Electronic Research Archive

ERA, 33(4): 2618–2667.
DOI: 10.3934/era.2025117
Received: 07 February 2025
Revised: 08 April 2025
Accepted: 14 April 2025
Published: 28 April 2025

https://www.aimspress.com/journal/ERA

Research article

Enhanced dung beetle optimization algorithm and its application in 3D UAV path planning

Kaike Tu[1,2] and Jiatang Cheng[1,2,*]

[1] Key Laboratory of Advanced Manufacturing and Automation Technology, Guilin University of Technology, Education Department of Guangxi Zhuang Autonomous Region, Guilin 541006, China
[2] College of Mechanical and Control Engineering, Guilin University of Technology, Guilin 541006, China

* **Correspondence:** Email: chjt@163.com.

Abstract: Dung beetle optimization (DBO) algorithm is a population-based metaheuristic that excels in optimization tasks, demonstrating strong performance in finding high-quality solutions. However, when addressing complex optimization problems, the algorithm also suffers from issues such as low population diversity, a tendency to fall into local optima, and suboptimal convergence speed. To address these challenges, this paper proposed an enhanced dung beetle optimization (EDBO) algorithm. The core improvements included the use of the Sobol sequence for better population initialization and the introduction of a mutation mechanism based on probability ratios and random angles to enhance the global exploration ability and convergence accuracy of the algorithm. Furthermore, the convergence factor has been modified to ensure that the algorithm emphasizes global search in the early stages, while focusing on local exploitation in the later stages. Additionally, an adaptive T-distribution mutation method has been constructed to increase the algorithm's probability of escaping from local optima. To validate the performance of the enhanced dung beetle optimization, the algorithm has been tested on the CEC2014 and CEC2017 benchmark functions. The experimental results demonstrated that EDBO outclassed both the original DBO and other advanced optimization algorithms in terms of overall performance. Finally, the effectiveness of EDBO in solving real-world problems is verified through its application in the three-dimensional (3D) path planning of unmanned aerial vehicles. In the most challenging task scenarios, the EDBO algorithm successfully identified feasible paths. Compared with five benchmark algorithms, it achieved an average reduction in the overall cost of 17.1%, 9.5%, 22.9%, 14.5%, and 18.6%, respectively. These results further demonstrated the significant superiority and practical value of the EDBO algorithm in addressing the

unmanned aerial vehicles (UAVs) path planning problem.

Keywords: dung beetle optimization algorithm; Sobol sequence; adaptive T-distribution; path planning; unmanned aerial vehicles

1. Introduction

Optimization problems involve determining the optimal value of a function while satisfying specific constraints. Essentially, it is about selecting the best possible solution from a set of feasible alternatives [1]. In practical scenarios, we often aim to find the most optimal solutions to various problems, with the Traveling Salesman Problem being a classic example [2]. Optimization problems are prevalent across various fields and industries, and almost every domain has its own optimization challenges. For instance, in the field of image processing, optimization problems are commonly applied to tasks such as image denoising, image segmentation, object detection, and image encryption [3–5]. In the energy sector, optimization problems are involved in power system scheduling, energy distribution, and the integration of renewable energy sources [6–8]. In engineering design, optimization is used to enhance product performance, reduce costs, and improve efficiency [9]. In the field of unmanned aerial vehicles, optimization is widely applied to problems such as path planning, flight control, and task scheduling [10].

Traditional gradient-based optimization methods require that the objective function be differentiable, and their optimization accuracy largely depends on the location of the initial point [11]. These drawbacks result in traditional gradient-based methods performing well on small-scale problems but often getting stuck in local optima when dealing with more complex problems. In contrast, metaheuristic algorithms do not rely on the specific form of the problem, nor do they require derivative information of the objective function, which gives them strong generality and adaptability [12]. As a result, metaheuristic algorithms have received widespread research attention in recent years.

Metaheuristic algorithms [13] are a category of optimization techniques that draw inspiration from natural processes or systems. These algorithms do not depend on the detailed structural aspects of the problem, making them highly flexible and adaptable. They can be widely applied to various problems and demonstrate high efficiency in problem-solving, compensating for the limitations of exact optimization methods in certain complex problems. Genetic algorithms [14] and particle swarm optimization [15] are well-known instances of metaheuristic algorithms. Other classic metaheuristic algorithms, such as ant colony optimization [16], differential evolution [17], cuckoo search [18], and firefly algorithm [19], aim to address the shortcomings of exact optimization methods. In recent years, several new algorithms have also been proposed, including the grey wolf optimizer (GWO) [20], sine cosine algorithm (SCA) [21], crow search algorithm (CSA) [22], whale optimization algorithm (WOA) [23], sparrow search algorithm (SSA) [24], dung beetle optimization (DBO) [25], and moss growth optimization (MGO) [26].

Among the numerous available metaheuristic algorithms, the DBO algorithm is a novel population-based optimization algorithm proposed in 2022. It simulates the behavior of dung beetles in nature and constructs a search framework based on the "rolling-ovipositing-foraging-stealing" model. Compared to other state-of-the-art algorithms, the DBO algorithm has attracted widespread attention within the field of metaheuristic algorithms due to its excellent global search capability, unique design principles, and satisfactory solution accuracy. However, according to the "no free lunch

(NFL) theorem" [27], no population-based optimization method is capable of solving all types of optimization problems, and each population-based optimization algorithm has its own limitations and constraints. As problems become increasingly complex, better methods are always required, particularly when dealing with high-dimensional and dynamic problems. This paper focuses on enhancing the benchmark DBO algorithm, which has been extensively studied and exhibits stable performance. This approach contributes to achieving more rigorous and interpretable experimental results. In response to the shortcomings of the DBO algorithm, many researchers have introduced different strategies for improvement. Various improved versions of the DBO have also been widely applied to address real-world challenges. To some degree, these studies have improved the performance of the DBO, validating its advancement and effectiveness in applications, and providing guidance for future research. However, there are still some shortcomings in the existing research: First, these improved algorithms still exhibit certain limitations when applied in practice (such as insufficient optimization accuracy, low search efficiency, and the inability to avoid getting trapped in local optima), leaving room for further improvement. Second, many enhancements have been made by integrating concepts or formulas from other algorithms, taking advantage of their iterative nature, and further improvements are still possible. Furthermore, more thorough mathematical analysis of the algorithm's performance in solving multidimensional problems could be conducted, with the aim of further optimizing its exploration process.

This paper proposes an enhanced dung beetle optimization (EDBO) algorithm to overcome the limitations of the original DBO algorithm. EDBO combines various strategies to enhance the global optimization performance of the original DBO, while also improving its convergence accuracy and speed. The overall capability of the EDBO is comprehensively evaluated through experiments in different dimensions. In summary, this paper makes the following key contributions:

- Using the Sobol sequence to generate the initial solutions for the algorithm can better explore the solution space, making the initial population more random and diverse.
- A broad global search is conducted through random angles $(teta)$ and adaptive step sizes, while individual positions are finely adjusted based on the probability ratio (MOP) and linear transformations, thus improving the algorithm's global exploration ability and convergence precision.
- By improving the convergence factor, the algorithm exhibits stronger global exploration capability in the early stages, while improving the precision of local search in the later phases.
- An adaptive T-distribution mutation approach is developed to increase the algorithm's probability of escaping from local optima.

The structure of this paper is as follows: Section 2 outlines the essential background information related to the methods used, along with a comprehensive literature review. Section 3 introduces the basic DBO algorithm. Section 4 proposes the EDBO algorithm to address the limitations of the original DBO. Section 5 demonstrates the effectiveness of the proposed improvements through various experimental comparisons. Section 6 develops a 3D path planning model for unmanned aerial vehicles (UAVs). Section 7 applies the improved algorithm to UAV 3D path planning, further exploring its practical applicability. Section 8 concludes the paper.

2. Background and related works

This section first reviews the significant improvements made to metaheuristic algorithms in recent years by experts and scholars. It then provides a brief overview of metaheuristic optimization algorithms and their practical applications across various disciplines, aiming to provide the reader with a clear background and framework.

2.1. Improved algorithm

According to the NFL theorem, no single algorithm can be claimed to solve every problem. As a result, numerous scholars have made targeted improvements to address specific challenges. For example, Xu et al. [28] introduced a best point set initialization strategy, inertia weight, T-distribution, and inverse learning into the classic artificial bee colony (ABC) algorithm, which enhanced its ability to escape local optima. Attiya et al. [29] proposed an optimized algorithm combining the manta ray foraging optimization algorithm with the salp swarm algorithm. This approach aims to enhance the local exploitation ability of manta ray foraging optimization algorithm using salp swarm algorithm, thus achieving a good balance between global exploration and local exploitation, improving the algorithm's ability to find the optimal solution. The hybrid algorithm was ultimately tested on 15 different task scheduling instances and demonstrated significant performance advantages. Shen et al. [30] proposed a whale optimization algorithm variant based on multi-population evolution (MEWOA), which divided individuals into three equally-sized subpopulations based on their fitness: exploration subpopulation, exploitation subpopulation, and moderate subpopulation. Each subpopulation uses different movement strategies, with the exploration subpopulation responsible for global search, the exploitation subpopulation focusing on local search, and the moderate subpopulation randomly selecting between exploration and exploitation. Furthermore, to enhance global optimization capabilities and avoid local optima, an evolutionary strategy based on population dynamics was employed to boost the performance of the MEWOA. Several recent advances have also been made in improving DBO. For instance, Zhu et al. [31] proposed a quantum-based and multi-strategy fusion DBO (QHDBO) algorithm, which applied a good point set initialization strategy and introduced a dynamic convergence factor to enhance the algorithm's exploration capability during the initial phase. Additionally, a quantum-inspired T-distribution mutation technique was applied to modify the global best solution, helping to avoid the algorithm getting trapped in local optima. To further enhance DBO's global search capability, He et al. [32] combined Chebyshev chaotic mapping, curve-adaptive golden sine strategies, Lévy flights with Cauchy-T mutation strategies, and dynamic weight coefficients to propose a novel DBO (IDBO) algorithm. Wang et al. [33] introduced a DBO algorithm leveraging quasi-oppositional learning and q-learning, which incorporates quantum state update principles into quasi-oppositional learning to enhance the randomness of the initial population. Q-learning was used to select the behavior of the dung beetles, and a variable spiral local search method was designed through a spiral operator to enhance the exploration ability of the algorithm. Experiments showed that the improved algorithm performed well on the CEC2017 function test set (a standard benchmark suite for optimization algorithms) and also demonstrated superiority in some classic practical engineering design problems. Li et al. [34] devised an improved DBO (TDBO) algorithm, which initially enhances the dung beetle's rolling phase by eliminating the influence of the worst value and combining the current solution with the optimal one, all while preserving the benefits of the original algorithm. Then, a random factor was introduced into the optimal solution to address the issue where the dung beetle dance phase only considers the current solution information. Finally, a Jacobi curve was incorporated into the dung beetle's foraging phase to improve the algorithm's capacity to avoid local optima.

