applied. For foraging enhancement scenarios, the algorithm updates the location according to the specified equation. Then the optimal boundary control is performed to adjust the position of the beetles beyond the boundary. Next, the beetles' feeding areas, eggs and dung beetle locations were updated through reproductive behavior, and further updated through stealing behavior. The whole process continues until the maximum number of iterations is attained, at which point the algorithm returns the optimal solution.

---

**Algorithm 1** MSDBO

---

1. ***Begin***
- //Initialization
2.     Set the parameters:  $Max\_iter$  ,  $N$  ,  $Lb$  and  $Ub$
3.     Generate initial positions of the dung beetle population randomly
4.     Set the optimal solution  $X_{Best}$
5.     ***For***  $i = 1,2, \dots, Max\_iter$ :
- //Adaptive demarcation
6.         Calculate rolling proportion of population  $p$  and  $N_r$  based on Eq. (14) and Eq. (13)
7.         Choose between the two scenarios based random number generation outcome
8.             Scenario 1: Base Case
9.                 ***If*** obstacle-free:
10.                     Update positions based on Eq. (2) and (3)
11.                 ***Else***
12.                     Update positions based on Eq. (4)
13.             Scenario 2: Foraging Enhancement
14.                     Update positions based on Eq. (17) and Eq. (18)
- //Optimal Boundary Control
15.         For positions below lower bound:
16.             Update positions based on Eq. (15)
17.         For positions above upper bound:
18.             Update positions based on Eq. (16)
- //Reproductive Behavior
19.         Update  $Lb^*$ ,  $Ub^*$  of the spawning area based on Eq. (5), Eq. (6), and Eq. (7)
20.         Update the oospheres based on Eq. (8)
21.         Update  $Lb^b$ ,  $Ub^b$  of the optimal feeding area based on Eq. (9), and Eq. (10)
22.         Update positions of little dung beetles based on Eq. (11)
- //Stealing Behavior
23.         Update positions based on Eq. (12)
24.         Update  $X_{Best}$  if needed
25.     ***End***
26.     Return the best solution  $X_{Best}$

---

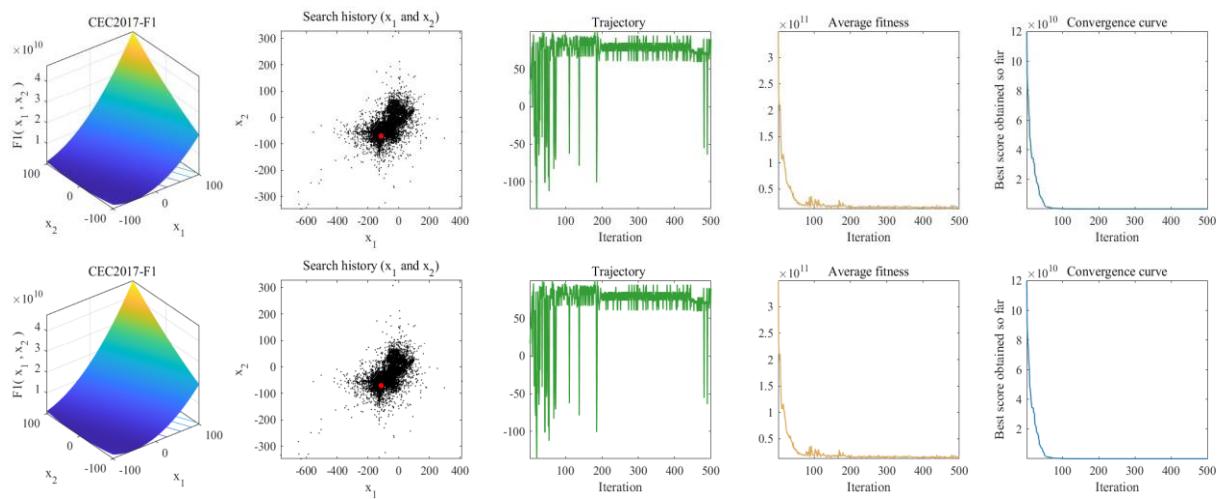
### 3. Experiment on qualitative analysis

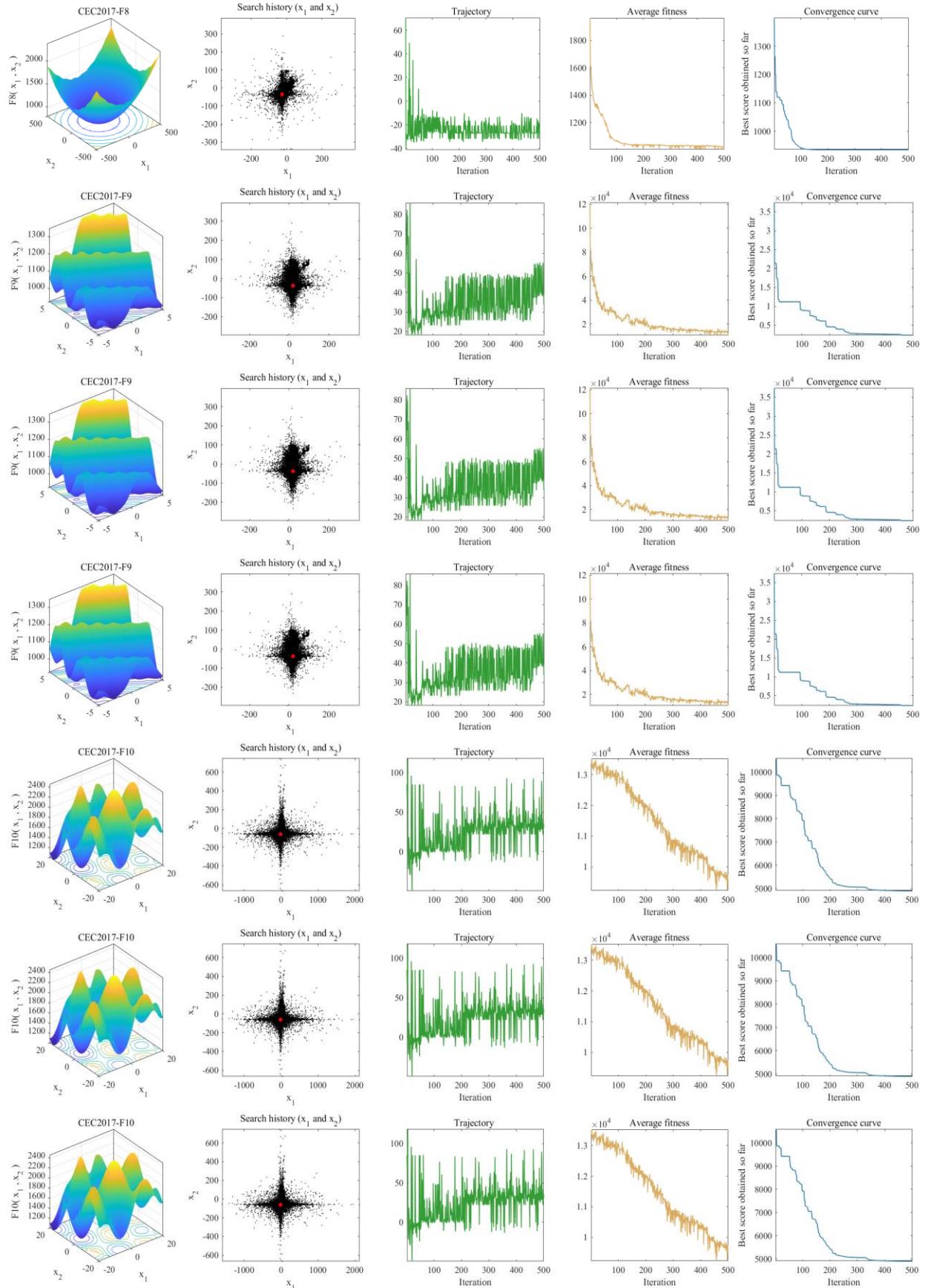
This section verifies the performance of MSDBO by conducting experiments analyzing exploration and exploitation, and convergence behavior.

#### 3.1 Convergence behavior analysis

To illustrate the convergence of MSDBO, we display the convergence behaviors in **Fig. 3**, which displays 9 images corresponding to various test functions in total. The first column displays the 3-dimensional contour of the benchmark function. The second column depicts the final locations of the search agents, highlighted by a red dot which marks the optimal solution. Observations indicate that while the agents are spread throughout the parameter space, they primarily converge around the optimal solution. This convergence underscores MSDBO's superior exploration and exploitation abilities.

Additionally, the third column shows the solution change trajectory of the algorithm during the iterative process, which reflects the dynamic search process and stability of the algorithm. It can be seen that there are considerable fluctuations in the initial stages of the iteration process, but these gradually diminish over time, and tends to be stable after 100 iterations, indicating that MSDBO has undergone several adjustments (early fluctuation) in the search process, avoiding the local optimal solution, and transiting from the initial extensive exploration to the later elaborate development. In addition, the fourth column represents the change of the average fitness value in the whole iteration process and the average performance of the algorithm, providing an intuitive reflection of the optimization effect of the algorithm. Although the initial average fitness value is high, the fitness has a significant decline and then becomes stable, indicating that the algorithm can stabilize in a good solution after a period of time.





**Fig. 3.** Convergence behaviors of MSDBO in the search process.

### 3.2 Assessment of exploration and exploitation

As discussed previously, the MSDBO algorithm is designed to split the search process into two distinct phases: exploration and exploitation. Achieving a balance between the two is essential for improving the robustness of the algorithm. The percentage of exploration and exploitation is determined by Eq. (19) and Eq. (20) accordingly. Within the two equations,  $Div(t)$  denotes the measure of dimensional diversity computed by Eq. (21), and  $Div_{max}$  denotes the highest diversity recorded during the whole iteration.

$$Exploration(\%) = \frac{Div(t)}{Div_{max}} \times 100 \quad (19)$$

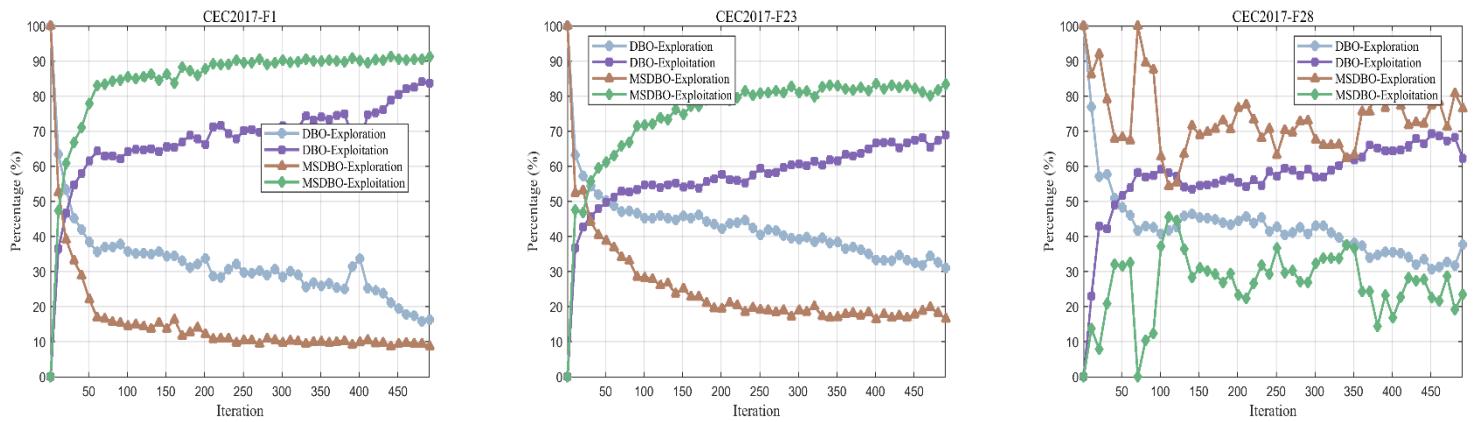
$$Exploitation(\%) = \frac{|Div(t) - Div_{max}|}{Div_{max}} \times 100 \quad (20)$$

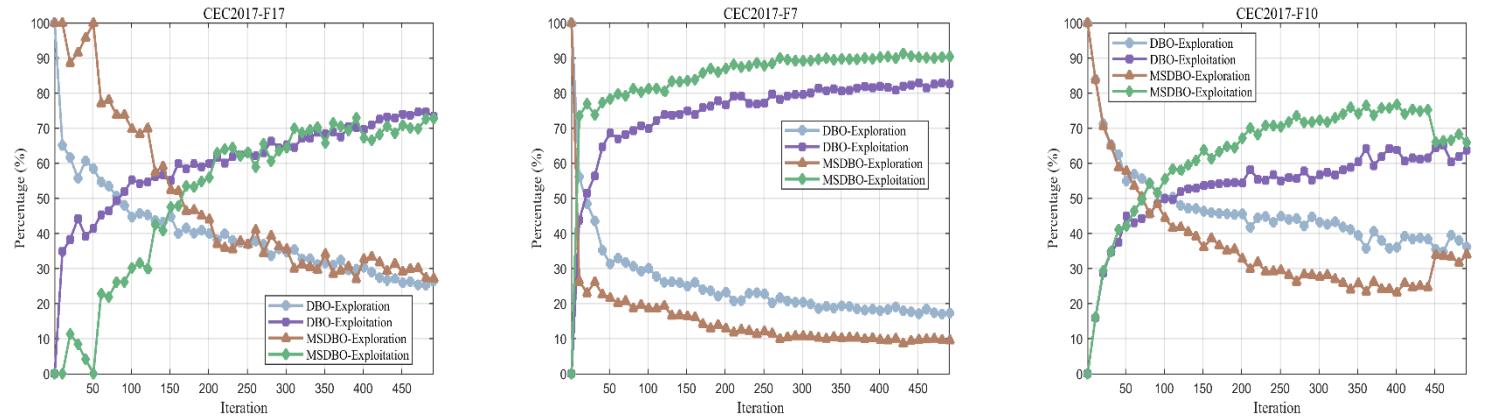
$$Div(t) = \frac{1}{dim} \sum_{d=1}^{dim} \frac{1}{n} \sum_{i=1}^n |median(x_d(t)) - x_{id}(t)| \quad (21)$$

The experimental outcomes are presented in 错误!未找到引用源。. As the iterations increase, across the entire CEC2017 test suite, DBO and MSDBO algorithms demonstrate a significant performance difference. DBO's exploitation performance rapidly increases early in the iterations and stabilizes at around 70%. In contrast, MSDBO's exploitation performance is more remarkable, quickly rising to near 90% and maintaining a high level of stability. The results demonstrate that MSDBO has a clear superiority in identifying and optimizing well-known favorable areas, enabling it to more effectively approach the global optimal solution.

Regarding exploration performance, MSDBO's exploration decreases faster than DBO, suggesting that MSDBO can reduce exploration of new areas earlier and concentrate on in-depth exploitation in known potential areas. This strategy enables the algorithm to avoid stagnation at local optima and accelerates the optimization process.

Overall, MSDBO exhibits superior performance and an effective balance between exploration and exploitation across the entire CEC test suite. This efficient balance demonstrates its applicability and effectiveness in complex optimization environments, showcasing its superior performance on multiple benchmark functions.





**Fig. 4.** The exploration and exploitation of MSDBO and DBO.

**Table 1** CEC2017 benchmark functions.

CEC2017 Benchmark Functions			
Type	No.	Functions Description	$F_t^*$
Unimodal Functions	F1	Shifted and Rotated Bent Cigar Function	100
	F2	Shifted and Rotated Sum of Different Power Function	200
	F3	Shifted and Rotated Zakharov Function	300
	F4	Shifted and Rotated Rosenbrock's Function	400
	F5	Shifted and Rotated Rastrigin's Function	500
	F6	Shifted and Rotated Expanded Scaffer's F6 Function	600
	F7	Shifted and Rotated Lunacek Bi_Rastrigin Function	700
	F8	Shifted and Rotated Non-Continuous Rastrigin's Function	800
Simple Multimodal Functions	F9	Shifted and Rotated Levy Function	900
	F10	Shifted and Rotated Schwefel's Function	1000
	F11	Hybrid Function 1 (N=3)	1100
	F12	Hybrid Function 2 (N=3)	1200
	F13	Hybrid Function 3 (N=3)	1300
	F14	Hybrid Function 4 (N=3)	1400
	F15	Hybrid Function 5 (N=3)	1500
	F16	Hybrid Function 6 (N=3)	1600
	F17	Hybrid Function 6 (N=3)	1700
	F18	Hybrid Function 6 (N=3)	1800
Hybrid Functions	F19	Hybrid Function 6 (N=3)	1900
	F20	Hybrid Function 6 (N=3)	2000
	F21	Composition Function 1 (N=3)	2100
	F22	Composition Function 2 (N=3)	2200
	F23	Composition Function 3 (N=3)	2300
	F24	Composition Function 4 (N=3)	2400
	F25	Composition Function 5 (N=3)	2500
	F26	Composition Function 6 (N=3)	2600
	F27	Composition Function 7 (N=3)	2700

F28	Composition Function 8 (N=3)	2800
F29	Composition Function 9 (N=3)	2900
F30	Composition Function 10 (N=3)	3000

Search Range: [-100,100]<sup>D</sup>

\*Please Note: These problems should be treated as black-box problems. The explicit-equations of the problems are not to be used.

**Table 2** CEC2022 benchmark functions.

CEC2022 Benchmark Functions			
Type	No.	Functions Description	F <sub>t</sub> *
Unimodal Function Basic Functions Hybrid Functions Composition Functions	1	Shifted and full Rotated Zakharov Function	300
	2	Shifted and full Rotated Rosenbrock's Function	400
	3	Shifted and full Rotated Expanded Schaffer's f6 Function	600
	4	Shifted and full Rotated Non-Continuous Rastrigin's Function	800
	5	Shifted and full Rotated Levy Function	900
	6	Hybrid Function 1 (N= 3)	1800
	7	Hybrid Function 2 (N= 6)	200
	8	Hybrid Function 3 (N= 5)	2200
	9	Composition Function 1 (N=5)	2300
	10	Composition Function 2 (N=4)	2400
	11	Composition Function 3 (N=5)	2600
	12	Composition Function 4 (N=6)	2700

Search range: [-100, 100]<sup>D</sup>

\*Please Note: These problems should be treated as black-box problems. The explicit equations of the problems are not to be used.

**Table 3** Parameter settings.

No.	Algorithm	Name of parameters	Value of parameters
1	QHDBO	$k, \lambda, b, S$	0.1,0.1, 1.3, 0.5
2	IDBO	$k, \lambda, b, S$	0.1,0.1, 1.3, 0.5
3	DBO	$k, \lambda, b, S$	0.1,0.1, 1.3, 0.5
4	WOA	$a, b$	Decreased from 2 to 0, 2
5	GWO	$a$	Decreases linearly from 2 to 0
6	HHO	$E_0$	[-1, 1]
7	EVO	$bw, HMCR, PAR$	0.2, 0.95, 0.3
8	PO	$C_1, C_2, \mu, \sigma$	0.2, 3, 25, 3
9	NRBO	$p, q, CF, \varepsilon$	[2, 5], [10, n], [0, 1], 0.5
10	CPO	N, a, N <sub>min</sub> , Tf, T	120, 0.1, 80, 0.5, 2

11	SWO	$Tr, Cr, Nmin$	0.3,0.2,20
12	MSDBO	$k, \lambda, b, S$	0.1,0.1,1.3,0.5

#### 4. Experiment on quantitative analysis

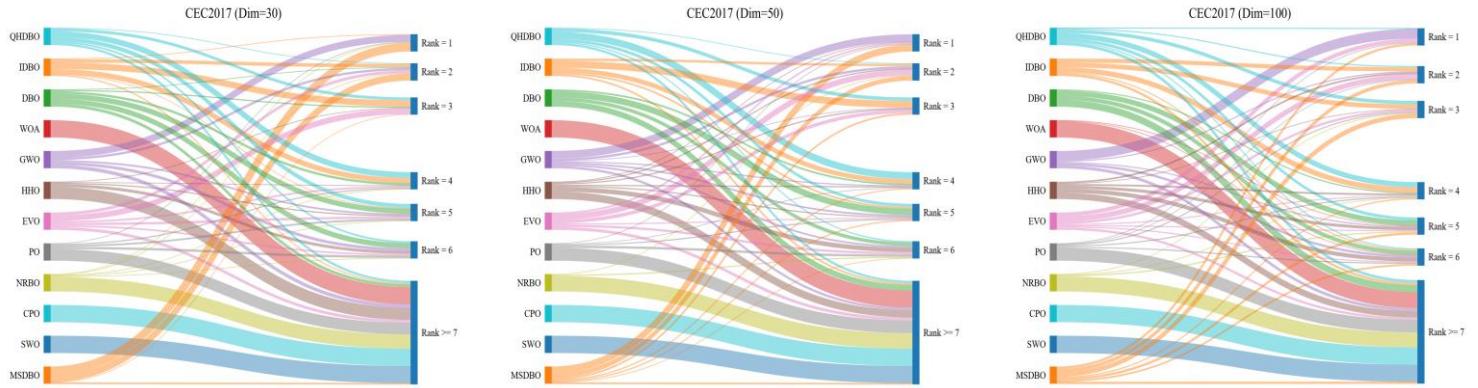
This section compares MSDBO's results to 11 other popular algorithms by evaluating optimization efficiency of various competitive methods with the CEC2017 and CEC2022 numerically. Moreover, two non-parametric tests, namely the Wilcoxon rank sum test and the Friedman mean rank test, were employed to examine the differences and general effectiveness of the candidates.