2.2. Practical applications

In the past few decades, metaheuristic algorithms have been widely studied and applied across various fields. We have discussed some of the latest applications of metaheuristic algorithms. For

instance, Bukumira et al. [35] developed an enhanced variant of the sine-cosine algorithm to optimize the XGBoost machine learning model for solving the identification and classification problem of naval vessels. Extensive testing was conducted on the automatic identification system (AIS) dataset to validate its effectiveness. Radomirovic et al. [36] combined the particle swarm optimization (PSO) algorithm with genetic mechanisms to propose a genetic-enhanced PSO (GEPSO) for optimizing artificial intelligence model parameters. The optimized model was applied to solar flare classification and achieved excellent results on a widely recognized dataset, demonstrating the significant importance of integrating metaheuristic algorithms. Lopez-Franco et al. [37] proposed a general method for solving the trajectory tracking problem of robotic manipulators using metaheuristic optimization algorithms. Seven advanced metaheuristic algorithms were selected and tested on different types of robots. The final results demonstrated the feasibility of the proposed general method. Bencherqui et al. [38] integrated the Archimedes optimization (AO) algorithm with DNA coding and hyperchaotic systems to address the compression and encryption of large-scale medical images, fully leveraging the advantages of the AO algorithm in enhancing image reconstruction quality, encryption effectiveness, and security. Kmich et al. [39] developed a chaotic Puma optimizer algorithm (CPOA) by incorporating chaotic maps into the optimization phases of exploration and exploitation, aiming to control nonlinear and time-varying wheeled mobile robots. Karmouni et al. [40] proposed the aquila optimization algorithm (AOA) to optimize the parameters of discrete orthogonal Racah moment transformations (DORMT) and dynamically adjust the scaling factor, applied to watermarking and encryption systems for images. El Ghouate et al. [41] designed a chaotic Kepler optimization algorithm (CKOA), which enhances the algorithm's global search capability through a dynamic diversification strategy based on chaotic maps, and it has been validated on three complex engineering problems.

3. Dung beetle optimization algorithm

The basic DBO algorithm is inspired by the rolling, dancing, foraging, stealing, and breeding behaviors of dung beetles in nature. Based on these behaviors, four population update strategies have been designed.

3.1. Rollerball dung beetles

In nature, dung beetles use solar navigation to maintain a straight path while rolling their dung balls. In the original paper, Eq (1) is applied to modify the position of the rolling dung beetle:

$$\begin{aligned} x_i(t+1) &= x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x \\ \Delta x &= |x_i(t) - X^{\omega}| \end{aligned} \tag{1}$$

where t represents the current number of iterations, and $x_i(t)$ represents the location of the i-th dung beetle at the t-th iteration. α indicates whether the dung beetle strays from its initial path, with its value determined probabilistically as either 1 or –1. $k \in (0,0.2]$ is a constant that represents the deflection coefficient, and b is a constant with a value range of $(0,1)$. X^{ω} denotes the global worst location, and Δx is used to simulate solar illumination. When a dung beetle comes across an obstacle and can no longer continue rolling, it must perform a dance to determine its new rolling direction. This dancing behavior is defined as follows:

$$x_i(t+1) = x_i(t) + tan(\theta)\,|x_i(t) - x_i(t-1)| \tag{2}$$

where $\theta \in [0, \pi]$, and the position is not updated when θ takes values such as 0, $\pi/2$, and π.

3.2. Breeding dung beetles

To ensure a secure environment for their young, dung beetles move the dung balls to secure locations and hide them before laying their eggs. A boundary selection strategy for simulating the egg-laying region of female dung beetles is proposed, as shown in Eq (3).

$$\begin{aligned} Lb^* &= max(X^* \times (1-R), Lb) \\ Ub^* &= min(X^* \times (1+R), Ub) \end{aligned} \tag{3}$$

where X^* represents the local optimal value, and $Lb^*$and Ub^* represent the lower bound and upper bound of the spawning region. $R = 1 - t/T_{max}$ and T_{max} represent the maximum number of iterations, and Lb and Ub denote the lower and upper limits of the optimization problem, respectively. Based on the above equation, the boundaries of the egg-laying region are dynamically determined by the variation of R. Consequently, the dung beetle's breeding position is updated continuously, as expressed by the following mathematical formulation:

$$B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \tag{4}$$

where $B_i(t)$ represents the position of the i-th sphere at the t-th iteration. b_1 and b_2 are two independent random vectors, each with a dimension of $1 \times D$, where D represents the dimensionality.

3.3. Foraging dung beetles

In nature, dung beetles, while foraging, select a safe area in a manner similar to when they lay their eggs. The original definition of this safe area has been redefined and is represented by the following equation:

$$\begin{aligned} Lb^b &= max(X^b \times (1-R),\ Lb) \\ Ub^b &= min(X^b \times (1+R),\ Ub) \end{aligned} \tag{5}$$

where X^b represents the global optimal position, and Lb^b and Ub^b denote the lower and upper limits of the optimal foraging region, respectively. Thus, the position of the small dung beetle is updated as follows:

$$x_i(t+1) = x_i(t) + C_1 \times (x_i(t) - Lb^b) + C_2 \times (x_i(t) - Ub^b) \tag{6}$$

C_1 is a random variable that follows a normal distribution, and C_2 represents a random variable in the interval $(0,1)$.

3.4. Stealing dung beetles

Thieving behavior involves stealing dung balls from other beetles. In the iterative process, the

position update mechanism for the thieving beetle is governed by Eq (7).

$$x_i(t+1) = X^b + S \times g \times (|x_i(t) - X^*| + |x_i(t) - X^b|) \tag{7}$$

where S represents a constant, and g is a random vector with magnitude following a normal distribution.

4. EDBO algorithm

When solving optimization problems, the performance of the DBO algorithm surpasses that of many intelligent optimization algorithms. However, obtaining the ideal optimal solution remains a challenge for certain problems. As a result, several enhancements have been introduced to improve the DBO algorithm's overall performance in optimization problem-solving.

4.1. Sobol sequence initialization

In the DBO algorithm, the initial population is generated randomly within the search space. This method has low coverage, and the individuals are unevenly distributed, making the distribution unpredictable. As a result, it can negatively impact the algorithm's performance to some extent.

Some researchers have optimized the initialization sequence using chaotic search to enhance the global search capability. For example, Lu [42] initialized the population using the Tent chaotic map, Ye et al. [43] used Latin hypercube sampling (LHS) for population initialization, and Zhang et al. [44] introduced the Cubic mapping algorithm for initialization. While these chaotic algorithms have some ability to escape local optima, they have two main drawbacks: First, they exhibit strong randomness, which introduces significant uncertainty during the algorithm's operation. Second, adjacent points are highly correlated, and if the algorithm converges to an unstable point, such as the small-period points (0.2, 0.4, 0.6, 0.8) and unstable points (0, 0.25, 0.5, 0.75) of the Tent map [45], it cannot continue running.

Unlike pseudo-random numbers, low-discrepancy sequences replace pseudo-random sequences with deterministic sequences, known as the quasi-Monte Carlo (QMC) method, which have low discrepancy. QMC achieves higher efficiency and uniformity in probabilistic problems by selecting appropriate sampling directions and filling points as evenly as possible within multidimensional hypercube cells. Among them, the Sobol sequence has a shorter computation cycle, faster sampling speed, and higher efficiency when handling high-dimensional sequences [46]. Therefore, this paper uses the Sobol sequence to map the initialization population.

Assume that the range of the optimal solution is $[x_{min}, x_{max}]$, and the random numbers generated by the Sobol sequence are $k_n \in [0,1]$. The initial population positions can be defined as:

$$x_n = x_{min} + K_n \times (x_{max} - x_{min}) \tag{8}$$

In order to demonstrate that the point set produced by the Sobol sequence exhibits better uniformity compared to a randomly generated set, the two-dimensional coordinate space is divided into 1×1 cells to check the evenness of point distribution within each cell. Figure 1 illustrates the initial distribution of 500 points in a two-dimensional space, generated via the Sobol sequence, while Figure 2 depicts the initial population distribution generated by a random strategy. As seen in Figure 3, the points produced by the Sobol sequence are distributed more evenly than those produced randomly.

This shows that incorporating the Sobol sequence can effectively increase the population diversity, which aids in preventing the algorithm from getting stuck in local optima.

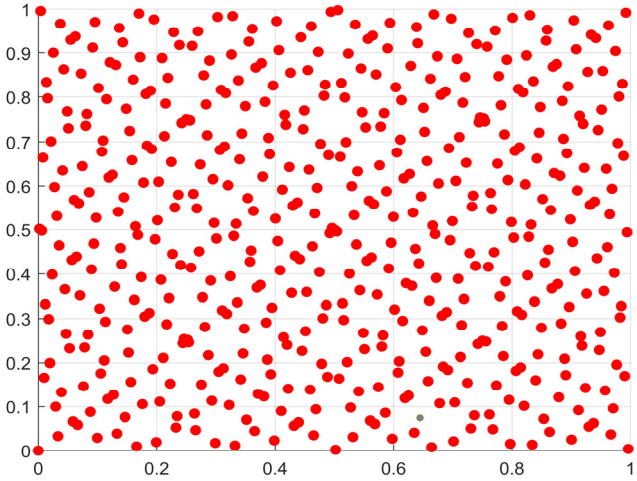


Figure 1. Sobol sequence generated point set plot.