##### 4.1 Benchmark test functions and parameters setting of competitor algorithms

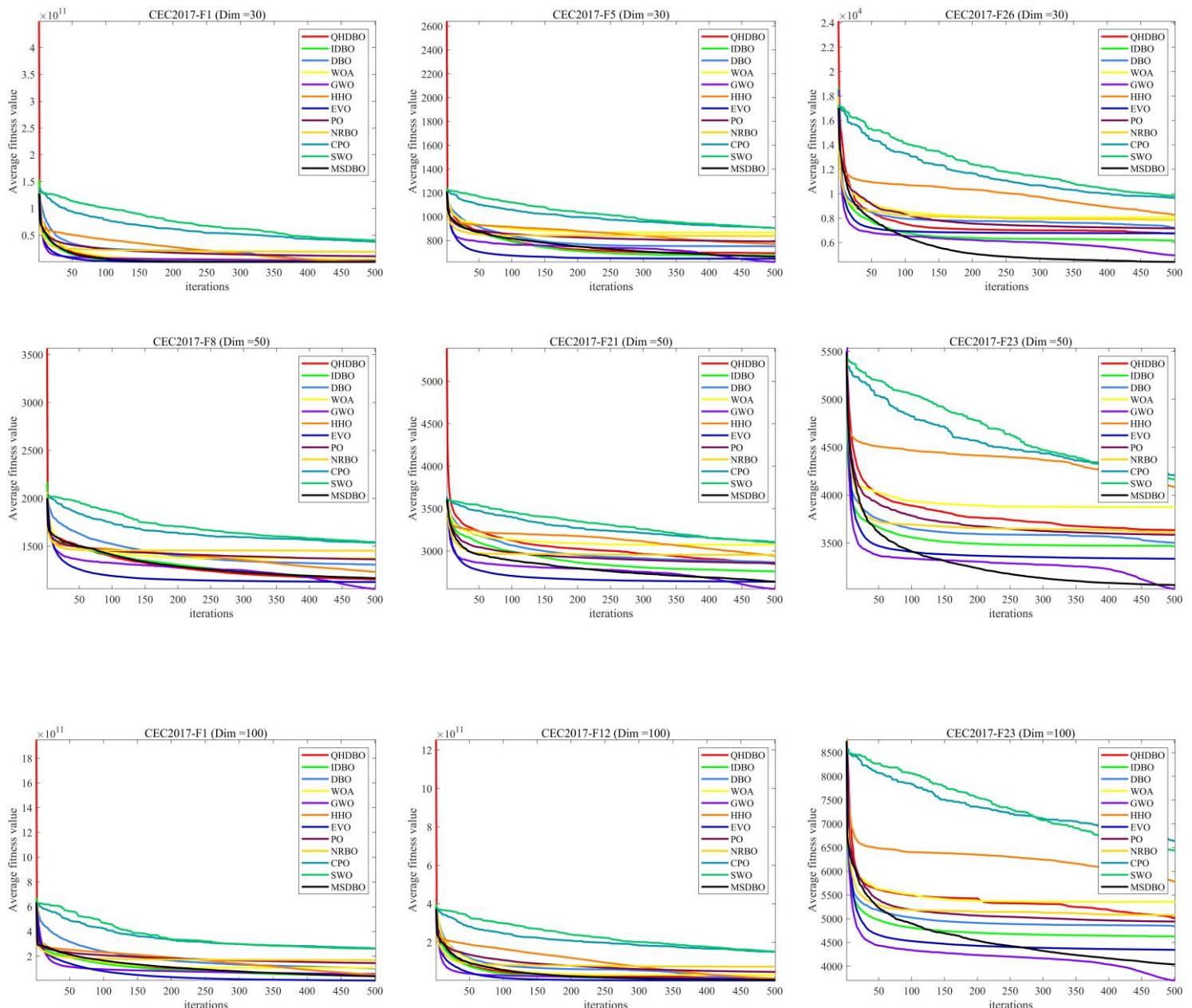
The benchmark test function is essential for determining an algorithm's feasibility, providing a platform to evaluate and compare various optimization methods. In this study, we use the CEC2017 and CEC2022 to evaluate the performance of MSDBO, with dimensions set to 30, 50, 100 (CEC2017) and 10, 20 (CEC2022) accordingly. As the dimension increases, the quantity of locally optimal solutions grows as well. In consequence, these test suites are highly effective in analyzing the algorithm's global optimization abilities and are widely regarded as standard benchmarks. **Table 1** and **Table 2** present the comprehensive details about both test suites accordingly. In **Table 3**, we summarize the parameters of these competitors. The maximum number of iterations was fixed at 500, and the population size at to 500 and 30, with 30 independent runs for all competitors.

##### 4.2 Comparison with other competitive algorithms on CEC2017

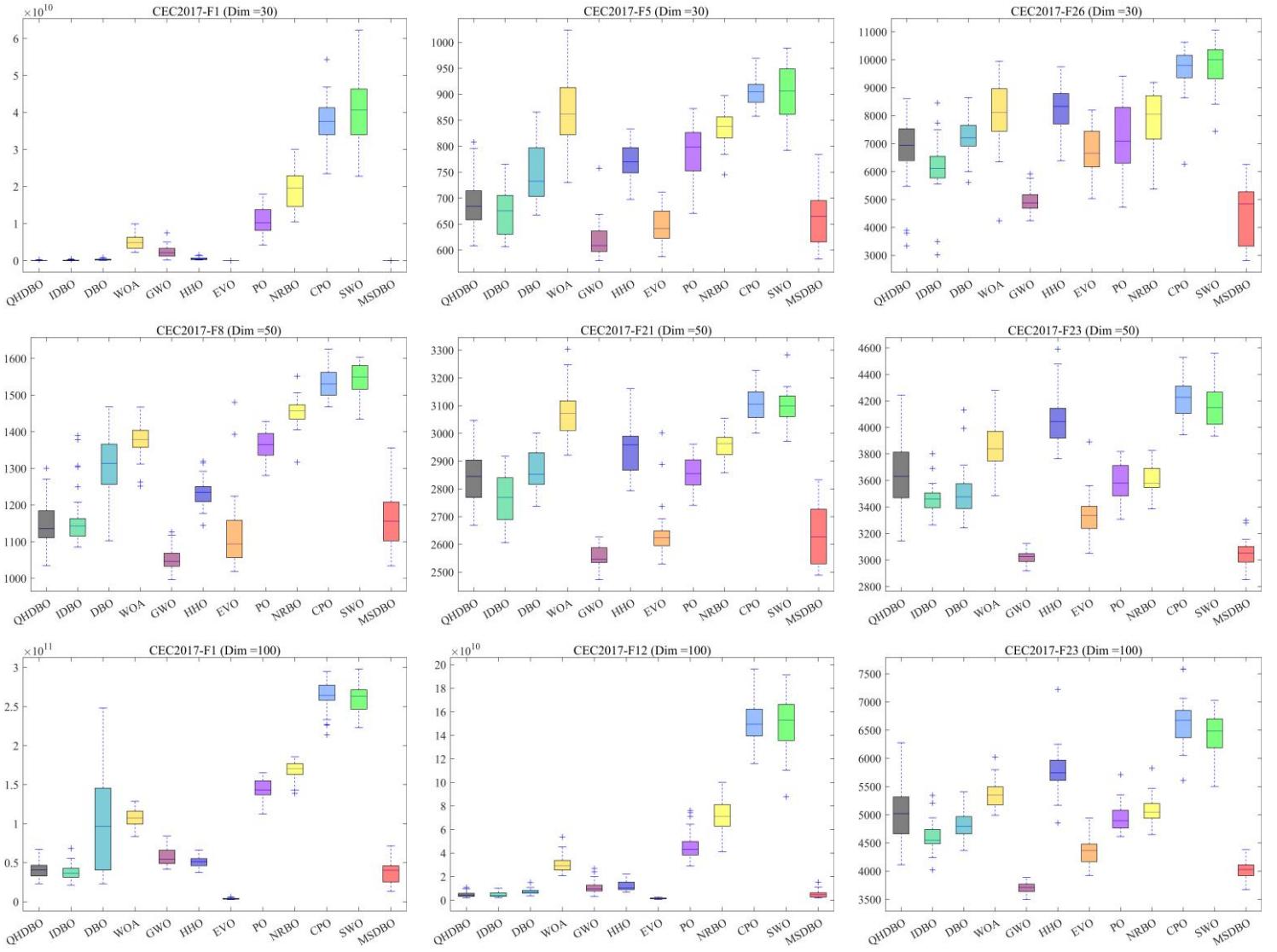
Here, we use the CEC2017 suite for performance testing. The sizes are set to 30, 50, and 100. To illustrate how each algorithm ranks, **Fig. 5** presents a Sankey ranking diagram of the different competitors. The Sankey ranking diagram can display the changes in rankings of different entities over multiple ranks. The left column shows different algorithms, while the right column displays their final rankings, with each entity marked in a different color. The lines illustrate the flow and changes in rankings of the entities, and their width reflects the quantity or proportion of change. This provides observers with an intuitive way to understand and analyze the ranking trends and performance differences of different algorithms. **Tables 4-6** include experimental results (including mean value (Ave) and standard deviation (Std)), with the top results from the 12 contestants highlighted in bold. Clearly, MSDBO consistently places within the top 5, demonstrating outstanding performance. Obviously, MSDBO ranks first in most of the 30 benchmark functions from F1 to F30 of CEC2017 in the 3 different setups, showing excellent performance similar to that of the CEC winners. In addition, it has been observed that MSDBO exhibits a lower sensitivity to dimensional changes. **Fig. 6** and **Fig. 7** depict the convergence curves of various candidates across the 3-dimensional setups. To minimize possible arbitrary variations, we generate the average iteration curve for each candidate with 30 iterations. The outcomes clearly indicate the quicker convergence and higher precision achieved by the newly introduced method. In general, MSDBO is superior to the most advanced optimization algorithms.



**Fig. 5.** The ranking chart of convergence behavior of MSDBO.



**Fig. 6.** Comparison of the convergence speed of different competitors on CEC2017.



**Fig. 7.** Comparison of the box plots of different competitors on CEC2017.

**Table 4** Comparison results on CEC2017 (Dim=30).

Algorithm	Metric	QHDBO	IDBO	DBO	WOA	GWO	HHO	EVO	PO	NRBO	CPO	SWO	MSDBO
CEC2017-F1	Avg	8.804918E+07	1.420709E+08	2.698453E+08	5.820422E+09	2.703753E+09	4.161971E+08	2.080641E+09	1.004244E+10	1.815089E+10	3.917536E+10	3.845969E+10	<b>8.258386E+06</b>
	Std	8.585616E+07	1.935940E+08	1.678093E+08	2.490514E+09	1.886229E+09	2.142458E+08	1.135094E+10	4.296522E+09	4.300666E+09	5.849443E+09	8.154481E+09	<b>3.288988E+07</b>
CEC2017-F2	Avg	1.945579E+35	2.831425E+33	1.801476E+32	1.505739E+36	3.073659E+32	5.619656E+35	1.293715E+48	3.002728E+34	4.931394E+34	6.358698E+43	1.132343E+45	<b>9.332252E+28</b>
	Std	1.065041E+36	9.562791E+33	9.023177E+32	5.386872E+36	1.399960E+33	3.066880E+36	7.085966E+48	1.088148E+35	1.788330E+35	2.654572E+44	5.898484E+45	<b>3.569646E+29</b>
CEC2017-F3	Avg	9.844628E+04	6.551174E+04	8.689158E+04	2.560384E+05	6.764182E+04	5.457453E+04	4.229434E+05	<b>5.177936E+04</b>	5.334465E+04	1.816314E+05	1.308467E+05	9.246078E+04
	Std	2.747528E+04	9.969318E+03	2.247601E+04	5.684709E+04	1.537632E+04	<b>8.465677E+03</b>	8.724751E+05	9.425580E+03	9.417591E+03	5.798339E+04	6.573666E+04	2.384214E+04
CEC2017-F4	Avg	5.567024E+02	5.780421E+02	6.940424E+02	1.284549E+03	6.314071E+02	7.062902E+02	5.590469E+02	1.071763E+03	2.426838E+03	9.682990E+03	8.687220E+03	<b>5.026519E+02</b>
	Std	5.223422E+01	9.389112E+01	1.769147E+02	3.124596E+02	6.700587E+01	1.017334E+02	<b>2.874027E+01</b>	4.345675E+02	1.083769E+03	2.188216E+03	2.774953E+03	5.301130E+01
CEC2017-F5	Avg	6.960870E+02	6.823947E+02	7.365795E+02	8.623154E+02	<b>6.261322E+02</b>	7.624175E+02	6.558231E+02	7.827548E+02	8.481256E+02	9.120201E+02	9.064446E+02	6.669671E+02
	Std	4.628748E+01	3.570089E+01	5.714649E+01	6.213861E+01	<b>2.765114E+01</b>	4.377473E+01	6.140355E+01	3.878300E+01	4.229580E+01	3.076451E+01	4.253635E+01	5.124905E+01
CEC2017-F6	Avg	6.411776E+02	6.335906E+02	6.499308E+02	6.797732E+02	<b>6.142009E+02</b>	6.679266E+02	6.379013E+02	6.647706E+02	6.714497E+02	6.845480E+02	6.863294E+02	6.202966E+02
	Std	1.046783E+01	9.351662E+00	1.012267E+01	1.174272E+01	<b>4.564769E+00</b>	7.205888E+00	8.833515E+00	1.045792E+01	9.466616E+00	7.433426E+00	8.543285E+00	1.037866E+01
CEC2017-F7	Avg	1.000341E+03	9.550293E+02	1.018871E+03	1.314484E+03	<b>8.947890E+02</b>	1.316295E+03	1.041248E+03	1.237978E+03	1.240678E+03	1.478505E+03	1.460956E+03	9.369297E+02
	Std	8.364982E+01	6.651710E+01	8.363563E+01	7.637637E+01	<b>3.941823E+01</b>	6.928198E+01	8.716929E+01	5.563320E+01	8.845800E+01	7.506262E+01	8.757839E+01	7.505514E+01
CEC2017-F8	Avg	9.638259E+02	9.500866E+02	1.037936E+03	1.059645E+03	<b>9.143171E+02</b>	9.848875E+02	9.452593E+02	1.026320E+03	1.091838E+03	1.165360E+03	1.159294E+03	9.250240E+02
	Std	3.603018E+01	4.675883E+01	5.985700E+01	4.397461E+01	3.733435E+01	<b>1.979695E+01</b>	6.543287E+01	3.040368E+01	2.211258E+01	3.362978E+01	2.491834E+01	4.166011E+01
CEC2017-F9	Avg	6.936148E+03	6.844828E+03	6.094814E+03	1.096279E+04	<b>2.710884E+03</b>	8.617113E+03	4.500845E+03	6.975879E+03	7.618955E+03	1.446572E+04	1.437291E+04	3.691886E+03
	Std	1.836231E+03	3.111436E+03	2.186395E+03	2.849535E+03	1.261271E+03	<b>9.284708E+02</b>	1.766304E+03	1.002305E+03	1.543530E+03	2.658430E+03	2.420447E+03	1.883341E+03
CEC2017-F10	Avg	6.718570E+03	7.853533E+03	6.401623E+03	7.775069E+03	<b>5.232101E+03</b>	6.258549E+03	5.470435E+03	6.775295E+03	8.118367E+03	9.717977E+03	9.386342E+03	7.747331E+03
	Std	1.281405E+03	9.107449E+02	1.052680E+03	5.363943E+02	1.449851E+03	9.640529E+02	1.152293E+03	7.014830E+02	4.577012E+02	<b>3.325950E+02</b>	3.713715E+02	1.309136E+03
CEC2017-F11	Avg	1.745080E+03	1.493735E+03	1.857402E+03	9.557787E+03	2.639971E+03	1.626704E+03	2.436133E+03	2.007961E+03	2.693133E+03	1.425828E+04	9.217875E+03	<b>1.397049E+03</b>
	Std	5.871966E+02	2.254127E+02	3.892945E+02	3.629905E+03	1.134754E+03	<b>1.619845E+02</b>	5.284368E+03	7.665184E+02	4.993119E+02	3.674768E+03	3.395160E+03	1.981752E+02
CEC2017-F12	Avg	3.750760E+07	4.903081E+07	7.278053E+07	5.557436E+08	1.042051E+08	8.510728E+07	4.057464E+08	3.531773E+08	1.599540E+09	6.122999E+09	5.491332E+09	<b>5.219535E+06</b>
	Std	4.989654E+07	1.077612E+08	1.038161E+08	3.795274E+08	1.030833E+08	7.143414E+07	1.905736E+09	3.557487E+08	1.019520E+09	2.163399E+09	2.100370E+09	<b>5.180033E+06</b>

CEC2017-F13		Avg	4.351309E+06	4.657244E+06	2.056624E+07	1.348298E+07	6.733500E+06	1.185080E+06	1.294593E+05	3.455891E+07	3.634091E+08	3.121576E+09	3.599343E+09	<b>3.324890E+04</b>	
		Std	1.340009E+07	1.387539E+07	5.300083E+07	1.371200E+07	1.567748E+07	1.221622E+06	1.997002E+05	7.618673E+07	3.119554E+08	1.534566E+09	2.051658E+09	<b>3.416063E+04</b>	
CEC2017-F14	Avg	5.291999E+05	2.611803E+05	3.158146E+05	3.222276E+06	7.319406E+05	8.520504E+05	8.096432E+05	5.831600E+05	<b>1.166957E+05</b>	5.298937E+06	3.079789E+06	2.060204E+05		
	Std	8.861813E+05	3.311188E+05	2.915151E+05	4.369830E+06	7.769544E+05	8.081693E+05	9.795033E+05	6.480505E+05	<b>1.862455E+05</b>	3.271356E+06	3.011030E+06	2.513195E+05		
CEC2017-F15	Avg	6.702275E+05	4.413900E+04	6.800135E+06	6.575973E+06	1.099026E+06	1.050393E+05	1.868668E+06	7.572349E+05	2.268630E+06	2.672524E+08	3.314110E+08	<b>6.552502E+03</b>		
	Std	2.757674E+06	5.121020E+04	3.509840E+07	8.997857E+06	2.303049E+06	5.881194E+04	1.001291E+07	1.427138E+06	5.252418E+06	1.485187E+08	2.875924E+08	<b>7.372838E+03</b>		
CEC2017-F16	Avg	3.387392E+03	3.211737E+03	3.353625E+03	4.201358E+03	<b>2.744307E+03</b>	3.894252E+03	3.257865E+03	3.582308E+03	3.872361E+03	5.154195E+03	4.910805E+03	3.121575E+03		
	Std	5.043709E+02	3.745347E+02	4.164559E+02	4.863781E+02	<b>3.221952E+02</b>	5.172804E+02	3.349236E+02	4.632822E+02	4.048429E+02	3.634853E+02	4.523077E+02	5.312227E+02		
CEC2017-F17	Avg	2.684875E+03	2.672842E+03	2.663232E+03	2.779645E+03	<b>2.118364E+03</b>	2.686616E+03	2.394513E+03	2.615614E+03	2.635372E+03	3.465446E+03	3.371558E+03	2.260258E+03		
	Std	2.872473E+02	2.258917E+02	3.413284E+02	3.456518E+02	2.564650E+02	2.817392E+02	2.377056E+02	2.537961E+02	2.948718E+02	2.156827E+02	4.190116E+02	<b>1.917028E+02</b>		
CEC2017-F18	Avg	3.068283E+06	2.066027E+06	4.007869E+06	1.678225E+07	2.707026E+06	4.736536E+06	2.011911E+07	4.643032E+06	2.359535E+06	4.828590E+07	2.822598E+07	<b>1.612633E+06</b>		
	Std	5.591089E+06	3.119731E+06	6.406912E+06	1.846664E+07	5.033320E+06	4.403375E+06	5.494327E+07	6.551155E+06	2.937494E+06	3.031069E+07	2.801001E+07	<b>2.085116E+06</b>		
CEC2017-F19	Avg	3.863238E+05	5.481815E+05	3.135131E+06	2.177933E+07	1.109605E+07	1.823544E+06	1.544386E+06	3.710213E+06	1.832971E+07	4.310417E+08	4.085342E+08	<b>8.390003E+03</b>		
	Std	1.202729E+06	1.960183E+06	7.065724E+06	1.808808E+07	3.460854E+07	1.521706E+06	2.491942E+06	3.102887E+06	1.994647E+07	2.898385E+08	4.741370E+08	<b>1.214788E+04</b>		
CEC2017-F20	Avg	2.800249E+03	2.644376E+03	2.772349E+03	2.982504E+03	<b>2.448760E+03</b>	2.738547E+03	2.768999E+03	2.724660E+03	2.780438E+03	3.373032E+03	3.250005E+03	2.551436E+03		
	Std	2.288172E+02	1.821408E+02	2.608644E+02	2.143078E+02	1.568664E+02	2.405073E+02	2.171640E+02	1.634484E+02	1.859990E+02	<b>1.397235E+02</b>	2.243691E+02	2.075192E+02		
CEC2017-F21	Avg	2.583391E+03	2.489708E+03	2.539754E+03	2.638998E+03	<b>2.420416E+03</b>	2.611366E+03	2.447700E+03	2.543403E+03	2.597342E+03	2.693103E+03	2.681067E+03	2.428761E+03		
	Std	6.057489E+01	4.927920E+01	6.416579E+01	5.870633E+01	4.747989E+01	5.050220E+01	4.805263E+01	5.737554E+01	3.695020E+01	<b>2.736033E+01</b>	4.080350E+01	5.799362E+01		
CEC2017-F22	Avg	4.980810E+03	<b>2.403515E+03</b>	5.786622E+03	8.479739E+03	5.809224E+03	7.264783E+03	5.418700E+03	5.164940E+03	5.603190E+03	8.628425E+03	8.142742E+03	3.249022E+03		
	Std	2.882524E+03	<b>8.403600E+01</b>	2.515005E+03	1.238586E+03	2.337972E+03	1.311618E+03	2.485034E+03	1.924304E+03	2.150834E+03	1.750165E+03	1.787303E+03	2.274491E+03		
CEC2017-F23	Avg	2.985158E+03	2.938472E+03	3.027371E+03	3.155023E+03	<b>2.790697E+03</b>	3.244556E+03	2.906237E+03	2.987902E+03	3.055690E+03	3.336935E+03	3.352796E+03	2.829642E+03		
	Std	1.070977E+02	6.737383E+01	8.666356E+01	1.123581E+02	<b>4.357417E+01</b>	1.203508E+02	6.878464E+01	5.699494E+01	5.806800E+01	8.988946E+01	7.823499E+01	5.300580E+01		
CEC2017-F24	Avg	3.203148E+03	3.110895E+03	3.219475E+03	3.273889E+03	<b>2.953534E+03</b>	3.513526E+03	3.071447E+03	3.170368E+03	3.204722E+03	3.630834E+03	3.564994E+03	3.010091E+03		
	Std	2.123549E+02	7.468597E+01	8.378525E+01	1.114353E+02	<b>5.547846E+01</b>	1.821834E+02	9.614347E+01	8.822255E+01	6.437450E+01	1.091586E+02	1.434154E+02	6.145103E+01		
CEC2017-F25	Avg	2.937588E+03	2.921795E+03	3.041362E+03	3.227336E+03	3.011804E+03	3.015531E+03	2.956633E+03	3.160546E+03	3.393887E+03	4.753142E+03	4.853039E+03	<b>2.906892E+03</b>		
	Std	3.996477E+01	2.748475E+01	2.563107E+02	6.881580E+01	7.105650E+01	3.430881E+01	3.194179E+01	1.261520E+02	1.846468E+02	5.491906E+02	6.293990E+02	<b>1.926104E+01</b>		

CEC2017-F26	Avg	6.481990E+03	6.454545E+03	6.952406E+03	8.354319E+03	5.092178E+03	8.420884E+03	6.343142E+03	7.744973E+03	8.205186E+03	1.008085E+04	9.821744E+03	<b>4.305358E+03</b>
	Std	1.566202E+03	9.067507E+02	8.319101E+02	1.058528E+03	<b>4.702285E+02</b>	1.423150E+03	1.107792E+03	1.101381E+03	8.647271E+02	5.990350E+02	5.829537E+02	1.069319E+03
CEC2017-F27	Avg	3.340058E+03	3.321180E+03	3.321242E+03	3.536601E+03	3.271540E+03	3.700653E+03	3.393973E+03	3.401490E+03	3.453437E+03	4.153995E+03	4.026203E+03	<b>3.200007E+03</b>
	Std	9.027556E+01	5.710447E+01	5.460264E+01	1.694918E+02	2.736895E+01	2.112840E+02	6.732522E+01	1.196514E+02	9.573257E+01	2.100721E+02	1.767740E+02	<b>1.505347E-04</b>
CEC2017-F28	Avg	3.526261E+03	3.333365E+03	3.525085E+03	3.928372E+03	3.484752E+03	3.488748E+03	3.350186E+03	3.868729E+03	4.143613E+03	6.264680E+03	6.200610E+03	<b>3.299102E+03</b>
	Std	6.885180E+02	5.225742E+01	4.566234E+02	2.494602E+02	8.024522E+01	8.030046E+01	3.013915E+01	3.420535E+02	4.873035E+02	5.277285E+02	8.074749E+02	<b>2.949285E+00</b>
CEC2017-F29	Avg	4.398695E+03	4.381743E+03	4.631366E+03	5.342228E+03	3.980328E+03	4.905660E+03	4.716198E+03	5.097114E+03	5.069491E+03	6.337536E+03	6.297210E+03	<b>3.794700E+03</b>
	Std	3.294130E+02	3.664871E+02	4.554466E+02	4.121243E+02	<b>2.025927E+02</b>	3.611179E+02	4.806182E+02	5.229605E+02	5.052110E+02	3.278414E+02	7.043987E+02	2.850903E+02
CEC2017-F30	Avg	5.595657E+06	4.750339E+06	3.528741E+06	7.528435E+07	1.222017E+07	1.673533E+07	9.228745E+06	3.574540E+07	9.601718E+07	4.126334E+08	3.608444E+08	<b>9.440094E+03</b>
	Std	5.865951E+06	9.470750E+06	3.987230E+06	6.128732E+07	1.170912E+07	2.560689E+07	7.139553E+06	4.005646E+07	5.390076E+07	2.227698E+08	2.244302E+08	<b>9.819181E+03</b>

**Table 5** Comparison results on CEC2017 (Dim=50).

Algorithm	Metric	QHDBO	IDBO	DBO	WOA	GWO	HHO	EVO	PO	NRBO	CPO	SWO	MSDBO
CEC2017-F1	Avg	1.707585E+09	1.789480E+09	3.532538E+09	2.242237E+10	1.119265E+10	5.212697E+09	<b>1.156783E+08</b>	3.759048E+10	5.210520E+10	9.273565E+10	9.970994E+10	9.876995E+08
	Std	1.322279E+09	1.548335E+09	2.468663E+09	4.065887E+09	3.749767E+09	1.868007E+09	<b>4.189236E+07</b>	7.170380E+09	1.025352E+10	1.010766E+10	1.124319E+10	6.432059E+08
CEC2017-F2	Avg	1.426593E+64	<b>1.423762E+57</b>	5.802872E+66	5.090354E+80	2.451696E+59	3.952250E+64	2.954514E+87	5.175076E+66	4.592441E+63	9.533090E+79	8.883605E+77	1.060173E+60
	Std	7.809998E+64	<b>5.466355E+57</b>	2.332715E+67	2.524153E+81	1.341982E+60	1.770776E+65	1.596808E+88	2.813012E+67	1.162336E+64	5.216136E+80	4.446071E+78	5.806576E+60
CEC2017-F3	Avg	2.974301E+05	2.360564E+05	2.850010E+05	3.223124E+05	1.836194E+05	1.724682E+05	6.868126E+06	<b>1.389491E+05</b>	1.720340E+05	1.077738E+06	3.121473E+05	2.532386E+05
	Std	1.178253E+05	4.746867E+04	8.352488E+04	1.246870E+05	3.735911E+04	2.235821E+04	3.256301E+07	<b>1.799893E+04</b>	4.142511E+04	3.496487E+06	9.386966E+04	4.568379E+04
CEC2017-F4	Avg	9.715102E+02	9.267391E+02	1.900984E+03	4.963250E+03	1.577335E+03	2.071618E+03	7.921878E+02	4.700431E+03	8.294935E+03	2.681265E+04	2.657692E+04	<b>7.830727E+02</b>
	Std	2.212734E+02	3.186820E+02	1.837127E+03	1.815036E+03	7.686192E+02	7.459291E+02	<b>8.444603E+01</b>	1.561029E+03	2.238156E+03	4.241655E+03	5.318095E+03	1.092783E+02
CEC2017-F5	Avg	8.635174E+02	8.763665E+02	9.863129E+02	1.085158E+03	<b>7.620064E+02</b>	9.365479E+02	7.825969E+02	1.022674E+03	1.123968E+03	1.230765E+03	1.227150E+03	8.781887E+02
	Std	9.015080E+01	1.008223E+02	1.103849E+02	8.447738E+01	3.357580E+01	3.023704E+01	5.049327E+01	4.900256E+01	<b>2.529600E+01</b>	4.610862E+01	4.974178E+01	9.237752E+01
CEC2017-F6	Avg	6.578725E+02	6.492542E+02	6.690145E+02	6.962271E+02	<b>6.255230E+02</b>	6.801234E+02	6.476803E+02	6.833563E+02	6.914501E+02	7.035159E+02	7.050999E+02	6.378202E+02
	Std	7.968435E+00	8.932575E+00	8.686637E+00	1.097260E+01	5.896974E+00	<b>4.109126E+00</b>	4.733738E+00	7.268613E+00	7.124693E+00	6.462146E+00	8.741891E+00	9.092192E+00
CEC2017-F7	Avg	1.397543E+03	1.302993E+03	1.445406E+03	1.897352E+03	<b>1.131713E+03</b>	1.912835E+03	1.443001E+03	1.846796E+03	2.133005E+03	2.125564E+03	1.316454E+03	

		Std	1.208775E+02	1.250425E+02	1.589882E+02	9.350962E+01	<b>7.146547E+01</b>	7.445519E+01	1.658267E+02	8.285966E+01	1.306112E+02	1.087097E+02	1.586652E+02	1.319069E+02
CEC2017-F8	Avg	1.165016E+03	1.147526E+03	1.269778E+03	1.389936E+03	<b>1.081474E+03</b>	1.243393E+03	1.137559E+03	1.351225E+03	1.450589E+03	1.530090E+03	1.569622E+03	1.148326E+03	
	Std	9.208152E+01	8.423117E+01	1.078834E+02	6.007365E+01	7.250701E+01	<b>2.785944E+01</b>	1.437654E+02	4.814298E+01	4.321012E+01	4.428089E+01	5.301440E+01	8.674430E+01	
CEC2017-F9	Avg	2.593396E+04	2.998162E+04	2.984360E+04	3.778499E+04	<b>1.211111E+04</b>	3.153094E+04	1.529581E+04	2.507610E+04	3.018140E+04	4.753874E+04	4.669644E+04	2.197911E+04	
	Std	7.957461E+03	6.756274E+03	6.233481E+03	1.034650E+04	4.824112E+03	<b>2.506876E+03</b>	9.407601E+03	4.050070E+03	4.817347E+03	6.767049E+03	5.253005E+03	7.239381E+03	
CEC2017-F10	Avg	1.285782E+04	1.400677E+04	1.103458E+04	1.331244E+04	<b>9.895132E+03</b>	1.033179E+04	1.012142E+04	1.197323E+04	1.430067E+04	1.642852E+04	1.593195E+04	1.404354E+04	
	Std	2.978751E+03	1.814630E+03	2.389247E+03	1.050472E+03	3.139094E+03	1.070445E+03	3.394773E+03	9.375403E+02	8.798723E+02	6.522045E+02	<b>6.044466E+02</b>	2.155817E+03	
CEC2017-F11	Avg	3.293997E+03	3.261828E+03	4.492127E+03	8.283462E+03	7.171508E+03	<b>2.966192E+03</b>	1.177624E+04	6.304360E+03	1.050782E+04	2.859214E+04	2.481707E+04	3.693370E+03	
	Std	1.477869E+03	1.041109E+03	2.955650E+03	1.988455E+03	3.473403E+03	<b>6.719990E+02</b>	1.663706E+04	1.695380E+03	2.689775E+03	6.506839E+03	5.010681E+03	1.453036E+03	
CEC2017-F12	Avg	4.894502E+08	5.614319E+08	8.509433E+08	4.009488E+09	2.119555E+09	1.151246E+09	3.803442E+08	7.488231E+09	1.563619E+10	4.568860E+10	4.219357E+10	<b>2.118630E+08</b>	
	Std	3.578929E+08	5.546367E+08	6.513925E+08	1.534295E+09	2.687348E+09	9.183436E+08	<b>2.379848E+08</b>	4.904238E+09	4.838321E+09	1.147006E+10	1.086547E+10	6.038743E+08	
CEC2017-F13	Avg	2.440915E+07	9.552554E+07	1.240359E+08	5.574850E+08	4.302511E+08	1.029958E+08	<b>3.848810E+05</b>	1.053513E+09	3.878029E+09	2.088555E+10	1.858213E+10	1.008310E+06	
	Std	4.119728E+07	1.103206E+08	1.109565E+08	3.432301E+08	1.385755E+09	2.757136E+08	<b>3.515773E+05</b>	2.022465E+09	2.170567E+09	4.886151E+09	6.777541E+09	1.362130E+06	
CEC2017-F14	Avg	3.465554E+06	3.024541E+06	3.774709E+06	6.873685E+06	1.673904E+06	6.408524E+06	1.673861E+06	3.279990E+06	2.623704E+06	4.912069E+07	4.732922E+07	<b>1.522749E+06</b>	
	Std	3.808001E+06	3.483766E+06	3.671164E+06	8.589954E+06	<b>1.226242E+06</b>	5.032920E+06	1.367843E+06	2.487031E+06	2.047105E+06	2.259577E+07	3.168424E+07	1.311110E+06	
CEC2017-F15	Avg	7.883913E+05	1.634117E+06	3.129719E+07	9.378632E+07	1.823997E+07	4.211032E+06	<b>6.145866E+04</b>	9.020849E+07	3.048364E+08	3.223547E+09	3.665918E+09	1.381485E+05	
	Std	3.698250E+06	8.240912E+06	7.055705E+07	1.885660E+08	2.507349E+07	8.435582E+06	<b>3.645815E+04</b>	1.410635E+08	1.722250E+08	1.539756E+09	2.096253E+09	2.662913E+05	
CEC2017-F16	Avg	4.633607E+03	4.372867E+03	5.008200E+03	6.512322E+03	<b>3.273824E+03</b>	4.914337E+03	4.306073E+03	5.471472E+03	5.941602E+03	8.056165E+03	7.772467E+03	4.802746E+03	
	Std	6.468739E+02	5.922992E+02	7.340971E+02	1.261429E+03	<b>4.573600E+02</b>	7.323325E+02	8.550082E+02	9.099298E+02	6.627317E+02	7.327798E+02	6.149292E+02	9.431321E+02	
CEC2017-F17	Avg	3.903572E+03	3.743295E+03	4.213643E+03	4.735557E+03	<b>3.254083E+03</b>	3.908365E+03	3.722836E+03	4.485870E+03	4.664804E+03	6.349958E+03	7.861211E+03	3.628645E+03	
	Std	4.701407E+02	<b>3.948128E+02</b>	5.302899E+02	4.947551E+02	4.996960E+02	4.684361E+02	5.323269E+02	7.607534E+02	5.786658E+02	8.085431E+02	2.896249E+03	5.011111E+02	
CEC2017-F18	Avg	1.276332E+07	6.407208E+06	1.173275E+07	6.163298E+07	1.338321E+07	1.006716E+07	1.466272E+07	1.563695E+07	2.254792E+07	1.598865E+08	9.726689E+07	<b>5.967638E+06</b>	
	Std	1.492485E+07	5.758518E+06	1.000981E+07	5.007254E+07	1.588828E+07	8.377699E+06	2.772749E+07	1.442160E+07	1.840319E+07	7.148036E+07	4.398625E+07	<b>5.120015E+06</b>	
CEC2017-F19	Avg	1.594230E+06	2.899825E+06	1.241444E+07	1.630904E+07	9.377413E+06	1.917043E+06	7.477233E+06	4.474884E+07	2.140084E+08	1.494540E+09	1.606843E+09	<b>2.414505E+04</b>	
	Std	2.536024E+06	8.885352E+06	1.394116E+07	1.596870E+07	2.768847E+07	1.692165E+06	2.165039E+07	6.109496E+07	1.189543E+08	6.239685E+08	1.061650E+09	<b>1.909890E+04</b>	
CEC2017-F20	Avg	3.775127E+03	3.581036E+03	3.600447E+03	3.954575E+03	<b>3.227337E+03</b>	3.564959E+03	3.881149E+03	3.633909E+03	3.804837E+03	4.868715E+03	4.795865E+03	3.566340E+03	

	Std	4.620757E+02	4.751163E+02	3.524353E+02	3.852998E+02	5.024418E+02	3.460775E+02	6.072093E+02	3.792896E+02	2.784959E+02	<b>1.714312E+02</b>	2.415974E+02	4.676105E+02
CEC2017-F21	Avg	2.910473E+03	2.762451E+03	2.881739E+03	3.061044E+03	<b>2.569397E+03</b>	2.939343E+03	2.602516E+03	2.854584E+03	2.979332E+03	3.123390E+03	3.074193E+03	2.582122E+03
	Std	1.050846E+02	1.031456E+02	9.642135E+01	1.218705E+02	7.857926E+01	7.764615E+01	7.145644E+01	7.355864E+01	<b>5.999725E+01</b>	6.639366E+01	6.962623E+01	9.960482E+01
CEC2017-F22	Avg	1.327667E+04	1.530788E+04	1.317712E+04	1.479347E+04	<b>9.891130E+03</b>	1.221708E+04	1.204315E+04	1.375899E+04	1.533639E+04	1.824825E+04	1.785151E+04	1.594233E+04
	Std	2.302595E+03	2.091491E+03	3.030329E+03	8.181273E+02	9.868432E+02	1.086290E+03	2.750406E+03	8.058701E+02	2.037238E+03	5.356470E+02	<b>4.886941E+02</b>	1.906809E+03
CEC2017-F23	Avg	3.568101E+03	3.450856E+03	3.530595E+03	3.890480E+03	<b>3.039882E+03</b>	4.035143E+03	3.358832E+03	3.594931E+03	3.666858E+03	4.248959E+03	4.179467E+03	3.063949E+03
	Std	2.195329E+02	1.180742E+02	1.378293E+02	1.593504E+02	<b>5.631405E+01</b>	1.839579E+02	1.290516E+02	1.165205E+02	1.619620E+02	1.290598E+02	1.800877E+02	8.841493E+01
CEC2017-F24	Avg	3.905074E+03	3.559284E+03	3.671189E+03	3.900822E+03	<b>3.207615E+03</b>	4.420266E+03	3.548248E+03	3.692159E+03	3.729231E+03	4.613612E+03	4.514784E+03	3.362631E+03
	Std	5.363721E+02	1.062964E+02	1.138960E+02	1.683454E+02	<b>7.875374E+01</b>	2.137519E+02	1.642540E+02	1.081382E+02	1.166197E+02	1.847591E+02	1.966005E+02	1.123353E+02
CEC2017-F25	Avg	3.322483E+03	3.276762E+03	4.066015E+03	5.277730E+03	3.843325E+03	3.875922E+03	3.286115E+03	5.880974E+03	7.065279E+03	1.412294E+04	1.519606E+04	<b>3.216813E+03</b>
	Std	1.472805E+02	9.553682E+01	1.849047E+03	6.598467E+02	3.565284E+02	2.217353E+02	1.554236E+02	7.514373E+02	1.047563E+03	1.772861E+03	2.091639E+03	<b>9.525426E+01</b>
CEC2017-F26	Avg	9.299048E+03	8.305697E+03	1.114040E+04	1.451950E+04	7.418237E+03	1.192847E+04	1.064748E+04	1.315744E+04	1.319862E+04	1.735846E+04	1.742380E+04	<b>6.716979E+03</b>
	Std	2.491482E+03	2.438972E+03	1.622715E+03	1.568885E+03	<b>6.357104E+02</b>	8.059702E+02	1.426744E+03	1.243250E+03	1.366381E+03	9.662958E+02	1.349272E+03	2.185826E+03
CEC2017-F27	Avg	4.023237E+03	3.937175E+03	4.051675E+03	5.094608E+03	3.708068E+03	5.219637E+03	4.167506E+03	4.172696E+03	4.350667E+03	6.561059E+03	6.223945E+03	<b>3.200012E+03</b>
	Std	2.827797E+02	2.321422E+02	3.289726E+02	6.724752E+02	1.110030E+02	6.877168E+02	2.616286E+02	3.051110E+02	3.246974E+02	3.977990E+02	4.442623E+02	<b>1.538898E-04</b>
CEC2017-F28	Avg	4.346822E+03	3.840750E+03	6.842801E+03	6.333743E+03	4.659122E+03	4.833227E+03	3.736897E+03	6.188574E+03	7.260811E+03	1.152787E+04	1.102574E+04	<b>3.298334E+03</b>
	Std	1.334561E+03	3.202887E+02	2.376666E+03	5.203596E+02	4.694919E+02	3.578577E+02	1.387064E+02	7.141934E+02	8.052978E+02	1.176439E+03	1.219088E+03	<b>4.663228E+00</b>
CEC2017-F29	Avg	6.306806E+03	5.742767E+03	6.449140E+03	1.002111E+04	<b>4.949638E+03</b>	7.367563E+03	7.051246E+03	8.202388E+03	8.316831E+03	1.889855E+04	1.725803E+04	5.124038E+03
	Std	8.551711E+02	5.627602E+02	8.753202E+02	2.701426E+03	<b>4.094906E+02</b>	1.303178E+03	8.996062E+02	1.340175E+03	1.170047E+03	1.054719E+04	6.252451E+03	6.690235E+02
CEC2017-F30	Avg	2.439763E+07	1.786071E+07	4.388381E+07	3.387172E+08	1.683015E+08	1.386770E+08	1.856340E+08	3.379200E+08	6.375205E+08	3.261749E+09	2.968821E+09	<b>2.024526E+04</b>
	Std	2.325963E+07	1.641486E+07	4.454887E+07	1.525461E+08	5.364127E+07	5.676849E+07	7.043096E+07	1.369678E+08	2.533603E+08	9.057208E+08	1.677337E+09	<b>1.891581E+04</b>

Table 6 Comparison results on CEC2017 (Dim=100).

Algorithm	Metric	QHDBO	IDBO	DBO	WOA	GWO	HHO	EVO	PO	NRBO	CPO	SWO	MSDBO
CEC2017-F1	Avg	3.894559E+10	3.623981E+10	7.668947E+10	1.134098E+11	5.575461E+10	5.003825E+10	<b>3.666708E+09</b>	1.395055E+11	1.726018E+11	2.588626E+11	2.596613E+11	3.534394E+10
	Std	1.081013E+10	9.224652E+09	6.776763E+10	1.211427E+10	1.081299E+10	7.127012E+09	<b>6.948107E+08</b>	1.503164E+10	1.566240E+10	1.737916E+10	1.737721E+10	7.489134E+09

CEC2017-F2	Avg	7.683027E+146	3.009734E+153	5.986289E+154	3.214125E+175	<b>1.499460E+140</b>	1.840543E+155	2.216409E+179	1.057428E+157	8.948116E+152	5.754057E+175	5.644259E+169	6.690653E+142
	Std	3.407566E+147	-	-	-	<b>8.212878E+140</b>	-	-	-	-	-	-	3.540718E+143
CEC2017-F3	Avg	6.905723E+05	3.740522E+05	6.821134E+05	9.173039E+05	5.405730E+05	3.577050E+05	4.138314E+07	<b>3.212858E+05</b>	4.326021E+05	7.698670E+06	1.060003E+06	5.245244E+05
	Std	2.444287E+05	2.564180E+04	2.388248E+05	1.476002E+05	8.081043E+04	8.297649E+04	1.776632E+08	<b>1.700392E+04</b>	7.458119E+04	2.413744E+07	1.108188E+06	1.030571E+05
CEC2017-F4	Avg	5.384188E+03	4.310620E+03	1.627960E+04	2.238033E+04	6.258514E+03	9.955090E+03	<b>1.823637E+03</b>	2.011845E+04	3.243816E+04	8.664199E+04	8.122191E+04	5.097597E+03
	Std	2.377822E+03	1.119625E+03	1.540398E+04	4.489868E+03	1.929458E+03	1.815630E+03	<b>2.174465E+02</b>	5.499505E+03	5.910291E+03	1.169969E+04	1.630127E+04	1.705612E+03
CEC2017-F5	Avg	1.527614E+03	1.525852E+03	1.713459E+03	1.925271E+03	<b>1.282828E+03</b>	1.667651E+03	1.354355E+03	1.842137E+03	2.003185E+03	2.189206E+03	2.192412E+03	1.598224E+03
	Std	1.032472E+02	1.574593E+02	2.125308E+02	9.633962E+01	1.480486E+02	5.319022E+01	2.230020E+02	<b>4.995088E+01</b>	5.877700E+01	6.146786E+01	7.757433E+01	2.006761E+02
CEC2017-F6	Avg	6.715596E+02	6.688379E+02	6.772766E+02	7.096604E+02	<b>6.463888E+02</b>	6.909745E+02	6.625775E+02	6.969535E+02	7.054181E+02	7.179614E+02	7.169933E+02	6.687525E+02
	Std	5.448950E+00	9.945029E+00	1.166317E+01	1.110498E+01	4.707788E+00	4.138178E+00	6.001783E+00	4.159283E+00	<b>3.971491E+00</b>	6.672388E+00	6.009172E+00	1.086179E+01
CEC2017-F7	Avg	2.925644E+03	2.893137E+03	2.951295E+03	3.820042E+03	<b>2.238505E+03</b>	3.802613E+03	2.764095E+03	3.675514E+03	3.714166E+03	4.291289E+03	4.140179E+03	2.799937E+03
	Std	2.580886E+02	2.506849E+02	2.508481E+02	1.828356E+02	1.903302E+02	1.295627E+02	2.062297E+02	<b>1.153993E+02</b>	1.914106E+02	2.790469E+02	1.528151E+02	2.049224E+02
CEC2017-F8	Avg	1.871008E+03	1.904075E+03	2.166679E+03	2.366980E+03	<b>1.548746E+03</b>	2.124621E+03	1.761192E+03	2.276531E+03	2.467865E+03	2.621456E+03	2.632341E+03	1.972765E+03
	Std	9.596977E+01	1.544053E+02	2.220117E+02	1.186234E+02	6.815880E+01	6.400868E+01	2.076934E+02	8.746236E+01	8.974131E+01	<b>6.154172E+01</b>	9.186843E+01	1.921514E+02
CEC2017-F9	Avg	7.516433E+04	7.787044E+04	7.204527E+04	7.954670E+04	4.281459E+04	6.861949E+04	<b>3.902846E+04</b>	5.938523E+04	7.107292E+04	1.054800E+05	1.012567E+05	6.929681E+04
	Std	1.331288E+04	<b>4.581810E+03</b>	1.156367E+04	1.639826E+04	1.242636E+04	5.023517E+03	2.229239E+04	5.056565E+03	5.547023E+03	9.093935E+03	9.972769E+03	1.309629E+04
CEC2017-F10	Avg	3.090916E+04	3.178671E+04	2.729195E+04	2.894034E+04	<b>1.829923E+04</b>	2.452404E+04	2.168426E+04	2.743714E+04	3.078410E+04	3.443292E+04	3.381768E+04	3.130183E+04
	Std	3.583024E+03	1.414455E+03	5.213981E+03	1.243901E+03	1.670015E+03	1.102359E+03	5.138687E+03	1.339038E+03	1.348681E+03	<b>8.373504E+02</b>	8.882550E+02	2.106124E+03
CEC2017-F11	Avg	3.041797E+05	2.320566E+05	2.403934E+05	3.072107E+05	<b>9.290577E+04</b>	1.427446E+05	3.403788E+05	1.038095E+05	1.238100E+05	3.814184E+05	2.868037E+05	2.222297E+05
	Std	1.005516E+05	5.560193E+04	7.358196E+04	9.564029E+04	<b>1.701892E+04</b>	3.703889E+04	1.716822E+05	2.094959E+04	2.328985E+04	1.260383E+05	8.583816E+04	3.265157E+04
CEC2017-F12	Avg	4.432800E+09	4.478235E+09	6.793130E+09	2.974002E+10	1.214904E+10	1.179954E+10	<b>1.691500E+09</b>	4.203125E+10	7.403439E+10	1.486182E+11	1.546611E+11	3.685781E+09
	Std	1.966953E+09	3.376485E+09	2.297893E+09	5.926989E+09	4.885879E+09	3.477497E+09	<b>8.877319E+08</b>	1.034170E+10	1.618411E+10	1.837262E+10	1.947016E+10	1.681317E+09
CEC2017-F13	Avg	7.944067E+07	1.132271E+08	3.014525E+08	2.888850E+09	1.494156E+09	3.696319E+08	<b>1.508596E+06</b>	8.555930E+09	1.499427E+10	3.358056E+10	3.387108E+10	6.783879E+07
	Std	6.477896E+07	9.261607E+07	3.675453E+08	9.080854E+08	1.430420E+09	3.065123E+08	<b>1.179198E+06</b>	3.817471E+09	3.935080E+09	5.098236E+09	7.161207E+09	4.747322E+07
CEC2017-F14	Avg	<b>5.682942E+06</b>	8.303071E+06	1.499186E+07	2.106751E+07	9.525866E+06	9.344896E+06	1.023491E+07	1.645644E+07	2.401449E+07	1.231836E+08	9.151883E+07	8.284656E+06
	Std	<b>3.086421E+06</b>	7.523431E+06	9.876538E+06	9.315999E+06	4.918184E+06	3.316577E+06	8.533801E+06	9.999891E+06	1.274054E+07	4.670323E+07	5.382531E+07	3.606853E+06