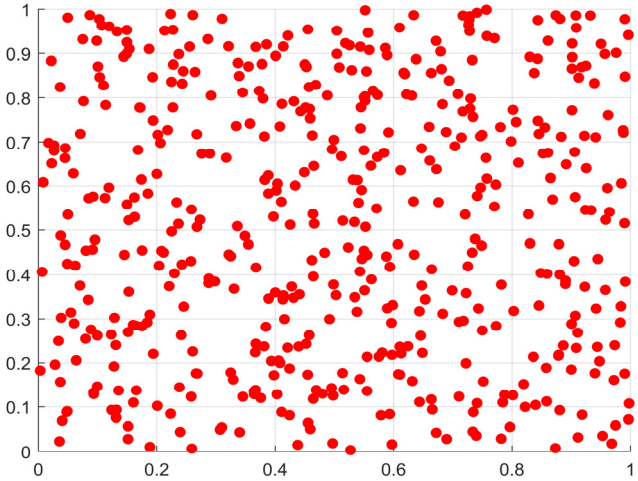


Figure 2. Random strategy generated point set plot.

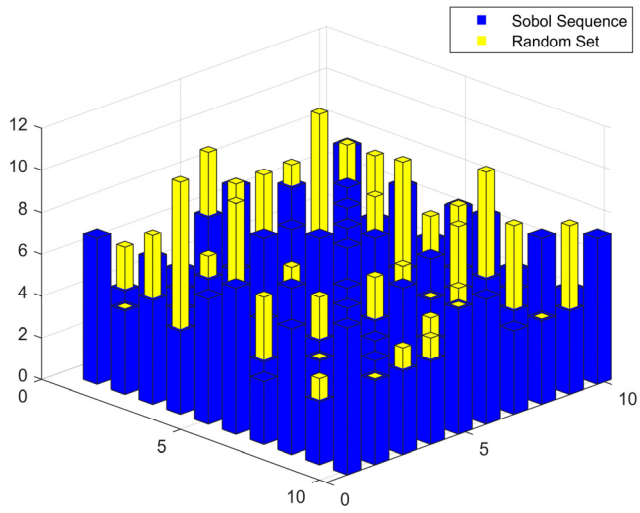


Figure 3. Point arrangement diagram.

4.2. Enhanced search strategy

During the iteration process, as the population gradually converges to the optimal individual, the population diversity begins to diminish. To improve the adaptability of the algorithm, the improved method proposed in this paper introduces the probability ratio (MOP) and random angle mutation. By dynamically adjusting the update strategy of individuals, the population can achieve better exploration and exploitation capabilities at different stages of the iteration. During each generation of the iteration process, the algorithm dynamically adjusts the probability ratio as shown by the following formula:

$$MOP(t) = 1 - (\frac{t}{T})^{\frac{1}{\alpha}} \tag{9}$$

where t is the current iteration number, T is the maximum iteration number, and α is a control parameter, typically set to 3. As the number of iterations increases, the MOP value gradually decreases, meaning that in the early stages of the algorithm, individual updates are more randomized, allowing for broader coverage of the search space, while in the later stages, the population gradually converges.

During the iteration, each individual X_i undergoes mutation based on a random selection strategy and random angle mutation strategy. The algorithm uses a probability of $1 - \sqrt{t/T_{max}}$ to determine which strategy to apply. q is a random number between $[0,1]$, and when q is smaller than this probability value, the position of the individual X_i is updated according to the following formula:

$$X_i = X_i \times MOP \times ((Lb - Ub) \times Mu + Ub) \tag{10}$$

where Mu is the adjustment factor, typically set to 0.5. By modifying the MOP value, the algorithm can control the global search capability of the individual, increasing the probability of finding better solutions in the solution space. When q is greater than the above probability, the individual will be updated according to the following mutation formula:

$$X_i = X_i + step \times \tan(\theta) \times |X^* - X_i| \tag{11}$$

where the step size $step$ is calculated as follows:

$$step = sign(rand - 0.5) \times norm(X^*) \times \log(1 + \frac{10 \times D}{t}) \tag{12}$$

The algorithm updates the position by calculating the difference between the current individual and the best individual X^*, combined with a random angle θ, using a large step size for the update. This approach helps the algorithm escape local optima during later stages of exploration, thereby accelerating global convergence.

Through the combination of both strategies, the algorithm demonstrates strong search capability during the global exploration phase, enabling a wide exploration of the solution space. During the local convergence phase, the algorithm enhances its local exploitation ability through random angle mutations, thus improving both convergence speed and optimization performance.

4.3. Improved convergence factor

According to the formula $R = 1 - t/T_{max}$, it is evident that as the convergence factor R reduces, the scope of the optimal breeding and foraging areas gradually diminishes. During the initial stage of the algorithm, this area is large, which contributes to improving the algorithm's global search ability.

To enhance global search capability, the algorithm is designed with a larger optimal breeding and foraging area in the initial stages. At later stages, as the area narrows, the algorithm can perform more effective local exploration. The goal is for the algorithm to have stronger global exploration ability in the early stages to better discover high-quality solutions, and to enhance local search ability in the later stages to accelerate convergence. Therefore, the factor must be adjusted to decrease slowly in the early stages and rapidly in the later stages. This paper presents an improvement to the inertia factor in the original formulation, and the new convergence factor R is given by the formula (13).

$$R = \sin^2((1 - \frac{t}{T_{max}}) \times \frac{\pi}{2}) \tag{13}$$

4.4. Adaptive T-distribution mutation

To avoid premature convergence that may lead the DBO algorithm into local optima, it is essential to allow the population to move away from its current state. While the original DBO applies a normal distribution for mutation, it overlooks the fact that the capacity of individuals to escape their current position changes at different stages. Therefore, an adaptive T-distribution mutation technique is introduced to enhance the algorithm's ability to escape from local optima.

The T-distribution is a statistical sampling distribution commonly used in hypothesis testing [47], and its probability density function is as follows:

$$f(x, n) = \frac{\Gamma\left(\frac{n+1}{2}\right)}{\sqrt{n\pi}}\left(1 + \frac{x^2}{n}\right)^{\frac{n+1}{2}}, -\infty < x < \infty \tag{14}$$

where n represents the degrees of freedom. The T-distribution is illustrated in Figure 4. Building upon the T-distribution, we introduce an adaptive factor and integrate the foraging phase from the original DBO algorithm. This factor enhances the exploration probability during the initial iterations, encouraging the population to explore the solution space more extensively. As the algorithm progresses,

the mutation factor gradually shifts its focus toward guiding the exploitation phase, facilitating a smooth transition from global exploration to local exploitation. Additionally, the adaptive mutation mechanism of the T-distribution helps address the issue of the optimal foraging region contracting due to the reduction of the convergence factor R in later stages.

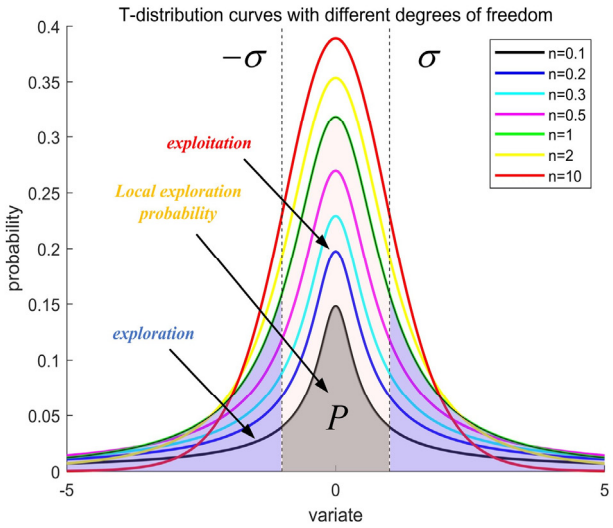


Figure 4. T-distribution image.

Excessive contraction of this region can lead to individuals clustering within a restricted area, thereby diminishing population diversity and causing the algorithm to get stuck in local optima. By introducing the adaptive T-distribution and its disturbance strategy, premature convergence is effectively avoided. The specific formula is as follows:

$$T = T(\frac{a^t}{m}) = T(\frac{1}{m} \times \exp(\frac{\ln t_c}{t_c})^t) \tag{15}$$

where $t_c = T_{max}/5$, and the expression inside the parentheses denotes the degrees of freedom. t is the current iteration number, t_c plays a pivotal role in the iteration process, and m is the scaling factor, which is set to 10 in this study. In the foraging process of the dung beetle, if the disturbance is too strong, the population may fall into a state of chaos; whereas if the disturbance is too weak, the disturbance strategy may not be effective. Therefore, a pseudo-random number generated within the interval (0, 1) is used to determine whether disturbance is needed. When the random number is greater than 0.5, it indicates significant clustering within the optimal foraging area, and disturbance is required to prevent blind clustering. When the random number is less than or equal to 0.5, it indicates that no clustering has occurred in the optimal foraging area. The specific update formula for the new position of the dung beetle is as follows:

$$x_i(t+1) = \begin{cases} T \times \exp\left(\frac{X^\omega - x_i(t)}{i^2}\right), & rand > 0.5 \\ x_i(t) + T \times (x_i(t) - Lb^b) + T \times (x_i(t) - Ub^b), & rand \le 0.5 \end{cases} \tag{16}$$

The symbols in this formula have the same meaning as those described earlier. The pseudocode of EDBO is shown in Algorithm 1. To present the process intuitively, Figure 5 demonstrates the working principle of EDBO in the form of a flowchart.

Algorithm 1: The framework of the EDBO algorithm.

Input: The maximum iteration T_{max}, the population size N. obtain an initialized population X of dung beetles using the Sobol sequence strategy.

Output: Optimal position X^b and its fitness value f_b.

while $t \leq T_{max}$ **do**
  for i = 1 to $Number\ of\ Rollerball\ \ dung\ beetles$ **do**
    if $q < 1 - \sqrt{t/T_{max}}$ **then**
      Update Rollerball dung beetle by Eq (10).
    else
      Rolling the ball in the encounter of obstacles by Eqs (11) and (12).
    end if
  end for
  The value of the nonlinear convergence factor is calculated by Eq (13).
  for i = 1 to $Number\ of\ Breeding\ dung\ beetles$ **do**
    Update Breeding dung beetles by Eqs (3) and (4).
  end for
  for i = 1 to $Number\ of\ Foraging\ dung\ beetles$ **do**
    Update Foraging dung beetles by Eqs (15) and (16).
  end for
  for i = 1 to S$tealing\ \ dung\ \ beetles$ **do**
    Update the position of the Stealing dung beetle using Eq (7).
  end for
  The global optimal solution is modified through adaptive t-distribution interference of the mutation factor, and the optimal point is determined via a reinforced search strategy.
end while
return X^b and its fitness value f_b.