CEC2017-F15		Avg	2.206414E+07	2.887618E+07	7.871464E+07	4.521131E+08	4.363921E+08	1.503724E+07	<b>2.989466E+06</b>	2.298175E+09	4.496718E+09	1.390780E+10	1.234606E+10	6.583231E+06
		Std	3.770291E+07	5.289769E+07	8.977571E+07	3.679886E+08	9.736007E+08	7.137451E+06	1.384191E+07	1.599200E+09	1.896491E+09	2.651480E+09	5.829050E+09	<b>4.533236E+06</b>
CEC2017-F16	Avg	7.931017E+03	7.442925E+03	9.182855E+03	1.611276E+04	<b>6.756727E+03</b>	1.066860E+04	9.122028E+03	1.302296E+04	1.432673E+04	1.959634E+04	1.852928E+04	1.048729E+04	
	Std	1.064633E+03	8.756286E+02	9.524860E+02	1.857486E+03	<b>8.147878E+02</b>	1.110821E+03	1.685528E+03	2.000905E+03	1.585508E+03	1.730046E+03	2.227673E+03	2.433881E+03	
CEC2017-F17	Avg	8.482517E+03	8.814414E+03	8.939787E+03	3.970870E+04	<b>5.518380E+03</b>	8.614716E+03	6.744790E+03	4.002565E+04	6.951730E+04	1.459396E+06	8.775425E+05	7.738205E+03	
	Std	1.048704E+03	1.631889E+03	1.348739E+03	5.146605E+04	<b>6.807203E+02</b>	2.206643E+03	8.627705E+02	5.936426E+04	6.918058E+04	1.351673E+06	6.848238E+05	9.262195E+02	
CEC2017-F18	Avg	1.538393E+07	1.173666E+07	3.025811E+07	1.872253E+07	<b>8.256471E+06</b>	1.153576E+07	1.509918E+07	1.769979E+07	3.797083E+07	1.926766E+08	1.757832E+08	1.230534E+07	
	Std	1.467701E+07	1.087461E+07	1.820369E+07	7.691342E+06	<b>4.378759E+06</b>	5.773958E+06	2.388598E+07	1.054225E+07	1.943255E+07	8.898619E+07	7.276621E+07	7.066624E+06	
CEC2017-F19	Avg	4.032858E+07	3.593648E+07	9.720432E+07	5.369191E+08	4.398848E+08	3.575412E+07	1.562792E+07	1.356696E+09	4.059444E+09	1.430617E+10	1.354777E+10	<b>1.965009E+06</b>	
	Std	5.583222E+07	2.716482E+07	9.203941E+07	3.100754E+08	5.550617E+08	1.774636E+07	1.351952E+07	1.311860E+09	1.745668E+09	3.437018E+09	4.244892E+09	<b>2.841147E+06</b>	
CEC2017-F20	Avg	7.548842E+03	7.446289E+03	7.169248E+03	7.221199E+03	<b>5.770400E+03</b>	6.157816E+03	6.918263E+03	6.707881E+03	7.112276E+03	8.773671E+03	8.718392E+03	7.605297E+03	
	Std	9.537661E+02	4.876876E+02	9.419518E+02	5.324855E+02	1.280161E+03	5.254766E+02	1.216698E+03	5.218674E+02	5.411011E+02	3.718419E+02	<b>2.983519E+02</b>	4.795222E+02	
CEC2017-F21	Avg	3.959602E+03	3.857835E+03	4.040265E+03	4.513119E+03	<b>3.082394E+03</b>	4.342914E+03	3.335161E+03	4.032798E+03	4.141026E+03	4.577112E+03	4.504894E+03	3.424214E+03	
	Std	1.833397E+02	1.968080E+02	1.771942E+02	2.099177E+02	<b>8.024961E+01</b>	1.957610E+02	1.797317E+02	1.584435E+02	1.679886E+02	1.073996E+02	1.355916E+02	2.044698E+02	
CEC2017-F22	Avg	3.025375E+04	3.274750E+04	3.112439E+04	3.173541E+04	<b>2.340918E+04</b>	2.759862E+04	2.573486E+04	3.054585E+04	3.366943E+04	3.717629E+04	3.616967E+04	3.448975E+04	
	Std	4.870247E+03	3.247036E+03	4.330972E+03	1.271139E+03	4.805447E+03	1.203506E+03	5.940779E+03	1.605963E+03	9.696402E+02	<b>6.166972E+02</b>	8.817089E+02	1.404317E+03	
CEC2017-F23	Avg	4.935537E+03	4.502322E+03	4.828470E+03	5.330784E+03	<b>3.745994E+03</b>	5.868881E+03	4.403746E+03	4.947706E+03	4.935197E+03	6.617589E+03	6.360535E+03	3.988850E+03	
	Std	3.499070E+02	1.611752E+02	2.826544E+02	2.464917E+02	<b>1.500375E+02</b>	3.187878E+02	2.469738E+02	1.971356E+02	2.505530E+02	4.035752E+02	3.413086E+02	2.249940E+02	
CEC2017-F24	Avg	6.335639E+03	5.877520E+03	6.078522E+03	6.737819E+03	<b>4.535108E+03</b>	8.289124E+03	5.938197E+03	6.314653E+03	6.312244E+03	1.038620E+04	9.813162E+03	4.952981E+03	
	Std	9.785604E+02	4.299454E+02	3.846615E+02	4.884530E+02	<b>1.887465E+02</b>	6.789909E+02	3.520620E+02	4.044765E+02	3.157435E+02	7.795673E+02	5.638296E+02	2.274355E+02	
CEC2017-F25	Avg	6.347471E+03	5.945920E+03	1.072013E+04	1.113977E+04	7.086848E+03	6.736290E+03	<b>4.484536E+03</b>	1.277902E+04	1.575157E+04	2.916878E+04	2.816145E+04	6.095319E+03	
	Std	9.763935E+02	6.052039E+02	7.026187E+03	1.232748E+03	8.535528E+02	5.256265E+02	<b>1.885388E+02</b>	1.693645E+03	2.076616E+03	2.936723E+03	2.856064E+03	7.429688E+02	
CEC2017-F26	Avg	2.404800E+04	2.349863E+04	2.653905E+04	3.843389E+04	<b>1.772465E+04</b>	3.204588E+04	2.748505E+04	3.752922E+04	3.793768E+04	5.001422E+04	5.156442E+04	2.381858E+04	
	Std	3.472371E+03	3.031163E+03	3.827266E+03	3.157071E+03	1.802755E+03	<b>1.800922E+03</b>	3.130692E+03	3.128551E+03	3.076454E+03	3.756009E+03	4.483438E+03	3.288725E+03	
CEC2017-F27	Avg	4.606239E+03	4.400691E+03	4.798577E+03	6.144894E+03	4.256319E+03	7.299195E+03	5.275683E+03	5.308473E+03	6.311439E+03	1.166937E+04	1.112862E+04	<b>3.200024E+03</b>	
	Std	4.642973E+02	4.042727E+02	5.767023E+02	8.372261E+02	1.698829E+02	1.760240E+03	5.120567E+02	4.787272E+02	6.237495E+02	1.011400E+03	9.645133E+02	<b>1.785314E-04</b>	

CEC2017-F28	Avg	7.501210E+03	7.382538E+03	1.770759E+04	1.496730E+04	9.694703E+03	9.537185E+03	5.335188E+03	1.502389E+04	1.909705E+04	3.158227E+04	3.310385E+04	<b>5.164942E+03</b>
	Std	1.482477E+03	1.009232E+03	6.283932E+03	1.120385E+03	1.494425E+03	1.048869E+03	<b>4.406197E+02</b>	1.344070E+03	1.557051E+03	2.657200E+03	3.692442E+03	2.607584E+03
CEC2017-F29	Avg	1.038938E+04	9.304387E+03	1.225397E+04	2.162398E+04	9.324750E+03	1.332025E+04	1.314465E+04	1.991247E+04	2.416306E+04	2.140440E+05	2.060716E+05	<b>8.263944E+03</b>
	Std	2.151237E+03	8.513302E+02	3.058491E+03	4.566096E+03	<b>7.810177E+02</b>	1.552816E+03	2.085498E+03	4.056711E+03	7.725129E+03	1.441707E+05	2.385283E+05	1.365406E+03
CEC2017-F30	Avg	1.074404E+08	1.990229E+08	2.523369E+08	3.299251E+09	1.513202E+09	8.072209E+08	3.311681E+08	6.513336E+09	1.320001E+10	2.468105E+10	2.498641E+10	<b>6.479656E+07</b>
	Std	<b>6.235657E+07</b>	1.254730E+08	1.520445E+08	1.338053E+09	1.310060E+09	4.807608E+08	1.329038E+08	3.214654E+09	4.227885E+09	5.765800E+09	6.043809E+09	9.472246E+07

**Table 7** Comparison results on CEC2022 (Dim=10).

Algorithm	Metric	QHDBO	IDBO	DBO	WOA	GWO	HHO	EVO	PO	NRBO	CPO	SWO	MSDBO
CEC2022-F1	Avg	1.883854E+03	5.431698E+02	1.782235E+03	3.009062E+04	3.683027E+03	1.039802E+03	1.625080E+04	<b>4.073885E+02</b>	1.225572E+03	1.652116E+04	1.257442E+04	7.470457E+02
	Std	2.135386E+03	3.391811E+02	1.800075E+03	1.357139E+04	2.817465E+03	5.413206E+02	4.629169E+04	<b>5.578799E+01</b>	7.582993E+02	5.300492E+03	6.566635E+03	6.516134E+02
CEC2022-F2	Avg	4.228485E+02	4.271765E+02	4.305090E+02	4.631905E+02	4.279436E+02	4.754158E+02	4.224121E+02	4.385114E+02	4.647074E+02	6.745200E+02	6.704044E+02	<b>4.071138E+02</b>
	Std	2.829505E+01	3.045569E+01	4.290758E+01	8.143826E+01	2.308585E+01	6.014427E+01	2.920483E+01	3.072949E+01	3.493261E+01	9.731404E+01	1.745264E+02	<b>1.337409E+01</b>
CEC2022-F3	Avg	6.101025E+02	6.075532E+02	6.117215E+02	6.432218E+02	<b>6.017028E+02</b>	6.367360E+02	6.117401E+02	6.229669E+02	6.248310E+02	6.398673E+02	6.412988E+02	6.017906E+02
	Std	8.038276E+00	8.169577E+00	7.782422E+00	1.478698E+01	2.101577E+00	1.274771E+01	5.970649E+00	8.819563E+00	9.584380E+00	8.193348E+00	1.020479E+01	<b>1.606733E+00</b>
CEC2022-F4	Avg	8.278023E+02	8.237528E+02	8.303216E+02	8.414857E+02	<b>8.130897E+02</b>	8.247877E+02	8.247407E+02	8.269712E+02	8.317929E+02	8.667695E+02	8.675409E+02	8.183921E+02
	Std	9.899679E+00	9.526894E+00	1.170684E+01	1.466836E+01	<b>5.051427E+00</b>	6.575189E+00	1.159279E+01	8.222244E+00	8.590361E+00	1.082571E+01	7.763023E+00	8.133885E+00
CEC2022-F5	Avg	9.891669E+02	9.851668E+02	9.965609E+02	1.629402E+03	9.195655E+02	1.367437E+03	1.034654E+03	1.085351E+03	1.032948E+03	1.467106E+03	1.569358E+03	<b>9.078105E+02</b>
	Std	8.964213E+01	9.129483E+01	9.484810E+01	4.753263E+02	2.314540E+01	1.676400E+02	1.210312E+02	1.592943E+02	1.118490E+02	3.006512E+02	3.430684E+02	<b>1.265039E+01</b>
CEC2022-F6	Avg	<b>3.322382E+03</b>	5.357001E+03	4.583023E+03	7.825431E+03	5.813789E+03	7.954712E+03	3.690740E+03	5.038128E+03	3.930205E+03	2.425358E+07	1.748121E+07	7.216054E+03
	Std	1.915438E+03	2.540983E+03	2.319623E+03	1.202828E+04	2.074469E+03	4.975269E+03	<b>1.859237E+03</b>	2.247786E+03	1.927978E+03	2.200624E+07	1.602792E+07	8.385682E+03
CEC2022-F7	Avg	2.039787E+03	2.031168E+03	2.041150E+03	2.080551E+03	2.035004E+03	2.086457E+03	2.037937E+03	2.057110E+03	2.063841E+03	2.110452E+03	2.099189E+03	<b>2.028592E+03</b>
	Std	2.230661E+01	1.477980E+01	2.148106E+01	3.263590E+01	1.463116E+01	3.262046E+01	1.356534E+01	1.887478E+01	1.980523E+01	2.247317E+01	2.580095E+01	<b>8.632874E+00</b>
CEC2022-F8	Avg	2.229801E+03	2.224634E+03	2.235267E+03	2.235265E+03	2.233368E+03	2.238410E+03	2.238800E+03	2.231579E+03	2.239226E+03	2.250704E+03	2.253143E+03	<b>2.222510E+03</b>
	Std	2.132166E+01	5.188920E+00	3.157446E+01	8.106597E+00	3.059682E+01	1.658634E+01	3.263008E+01	<b>3.708138E+00</b>	3.212807E+01	1.610685E+01	2.481101E+01	3.970009E+00
CEC2022-F9	Avg	2.537450E+03	2.543511E+03	2.571761E+03	2.613566E+03	2.580470E+03	2.613886E+03	2.544130E+03	2.586275E+03	2.574848E+03	2.694300E+03	2.684494E+03	<b>2.485509E+03</b>

	Std	1.703921E+01	3.390670E+01	5.391092E+01	4.613421E+01	3.459335E+01	4.184159E+01	3.054847E+01	5.214855E+01	4.212420E+01	2.758723E+01	3.331247E+01	<b>2.253752E-02</b>
CEC2022-F10	Avg	<b>2.543192E+03</b>	2.547658E+03	2.558704E+03	2.665019E+03	2.570435E+03	2.598947E+03	2.563612E+03	2.545568E+03	2.590098E+03	2.555483E+03	2.576631E+03	2.543805E+03
	Std	6.155061E+01	6.252166E+01	6.753490E+01	3.081395E+02	6.798470E+01	7.216802E+01	6.476778E+01	6.214169E+01	6.905045E+01	7.143827E+01	7.583274E+01	<b>5.794793E+01</b>
CEC2022-F11	Avg	2.721043E+03	2.753604E+03	2.756553E+03	2.863675E+03	2.803459E+03	2.807383E+03	2.732774E+03	2.700764E+03	2.887414E+03	2.949744E+03	3.064032E+03	<b>2.677187E+03</b>
	Std	1.698875E+02	1.130285E+02	1.491413E+02	1.734866E+02	1.731609E+02	1.838376E+02	1.818854E+02	<b>4.174856E+01</b>	2.043833E+02	1.195172E+02	2.213451E+02	1.233615E+02
CEC2022-F12	Avg	2.873511E+03	2.879424E+03	2.874324E+03	2.907097E+03	2.868547E+03	2.942243E+03	2.876431E+03	<b>2.867277E+03</b>	2.879453E+03	2.959489E+03	2.945027E+03	2.888180E+03
	Std	1.298531E+01	1.792870E+01	1.393524E+01	3.983669E+01	8.041097E+00	6.182789E+01	1.931243E+01	<b>5.436865E+00</b>	2.780365E+01	2.918193E+01	2.834998E+01	2.180909E+01

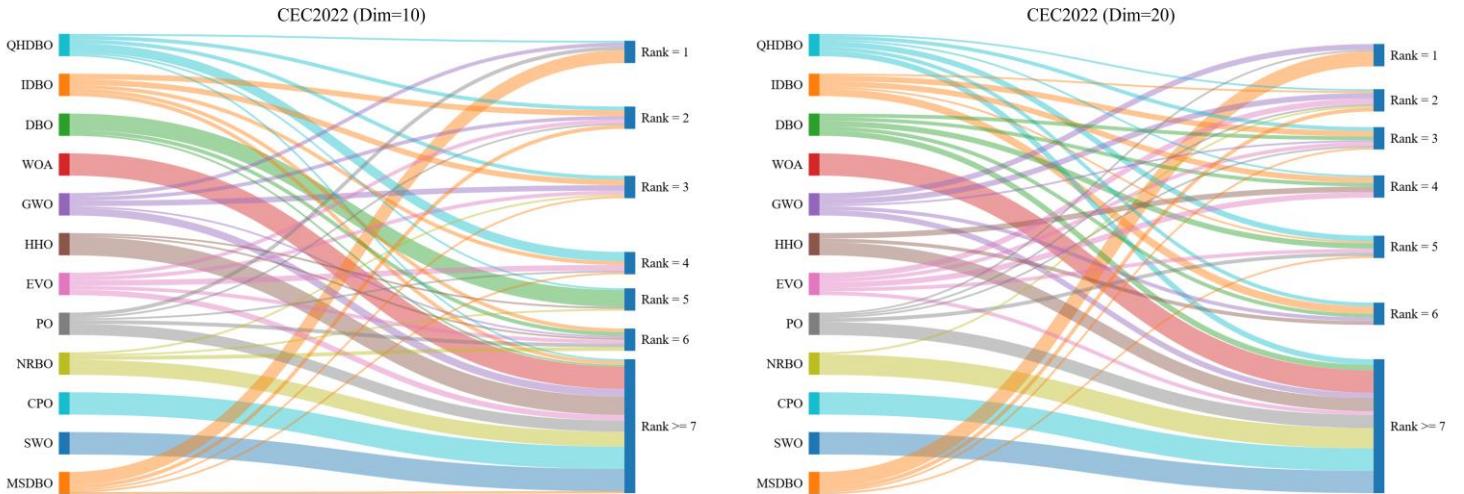
**Table 8** Comparison results on CEC2022 (Dim=20).

Algorithm	Metric	QHDBO	IDBO	DBO	WOA	GWO	HHO	EVO	PO	NRBO	CPO	SWO	MSDBO
CEC2022-F1	Avg	4.827829E+04	3.375153E+04	3.432516E+04	3.849928E+04	1.494941E+04	2.740063E+04	5.847864E+04	<b>1.347027E+04</b>	1.401057E+04	5.908760E+04	4.368295E+04	2.779020E+04
	Std	1.548016E+04	8.901759E+03	8.974579E+03	1.379301E+04	5.914193E+03	8.899790E+03	2.016609E+04	<b>4.029413E+03</b>	4.310884E+03	1.555806E+04	1.194019E+04	8.538660E+03
CEC2022-F2	Avg	4.840033E+02	4.786397E+02	5.257198E+02	5.993012E+02	5.132307E+02	5.587132E+02	4.824265E+02	6.303301E+02	7.348303E+02	1.597807E+03	1.415097E+03	<b>4.442429E+02</b>
	Std	4.929079E+01	3.703454E+01	8.840589E+01	6.965688E+01	5.592146E+01	5.469157E+01	3.325341E+01	6.908063E+01	8.624861E+01	3.194263E+02	4.311624E+02	<b>3.101813E+01</b>
CEC2022-F3	Avg	6.331485E+02	6.244001E+02	6.386832E+02	6.714866E+02	<b>6.073335E+02</b>	6.625920E+02	6.285885E+02	6.547691E+02	6.570620E+02	6.712201E+02	6.700145E+02	6.118779E+02
	Std	1.269725E+01	6.805884E+00	1.037506E+01	1.320075E+01	<b>4.564939E+00</b>	9.751414E+00	8.616707E+00	1.141355E+01	1.197697E+01	8.254040E+00	1.084805E+01	7.565003E+00
CEC2022-F4	Avg	8.893425E+02	8.923718E+02	9.070563E+02	9.405470E+02	<b>8.589071E+02</b>	8.836791E+02	8.678795E+02	9.049328E+02	9.421633E+02	9.953224E+02	9.984525E+02	8.860679E+02
	Std	2.852785E+01	3.413359E+01	3.524431E+01	3.970848E+01	3.106726E+01	1.321790E+01	3.233921E+01	2.015331E+01	2.256622E+01	<b>1.265440E+01</b>	1.852682E+01	3.577014E+01
CEC2022-F5	Avg	2.180602E+03	2.144746E+03	2.239670E+03	4.322317E+03	<b>1.360642E+03</b>	3.035397E+03	2.023811E+03	2.545022E+03	2.561634E+03	4.667903E+03	4.788095E+03	1.450134E+03
	Std	5.693582E+02	6.272261E+02	7.207274E+02	1.720365E+03	<b>3.072474E+02</b>	3.128061E+02	6.549256E+02	3.995699E+02	5.910700E+02	8.487176E+02	8.398272E+02	4.201845E+02
CEC2022-F6	Avg	2.380261E+05	7.372634E+04	7.318739E+05	5.884603E+06	6.631381E+06	2.161482E+05	5.929342E+03	2.953669E+06	2.808065E+07	7.365286E+08	6.092670E+08	<b>5.682200E+03</b>
	Std	5.169532E+05	2.832847E+05	2.352722E+06	7.057341E+06	1.456149E+07	1.025146E+05	4.630879E+03	5.753838E+06	2.562532E+07	2.667007E+08	3.957047E+08	<b>4.313981E+03</b>
CEC2022-F7	Avg	2.158823E+03	2.098223E+03	2.147610E+03	2.237576E+03	2.097986E+03	2.201860E+03	2.183621E+03	2.148690E+03	2.181047E+03	2.290507E+03	2.247363E+03	<b>2.080718E+03</b>
	Std	6.779680E+01	3.817009E+01	5.343623E+01	6.868216E+01	4.813730E+01	8.523141E+01	6.748240E+01	3.673096E+01	3.391276E+01	5.444259E+01	6.838715E+01	<b>2.629454E+01</b>
CEC2022-F8	Avg	2.320334E+03	2.289063E+03	2.336342E+03	2.295501E+03	2.272504E+03	2.302509E+03	2.295537E+03	2.278557E+03	2.294405E+03	2.448657E+03	2.439198E+03	<b>2.242973E+03</b>
	Std	9.043706E+01	6.453042E+01	8.457044E+01	7.627435E+01	5.683319E+01	9.863840E+01	7.128061E+01	4.827966E+01	6.177857E+01	9.945655E+01	1.374350E+02	<b>2.463281E+01</b>

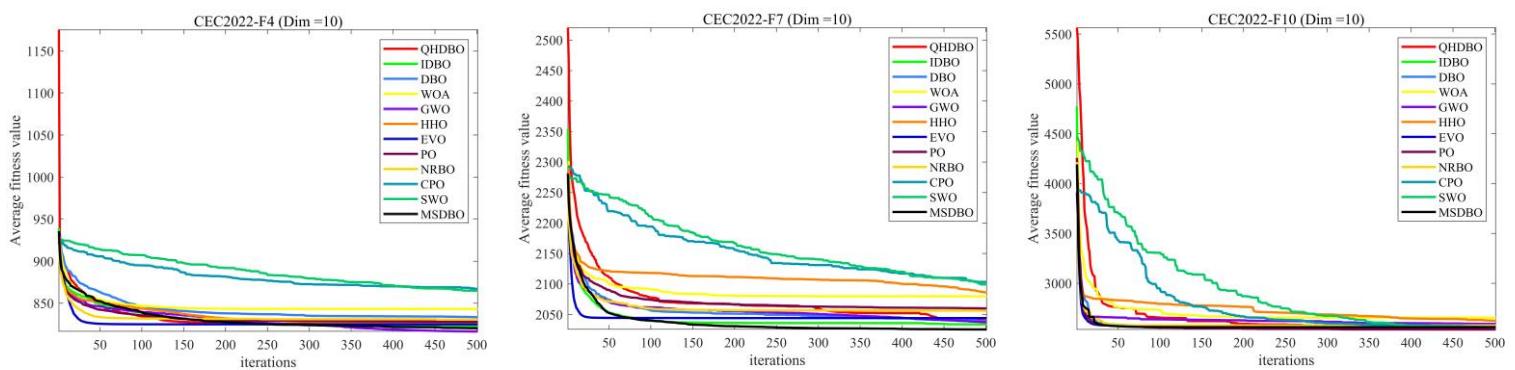
CEC2022-F9	Avg	2.501092E+03	2.510902E+03	2.509726E+03	2.600633E+03	2.523112E+03	2.538990E+03	2.513774E+03	2.584693E+03	2.615841E+03	2.854200E+03	2.850865E+03	<b>2.465852E+03</b>
	Std	3.060486E+01	2.544109E+01	3.182047E+01	5.528551E+01	3.129137E+01	4.171100E+01	3.406242E+01	5.510202E+01	5.302596E+01	1.081965E+02	1.077580E+02	<b>6.278508E-01</b>
CEC2022-F10	Avg	3.997123E+03	3.417408E+03	3.090026E+03	4.253065E+03	3.936646E+03	4.312453E+03	3.333539E+03	<b>2.589948E+03</b>	4.893923E+03	4.931534E+03	5.291008E+03	2.790013E+03
	Std	1.456791E+03	1.109031E+03	1.032974E+03	1.533739E+03	1.140350E+03	7.946109E+02	6.033408E+02	<b>4.160078E+02</b>	1.638640E+03	2.089445E+03	1.889637E+03	5.223765E+02
CEC2022-F11	Avg	3.052917E+03	3.062090E+03	3.113761E+03	4.125870E+03	3.432728E+03	3.471864E+03	2.956022E+03	4.783201E+03	4.894377E+03	6.931705E+03	7.099438E+03	<b>2.928535E+03</b>
	Std	1.533882E+02	1.797933E+02	1.698051E+02	7.662635E+02	2.482457E+02	5.588397E+02	1.066406E+02	7.683163E+02	7.241138E+02	7.242449E+02	1.067153E+03	<b>9.350363E+01</b>
CEC2022-F12	Avg	3.085643E+03	3.035358E+03	3.026036E+03	3.085246E+03	2.982198E+03	3.268707E+03	3.029162E+03	3.022118E+03	3.063627E+03	3.555303E+03	3.528874E+03	<b>2.900004E+03</b>
	Std	1.046797E+02	5.065865E+01	4.342407E+01	8.677056E+01	2.803355E+01	1.657149E+02	4.535650E+01	4.198820E+01	6.136939E+01	1.214266E+02	1.601193E+02	<b>1.711242E-04</b>

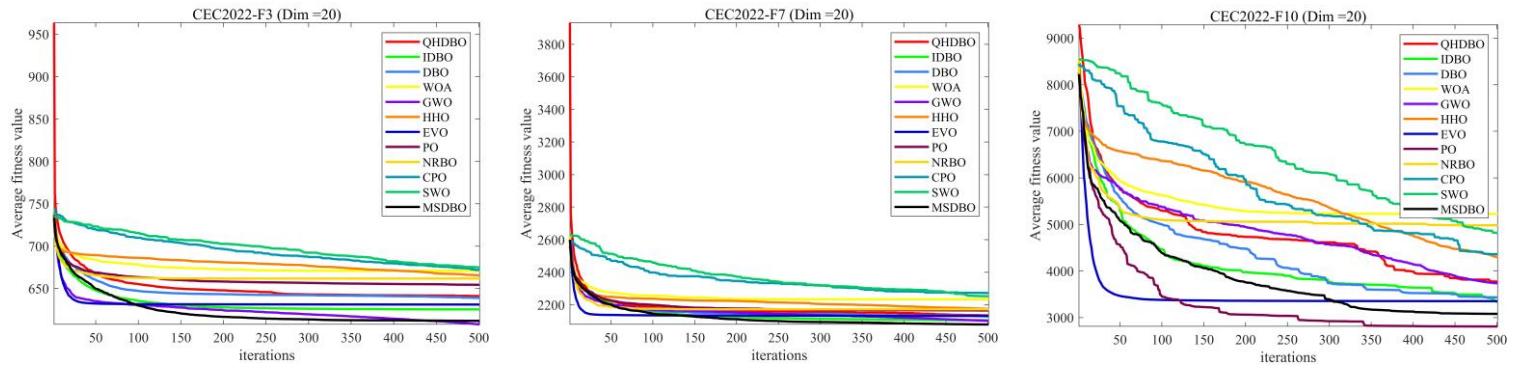
### 4.3 Comparison with other competitive algorithms on CEC2022

Here, we use CEC2022 for performance testing. The sizes are set to 10 and 20, respectively. To illustrate how each algorithm ranks, we present the Sankey diagram of the optimizers within Fig. 8, showing that MSDBO consistently ranks within the best 3 across various test functions. Table 7 and Table 8 present the experimental outcomes (including mean value (Ave) and standard deviation (Std)), with the top results from the 12 contestants highlighted in bold. It is obvious that most of the 12 benchmark functions from F1 to F12 in the three dimensions of CEC2022 proposed by MSDBO are ranked in the top three, showing excellent performance similar to that of CEC winners. In addition, it has been observed that MSDBO exhibits a lower sensitivity to size changes. Figs. 9 and 10 depict the convergence curves of different algorithms in different dimensions. To minimize possible arbitrary variations, we generate the average iteration curve for each candidate with 30 iterations. The outcomes clearly indicate the quicker convergence and higher precision achieved by the newly introduced method. Overall, MSDBO is superior to the most advanced optimizers.

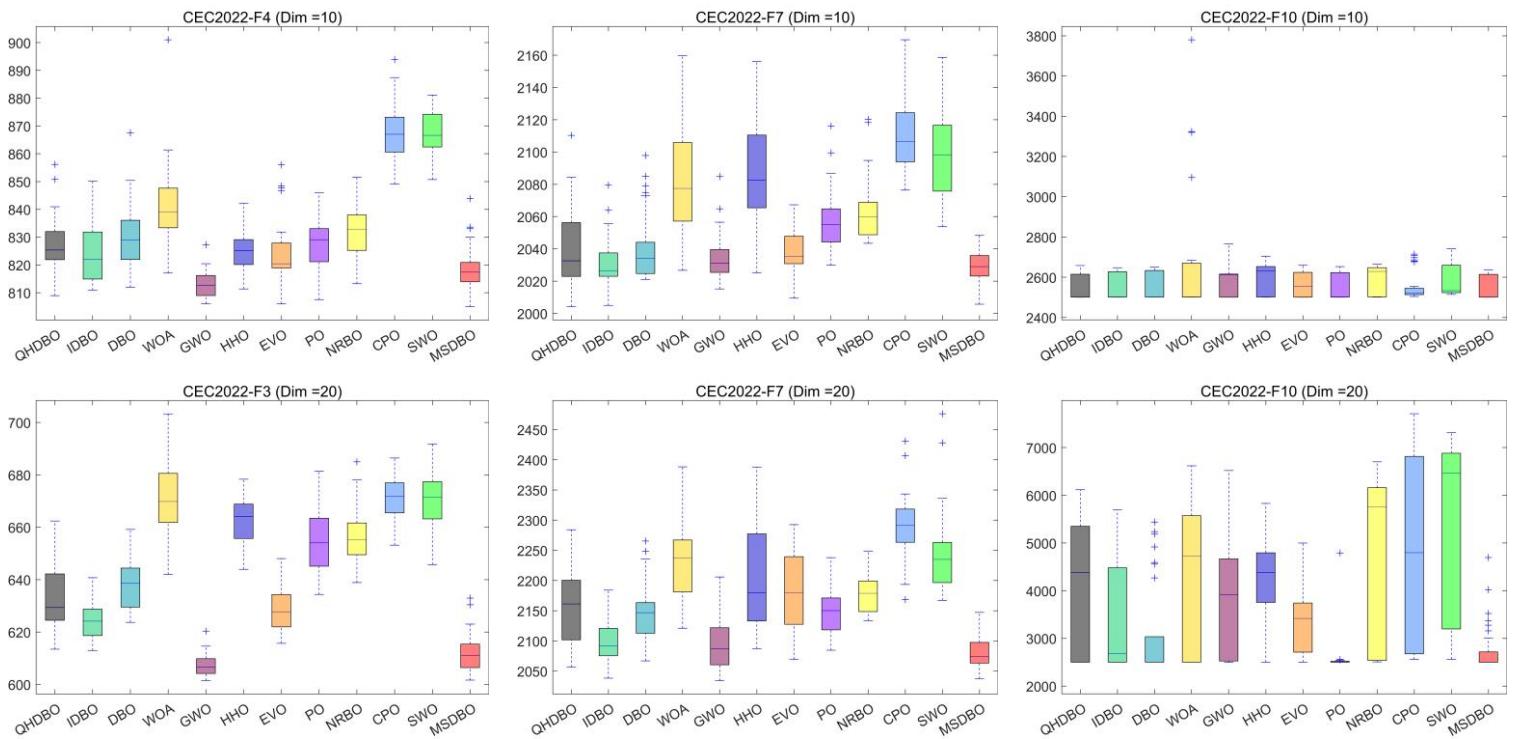


**Fig. 8.** The ranking bar chart of convergence behavior of MSDBO.





**Fig. 9.** Comparison of the convergence speed of different competitors on CEC2022.



**Fig. 10.** Comparison of the box plots of different competitors on CEC2022.

#### 4.4 Statistical analysis

##### 4.4.1 Wilcoxon signed-rank test

The Wilcoxon signed-rank test, a nonparametric method, is employed to analyze the disparities between MSDBO and its rival algorithms, with the specific findings depicted in **Table 9**. A p-value below 0.05 signifies a significant variance between MSDBO and the competing algorithm, while a lack of significant difference is accordingly noted if the p-value is above 0.05. The symbols '+/-/-' serve to denote whether MSDBO outperforms, matches, or underperforms in comparison to its competitors. The Wilcoxon signed-rank results from **Table 9** provide a clear demonstration of the superior performance of MSDBO across various dimensions of CEC2017 and CEC2022. In the 30, 50, and 100 dimensions of CEC2017, MSDBO consistently outperforms or matches its competitors, with a 30/0/0 score against algorithms like CPO and SWO across all three dimensions. This excellence continues robustly in the CEC2022, where MSDBO exhibits exceptional performance in

both 10 and 20 dimensions, maintaining dominant against most competitors and underscoring its effectiveness and efficiency in rapidly converging to optimal solutions even in less complex scenarios. Overall, the aggregated results across all tested dimensions confirm the versatility, the robust design, and the capability of MSDBO to effectively manage and excel in complex optimization environments. This makes MSDBO not only a versatile and powerful tool for evolutionary computation but also a leading choice for academic and practical applications aimed at solving a wide array of optimization problems.

**Table 9** The Wilcoxon signed-rank test statistical results on CEC2017 and CEC2022.

MSDBO VS.	CEC2017	CEC2017	CEC2017	CEC2022	CEC2022
	Dim=30	Dim=50	Dim=100	Dim=10	Dim=20
QHDBO	12/14/4	25/4/1	19/10/1	7/4/1	11/1/0
IDBO	10/15/5	21/7/2	15/14/1	7/4/1	10/2/0
DBO	24/3/3	26/3/1	24/4/2	10/1/1	11/1/0
WOA	26/1/3	29/1/0	28/0/2	11/1/0	12/0/0
GWO	11/2/17	16/4/10	12/6/12	5/5/2	6/3/3
HHO	19/6/5	28/0/2	24/3/3	12/0/0	10/2/0
EVO	10/5/15	26/3/1	15/9/6	10/1/1	10/1/1
PO	24/0/6	28/0/2	26/1/3	9/1/2	10/1/1
NRBO	24/1/5	26/3/1	27/1/2	10/1/1	11/0/1
CPO	30/0/0	30/0/0	30/0/0	12/0/0	12/0/0
SWO	30/0/0	30/0/0	30/0/0	12/0/0	12/0/0
<b>Overall (+/-)</b>	<b>220/47/63</b>	<b>285/25/20</b>	<b>250/48/32</b>	<b>105/18/9</b>	<b>115/11/6</b>

#### 4.4.2 Friedman mean rank test

This part utilizes the Friedman mean rank test to evaluate and rank the performance of MSDBO relative to competing algorithms on CEC2017 and CEC2022. Findings from this evaluation are detailed in **Table 10**. The experimental data clearly shows that MSDBO maintains a high rank throughout. This outcome underscores the exceptional efficacy of the MSDBO algorithm we proposed, as demonstrated in the test suites, against other competitors.

The test outcomes shown in **Table 10** reveal the MSDBO's excellent performance across the various dimensions of the CEC2017 and CEC2022 benchmarks. In the CEC2017 benchmark, MSDBO ranks highest in the 30 and 50 dimensions, with an average Friedman ranking of first place, and remains highly competitive in the 100 dimensions, with an average Friedman ranking of third, showcasing its robustness and adaptability. In the new CEC2022 benchmark, MSDBO continues to excel, ranking first in the Friedman average across both 10 and 20 dimensions, further demonstrating its excellence and strategic advantages in tackling dynamic and complex optimization challenges. Overall, these results demonstrate the superior design and performance of MSDBO, affirming its status as a top algorithm in evolutionary computing and highlighting its potential to efficiently and reliably solve a wide range of complex global optimization problems.

**Table 10** Friedman mean rank test statistical results on CEC2017 and CEC2022.

Dimensions	CEC2017			CEC2022	
	30	50	100	10	20

Algorithms	Average	Final								
	Rank	Rank								
QHDBO	4.8	5	4.5	5	4.7	5	3.6	2	5.4	5
IDBO	3.4	2	3.8	3	4.0	4	4.1	3	4.3	3
DBO	5.4	6	5.9	6	6.1	7	5.4	6	5.4	5
WOA	9.6	10	9.0	10	8.6	10	9.6	10	8.8	10
GWO	3.6	3	3.1	2	3.0	1	4.3	4	3.8	2
HHO	7.3	8	6.2	7	5.9	6	8.8	9	7.0	8
EVO	4.3	4	3.9	4	3.5	2	4.7	5	4.3	3
PO	6.8	7	7.2	8	7.0	8	5.4	6	6.1	7
NRBO	8.0	9	8.7	9	8.6	9	7.6	8	8.4	9
CPO	11.7	12	11.7	12	11.7	12	11.2	11	11.5	12
SWO	11.0	11	11.3	11	11.1	11	11.2	11	11.3	11
<b>MSDBO</b>	<b>1.9</b>	<b>1</b>	<b>2.7</b>	<b>1</b>	<b>3.8</b>	<b>3</b>	<b>2.2</b>	<b>1</b>	<b>1.7</b>	<b>1</b>

#### 4. 5 Impact analysis of the proposed modifications to MSDBO

This section presents an in-depth experimental study of the MSDBO algorithm, aiming to validate the effectiveness of its three improvement strategies. The experiments were first conducted on 23 benchmark functions, with the results demonstrating the performance of MSDBO across different optimization problems. Subsequently, ablation experiments were performed by sequentially removing each of the three core strategies of MSDBO to analyze their specific impact on the algorithm's performance. These experimental results provide strong evidence of the role of each strategy in the optimization process and offer valuable empirical support for further improvements and optimizations of the algorithm.

##### 4. 5. 1 The review of 23 classical benchmark functions and its experimental results

In this study, we conducted performance validation of the MSDBO algorithm on 23 benchmark functions, as shown in **Table 11**. The 23 benchmark functions cover different types of problems, including high-dimensional problems, unimodal functions, and multimodal functions. Specifically, some functions are unimodal, such as the step function (which has a single minimum and is discontinuous) and the noisy quartic function (where the noise term is a uniformly distributed random variable). Additionally, there are multimodal functions, where the number of local minima increases exponentially with the problem's dimensionality. These problems are generally the most challenging for many optimization algorithms (including MSDBO). Some of the functions are low-dimensional, having only a few local minima. For unimodal functions, the convergence rate of MSDBO is more important than the final optimization results, The visualization results of the 23 benchmark functions are shown in **Fig. 11**.

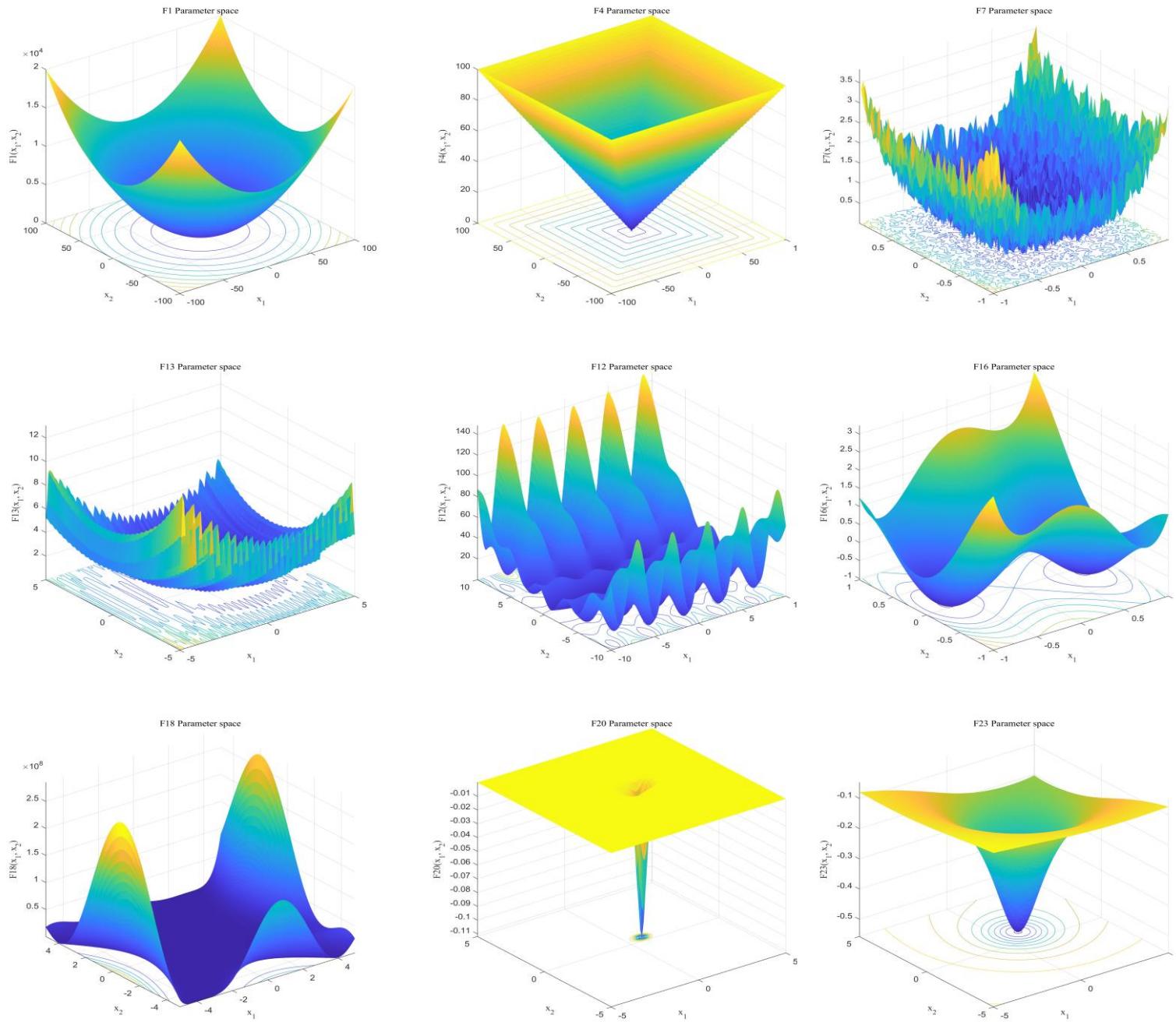
**Table 11** The description of 23 classical benchmark functions.

Test function	n	s	f <sub>min</sub>
$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100,100]^n$	0
$f_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	30	$[-10,10]^n$	0

---

$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$[-100,100]^n$	0
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	$[-100,100]^n$	0
$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30,30]^n$	0
$f_6(x) = \sum_{i=1}^n (x_i + 0.5)^2$	30	$[-100,100]^n$	0
$f_7(x) = \sum_{i=1}^n i x_i^4 + \text{random}[0,1)$	30	$[-1.28,1.28]^n$	0
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$[-500,500]^n$	-12569.5
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12,5.12]^n$	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right)$	30	$[-32,32]^n$	0
$f_{11}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^{n \prod \left(\frac{x_i}{\sqrt{i}}\right)} \cos$	30	$[-600,600]^n$	0
$f_{12}(x)$ $= \frac{\pi}{n} \left\{ \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] \right\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4)$	30	$[-50,50]^n$	0
$y_i = 1 + \frac{1}{4}(x_i + 1)$			
$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a, \\ 0, & -a \leq x_i \leq a, \\ k(-x_i - a)^m, & x_i < -a. \end{cases}$			
$f_{13}(x)$ $= 0.1 \left\{ \sin^2(3\pi x_1) + \sum_{n=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(x_n - 1)[1 + \sin^2(2\pi x_n)] + \sum_n y(x_i, 5, 100, 4)$	30	$[-50,50]^n$	0

$f_{14}(x)$				
$= \left[ \frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^2 (x_i - a_{ij})^6} \right]^{-1}$	2	$[-65.536, 65.536]^n$	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	$[-5, 5]^n$	0.00030	75
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2$ $- 4x_2^2 + 4x_2^4$	2	$[-5, 5]^n$	1.03162	85
$f_{17}(x) = \left( x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2$ $+ 10 \left( 1 - \frac{1}{8\pi} \right) \cos x_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.398	
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1 x_2 + 3x_2^2)]$ $\times [30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1 x_2 + 27x_2^2)]$	2	$[-2, 2]^n$	3	
$f_{19}(x) = - \sum_{i=1}^4 c_i \exp \left[ - \sum_{j=1}^4 a_{ij} (x_j - p_{ij})^2 \right]$	4	$[0, 1]^n$	-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	$[0, 1]^n$	-3.32	
$f_{21}(x) = - \sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10	
$f_{22}(x) = - \sum_{i=1}^7 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10	
$f_{23}(x) = - \sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10	



**Fig. 11.** Visualization of 23 Benchmark Functions.

#### 4. 5. 2 Ablation study in MSDBO

An Ablation Study is a commonly used experimental method in machine learning and optimization algorithms. The basic principle involves systematically removing or disabling certain components or strategies within a model or algorithm to observe the impact of each part on overall performance. This approach helps identify the contribution of each component or strategy to the overall algorithm performance, providing valuable insights for further optimization and improvements. The typical steps in an ablation study include:

- 1) **Baseline Model:** Initially, a full model (or algorithm) containing all components and strategies is trained and evaluated. The performance of this baseline model serves as a reference for comparing other variations.
- 2) **Step-by-Step Removal of Strategies or Components:** After establishing the baseline, the experiment proceeds by sequentially removing specific strategies or components from the model. This step is conducted one at a time to observe the effects of each removed part on performance.
- 3) **Performance Comparison:** The performance of the modified model is then compared with the baseline model. If removing a particular strategy causes a significant drop in performance, it indicates that the strategy plays a crucial role. Conversely, if the removal has little effect, the strategy may not be as important for the overall performance.