4.5. Complexity assessment of the algorithm

In this study, the population size is denoted by N, the dimensionality by D, and the maximum number of iterations by max_it. The population is divided into four behavioral groups: rolling, reproducing, foraging, and stealing, with corresponding proportions z_1, z_2, z_3, and z_4, respectively, where $z_1 + z_2 + z_3 + z_4 = 1$. The time complexity of population initialization in EDBO using the Sobol sequence is $O(N \times D)$. The time complexity for updating the rolling behavior (employing a probability ratio and random angle mutation strategy) is $O(z_1 \times N \times D \times max\_it)$, and for the reproducing behavior, it is $O(z_2 \times N \times D \times max\_it)$. The foraging behavior (using an adaptive T-distribution mutation strategy) has a time complexity of $O(z_3 \times N \times D \times max\_it)$, while the stealing behavior has a time complexity of $O(z_4 \times N \times D \times max\_it)$. The time complexity of updating the global best solution is $O(D \times max\_it)$, and the complexity for evaluating individual fitness is $O((N + 1) \times max\_it)$. Therefore, the overall time complexity of EDBO can be expressed as:$O(N \times D + (N + 1) \times D \times max\_it + (N + 1) \times max\_it)$.In comparison, the standard DBO has a time complexity of: $O(N \times D + N\ \ D \times max\_it + N \times max\_it)$.In summary, the time complexity of EDBO is of the same order of magnitude as that of the standard DBO and does not introduce a significant increase in computational cost.

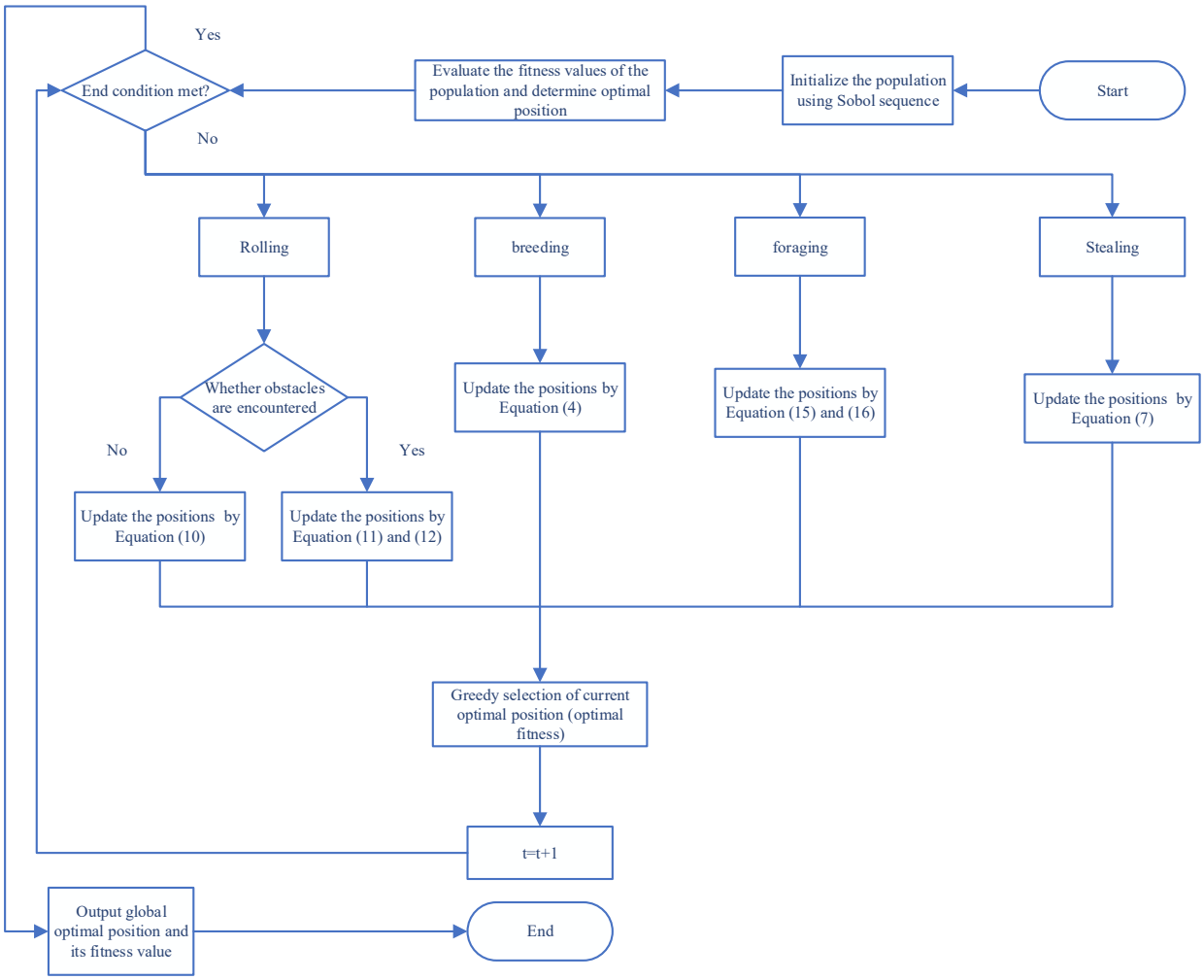


Figure 5. The flowchart of EDBO.

5. Simulation experiment analysis

To thoroughly assess the performance of the enhanced algorithm, two sets of benchmark functions are selected: CEC2014 [49] and CEC2017 [50]. The CEC2014 set includes three unimodal functions F1–F3, thirteen simple multimodal functions F4–F16, six hybrid functions F17–F22, and eight composite functions F23–F30. The main components of CEC2017 consist of 30 test functions, but due to the poor stability of F2, not all experiments were conducted on F2. In this set of functions, F1 and F3 are unimodal functions, containing only the global minimum without any local minima, mainly employed to assess the algorithm's effectiveness in locating the optimal solution. F4–F10 are multimodal functions with multiple local extrema, utilized to evaluate the algorithm's ability to avoid local optima. F11–F20 are hybrid functions, each composed of at least three CEC2017 benchmark functions that have been rotated or shifted, with specific weights assigned to each sub-function. F21–F30 are composite functions, made up of at least three hybrid functions or CEC2017 benchmark functions that have undergone rotation and displacement, with additional bias values and weights added to each sub-function, substantially enhancing the optimization complexity of the algorithm.

5.1. Sensitivity testing of parameters

In the EDBO algorithm, two key parameters are introduced: α and m. The probability ratio (MOP) in Eq (9) is determined by the parameter α, while the adaptiveT-distribution mutation factor in Eq (15) is governed by the parameter m. The CEC2014 test functions are selected as benchmarks to analyze the sensitivity of the EDBO algorithm to these parameters. In the experiments, the value of α ranges from 1 to 5 with a step size of 1, and the value of m ranges from 5 to 25 with a step size of 5. The population size N, problem dimension D, and maximum number of function evaluations ($maxFEs$) are set to 50, 30, and 300,000, respectively. Each function is independently executed 30 times. The statistical results are presented in Tables 1 and 2, with the best results highlighted in bold.

Table 1. Mean errors using various α values ($m = 10$).

Fun	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$	$\alpha = 5$
F1	4.06×10^{5}	4.06×10^{5}	4.06×10^{5}	4.06×10^{5}	4.06×10^{5}
F2	7.16×10^{-2}	7.62×10^{-2}	$\mathbf{2.58\times10^{-3}}$	7.69×10^{-3}	7.57×10^{-2}
F3	4.72×10^{1}	4.64×10^{1}	$\mathbf{5.00\times10^{0}}$	4.44×10^{2}	$\mathbf{5.00\times10^{0}}$
F4	$\mathbf{1.87\times10^{1}}$	3.65×10^{1}	2.15×10^{1}	2.75×10^{1}	3.87×10^{1}
F5	2.04×10^{1}	$\mathbf{2.02\times10^{1}}$	2.04×10^{1}	2.03×10^{1}	2.04×10^{1}
F6	2.42×10^{1}	2.34×10^{1}	$\mathbf{2.05\times10^{1}}$	2.31×10^{1}	2.42×10^{1}
F7	1.60×10^{-2}	9.93×10^{-3}	1.60×10^{-2}	$\mathbf{8.77\times10^{-3}}$	1.60×10^{-2}
F8	1.13×10^{2}	1.11×10^{2}	$\mathbf{9.62\times10^{1}}$	1.07×10^{2}	1.13×10^{2}
F9	1.84×10^{2}	$\mathbf{1.77\times10^{2}}$	1.81×10^{2}	1.79×10^{2}	1.84×10^{2}
F10	2.86×10^{3}	2.84×10^{3}	$\mathbf{2.57\times10^{3}}$	2.72×10^{3}	2.86×10^{3}
F11	4.01×10^{3}	4.06×10^{3}	$\mathbf{3.81\times10^{3}}$	4.05×10^{3}	4.01×10^{3}
F12	4.81×10^{-1}	5.70×10^{-1}	4.81×10^{-1}	$\mathbf{4.61\times10^{-1}}$	4.81×10^{-1}
F13	5.67×10^{-1}	6.01×10^{-1}	5.67×10^{-1}	5.67×10^{-1}	$\mathbf{5.61\times10^{-1}}$
F14	5.41×10^{-1}	$\mathbf{5.16\times10^{-1}}$	5.41×10^{-1}	5.54×10^{-1}	5.41×10^{-1}
F15	1.55×10^{1}	$\mathbf{1.43\times10^{1}}$	1.55×10^{1}	1.63×10^{1}	1.55×10^{1}
F16	1.19×10^{1}	1.19×10^{1}	$\mathbf{1.17\times10^{1}}$	1.19×10^{1}	$\mathbf{1.17\times10^{1}}$
F17	1.32×10^{5}	1.62×10^{5}	$\mathbf{1.23\times10^{5}}$	2.00×10^{5}	1.87×10^{5}
F18	$\mathbf{5.84\times10^{3}}$	7.36×10^{3}	$\mathbf{5.84\times10^{3}}$	5.94×10^{3}	$\mathbf{5.84\times10^{3}}$
F19	$\mathbf{1.04\times10^{1}}$	1.16×10^{1}	1.16×10^{1}	1.25×10^{1}	1.17×10^{1}
F20	1.29×10^{3}	1.33×10^{3}	1.14×10^{3}	$\mathbf{9.18\times10^{2}}$	1.14×10^{3}
F21	6.48×10^{4}	$\mathbf{6.01\times10^{4}}$	6.48×10^{4}	6.49×10^{4}	6.06×10^{4}
F22	3.64×10^{2}	3.82×10^{2}	3.64×10^{2}	$\mathbf{3.37\times10^{2}}$	3.64×10^{2}
F23	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F24	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F25	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F26	1.01×10^{2}	1.01×10^{2}	1.01×10^{2}	1.01×10^{2}	1.01×10^{2}
F27	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F28	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F29	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F30	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}

According to Table 1, the parameter α does not exhibit an overly pronounced impact on the precision of EDBO. Specifically, as the parameter value increases, the algorithm demonstrates a trend of initially increasing and subsequently decreasing convergence accuracy on functions F2, F3, F6, F8, F10, and F11. Conversely, it shows an opposite pattern on functions F4, F5, F9, F13, F14, and F17. For functions F7 and F18, EDBO demonstrates unstable convergence characteristics, while the convergence performance of other functions remains relatively unaffected by the parameter α settings. When α is set to 3, EDBO achieves superior results compared to other parameter values on nine benchmark functions. Therefore, setting α to 3 represents an optimal configuration choice.

Table 2. Mean errors using various m values $(\alpha = 3)$.

Fun	$m = 5$	$m = 10$	$m = 15$	$m = 20$	$m = 25$
F1	3.77×10^{5}	4.06×10^{5}	3.77×10^{5}	4.55×10^{5}	$\mathbf{3.73\times10^{5}}$
F2	7.69×10^{1}	$\mathbf{2.58\times10^{-3}}$	7.69×10^{1}	7.69×10^{1}	7.80×10^{-3}
F3	6.68×10^{0}	$\mathbf{5.00\times10^{0}}$	6.68×10^{0}	3.24×10^{2}	5.08×10^{2}
F4	4.13×10^{1}	2.15×10^{1}	4.13×10^{1}	$\mathbf{1.80\times10^{1}}$	2.59×10^{1}
F5	2.05×10^{1}	$\mathbf{2.04\times10^{1}}$	$\mathbf{2.04\times10^{1}}$	$\mathbf{2.04\times10^{1}}$	2.17×10^{1}
F6	2.29×10^{1}	$\mathbf{2.05\times10^{1}}$	2.29×10^{1}	2.32×10^{1}	2.36×10^{1}
F7	$\mathbf{7.96\times10^{-3}}$	1.60×10^{-2}	$\mathbf{7.96\times10^{-3}}$	9.68×10^{-3}	1.27×10^{-2}
F8	1.16×10^{2}	$\mathbf{9.62\times10^{1}}$	1.16×10^{2}	1.25×10^{2}	1.25×10^{2}
F9	1.86×10^{2}	1.81×10^{2}	1.86×10^{2}	$\mathbf{1.69\times10^{2}}$	1.77×10^{2}
F10	2.67×10^{3}	$\mathbf{2.57}\times10^{3}$	2.67×10^{3}	2.99×10^{3}	2.92×10^{3}
F11	4.08×10^{3}	3.81×10^{3}	4.08×10^{3}	3.86×10^{3}	$\mathbf{3.72\times10^{3}}$
F12	5.41×10^{-1}	$\mathbf{4.81\times10^{-1}}$	5.41×10^{-1}	6.00×10^{-1}	5.89×10^{-1}
F13	5.70×10^{-1}	5.67×10^{-1}	5.70×10^{-1}	$\mathbf{5.60\times10^{-1}}$	5.91×10^{-1}
F14	$\mathbf{5.29\times10^{-1}}$	5.41×10^{-1}	$\mathbf{5.29}\times10^{-1}$	6.44×10^{-1}	6.33×10^{-1}
F15	1.54×10^{1}	1.55×10^{1}	1.54×10^{1}	1.70×10^{1}	$\mathbf{1.51\times10^{1}}$
F16	1.17×10^{1}	1.17×10^{1}	1.17×10^{1}	1.17×10^{1}	1.17×10^{1}
F17	1.64×10^{5}	$\mathbf{1.23\times10^{5}}$	1.64×10^{5}	1.32×10^{5}	1.55×10^{5}
F18	6.82×10^{3}	$\mathbf{5.84\times10^{3}}$	8.82×10^{3}	8.62×10^{3}	1.13×10^{4}
F19	$\mathbf{1.11\times10^{1}}$	1.16×10^{1}	1.24×10^{1}	1.68×10^{1}	1.50×10^{1}
F20	$\mathbf{1.22\times10^{3}}$	1.14×10^{3}	$\mathbf{1.11\times10^{3}}$	1.25×10^{3}	1.54×10^{3}
F21	5.95×10^{4}	6.48×10^{4}	5.95×10^{4}	$\mathbf{5.15\times10^{4}}$	$\mathbf{5.15\times10^{4}}$
F22	4.09×10^{2}	3.64×10^{2}	4.09×10^{2}	$\mathbf{3.12\times10^{2}}$	3.39×10^{2}
F23	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F24	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F25	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F26	$\mathbf{1.01\times10^{2}}$	$\mathbf{1.01\times10^{2}}$	$\mathbf{1.01\times10^{2}}$	1.11×10^{2}	1.32×10^{2}
F27	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F28	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F29	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
F30	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}

According to Table 2, the value of the parameter m may vary depending on the characteristics of the problem. Therefore, it is difficult to obtain the truly optimal parameter for a general optimization

problem. The experimental results for the CEC2014 benchmark functions indicate that the performance of EDBO is closely related to the value of mmm. Specifically, when $m = 10$, it achieves relatively ideal solutions for functions F2, F3, F5, F6, F8, F10, F12, F17, F18, and F26. On the other hand, $m = 20$ demonstrates outstanding solving capabilities for problems F4, F5, F9, F13, F21, and F22. Additionally, when $m = 5$ and $m = 15$, reasonable solutions are provided for five problems in the CEC2014 set. $m = 25$ performs well for four problems. It is worth noting that variations in the value of mmm do not significantly affect the solution accuracy of EDBO for functions F16, F23, F24, F25, F27, F28, F29, and F30. In conclusion, setting the parameters α and m to 3 and 10, respectively, more effectively balances the optimization performance of the EDBO algorithm.

5.2. Comparison with advanced DBO variants

EDBO is compared with the standard DBO and three advanced DBO variants, namely, QHDBO, IDBO, and TDBO. To ensure a fair comparison, the population size N and the maximum number of function evaluations ($maxFEs$) are set to 50 and $10000 \times D$, where D denotes the problem dimension. Each algorithm is independently run 30 times to reduce random errors. In the experiments, for a more comprehensive comparison, the best value (Best), worst value (Worst), mean value (Mean), and standard deviation (Std) of each algorithm are reported, and the best results are highlighted in bold. In addition, the Wilcoxon's rank-sum test, with a significance level of 0.05, is applied to assess whether there are significant differences between the algorithms. A Friedman test is also performed to further evaluate the performance disparities among the methods, presenting the average rank value (ARV) and the final ranking (Rank). The parameter settings for the above algorithms are shown in Table 3.

Table 3. Parameter setup of the comparison algorithms.

Algorithm	setup
DBO	$k = 0.1, b = 0.3$
QHDBO	$k = 0.1, b = 0.3$
IDBO	$k = 0.1, b = 0.3$
TDBO	$k = 0.1, b = 0.3, \alpha = \beta = 0.5$
EDBO	$k = 0.1, b = 0.3, \alpha = 3, m = 10$

5.2.1. Analysis of CEC2014 statistical results

In this section, the CEC2014 test functions are chosen as benchmarks to evaluate the superiority of the EDBO algorithm in 30D and 50D. The results are presented in Tables 4 and 5, with the best results for each function highlighted in bold. The performance of EDBO compared to other algorithms is summarized using the symbols "+/≈/–", indicating in how many functions EDBO outperforms, matches, or underperforms the other comparative algorithms. Additionally, to vividly demonstrate the advantages of the proposed algorithm, Figure 6 illustrates the convergence curves for selected benchmark problems in the 30-dimensional scenario. As the 50-dimensional results exhibit similar convergence patterns, they are omitted for brevity.

As shown in Table 4, EDBO demonstrates outstanding performance in terms of solution quality. Specifically, EDBO achieves the highest solution accuracy in 21 functions, namely, F1–F5, F7, F10, F12, F17, and F19–F30. At the same time, EDBO performs second best in 4 functions, namely, F6,

F11, F16, and F18. Notably, IDBO is also highly competitive. IDBO achieves the best performance in 8 functions, namely, F16, F18, F23–F25, and F27–F29, and ranks second in 16 functions, namely, F1–F8, F10, F13–F15, F17, F19, F21, and F30. As for DBO, it provides better solutions for F24, while QHDBO excels in solving F6, F8, F9, F13–F15, and F24. TDBO exceeds other algorithms in F11, F24, and F25. The statistical results from the Friedman test show that EDBO has the lowest average ranking value of 1.7500, followed by IDBO with a ranking value of 2.2167. Thus, it is evident that EDBO performs the best and ranks first among these advanced DBO variants, proving the superiority of the proposed algorithm.

As the dimensionality of the optimization problem increases, the search accuracy often decreases. As shown in Table 5, EDBO demonstrates highly competitive optimization performance when handling 50-dimensional problems. Specifically, EDBO performs best in 22 functions, namely, F1–F5, F7, F10, F12, F13, F15, F17–F19, F21–F25, and F27–F30, and ranks second in 6 functions, namely, F6, F8, F11, F14, F16, and F20. Notably, IDBO achieves the best result in 9 functions, namely, F11, F14, F16, F23–F25, and F27–F29. For DBO, QHDBO, and TDBO, none of them achieves the best result in more than four functions. In conclusion, the improved algorithm EDBO exhibits outstanding performance across problems of various dimensions.