Based on the three strategies in the MSDBO algorithm and the corresponding MSDBO-1, MSDBO-2, and MSDBO-3 variants, the steps for the ablation study can be outlined as follows:

- **MSDBO-1:** Removal of the First Strategy (Sub-population Strategy): In the MSDBO-1 experiment, the sub-population strategy is removed, meaning the algorithm no longer uses multiple sub-populations to conduct optimization. Instead, a single population may be used for the search. The performance of MSDBO-1 is evaluated on the benchmark functions, and the effect of the sub-population strategy on overall optimization capability is analyzed.
- **MSDBO-2:** Removal of the Second Strategy (New Boundary Control): In MSDBO-2, the new boundary control strategy is removed, and the algorithm reverts to a more traditional boundary control method. Boundary control strategies are critical for ensuring solution feasibility and guiding the search process within valid regions.
- **MSDBO-3:** Removal of the Third Strategy (Foraging enhancement strategy): In the MSDBO-3 experiment, the foraging enhancement strategy is removed, meaning the algorithm no longer employs specific tactics to enhance the search process by improving individual movements. As a result, the population is updated without the additional refinement provided by the foraging mechanism. The performance of MSDBO-3 is evaluated on benchmark functions, and the effect of the foraging enhancement strategy on optimization performance is analyzed.

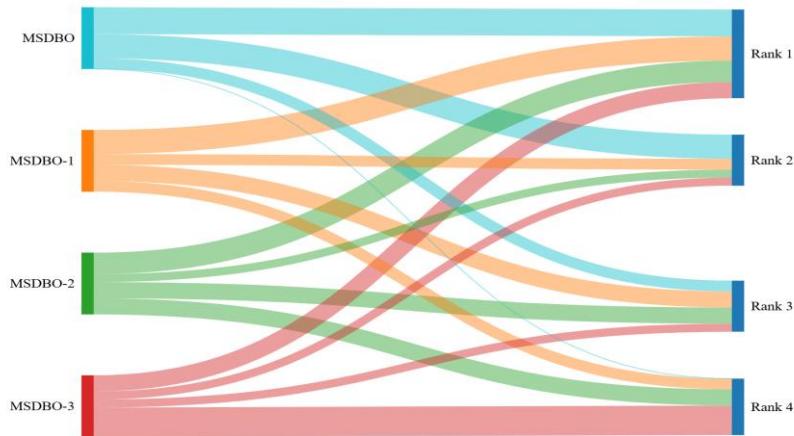
#### 4. 5. 3 Results and analysis of the ablation study on MSDBO

**Table 12** Friedman mean rank test statistical results on 23 classical benchmark functions.

Benchmark Functions	MSDBO-1	MSDBO-2	MSDBO-3	MSDBO
<b>F1</b>	2.2	2.0	4.0	1.9
<b>F2</b>	2.2	2.2	4.0	1.6
<b>F3</b>	1.8	2.1	4.0	2.1
<b>F4</b>	2.3	1.8	4.0	1.9
<b>F5</b>	2.8	2.2	2.5	2.5
<b>F6</b>	2.8	2.3	2.7	2.2
<b>F7</b>	2.2	1.9	3.8	2.1
<b>F8</b>	4.0	2.4	1.7	1.9
<b>F9</b>	2.3	2.3	3.0	2.3
<b>F10</b>	2.5	2.5	2.5	2.5
<b>F11</b>	2.4	2.4	2.8	2.4
<b>F12</b>	2.2	2.1	3.3	2.4
<b>F13</b>	2.1	3.0	2.4	2.5

<b>F14</b>	2.9	2.8	2.3	2.1
<b>F15</b>	2.0	2.6	2.9	2.6
<b>F16</b>	2.5	2.6	2.4	2.6
<b>F17</b>	2.5	2.5	2.6	2.5
<b>F18</b>	2.3	2.7	2.5	2.5
<b>F19</b>	2.5	2.5	2.7	2.3
<b>F20</b>	2.5	2.5	2.5	2.5
<b>F21</b>	2.7	3.0	2.0	2.3
<b>F22</b>	2.6	3.1	1.8	2.6
<b>F23</b>	2.4	2.9	2.3	2.4
<b>Average rank</b>	2.4616	2.4565	2.8029	2.2790
<b>Final rank</b>	3	2	4	<b>1</b>

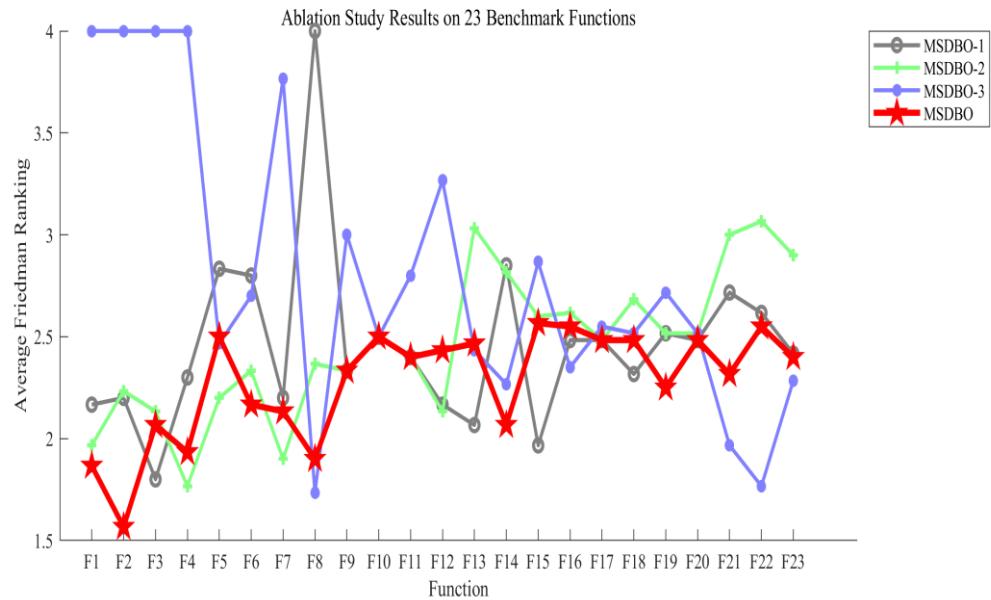
According to the results in **Table 12 and Fig. 12 (a)** from the Friedman mean rank test, the full version of the improved model, MSDBO, demonstrated the best performance with a final average rank of 2.2790, significantly outperforming the variant models with removed strategies. In comparison, the average ranks of MSDBO-1 (removal of the sub-population strategy), MSDBO-2 (removal of the new boundary control strategy), and MSDBO-3 (removal of the foraging enhancement strategy) were 2.4616, 2.4565, and 2.8029, respectively, indicating that each strategy contributes positively to optimization performance. Among them, MSDBO-2 performed relatively better, while MSDBO-3 ranked the lowest. This suggests that the foraging enhancement strategy is more effective than the new boundary control strategy and the sub-population strategy in improving global search capability and avoiding local optima, as a higher ranking indicates a greater impact of the removed strategy on overall performance. Meanwhile, the impact of the new boundary control strategy is slightly greater than that of the sub-population strategy, indicating that the new boundary control strategy plays a more significant role in optimization performance than the sub-population strategy.



**Fig. 12 (a).** The ranking chart of MSDBO and 3 strategies.

Overall, these results validate the critical role of each strategy in the overall optimization performance of the MSDBO model. **Fig. 12 (b)** shows the results of the ablation study on 23 benchmark functions (F1-F23) using the average Friedman ranking. From the chart, we can observe

that the rankings vary significantly across the different methods. MSDBO (red line with stars) demonstrates relatively stable performance across most functions, with lower rankings compared to its variants (MSDBO-1, MSDBO-2, MSDBO-3), indicating its overall superior performance. MSDBO-3 (blue line) performs poorly on certain functions (e.g., F1-F4), achieving the highest rankings, which suggests weaker robustness. MSDBO-1 (gray line) and MSDBO-2 (green line) show good performance on specific functions (e.g., F1 and F3), but their overall consistency is not as strong as MSDBO. Overall, the results suggest that the MSDBO method achieves better performance across the benchmark functions compared to its variants.

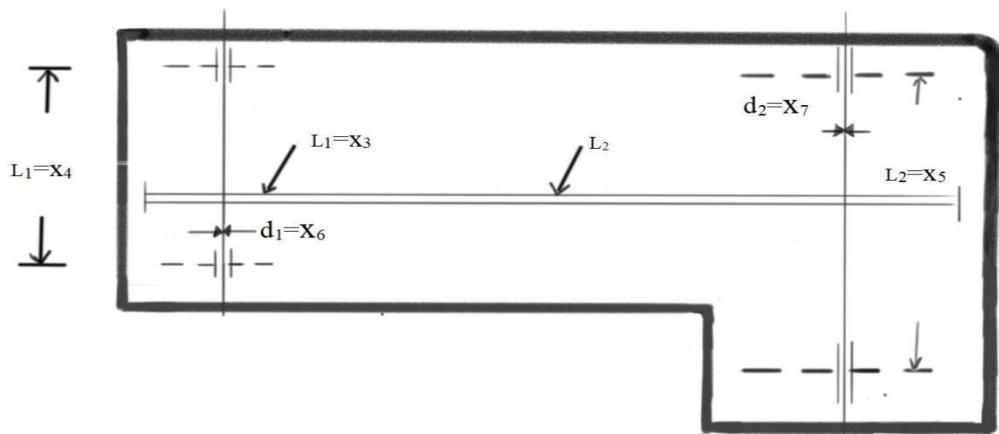


**Fig. 12 (b).** The average Friedman ranking of MSDBO and 3 strategies in 23 functions.

## 5. The application of engineering design problems

This section presents four engineering design problems that demonstrate the superior efficiency and real-world applicability of MSDBO

### 5. 1 Speed reducer design problem



**Fig. 13.** Schematic representation of the speed reducer design problem.

In mechanical design, as shown in **Fig. 13**, gearbox design is a typical optimization challenge. The key goal of this task is to minimize the gearbox's weight while meeting several constraints including gear bending stress, surface stress, shaft deflection, and shaft stress. The problem involves seven design variables, which are defined as follows:

- 1) Gear width (b) – The gear width, which directly affects the load capacity.
- 2) Gear module (m) – The gear module, determining the size and strength of the gear teeth.
- 3) Number of teeth on the pinion (p) – The number of teeth on the pinion, affecting the transmission ratio and efficiency.
- 4) Length between the first bearings (L1) – Defines the length of the first shaft.
- 5) Length between the second bearings (L2) – Defines the length of the second shaft.
- 6) Diameter of the first shaft (D1) – Affects the stiffness and strength of the shaft.
- 7) Diameter of the second shaft (D2) – Similar to the first shaft, affecting the stiffness and weight of the system.

The objective function for this optimization task is the weight of the gearbox, expressed as follows:

$$f(\mathbf{z}) = 0.7854z_1z_2^2(3.3333z_3^2 + 14.9334z_3 - 43.0934) - 1.508z_1(z_6^2 + z_7^2) + 7.4777(z_6^3 + z_7^3) + 0.7854(z_4z_6^2 + z_5z_7^2) \quad (22)$$

### Constraints:

$$g_1(\mathbf{z}) = \frac{27}{z_1z_2^2z_3} \leq 1$$

$$g_2(\mathbf{z}) = \frac{397.5}{z_1z_2^2z_3^2} \leq 1$$

$$g_3(\mathbf{z}) = \frac{1.93z_4^3}{z_2z_3z_7^4} \leq 1$$

$$g_4(\mathbf{z}) = \frac{1.93z_5^3}{z_2z_3z_6^4} \leq 1$$

$$g_5(\mathbf{z}) = \frac{\sqrt{\left(\frac{z_4}{z_2z_3}\right)^2 + 16.9 \times 10^6}}{110z_6^3} \leq 1$$

$$g_6(\mathbf{z}) = \frac{\sqrt{\left(\frac{z_5}{z_2z_3}\right)^2 + 157.5 \times 10^6}}{85z_7^3} \leq 1 \quad (23)$$

$$g_7(\mathbf{z}) = \frac{z_2z_3}{40} \leq 1$$

$$g_8(\mathbf{z}) = \frac{5z_2}{z_1} \leq 1$$

$$g_9(\mathbf{z}) = \frac{z_1}{12z_2} \leq 1$$

$$g_{10}(\mathbf{z}) = \frac{1.5z_6 + 1.9}{z_4} \leq 1$$

$$g_{11}(\mathbf{z}) = \frac{1.1z_7 + 1.9}{z_5} \leq 1$$

Bounds on design variables:

$$2.6 \leq z_1 \leq 3.6, \quad 0.7 \leq z_2 \leq 0.8, \quad 17 \leq z_3 \leq 28, \quad 7.3 \leq z_4 \leq 8.3,$$

$$7.3 \leq z_5 \leq 8.3, \quad 2.9 \leq z_6 \leq 3.9, \quad 5.0 \leq z_7 \leq 5.5$$

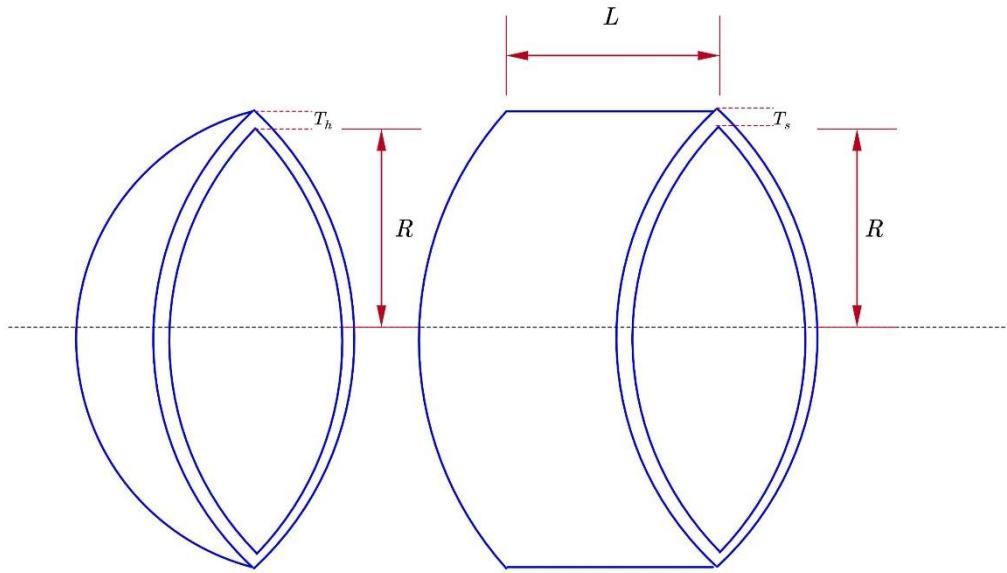
**Table 13** Results for speed reducer design problem.

Algorithm	b	m	p	L1	L2	D1	D2	Optimum cost
QHDBO	3.6000E+00	8.0000E-01	1.7000E+01	8.3000E+00	7.9500E+00	3.3505E+00	5.5000E+00	2.5000E+14
IDBO	3.5000E+00	7.0000E-01	1.7000E+01	7.3000E+00	7.7153E+00	3.3505E+00	5.2867E+00	2.9944E+03
DBO	3.5240E+00	7.0000E-01	1.7000E+01	8.0153E+00	8.3000E+00	3.3564E+00	5.2892E+00	3.0262E+03
WOA	3.5000E+00	7.0000E-01	2.7103E+01	8.2178E+00	8.2178E+00	3.8178E+00	5.4589E+00	5.4699E+03
GWO	3.5019E+00	7.0000E-01	1.7007E+01	8.1284E+00	7.8093E+00	3.3818E+00	5.2873E+00	3.0144E+03
HHO	3.5659E+00	7.0000E-01	1.7610E+01	8.2336E+00	8.2729E+00	3.3518E+00	5.2995E+00	3.1585E+03
EVO	3.5988E+00	7.1364E-01	2.6139E+01	7.3000E+00	8.0637E+00	3.3522E+00	5.2971E+00	5.2003E+03
PO	3.5102E+00	7.0000E-01	1.7000E+01	8.2997E+00	8.2283E+00	3.3568E+00	5.2870E+00	3.0204E+03
NRBO	3.5053E+00	7.0105E-01	1.7035E+01	7.3531E+00	8.3000E+00	3.4512E+00	5.2868E+00	3.0472E+03
CPO	3.5839E+00	7.1425E-01	2.7922E+01	8.0319E+00	7.8057E+00	3.3491E+00	5.3189E+00	5.7053E+03
SWO	3.5959E+00	7.0427E-01	2.1307E+01	7.9303E+00	8.1063E+00	3.8520E+00	5.3404E+00	4.1102E+03
MSDBO	5.1239E+00	6.0910E-01	1.4460E+01	6.9324E+00	7.7167E+00	3.3549E+00	5.2877E+00	<b>2.6142E+03</b>

Based on the **Table 13**, MSDBO achieved the optimal cost of 2.6142E+03 in the gearbox design optimization problem, which is the best among the algorithms presented. Compared to other algorithms like QHDBO, IDBO, and DBO, which have higher optimum costs (2.5000E+04, 2.9944E+03, and 3.0262E+03 respectively), MSDBO demonstrates superior performance in minimizing the total cost. Additionally, MSDBO also outperforms algorithms like WOA and GWO, which have optimal costs of 3.4699E+03 and 3.3144E+03, respectively.

MSDBO's success in optimizing the gearbox design is evident in its ability to balance all seven design variables (gear width, gear module, pinion teeth number, bearing lengths, and shaft diameters) while satisfying the various constraints, such as gear bending stress and shaft deflection. The optimization results show that MSDBO not only reduces the cost effectively but also provides a more efficient solution compared to the other algorithms tested.

## 5. 2 Pressure vessel design



**Fig.14.** The pressure vessel.

In the pressure vessel design problem, the primary goal is to minimize the aggregate manufacturing cost, which encompass expenses for materials, forming, and welding. As depicted in the structure shown in **Fig. 14** Four inequality constraints govern the problem, with one of them nonlinear and the other linear. The four variables are the thickness of the shell ( $x_1 = Ts$ ), the thickness of the head ( $x_2 = Th$ ), the inner radius ( $x_3 = R$ ), and the length of the cylindrical section of the vessel ( $x_4 = L$ ). The pressure vessel problem exemplifies a mixed integer problem, with the shell and the head thickness required to be integers. At the best known feasible solution, the objective function value is approximately  $f(x^*) = 6059.7143$ . The formulaic expression of this task is included below:

**Minimize:**

$$f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (24)$$

**Subject to:**

$$g_1(x) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(x) = -x_2 + 0.00954x_3 \leq 0$$

$$g_3(x) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0$$

$$g_4(x) = x_4 - 240 \leq 0 \quad (25)$$

**Where:**

$$Z_1 = 0.0625X_1, Z_2 = 0.0625X_2 \quad (26)$$

**With bound:**

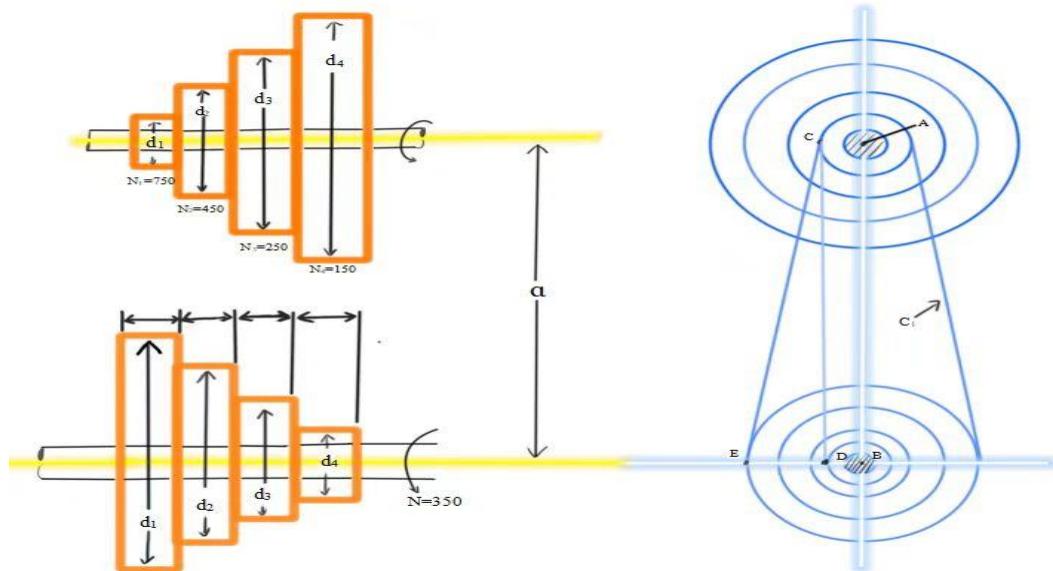
$$0 \leq x_1, x_2 \leq 99, 10 \leq x_3, x_4 \leq 200 \quad (27)$$

**Table 14** Result of pressure vessel design.

Algorithm	x1	x2	x3	x4	f(min)
QHDBO	1.7348E+01	9.3941E+00	5.5051E+01	6.2721E+01	6.7717E+03
IDBO	1.7404E+01	1.0028E+01	5.2936E+01	7.6897E+01	7.2665E+03
DBO	2.1082E+01	1.0114E+01	6.5225E+01	1.0000E+01	7.5445E+03
WOA	1.3777E+01	2.0307E+01	4.1877E+01	1.7940E+02	9.0602E+03
GWO	1.2990E+01	6.6782E+00	4.1622E+01	1.8335E+02	6.1352E+03
HHO	1.7973E+01	9.4123E+00	5.6271E+01	5.5257E+01	6.9785E+03
EVO	2.8388E+01	3.1177E+01	6.0742E+01	3.1658E+01	1.8803E+04
PO	1.8980E+01	1.0216E+01	5.9743E+01	3.5926E+01	7.3847E+03
NRBO	1.5391E+01	7.0984E+00	4.5859E+01	1.3769E+02	6.5032E+03
CPO	1.8354E+01	1.1759E+01	4.3317E+01	1.8931E+02	1.0090E+04
SWO	2.2602E+01	2.6338E+01	5.9666E+01	4.3014E+01	1.5310E+04
<b>MSDBO</b>	<b>1.4161E+01</b>	<b>7.3033E+00</b>	<b>4.5337E+01</b>	<b>1.4025E+02</b>	<b>6.0905E+03</b>

Based on the results in **Table 14**, MSDBO achieved the minimum objective function value of 6.0905E+03, showing a clear advantage. Compared to other algorithms, MSDBO yields better results in terms of the objective function value and outperforms several comparison algorithms such as QHDBO, IDBO, and DBO. Especially when compared to WOA, GWO, and HHO, MSDBO is able to approach the optimal solution more effectively, demonstrating its stronger performance in this optimization problem.