As shown in Figure 6, these curves indicate that, compared to the competing algorithms, EDBO exhibits faster convergence speed, lower volatility, and tends to stabilize. This means that EDBO can approach the optimal solution more quickly, improving the efficiency of problem solving while demonstrating better robustness. Furthermore, for most test problems, EDBO shows an accelerating convergence trend in the convergence curves, indicating that the search capability of EDBO improves progressively with iterations, enabling it to find better solutions more quickly. In conclusion, the improved EDBO algorithm demonstrates outstanding performance and is a more competitive optimization algorithm.

Table 4. Experimental results of DBO, QHDBO, IDBO, TDBO, and EDBO on CEC2014 at 30 dimensions.

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
F1	Best	7.26×10^{5}	5.32×10^{5}	1.60×10^{5}	5.46×10^{5}	1.25×10^{5}
	Worst	5.94×10^{7}	2.24×10^{7}	7.48×10^{6}	5.94×10^{7}	1.15×10^{6}
	Mean	1.78×10^{7}	6.15×10^{6}	4.69×10^{5}	1.44×10^{7}	$\mathbf{4.43\times10^{5}}$
	Std	1.38×10^{7}	6.10×10^{6}	3.00×10^{5}	1.53×10^{7}	2.55×10^{5}
F2	Best	2.21×10^{0}	3.35×10^{-1}	4.05×10^{-7}	1.06×10^{-4}	3.14×10^{-9}
	Worst	4.73×10^{5}	3.45×10^{4}	3.45×10^{4}	4.47×10^{5}	2.65×10^{-2}
	Mean	7.97×10^{4}	1.02×10^{4}	8.09×10^{3}	4.58×10^{4}	$\mathbf{2.58\times10^{-3}}$
	Std	1.32×10^{5}	1.50×10^{4}	1.48×10^{4}	9.60×10^{4}	6.81×10^{-3}
F3	Best	6.31×10^{1}	1.74×10^{3}	2.47×10^{-2}	6.45×10^{-3}	2.14×10^{-3}
	Worst	1.06×10^{4}	5.65×10^{4}	4.34×10^{1}	3.40×10^{3}	1.03×10^{2}
	Mean	3.39×10^{3}	1.10×10^{4}	1.41×10^{1}	9.18×10^{2}	$\mathbf{5.00\times10^{0}}$
	Std	2.71×10^{3}	1.06×10^{4}	2.34×10^{1}	1.09×10^{3}	5.57×10^{0}
F4	Best	1.11×10^{2}	1.54×10^{0}	3.92×10^{-2}	4.42×10^{0}	2.68×10^{-1}
	Worst	4.38×10^{2}	1.91×10^{2}	1.94×10^{2}	2.38×10^{2}	1.36×10^{2}
	Mean	1.76×10^{2}	9.41×10^{1}	4.44×10^{1}	1.15×10^{2}	$\mathbf{2.15\times10^{1}}$

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Std	6.75×10^{1}	4.39×10^{1}	3.64×10^{1}	4.38×10^{1}	3.12×10^{1}
F5	Best	2.04×10^{1}	2.06×10^{1}	2.01×10^{1}	2.02×10^{1}	2.00×10^{1}
	Worst	2.10×10^{1}	2.10×10^{1}	2.10×10^{1}	2.10×10^{1}	2.10×10^{1}
	Mean	2.08×10^{1}	2.09×10^{1}	2.07×10^{1}	2.08×10^{1}	$\mathbf{2.03\times10^{1}}$
	Std	1.84×10^{-1}	9.51×10^{-2}	3.07×10^{-1}	1.96×10^{-1}	3.61×10^{-1}
F6	Best	1.47×10^{1}	1.30×10^{1}	1.64×10^{1}	1.67×10^{1}	1.23×10^{1}
	Worst	2.85×10^{1}	3.07×10^{1}	3.07×10^{1}	3.11×10^{1}	2.77×10^{1}
	Mean	2.22×10^{1}	$\mathbf{2.04\times10^{1}}$	2.05×10^{1}	2.33×10^{1}	2.05×10^{1}
	Std	3.41×10^{0}	4.92×10^{0}	3.76×10^{0}	3.95×10^{0}	4.04×10^{0}
F7	Best	9.09×10^{-13}	7.37×10^{-3}	2.27×10^{-13}	2.77×10^{-6}	2.27×10^{-13}
	Worst	1.04×10^{0}	8.01×10^{-2}	6.86×10^{-2}	1.02×10^{0}	5.65×10^{-2}
	Mean	2.04×10^{-1}	3.76×10^{-2}	3.60×10^{-2}	2.71×10^{-1}	$\mathbf{1.40\times10^{-2}}$
	Std	3.53×10^{-1}	1.83×10^{-2}	8.33×10^{-2}	3.66×10^{-1}	1.56×10^{-2}
F8	Best	6.27×10^{1}	4.11×10^{1}	6.22×10^{1}	5.68×10^{1}	5.17×10^{1}
	Worst	2.06×10^{2}	1.70×10^{2}	1.58×10^{2}	1.81×10^{2}	1.60×10^{2}
	Mean	1.30×10^{2}	$\mathbf{8.98\times10^{1}}$	9.55×10^{1}	9.97×10^{1}	9.62×10^{1}
	Std	3.18×10^{1}	2.88×10^{1}	1.96×10^{1}	3.06×10^{1}	2.79×10^{1}
F9	Best	7.76×10^{1}	5.46×10^{1}	8.56×10^{1}	4.48×10^{1}	1.04×10^{2}
	Worst	2.99×10^{2}	2.69×10^{2}	2.08×10^{2}	2.39×10^{2}	2.68×10^{2}
	Mean	2.01×10^{2}	$\mathbf{1.44\times10^{2}}$	1.62×10^{2}	1.46×10^{2}	1.81×10^{2}
	Std	5.60×10^{1}	5.40×10^{1}	2.57×10^{1}	4.85×10^{1}	3.29×10^{1}
F10	Best	1.48×10^{3}	2.62×10^{2}	1.36×10^{3}	1.10×10^{3}	1.28×10^{3}
	Worst	3.90×10^{3}	4.86×10^{3}	5.46×10^{3}	4.45×10^{3}	4.36×10^{3}
	Mean	2.94×10^{3}	3.04×10^{3}	2.65×10^{3}	3.08×10^{3}	$\mathbf{2.46\times10^{3}}$
	Std	5.97×10^{2}	1.02×10^{3}	8.18×10^{2}	8.11×10^{2}	7.82×10^{2}
F11	Best	2.08×10^{3}	3.14×10^{3}	2.28×10^{3}	2.08×10^{3}	2.42×10^{3}
	Worst	5.98×10^{3}	7.01×10^{3}	4.80×10^{3}	4.74×10^{3}	4.78×10^{3}
	Mean	4.40×10^{3}	5.09×10^{3}	3.91×10^{3}	$\mathbf{3.77\times10^{3}}$	3.81×10^{3}
	Std	8.22×10^{2}	9.19×10^{2}	7.12×10^{2}	7.23×10^{2}	6.28×10^{2}
F12	Best	1.60×10^{-1}	8.41×10^{-1}	2.25×10^{-1}	3.08×10^{-1}	1.82×10^{-1}
	Worst	2.87×10^{0}	2.85×10^{0}	2.34×10^{0}	2.99×10^{0}	1.21×10^{0}
	Mean	9.86×10^{-1}	1.76×10^{0}	1.06×10^{0}	1.47×10^{0}	$\mathbf{4.95\times10^{-1}}$
	Std	7.55×10^{-1}	6.18×10^{-1}	5.86×10^{-1}	8.09×10^{-1}	2.06×10^{-1}
F13	Best	4.28×10^{-1}	2.69×10^{-1}	3.44×10^{-1}	3.25×10^{-1}	3.04×10^{-1}
	Worst	9.92×10^{-1}	7.61×10^{-1}	1.05×10^{0}	8.42×10^{-1}	9.13×10^{-1}
	Mean	5.90×10^{-1}	$\mathbf{5.36\times10^{-1}}$	5.47×10^{-1}	5.85×10^{-1}	5.67×10^{-1}
	Std	1.28×10^{-1}	1.18×10^{-1}	9.86×10^{-2}	1.18×10^{-1}	1.47×10^{-1}
F14	Best	3.11×10^{-1}	1.52×10^{-1}	1.87×10^{-1}	2.66×10^{-1}	2.19×10^{-1}
	Worst	1.20×10^{0}	8.39×10^{-1}	1.10×10^{0}	1.22×10^{0}	1.19×10^{0}
	Mean	9.01×10^{-1}	$\mathbf{2.78\times10^{-1}}$	4.37×10^{-1}	6.94×10^{-1}	5.41×10^{-1}
	Std	2.60×10^{-1}	1.16×10^{-1}	2.56×10^{-1}	3.32×10^{-1}	3.28×10^{-1}
F15	Best	5.67×10^{0}	4.62×10^{0}	6.33×10^{0}	7.00×10^{0}	6.10×10^{0}