### 5. 3 Step-cone pulley problem



**Fig. 15.** Step-cone pulley problem.

The step-cone pulley optimization problem, in **Fig. 15**, focuses on minimizing the weight of a four-step cone pulley system. The challenge involves five design variables, four of which correspond to the diameters of each step, while the fifth variable represents the width of the pulley.

This problem includes a total of 11 nonlinear constraints, primarily aimed at ensuring that the transmitted power remains at 0.75 hp.

The optimization of pulley design is crucial for numerous mechanical systems, as it impacts both efficiency and weight. Lightweight pulley systems contribute to energy savings and reduced material costs, while maintaining the required mechanical performance. In this regard, applying advanced optimization techniques is essential to find the optimal balance between weight reduction and constraint satisfaction. Mathematically, it can be expressed as:

$$f(x) = \rho \cdot d_1^2 \left( 11 + \left( \frac{N_1}{N} \right)^2 \right) + d_2^2 \left( 1 + \left( \frac{N_2}{N} \right)^2 \right) + d_3^2 \left( 1 + \left( \frac{N_3}{N} \right)^2 \right) + d_4^2 \left( 1 + \left( \frac{N_4}{N} \right)^2 \right) \quad (28)$$

where:

- $\rho$  represents the material density,
- $d_i$  for  $i = 1,2,3,4$  is the diameters of the pulley steps,
- $N_i$  are the corresponding speeds at each pulley step,
- $N$  is the base reference speed.

Constraints:

$$h_1(\bar{x}) = C_1 - C_2 = 0,$$

$$h_2(\bar{x}) = C_1 - C_3 = 0,$$

$$h_3(\bar{x}) = C_1 - C_4 = 0,$$

$$g_{i=1,2,3,4}(\bar{x}) = -R_i \leq 2,$$

$$g_{i=1,2,3,4}(\bar{x}) = (0.75 \times 745.6998) - P_i \leq 0 \quad (29)$$

$$C_i = \frac{\pi d_i}{2} \left( 1 + \frac{N_i}{N} - \frac{N_i - 1}{4a} \right), \quad i = (1,2,3,4),$$

$$R_i = \exp \left( \mu \left( \pi - 2 \sin^{-1} \left( \frac{N_i}{N} - \frac{d_1}{2a} \right) \right) \right), \quad i = (1,2,3,4),$$

$$P_i = \text{std}(1 - R_i) \frac{\pi x d_i N_i}{60}, \quad i = (1,2,3,4) ,$$

$$t = 8 \text{ mm}, \quad s = 1.75 \text{ MPa}, \quad \mu = 0.35, \quad \rho = 7200 \text{ kg/m}^3, \quad a = 3 \text{ mm}$$

- $C_i$  are coefficients related to pulley design,
- $R_i$  are the radii of the pulley steps,
- $P_i$  are power constraints at each pulley step,
- $R_{\max}$  and  $P_{\max}$  are the maximum allowable values for radii and power, respectively.

**Table 15** Result of step-cone pulley problem

Algorithm	$d_1$	$d_2$	$d_3$	$d_4$	$\rho$	Optimal weight
QHDBO	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	1.2512E+21
IDBO	4.0917E+01	5.6305E+01	7.5068E+01	9.0000E+01	9.0000E+01	1.8256E+01
DBO	4.0917E+01	5.6305E+01	7.5068E+01	9.0000E+01	9.0000E+01	1.8256E+01
WOA	4.0917E+01	5.6304E+01	7.5069E+01	8.9999E+01	8.9999E+01	4.8020E+04
GWO	4.0176E+01	5.5275E+01	7.3787E+01	8.8387E+01	9.0000E+01	4.8647E+07
HHO	3.9769E+01	5.4724E+01	7.2960E+01	8.7475E+01	8.7485E+01	1.7031E+01
EVO	4.0889E+01	5.6272E+01	7.5016E+01	8.9939E+01	8.7378E+01	4.2214E+05
PO	4.0738E+01	5.6055E+01	7.4738E+01	8.9612E+01	8.9419E+01	3.5195E+05
NRBO	4.0642E+01	5.5927E+01	7.4563E+01	8.9396E+01	8.9431E+01	1.7898E+01
CPO	4.5630E+01	4.8868E+01	5.8092E+01	8.9101E+01	8.7111E+01	7.9099E+12
SWO	3.8856E+01	5.1372E+01	8.3830E+01	8.9635E+01	8.9088E+01	1.2866E+12
MSDBO	3.9085E+01	5.3783E+01	7.1705E+01	8.5972E+01	8.8450E+01	<b>1.6371E+01</b>

In **Table 15**, MSDBO achieved the best optimal weight of 1.6371E+01 in the step-cone pulley optimization problem, which is significantly better than other comparison algorithms. Although QHDBO, IDBO, and DBO achieved similar optimal weights, they still did not reach the best performance of MSDBO. Additionally, while algorithms like WOA, GWO, and HHO also showed some performance, MSDBO more effectively met the constraints and optimization goals in terms of weight minimization. Overall, MSDBO demonstrated outstanding performance in this problem, balancing weight reduction and constraint satisfaction more precisely.

## 6. Discussion of application of the real-world problem (KELM enhanced by MSDBO for predicting bankruptcy)

### 6.1. The KELM model

The KELM model is an advanced version of ELM, based on SLFNs. Unlike its predecessor, which employs a conventional method, KELM transforms the hidden layer's feature with a kernel matrix. As an innovative method, KELM has demonstrated superior generalization abilities in comparison with ELM in different real-life challenges by the usage of kernel functions. To enhance comprehension, here, we provide a brief overview of the characteristics of KELM.

ELM's fundamental principle involves discovering different optimal solutions by arbitrarily assigning input weights and hidden layer biases within a neural network, rather than fine-tuning them. With  $N$  training samples  $(x_i, t_i)$ ,  $K$  hidden neurons, and an activation function  $g(x)$ , where each input  $x_i$  is a vector with dimension  $n \times 1$ ,  $x_i \in R^n$ , and  $t_i$  is a target vector with dimension  $m \times 1$ ,  $x_i \in R^m$ , we can use a linear system for the output from ELM. We denote the formulation of SLFNs mathematically as follows:

$$O_j = \sum_{i=1}^K \beta_i g(w_i \times x_i + b_i) \quad i = 1, 2, \dots, N; j = 1, 2, \dots, N \quad (30)$$

where,  $O_j$  denotes the output vector according to the  $j^{th}$  input,  $b_i$  represents the output weight vector that connecting the  $i^{th}$  hidden layer to the output layer,  $w_i$  the weight vector connecting

the  $i^{th}$  hidden neuron to the input layer,  $w_i \times x_i$  represents the scalar product of the two vectors, and  $g(w_i \times x_i + b_i)$  denotes the  $i^{th}$  hidden neuron's activation function in that hidden layer.

The learning process of ELM focuses on minimizing training errors. If SLFNs are able to precisely fit all N training samples without any error, the following conditions hold true:  $\sum_{i=1}^N \|t_i \times o_i\| = 0$ , where  $\beta_i w_i$  and  $b_i$  exist such that:

$$t_j = o_j = \sum_{i=1}^K \beta_i g(w_i \times x_i + b_i) \quad j = 1, 2, \dots, N \quad (31)$$

$$T = h(x)\beta = H \times \beta \quad (32)$$

where  $T = [t_1, t_2, \dots, t_N]^T$  and  $b = [b_1, b_2, \dots, b_K]^T$ ,  $h(x)$  denotes the transformation function converting data from the original input space into a K-dimensional feature space:

$$H = h(x) = \begin{bmatrix} h_1(x_j) \\ \vdots \\ h_K(x_j) \end{bmatrix}^T = \begin{bmatrix} g(w_1 \times x_1 + b_1) & \cdots & g(w_K \times x_1 + b_K) \\ \vdots & \ddots & \vdots \\ g(w_1 \times x_N + b_1) & \cdots & g(w_K \times x_N + b_K) \end{bmatrix}_{N \times K} \quad (33)$$

Suppose  $H$  is the matrix representing the outputs of the hidden layer within that neural network. Every column of  $H$  represents the output of a particular hidden neuron in response to the N-dimensional input. During the training of SLFNs, there is no need to adjust the weights or biases of that hidden layer; they will be assigned randomly. Analytical methods are used to calculate the network's output  $b$ . Given this, only the output weights require modifications for the network optimization. The following formulas outline how to set such weights:

$$\beta' = H^+ \times T \quad (34)$$

where  $H^+$  stands for the MP generalized inverse of  $H$  obtained through orthogonal projection, i.e.,  $H^+ = H^T(HH^T)^{-1}$ . Utilizing the MP inverse technique guarantees the minimum norm among all possible least squares solutions. Such technique is able to significantly improve the learning rate and contribute to more robust generalization abilities.

KELM aiming to enhance ELM's generalization capability was proposed by Huang et al, outperforming the traditional ELM built upon least squares. It is advisable to include a regularization term, usually a constant  $C$ , on the diagonal of  $H^T H$  when calculating the output weights  $b$ , as shown below:

$$\beta = H^T \left( \frac{I}{C} + HH^T \right)^{-1} T \quad (35)$$

where the coefficient  $C$  represents the penalty parameter, and  $I$  stands for the identity matrix. Hence, we express the output function as follows:

$$f(x) = h(x)H^T \left( \frac{I}{C} + HH^T \right)^{-1} T \quad (36)$$

The kernel matrix of ELM is computed in this formula below:

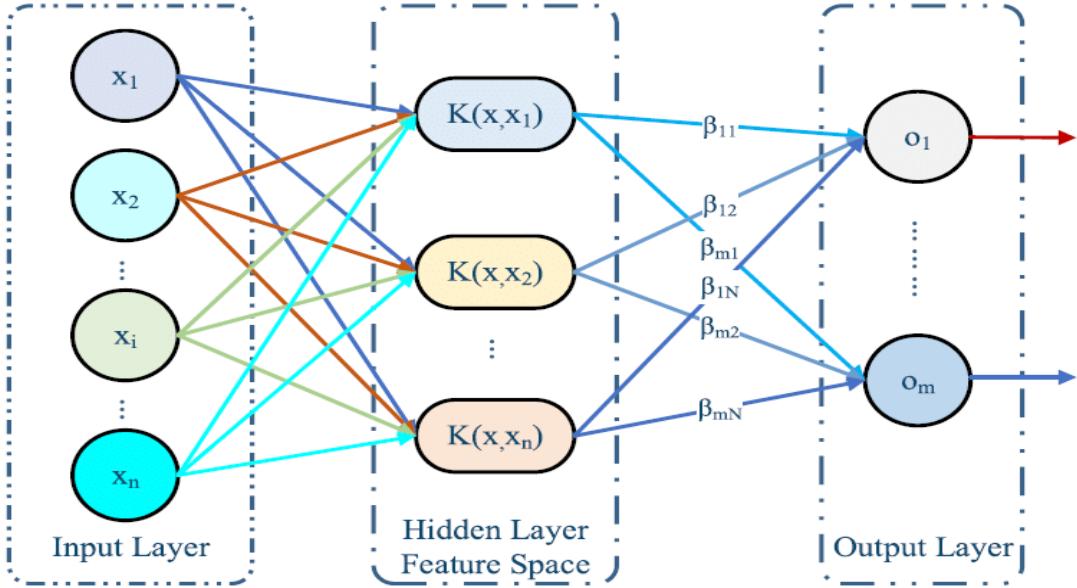
$$\Omega_{ELM} = HH^T: \Omega_{ELM_{ij}} = h(x_i)^2 = K(x_i, x_i) \quad (36)$$

where  $K(x_i, x_i)$  stands for some kernel function. The output function is denoted as:

$$f(x) = \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}^T \times \left( \frac{I}{C} + \Omega_{ELM} \right)^{-1} T \quad (37)$$

Furthermore, KELM enhances stability and generalization capabilities of the original model via the kernelization. **Fig. 16** provides a concise overview of the KELM model, where the kernel

function substitutes the conventional feature transformation function, facilitating the mapping from the input to the feature space. Consequently, the neural network's output is no longer reliant on the hidden layer's feature mapping but is directly governed by the kernel function. In addition, it is no longer necessary to predetermine the dimension of the feature space or mapping.



**Fig. 16.** Structure diagram of KELM model.

This study utilizes the Gaussian radial basis function as KELM's kernel function, denoted as:

$$K(u, v) = \exp(-\gamma \|u - v\|^2) \quad (38)$$

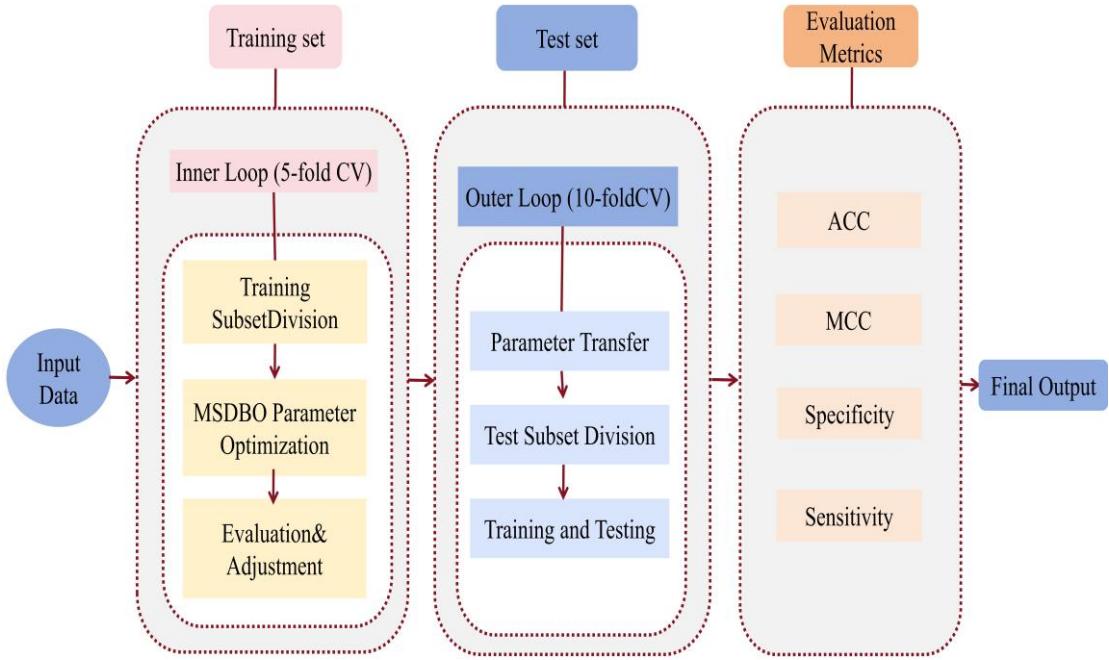
where the penalty parameter  $C$  and the kernel parameter  $\gamma$  is crucial in model definition.  $C$  balances minimizing fitting errors with controlling model complexity. At the same time,  $\gamma$  determines the nonlinear mapping from the input space to a specific high-dimensional hidden feature space. Typically, the KELM performance optimization requires choosing the aforementioned key parameters with suitable optimization methods.

## 6. 2. KELM enhanced by MSDBO

Here, we propose an improved version of KELM, named MSDBO-KELM. The model utilizes the MSDBO optimization algorithm previously discussed in this paper, which is able to dynamically compute 2 important parameters within KELM: the penalty parameter  $C$  and the kernel parameter  $\gamma$ . This model primarily involves 2 procedures. The first procedure concentrates on the inner loop parameter optimization, whereas the second procedure assesses the classification attributes in the outer loop. Throughout the process, MSDBO adaptively adjusts the KELM parameters. As soon as the optimal parameters are obtained, they are incorporated into the KELM classifier to tackle the classification task. The fitness function  $f$  is designed with a focus on classification accuracy:

$$f = avgAcc = \sum_i^K \frac{testAcc_i}{K} \quad (39)$$

where  $avgAcc$  represents the average test accuracy obtained by the KELM classifier through 5-fold CV within the parameter fine-tuning procedure.



**Fig. 17.** The general framework of the prediction model utilizing MSDBO-KELM.

The bankruptcy prediction framework built upon MSDBO-KELM is included in **Fig. 17**. In line with some other research, 10-fold CV is employed within the newly introduced prediction model to evaluate the classifier's effectiveness, seeking to achieve robust and unbiased experimental outcomes. Internally, each experiment uses 5-fold CV for fine-tuning the 2 classifier parameters. Within every experiment iteration built upon the 10-fold CV, 9 subsets are selected to create a particular sample, while the rest of the subsets serve as the test set. In addition, the fifth subset within the particular sample is assigned as the validation set, and the other subsets act as the training set. This method helps in achieving unbiased approximations of generalization accuracy and reliable outcomes. It is worth noting that we employ a stratified approach to divide the dataset, ensuring that each fold chooses its own sample based on the ratio of good to bad scores within the full dataset. Because of the inherent randomness within sampling, one 10-fold CV run may not always provide acceptable classification accuracy. Therefore, for every method being evaluated, we perform 30 runs of 10-fold CV and average the results from all runs to obtain the final outcome.

### 6. 3. Measures for performance assessment

We utilize standardized metrics to evaluate the effectiveness of the newly introduced method within multi-class classifying problems. The metrics include accuracy (ACC), Matthews correlation coefficient (MCC), sensitivity, and specificity. The 4 metrics are built upon the confusion matrix as shown below:

$$ACC = \frac{1}{n} \sum_{i=1}^n \frac{TP_i + TN_i}{TP_i + FP_i + FN_i + TN_i} \times 100\% \quad (40)$$

$$MCC = \frac{1}{n} \sum_{i=1}^n \frac{TP_i \times TN_i - FP_i \times FN_i}{\sqrt{(TP_i + FP_i) \times (TP_i + FN_i) \times (TN_i + FP_i) \times (TN_i + FN_i)}} \times 100\% \quad (41)$$

$$Sensitivity = \frac{1}{n} \sum_{i=1}^n \frac{TP_i}{TP_i + FN_i} \times 100\% \quad (42)$$

$$Specificity = \frac{1}{n} \sum_{i=1}^n \frac{TN_i}{TN_i + FP_i} \times 100\% \quad (43)$$

where  $TP_i$  stands for the true positives, the instances that are accurately categorized as "positive".  $TN_i$  represents true negatives, the instances that are accurately categorized as "negative".  $FP_i$  is the quantity of false positives, or negative cases incorrectly categorized as "positive". In a similar fashion,  $FN_i$  is the quantity of false negatives, or positive instances incorrectly categorized as "negative".  $MCC$  is a broadly accepted metric that measures the correlation coefficient between both observed and predicted values. A high  $MCC$  score indicates the model's great capability of managing imbalanced datasets. Sensitivity assesses the classifier's capability of detecting positives, while specificity evaluates its effectiveness in identifying negative cases.

## 6. 4. Experimental results and analysis in KELM

### 6. 4. 1 Analysis of four metrics (MCC, ACC, Specificity, Sensitivity)

As can be seen from **Fig. 18** and **Tables 16-18**, MSDBO showed good performance in various performance indicators of ACC, MCC, specificity and sensitivity mean. It is first noted that MSDBO shows a higher average in the ACC, which indicates that it outperforms other methods in terms of overall prediction correctness. Accuracy is crucial metric for measuring the overall efficiency of the model, and high accuracy means that the model can predict the positive and negative classes more accurately. Next, it is observed that the MCC, although MSDBO does not reach the maximum, is still high, showing a good balance between the predicted positive and negative classes. The MCC is a composite metric that takes into account the values of all four confusion matrices, so it provides a true representation of the model's performance even in very unbalanced data sets. Although MSDBO was not a leader in specificity, its results showed that the model was strong in excluding negative class samples, which was especially important for avoiding misidentification of negative class as positive class (false positive). Finally, MSDBO also performs well in the sensitivity mean index, which means that it shows high efficiency in identifying true positive samples.

From these boxplots in **Fig. 19**, the MSDBO method shows a high median and a small distribution range on all four indicators, indicating strong consistency and reliability of its results. In terms of accuracy and sensitivity, MSDBO performed particularly well, with maximum values approaching 84% and 82%. This suggests that MSDBO has advantages in maintaining high accuracy and sensitivity, while its small quartile spacing also accounts for the stability of its results. In addition, while MSDBO is not necessarily the highest in terms of MCC and specificity, its compact boxplot shows fewer outliers and fluctuations, which is very beneficial for scenarios that require stable prediction in real-world applications. Overall, MSDBO shows high stability and superiority in several key performance indicators.