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Worst	5.15×10^{1}	1.95×10^{1}	3.97×10^{1}	2.36×10^{1}	2.27×10^{1}
	Mean	1.80×10^{1}	$\mathbf{1.30\times10^{1}}$	1.34×10^{1}	1.47×10^{1}	1.50×10^{1}
	Std	1.12×10^{1}	4.19×10^{0}	4.37×10^{0}	4.50×10^{0}	4.35×10^{0}
F16	Best	1.02×10^{1}	1.10×10^{1}	1.01×10^{1}	1.07×10^{1}	9.43×10^{0}
	Worst	1.28×10^{1}	1.29×10^{1}	1.25×10^{1}	1.28×10^{1}	1.31×10^{1}
	Mean	1.17×10^{1}	1.19×10^{1}	$\mathbf{1.16\times10^{1}}$	1.17×10^{1}	1.17×10^{1}
	Std	6.31×10^{-1}	5.91×10^{-1}	5.91×10^{-1}	5.60×10^{-1}	7.43×10^{-1}
F17	Best	5.35×10^{4}	4.22×10^{4}	7.97×10^{3}	5.00×10^{4}	3.20×10^{4}
	Worst	7.72×10^{6}	5.99×10^{6}	7.75×10^{5}	4.77×10^{6}	5.92×10^{5}
	Mean	2.01×10^{6}	1.37×10^{6}	1.44×10^{5}	1.27×10^{6}	$\mathbf{1.23\times10^{5}}$
	Std	1.60×10^{6}	1.57×10^{6}	1.21×10^{5}	1.18×10^{6}	6.90×10^{4}
F18	Best	4.87×10^{2}	5.83×10^{2}	4.48×10^{2}	5.39×10^{2}	3.67×10^{2}
	Worst	1.12×10^{6}	2.58×10^{8}	1.28×10^{9}	1.18×10^{6}	2.63×10^{4}
	Mean	1.40×10^{5}	8.73×10^{6}	$\mathbf{3.28\times10^{3}}$	2.10×10^{5}	5.84×10^{3}
	Std	3.14×10^{5}	4.71×10^{7}	2.95×10^{3}	4.09×10^{5}	7.06×10^{3}
F19	Best	8.78×10^{0}	6.25×10^{0}	8.92×10^{0}	1.04×10^{1}	7.73×10^{0}
	Worst	5.91×10^{1}	1.23×10^{2}	3.41×10^{1}	7.82×10^{1}	1.71×10^{1}
	Mean	1.70×10^{1}	1.57×10^{1}	1.25×10^{1}	1.63×10^{1}	$\mathbf{1.04\times10^{1}}$
	Std	1.13×10^{1}	2.04×10^{1}	2.87×10^{0}	1.19×10^{1}	2.27×10^{0}
F20	Best	7.74×10^{2}	3.46×10^{3}	5.16×10^{2}	7.81×10^{2}	2.90×10^{2}
	Worst	2.29×10^{4}	8.34×10^{4}	8.52×10^{4}	6.03×10^{3}	4.70×10^{3}
	Mean	5.90×10^{3}	1.20×10^{4}	9.08×10^{3}	2.70×10^{3}	$\mathbf{1.14\times10^{3}}$
	Std	5.38×10^{3}	1.66×10^{4}	1.42×10^{4}	1.58×10^{3}	7.98×10^{2}
F21	Best	1.50×10^{4}	1.93×10^{4}	1.33×10^{4}	4.76×10^{4}	2.41×10^{3}
	Worst	2.94×10^{6}	8.96×10^{6}	1.61×10^{7}	1.46×10^{6}	3.56×10^{5}
	Mean	4.94×10^{5}	9.70×10^{5}	1.02×10^{5}	3.49×10^{5}	$\mathbf{6.35\times10^{4}}$
	Std	5.63×10^{5}	1.63×10^{6}	7.93×10^{4}	3.22×10^{5}	7.55×10^{4}
F22	Best	1.69×10^{2}	1.55×10^{2}	1.95×10^{2}	4.85×10^{1}	1.31×10^{2}
	Worst	1.18×10^{3}	1.06×10^{3}	1.40×10^{3}	1.27×10^{3}	6.76×10^{2}
	Mean	6.24×10^{2}	5.92×10^{2}	5.76×10^{2}	5.50×10^{2}	$\mathbf{3.64\times10^{2}}$
	Std	2.68×10^{2}	2.34×10^{2}	2.89×10^{2}	2.64×10^{2}	1.63×10^{2}
F23	Best	3.16×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	3.35×10^{2}	3.38×10^{2}	2.00×10^{2}	3.35×10^{2}	2.00×10^{2}
	Mean	3.20×10^{2}	2.63×10^{2}	$\mathbf{2.00\times10^{2}}$	3.09×10^{2}	$\mathbf{2.00\times10^{2}}$
	Std	4.53×10^{0}	6.44×10^{1}	0.00×10^{0}	3.71×10^{1}	0.00×10^{0}
F24	Best	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Mean	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$
	Std	9.81×10^{-5}	1.06×10^{-2}	0.00×10^{0}	5.64×10^{-5}	0.00×10^{0}
F25	Best	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	2.15×10^{2}	2.30×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Mean	2.05×10^{2}	2.01×10^{2}	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Std	5.48×10^{0}	5.90×10^{0}	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}
F26	Best	1.00×10^{2}	1.00×10^{2}	1.00×10^{2}	1.00×10^{2}	1.00×10^{2}
	Worst	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	1.01×10^{2}
	Mean	1.04×10^{2}	1.14×10^{2}	1.30×10^{2}	1.07×10^{2}	$\mathbf{1.01\times10^{2}}$
	Std	1.82×10^{1}	3.44×10^{1}	4.64×10^{1}	2.52×10^{1}	1.34×10^{-1}
F27	Best	4.02×10^{2}	2.00×10^{2}	2.00×10^{2}	4.01×10^{2}	2.00×10^{2}
	Worst	1.13×10^{3}	1.11×10^{3}	2.00×10^{2}	1.20×10^{3}	2.00×10^{2}
	Mean	4.42×10^{2}	6.06×10^{2}	$\mathbf{2.00\times10^{2}}$	6.22×10^{2}	$\mathbf{2.00\times10^{2}}$
	Std	1.31×10^{2}	3.95×10^{2}	0.00×10^{0}	3.15×10^{2}	0.00×10^{0}
F28	Best	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	2.33×10^{3}	2.29×10^{3}	2.00×10^{2}	1.92×10^{3}	2.00×10^{2}
	Mean	1.46×10^{3}	1.23×10^{3}	$\mathbf{2.00\times10^{2}}$	9.54×10^{2}	$\mathbf{2.00\times10^{2}}$
	Std	4.07×10^{2}	6.97×10^{2}	0.00×10^{0}	6.16×10^{2}	0.00×10^{0}
F29	Best	2.62×10^{3}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	4.01×10^{7}	2.95×10^{7}	2.00×10^{2}	3.35×10^{7}	2.00×10^{2}
	Mean	1.26×10^{7}	6.19×10^{6}	$\mathbf{2.00\times10^{2}}$	4.59×10^{6}	$\mathbf{2.00\times10^{2}}$
	Std	9.12×10^{6}	7.92×10^{6}	0.00×10^{0}	9.66×10^{6}	0.00×10^{0}
F30	Best	5.03×10^{3}	2.00×10^{2}	2.00×10^{2}	3.60×10^{3}	2.00×10^{2}
	Worst	1.80×10^{5}	2.37×10^{5}	4.08×10^{4}	1.98×10^{5}	2.00×10^{2}
	Mean	5.99×10^{4}	6.20×10^{4}	5.75×10^{3}	4.70×10^{4}	$\mathbf{2.00\times10^{2}}$
	Std	5.33×10^{4}	6.72×10^{4}	1.69×10^{4}	4.73×10^{4}	0.00×10^{0}
	+/≈/–	23/6/1	20/7/3	13/14/3	21/8/1	
	ARV	4.1500	3.4667	2.2167	3.4167	**1.7500**
	Rank	5	4	2	3	**1**

Table 5. Experimental results of DBO, QHDBO, IDBO, TDBO, and EDBO on CEC2014 at 50 dimensions.

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
F1	Best	6.61×10^{6}	2.31×10^{6}	1.16×10^{6}	7.64×10^{6}	9.89×10^{5}
	Worst	1.57×10^{8}	1.41×10^{8}	6.30×10^{6}	1.14×10^{8}	4.94×10^{6}
	Mean	6.04×10^{7}	3.48×10^{7}	2.45×10^{6}	4.89×10^{7}	$\mathbf{2.37\times10^{6}}$
	Std	3.60×10^{7}	3.85×10^{7}	9.80×10^{5}	2.80×10^{7}	9.04×10^{5}
F2	Best	2.13×10^{4}	2.81×10^{3}	1.14×10^{2}	1.41×10^{3}	3.03×10^{1}
	Worst	5.71×10^{7}	6.81×10^{8}	2.62×10^{4}	1.41×10^{8}	2.62×10^{4}
	Mean	1.29×10^{7}	2.27×10^{7}	9.24×10^{3}	1.87×10^{7}	$\mathbf{4.44\times10^{3}}$
	Std	1.30×10^{7}	1.24×10^{8}	1.48×10^{4}	2.86×10^{7}	7.58×10^{3}
F3	Best	1.16×10^{4}	2.74×10^{4}	2.01×10^{3}	3.78×10^{3}	1.14×10^{3}
	Worst	1.13×10^{5}	2.52×10^{5}	7.10×10^{4}	7.98×10^{4}	1.47×10^{4}
	Mean	5.61×10^{4}	1.66×10^{5}	1.37×10^{4}	2.17×10^{4}	$\mathbf{7.55\times10^{3}}$
	Std	2.24×10^{4}	4.69×10^{4}	1.34×10^{4}	2.15×10^{4}	3.47×10^{3}
F4	Best	1.08×10^{2}	7.70×10^{1}	9.42×10^{0}	7.88×10^{1}	2.14×10^{1}