**Table 16** The experimental comparison results in ACC.

Algorithm	ACC Mean	ACC STD	ACC Max	ACC Min	ACC Median
QHDBO	74.6908	0.5035	72.9011	78.3588	1.8596
IDBO	76.6902	0.5425	74.6739	80.3622	2.1452
DBO	76.0279	0.5289	74.2002	79.4277	1.4773
WOA	75.7329	0.5236	73.8185	79.3222	1.6369
GWO	76.1297	0.5312	74.3383	79.6481	1.1700
HHO	75.3888	0.5169	73.0359	79.4529	1.0850
EVO	75.9071	0.5258	73.7170	79.5583	0.9210
PO	75.9126	0.5264	74.1801	79.2493	0.9934
NRBO	75.8804	0.5267	73.7675	79.7083	1.2965
CPO	75.2047	0.5129	72.9323	79.1134	1.6952
SWO	74.4030	0.4972	72.0097	78.4412	1.6018
MSDBO	81.8786	0.6462	80.1434	85.2383	0.8896

**Table 17** The experimental comparison results in MCC.

Algorithm	MCC Mean	MCC STD	MCC Max	MCC Min	MCC Median
QHDBO	0.0393	2.2923	2.1367	77.9261	0.5643
IDBO	0.0425	2.3634	2.1088	83.5630	0.6808
DBO	0.0294	1.9127	1.3992	78.8775	0.5917
WOA	0.0331	1.9126	1.9147	78.7442	0.5800
GWO	0.0246	1.5922	1.3583	78.6674	0.5827
HHO	0.0225	1.5354	1.3053	77.1703	0.5501
EVO	0.0175	1.2178	0.9423	77.6232	0.5619
PO	0.0209	1.4719	1.4588	77.5000	0.5588
NRBO	0.0264	1.6280	1.5482	77.8674	0.5637
CPO	0.0346	1.9761	1.8519	78.0275	0.5646
SWO	0.0319	1.9559	1.6303	77.4551	0.5611
MSDBO	0.0170	1.3259	1.3803	83.4138	0.6727

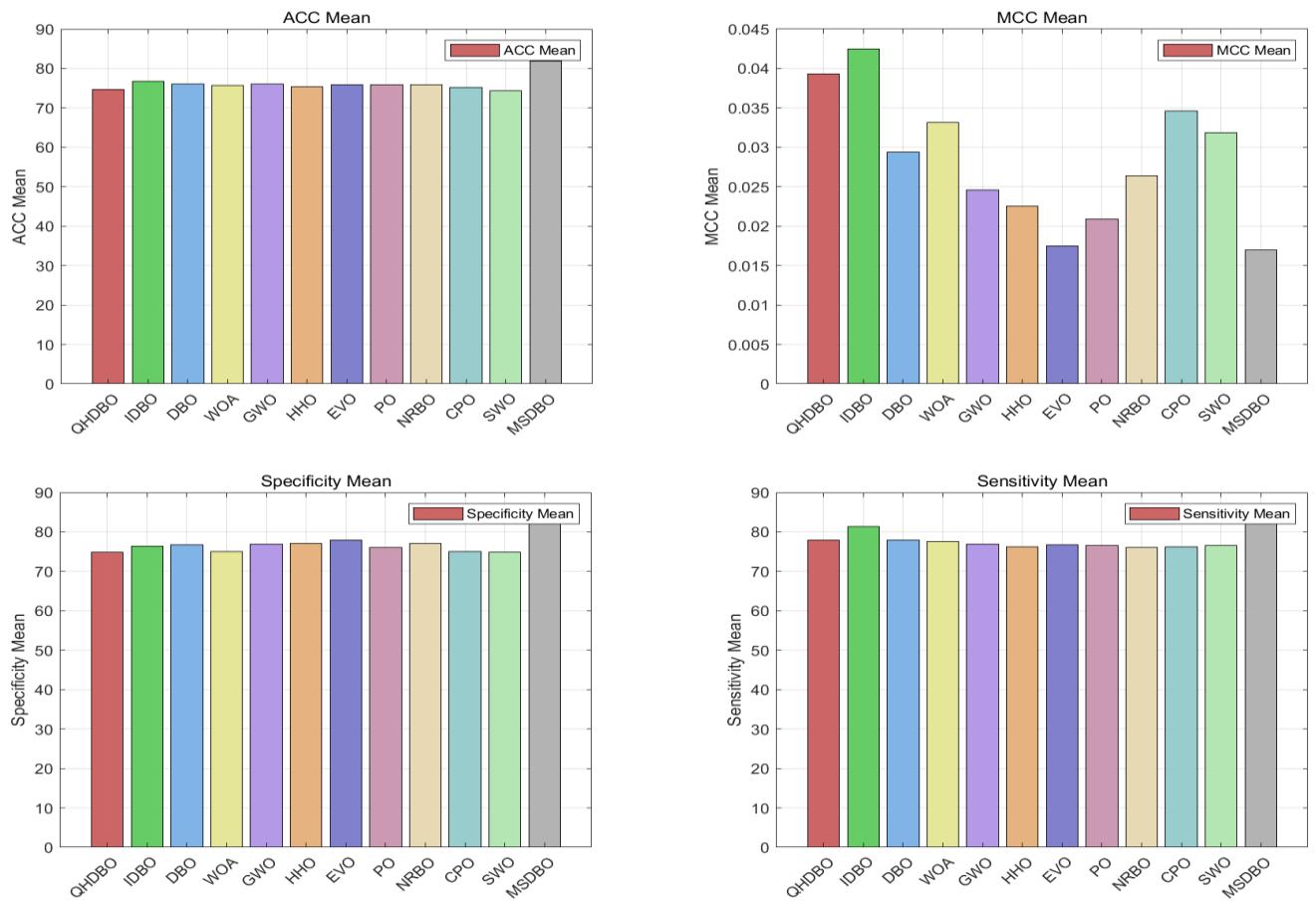
**Table 18** The experimental comparison results in Sensitivity.

Algorithm	Sensitivity	Sensitivity	Sensitivity	Sensitivity	Sensitivity
	Mean	STD	Max	Min	Median
QHDBO	77.8585	82.0495	72.0145	0.4435	69.1102
IDBO	81.2986	87.4376	73.2978	0.4736	71.0117
DBO	78.0179	82.5180	73.3174	0.4785	70.4621
WOA	77.5834	82.4271	72.1174	0.4493	69.1982
GWO	76.9608	82.6849	72.9928	0.4685	70.8580
HHO	76.2138	81.6097	73.3159	0.4749	70.2812
EVO	76.6997	81.4889	74.5609	0.4976	71.7672
PO	76.5184	81.4963	73.8036	0.4833	71.0108
NRBO	75.9912	84.4668	72.4899	0.4617	70.4309
CPO	76.2851	82.4576	71.1341	0.4261	67.9031

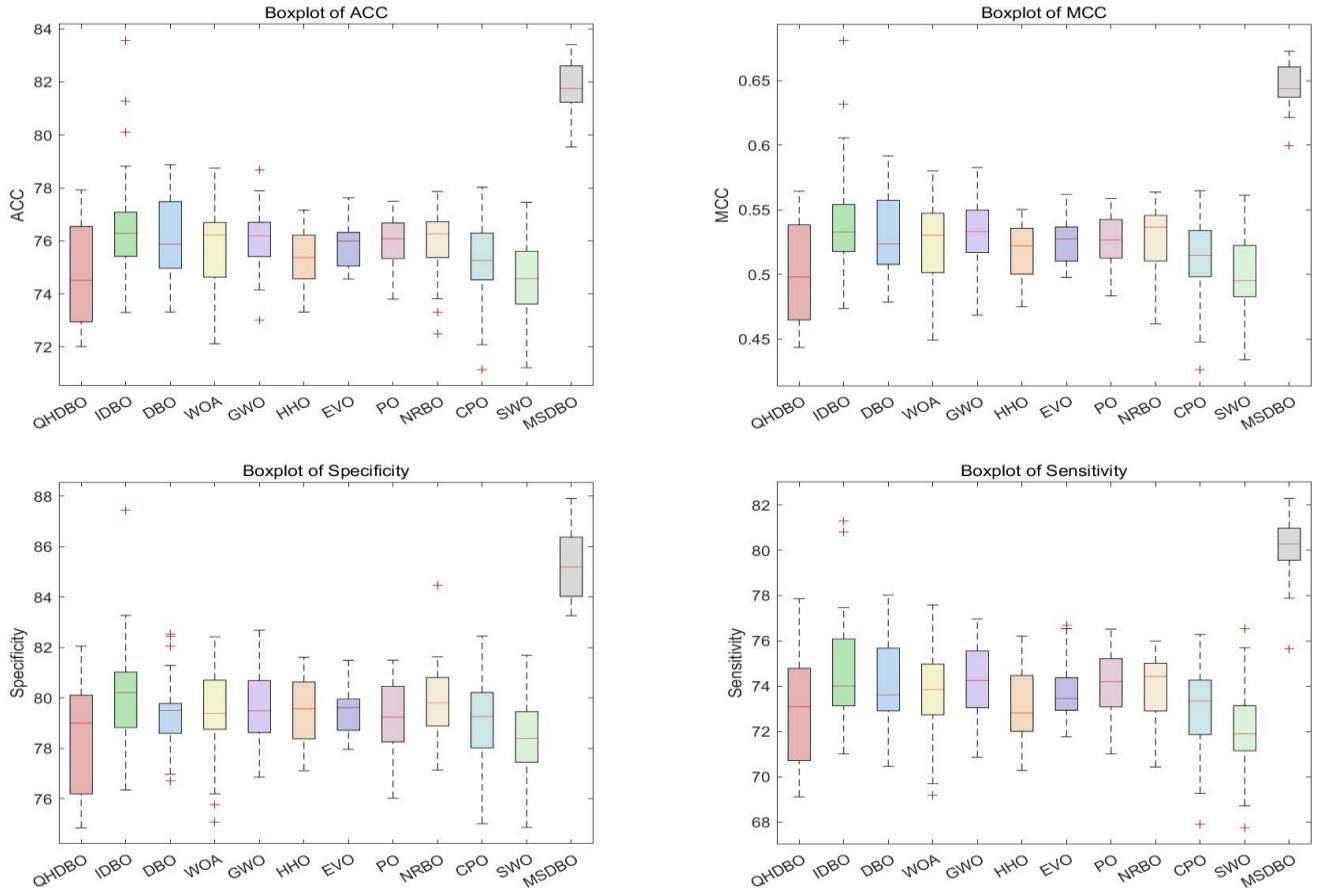
SWO	76.5427	81.6882	71.2130	0.4339	67.7456
MSDBO	82.2782	87.9053	79.5442	0.5997	75.6267

**Table 16** The experimental comparison results in Specificity.

Algorithm	Specificity	Specificity	Specificity	Specificity	Specificity
	Mean	STD	Max	Min	Median
QHDBO	74.8485	74.5174	0.4980	73.0929	79.0056
IDBO	76.3482	76.2960	0.5327	74.0073	80.2113
DBO	76.7061	75.8779	0.5235	73.6138	79.4982
WOA	75.0804	76.2301	0.5304	73.8465	79.3872
GWO	76.8571	76.2014	0.5331	74.2445	79.4886
HHO	77.1143	75.3688	0.5220	72.8135	79.5735
EVO	77.9569	75.9960	0.5273	73.4484	79.6134
PO	76.0235	76.0786	0.5267	74.2057	79.2380
NRBO	77.1413	76.2656	0.5365	74.4234	79.8028
CPO	75.0164	75.2634	0.5147	73.3358	79.2553
SWO	74.8675	74.5790	0.4952	71.8984	78.3978
MSDBO	83.2642	81.7518	0.6436	80.2679	85.1891



**Fig. 18.** The bar chart of result of mean in ACC, MCC, sensitivity and specificity.

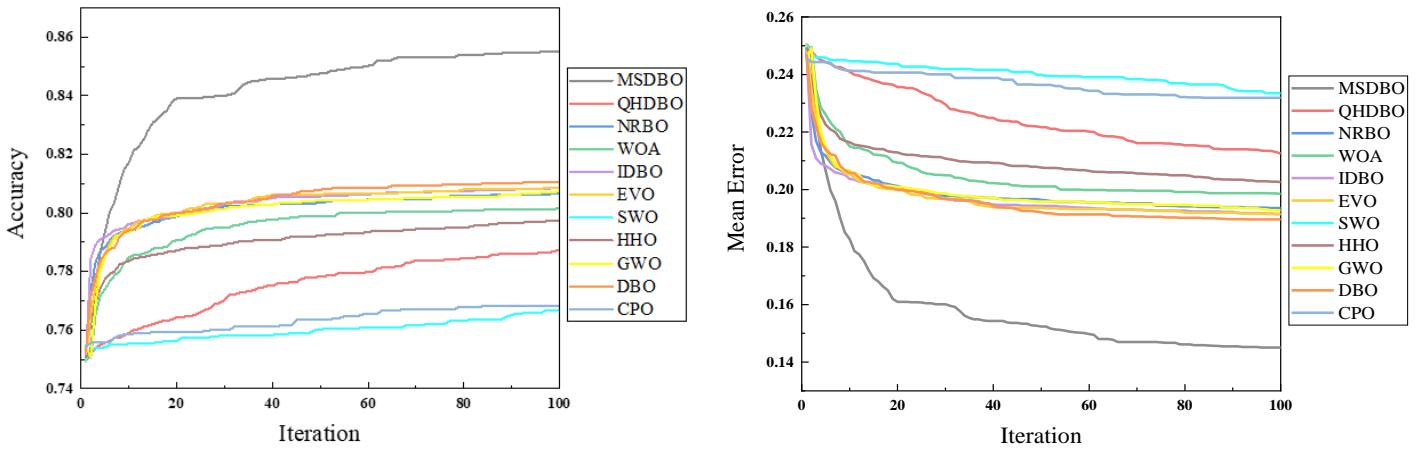


**Fig. 19.** The boxplot of result in ACC, MCC, sensitivity and specificity.

#### 6.4.2 Analysis of convergence curve in KELM

**Fig. 20** shows the average accuracy and average error changes of different methods during iteration. The MSDBO method quickly rose to the highest level in terms of accuracy and remained above 0.85, showing significant performance advantages. In terms of error, the average error of MSDBO also showed a rapid decline and tended to stabilize, and the final error was close to 0.14, which was much lower than other methods. These curves reflect that MSDBO not only rapidly converges to high accuracy, but also maintains a low error level, indicating that its optimization algorithm excels in both efficiency and stability. The high efficiency and stability make MSDBO especially useful in applications where rapid and accurate convergence is required.

Overall, MSDBO shows the strength of its overall performance with these results, especially its ability to balance other key performance indicators while maintaining high accuracy. This balanced performance makes MSDBO ideal for high-precision applications, and its improvements significantly improve the classification efficiency of the model.



**Fig. 20.** The average iteration curves.

## 7. Conclusion and prospect

The global financial crisis has heightened the focus on corporate bankruptcy prediction, highlighting the importance of accurate early warning systems that classify companies as solvent or insolvent. Modern machine learning algorithms generally outperform traditional statistical models, and researchers are increasingly interested in optimizing these algorithms for better performance. In this paper, we introduce the DBO algorithms and its three improved strategies (adaptive dung beetle demarcation strategy, optimal boundary control strategy, and foraging enhancement strategy) to solve engineering design problems and bankruptcy prediction problems. The search performance is enhanced by having an adaptive number of search agents, an adaptive weight variation, and inclusion of a sub-optimal solution. The paper also introduces the enhanced version of KELM model based on MSDBO, named MSDBO-KELM and compared it with other variations of KELM. We also review the performance of the newly introduced MSDBO-KELM. It performs significantly better regarding ACC, MCC, sensitivity and specificity. Also, it can reach a low fitness value within a few iterations. These performances can be seen as a strong indicator of availability of the purposed MSDBO-KELM in bankruptcy prediction problems. The purposed new algorithm can be considered as a reliable warning predictor, helping financial institutions and regulators make accurate decisions.

In the experimental sections, to evaluate the excellence of MSDBO in searching the global optimal values of a particular function, we compare MSDBO with many state-of-the-art competitors using both CEC2017 and CEC2022. The convergence behavior analysis showcases the excellent exploration and exploitation abilities of MSDBO, together with its superior stability. The quantitative analysis part demonstrates its constant top-ranking performance, fast convergence speed, and low variance. It also emphasizes the superiority of MSDBO to the original DBO in both exploration and exploitation. The statistical analysis part utilizes both Wilcoxon rank sum test and Friedman mean rank test to demonstrate how MSDBO outperforms the other competitors.

In further research, MSDBO-KELM can be utilized to solve more real-world classification tasks. The newly introduced improvements can be integrated with other feature selection methods to reduce computational resource requirement and improve both accuracy and time for the classification task. We also intend to apply the newly proposed enhanced method to more complex optimization tasks.

## Data Availability Statement

For results generated in this study can be obtained from the corresponding author on request.

### Competing Interest Statement

The authors declare that there is no conflict of interest regarding the publication of the article.

### Reference

- Abdel-Basset, M., Mohamed, R., & Abouhawwash, M. (2024). Crested Porcupine Optimizer: A new nature-inspired metaheuristic. *Knowledge-Based Systems*, 284, 111257.
- Abu Al-Haija, Q., Altamimi, S., & AlWadi, M. (2024). Analysis of Extreme Learning Machines (ELMs) for intelligent intrusion detection systems: A survey. *Expert Systems with Applications*, 253, 124317.
- Afzal, A. L., Nair, N. K., & Asharaf, S. (2021). Deep kernel learning in extreme learning machines. *Pattern Analysis and Applications*, 24, 11-19.
- Anandhakumar, C., Sakthivel Murugan, N. S., & Kumaresan, K. (2024). Extreme learning machine model with honey badger algorithm based state-of-charge estimation of lithium-ion battery. *Expert Systems with Applications*, 238, 121609.
- Antunes, F., Ribeiro, B., & Pereira, F. (2017). Probabilistic modeling and visualization for bankruptcy prediction. *Applied Soft Computing*, 60, 831-843.
- Azizi, M., Aickelin, U., A. Khorshidi, H., & Baghalzadeh Shishehgarkhaneh, M. (2023). Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. *Scientific Reports*, 13, 226.
- Barboza, F., Kimura, H., & Altman, E. (2017). Machine learning models and bankruptcy prediction. *Expert Systems with Applications*, 83, 405-417.
- Chen, H., Ahmadianfar, I., Liang, G., & Heidari, A. A. (2024). Robust kernel extreme learning machines with weighted mean of vectors and variational mode decomposition for forecasting total dissolved solids. *Engineering Applications of Artificial Intelligence*, 133, 108587.
- Chen, T.-K., Liao, H.-H., Chen, G.-D., Kang, W.-H., & Lin, Y.-C. (2023). Bankruptcy prediction using machine learning models with the text-based communicative value of annual reports. *Expert Systems with Applications*, 233, 120714.
- Chen, Z., Chen, W., & Shi, Y. (2020). Ensemble learning with label proportions for bankruptcy prediction. *Expert Systems with Applications*, 146, 113155.
- Citterio, A. (2024). Bank failure prediction models: Review and outlook. *Socio-Economic Planning Sciences*, 92, 101818.
- De Falco, D. E., Calabò, F., & Pragliola, M. (2024). Insights on the different convergences in Extreme Learning Machine. *Neurocomputing*, 599, 128061.
- Gao, Y., Luo, X., Gao, X., Yan, W., Pan, X., & Fu, X. (2024). Semantic segmentation of remote sensing images based on multiscale features and global information modeling. *Expert Systems with Applications*, 249, 123616.
- Guliyev, N. J., & Ismailov, V. E. (2018). On the approximation by single hidden layer feedforward neural networks with fixed weights. *Neural Networks*, 98, 296-304.
- Heidari, A. A., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, 97 <https://aliasgharheidari.com/HHO.html>, 849-872.

- Huang, G.-B., Zhou, H., Ding, X., & Zhang, R. (2012). Extreme Learning Machine for Regression and Multiclass Classification. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 42, 513-529.
- Jace, K., Koumanakos, D., & Tsagkanos, A. (2022). Bankruptcy Prediction in Social Enterprises. *Journal of Social Entrepreneurship*, 13, 205-220.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks* (Vol. 4, pp. 1942-1948 vol.1944).
- Kim, M. (2021). The generalized extreme learning machines: Tuning hyperparameters and limiting approach for the Moore–Penrose generalized inverse. *Neural Networks*, 144, 591-602.
- Kovačić, Đ., Radočaj, D., & Jurišić, M. (2024). Ensemble machine learning prediction of anaerobic co-digestion of manure and thermally pretreated harvest residues. *Bioresource Technology*, 402, 130793.
- Li, L., & Faff, R. (2019). Predicting corporate bankruptcy: What matters? *International Review of Economics & Finance*, 62, 1-19.
- Li, N., He, F., Ma, W., Wang, R., & Zhang, X. (2020). Wind Power Prediction of Kernel Extreme Learning Machine Based on Differential Evolution Algorithm and Cross Validation Algorithm. *IEEE Access*, 8, 68874-68882.
- Liu, X., Li, G., Yang, H., Zhang, N., Wang, L., & Shao, P. (2023). Agricultural UAV trajectory planning by incorporating multi-mechanism improved grey wolf optimization algorithm. *Expert Systems with Applications*, 233, 120946.
- Lohmann, C., & Möllenhoff, S. (2023). The bankruptcy risk matrix as a tool for interpreting the outcome of bankruptcy prediction models. *Finance Research Letters*, 55, 103851.
- Majumder, I., Dash, P. K., & Bisoi, R. (2018). Variational mode decomposition based low rank robust kernel extreme learning machine for solar irradiation forecasting. *Energy Conversion and Management*, 171, 787-806.
- Manthoulis, G., Doumpos, M., Zopounidis, C., & Galarotis, E. (2020). An ordinal classification framework for bank failure prediction: Methodology and empirical evidence for US banks. *European Journal of Operational Research*, 282, 786-801.
- Maria Jesi, P., & Antony Asir Daniel, V. (2024). Differential CNN and KELM integration for accurate liver cancer detection. *Biomedical Signal Processing and Control*, 95, 106419.
- Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95, 51-67.
- Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46-61.
- Pan, M., Su, T., Liang, K., Liang, L., & Yang, Q. (2024). Sensorless force estimation of teleoperation system based on multilayer depth Extreme Learning Machine. *Applied Soft Computing*, 157, 111494.
- Peng, Y., & Unluer, C. (2024). Interpretable machine learning-based analysis of hydration and carbonation of carbonated reactive magnesia cement mixes. *Journal of Cleaner Production*, 434, 140054.
- Quan, R., Liang, W., Wang, J., Li, X., & Chang, Y. (2024). An enhanced fault diagnosis method for fuel cell system using a kernel extreme learning machine optimized with improved sparrow search algorithm. *International Journal of Hydrogen Energy*, 50, 1184-1196.

- Son, H., Hyun, C., Phan, D., & Hwang, H. J. (2019). Data analytic approach for bankruptcy prediction. *Expert Systems with Applications*, 138, 112816.
- Sowmya, R., Premkumar, M., & Jangir, P. (2024). Newton-Raphson-based optimizer: A new population-based metaheuristic algorithm for continuous optimization problems. *Engineering Applications of Artificial Intelligence*, 128, 107532.
- Sun, H., Jiang, H., Gu, Z., Li, H., Wang, T., Rao, W., Wang, Y., Pei, L., Yuan, C., & Chen, L. (2024). A novel multiple kernel extreme learning machine model for remaining useful life prediction of lithium-ion batteries. *Journal of Power Sources*, 613, 234912.
- Suthaharan, S. (2016). Support Vector Machine. In S. Suthaharan (Ed.), *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning* (pp. 207-235). Boston, MA: Springer US.
- Veganzones, D., & Séverin, E. (2018). An investigation of bankruptcy prediction in imbalanced datasets. *Decision Support Systems*, 112, 111-124.
- Wang, Z., Huang, L., Yang, S., Li, D., He, D., & Chan, S. (2023). A quasi-oppositional learning of updating quantum state and Q-learning based on the dung beetle algorithm for global optimization. *Alexandria Engineering Journal*, 81, 469-488.
- Xue, J., & Shen, B. (2023). Dung beetle optimizer: a new meta-heuristic algorithm for global optimization. *The Journal of Supercomputing*, 79, 7305-7336.
- Zhang, Y., Ma, H., Wang, S., Li, S., & Guo, R. (2023). Indirect prediction of remaining useful life for lithium-ion batteries based on improved multiple kernel extreme learning machine. *Journal of Energy Storage*, 64, 107181.
- Zhang, Z., Zhang, L., Liu, D., Sun, N., Li, M., Faiz, M. A., Li, T., Cui, S., & Khan, M. I. (2024). Measurement and analysis of regional water-energy-food nexus resilience with an improved hybrid kernel extreme learning machine model based on a dung beetle optimization algorithm. *Agricultural Systems*, 218, 103966.
- Zheng, Y., Chen, B., Wang, S., Wang, W., & Qin, W. (2022). Mixture Correntropy-Based Kernel Extreme Learning Machines. *IEEE Transactions on Neural Networks and Learning Systems*, 33, 811-825.
- Zhou, T., Wei, Y., Jie, Y., & Zhang, Y. (2024). Prediction intervals for concrete face sandy gravel dam settlement using Kalman filter-based kernel extreme learning machine. *Measurement*, 236, 115094.
- Zhu, F., Li, G., Tang, H., Li, Y., Lv, X., & Wang, X. (2024). Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems. *Expert Systems with Applications*, 236, 121219.