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Worst	5.63×10^{2}	2.35×10^{2}	1.65×10^{2}	3.38×10^{2}	1.44×10^{2}
	Mean	2.51×10^{2}	1.44×10^{2}	1.08×10^{2}	2.11×10^{2}	$\mathbf{9.02\times10^{1}}$
	Std	9.66×10^{1}	4.57×10^{1}	3.23×10^{1}	6.84×10^{1}	1.99×10^{1}
F5	Best	2.04×10^{1}	2.09×10^{1}	2.03×10^{1}	2.08×10^{1}	2.00×10^{1}
	Worst	2.12×10^{1}	2.12×10^{1}	2.12×10^{1}	2.12×10^{1}	2.12×10^{1}
	Mean	2.11×10^{1}	2.11×10^{1}	2.11×10^{1}	2.11×10^{1}	$\mathbf{2.07\times10^{1}}$
	Std	2.20×10^{-1}	8.19×10^{-2}	2.24×10^{-1}	9.15×10^{-2}	4.45×10^{-1}
F6	Best	3.87×10^{1}	2.58×10^{1}	3.50×10^{1}	3.61×10^{1}	4.15×10^{1}
	Worst	6.72×10^{1}	5.84×10^{1}	5.68×10^{1}	6.20×10^{1}	5.75×10^{1}
	Mean	4.99×10^{1}	$\mathbf{4.40\times10^{1}}$	4.69×10^{1}	4.69×10^{1}	4.49×10^{1}
	Std	6.07×10^{0}	9.20×10^{0}	4.96×10^{0}	5.55×10^{0}	5.33×10^{0}
F7	Best	1.70×10^{-1}	3.59×10^{-1}	1.14×10^{-12}	2.96×10^{-12}	6.82×10^{-13}
	Worst	1.76×10^{0}	5.49×10^{0}	5.18×10^{-2}	4.67×10^{0}	1.72×10^{-20}
	Mean	1.08×10^{0}	1.03×10^{0}	6.24×10^{-2}	1.45×10^{0}	$\mathbf{4.76\times10^{-3}}$
	Std	3.06×10^{-1}	1.33×10^{0}	1.49×10^{-2}	1.02×10^{0}	5.96×10^{-3}
F8	Best	1.33×10^{2}	1.19×10^{2}	1.87×10^{2}	1.61×10^{2}	1.62×10^{2}
	Worst	4.76×10^{2}	2.61×10^{2}	3.12×10^{2}	2.73×10^{2}	3.07×10^{2}
	Mean	3.38×10^{2}	$\mathbf{1.95\times10^{2}}$	2.44×10^{2}	2.24×10^{2}	2.05×10^{2}
	Std	7.81×10^{1}	3.65×10^{1}	3.40×10^{1}	3.15×10^{1}	4.07×10^{1}
F9	Best	2.18×10^{2}	1.16×10^{2}	2.29×10^{2}	1.30×10^{2}	2.48×10^{2}
	Worst	6.01×10^{2}	5.34×10^{2}	4.24×10^{2}	3.67×10^{2}	4.45×10^{2}
	Mean	4.36×10^{2}	3.08×10^{2}	3.12×10^{2}	$\mathbf{2.67\times10^{2}}$	3.44×10^{2}
	Std	1.14×10^{2}	1.12×10^{2}	4.37×10^{1}	5.68×10^{1}	4.48×10^{1}
F10	Best	4.49×10^{3}	2.46×10^{3}	2.88×10^{3}	4.41×10^{3}	2.91×10^{3}
	Worst	9.76×10^{3}	8.89×10^{3}	8.19×10^{3}	8.41×10^{3}	8.22×10^{3}
	Mean	7.02×10^{3}	6.64×10^{3}	5.84×10^{3}	6.23×10^{3}	$\mathbf{5.69\times10^{3}}$
	Std	1.27×10^{3}	1.28×10^{3}	1.38×10^{3}	9.95×10^{2}	1.35×10^{3}
F11	Best	6.21×10^{3}	6.18×10^{3}	5.10×10^{3}	5.88×10^{3}	5.62×10^{3}
	Worst	1.15×10^{4}	1.41×10^{4}	1.11×10^{4}	1.13×10^{4}	9.07×10^{3}
	Mean	8.41×10^{3}	1.03×10^{4}	$\mathbf{7.17\times10^{3}}$	8.15×10^{3}	7.25×10^{3}
	Std	1.43×10^{3}	2.07×10^{3}	1.41×10^{3}	1.20×10^{3}	9.21×10^{2}
F12	Best	3.36×10^{-1}	1.35×10^{0}	5.23×10^{-1}	4.47×10^{-1}	2.45×10^{-1}
	Worst	4.00×10^{0}	3.78×10^{0}	3.23×10^{0}	3.92×10^{0}	2.95×10^{0}
	Mean	1.65×10^{0}	2.78×10^{0}	1.89×10^{0}	2.44×10^{0}	$\mathbf{7.65\times10^{-1}}$
	Std	1.20×10^{0}	7.68×10^{-1}	7.95×10^{-1}	1.30×10^{0}	2.40×10^{-1}
F13	Best	5.38×10^{-1}	5.73×10^{-1}	4.88×10^{-1}	4.98×10^{-1}	4.38×10^{-1}
	Worst	9.89×10^{-1}	1.05×10^{0}	9.75×10^{-1}	9.44×10^{-1}	9.47×10^{-1}
	Mean	7.59×10^{-1}	7.59×10^{-1}	7.17×10^{-1}	7.13×10^{-1}	$\mathbf{6.72\times10^{-1}}$
	Std	1.26×10^{-1}	1.12×10^{-1}	1.30×10^{-1}	1.13×10^{-1}	1.30×10^{-1}
F14	Best	2.40×10^{-1}	2.85×10^{-1}	2.30×10^{-1}	2.80×10^{-1}	2.47×10^{-1}
	Worst	1.32×10^{0}	1.16×10^{0}	1.08×10^{0}	1.38×10^{0}	1.06×10^{0}
	Mean	8.65×10^{-1}	4.71×10^{-1}	$\mathbf{4.03\times10^{-1}}$	7.22×10^{-1}	4.64×10^{-1}

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Std	3.60×10^{-1}	2.45×10^{-1}	2.16×10^{-1}	3.90×10^{-1}	2.63×10^{-1}
F15	Best	2.57×10^{1}	2.02×10^{1}	2.14×10^{1}	2.56×10^{1}	1.46×10^{1}
	Worst	6.82×10^{1}	5.82×10^{1}	1.16×10^{2}	8.79×10^{1}	7.01×10^{1}
	Mean	4.69×10^{1}	$\mathbf{3.79\times10^{1}}$	4.50×10^{1}	4.96×10^{1}	3.79×10^{1}
	Std	1.10×10^{1}	8.82×10^{0}	2.58×10^{1}	1.30×10^{1}	1.12×10^{1}
F16	Best	2.01×10^{1}	2.12×10^{1}	1.92×10^{1}	1.90×10^{1}	1.98×10^{1}
	Worst	2.28×10^{1}	2.32×10^{1}	2.22×10^{1}	2.26×10^{1}	2.25×10^{1}
	Mean	2.15×10^{1}	2.21×10^{1}	$\mathbf{2.09\times10^{1}}$	2.15×10^{1}	2.12×10^{1}
	Std	6.74×10^{-1}	4.83×10^{-1}	7.34×10^{-1}	8.78×10^{-1}	7.21×10^{-1}
F17	Best	5.43×10^{5}	1.14×10^{6}	2.71×10^{4}	1.14×10^{5}	1.06×10^{5}
	Worst	1.98×10^{7}	1.29×10^{8}	1.00×10^{8}	2.16×10^{7}	1.01×10^{6}
	Mean	9.54×10^{6}	1.68×10^{7}	3.81×10^{6}	6.34×10^{6}	$\mathbf{4.86\times10^{5}}$
	Std	5.94×10^{6}	2.56×10^{7}	1.83×10^{7}	5.80×10^{6}	2.37×10^{5}
F18	Best	2.18×10^{3}	1.54×10^{3}	8.35×10^{2}	1.48×10^{3}	6.83×10^{2}
	Worst	1.54×10^{5}	2.83×10^{8}	7.76×10^{3}	2.95×10^{6}	7.36×10^{3}
	Mean	1.87×10^{4}	9.45×10^{6}	3.73×10^{3}	2.10×10^{5}	$\mathbf{3.17\times10^{3}}$
	Std	3.92×10^{4}	5.17×10^{7}	1.80×10^{3}	6.20×10^{5}	1.56×10^{3}
F19	Best	2.46×10^{1}	1.61×10^{1}	1.44×10^{1}	2.23×10^{1}	1.68×10^{1}
	Worst	1.20×10^{2}	8.92×10^{1}	5.84×10^{1}	1.02×10^{2}	9.04×10^{1}
	Mean	5.10×10^{1}	3.24×10^{1}	3.15×10^{1}	4.48×10^{1}	$\mathbf{3.04\times10^{1}}$
	Std	2.76×10^{1}	1.46×10^{1}	1.02×10^{1}	2.72×10^{1}	2.45×10^{1}
F20	Best	4.74×10^{3}	2.10×10^{4}	1.92×10^{3}	2.10×10^{3}	2.38×10^{3}
	Worst	4.19×10^{4}	1.97×10^{5}	1.38×10^{5}	2.75×10^{4}	5.91×10^{4}
	Mean	2.03×10^{4}	7.50×10^{4}	2.55×10^{4}	$\mathbf{1.07\times10^{4}}$	1.72×10^{4}
	Std	9.73×10^{3}	4.00×10^{4}	3.19×10^{4}	7.58×10^{3}	1.72×10^{4}
F21	Best	2.10×10^{5}	3.22×10^{5}	4.60×10^{4}	1.02×10^{5}	9.85×10^{4}
	Worst	7.74×10^{6}	6.05×10^{6}	7.04×10^{5}	7.90×10^{6}	5.17×10^{5}
	Mean	3.26×10^{6}	2.25×10^{6}	3.34×10^{5}	2.31×10^{6}	$\mathbf{2.95\times10^{5}}$
	Std	2.35×10^{6}	1.74×10^{6}	1.84×10^{5}	2.39×10^{6}	1.30×10^{5}
F22	Best	8.96×10^{2}	5.43×10^{2}	8.55×10^{2}	5.84×10^{2}	7.18×10^{2}
	Worst	2.07×10^{3}	2.06×10^{3}	2.34×10^{3}	2.20×10^{3}	2.07×10^{3}
	Mean	1.44×10^{3}	1.46×10^{2}	1.60×10^{3}	1.40×10^{3}	$\mathbf{1.27\times10^{3}}$
	Std	3.26×10^{2}	3.33×10^{2}	4.40×10^{2}	4.17×10^{2}	4.12×10^{2}
F23	Best	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	4.08×10^{2}	2.00×10^{2}	2.00×10^{2}	3.98×10^{2}	2.00×10^{2}
	Mean	3.20×10^{2}	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$	2.65×10^{2}	$\mathbf{2.00\times10^{2}}$
	Std	6.62×10^{1}	8.32×10^{-13}	0.00×10^{0}	8.08×10^{1}	0.00×10^{0}
F24	Best	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	2.80×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Mean	2.03×10^{2}	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$
	Std	1.46×10^{1}	5.56×10^{-2}	0.00×10^{0}	2.94×10^{-5}	0.00×10^{0}
F25	Best	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Worst	2.45×10^{2}	2.20×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Mean	2.08×10^{2}	2.02×10^{2}	**2.00×10^{2}**	**2.00×10^{2}**	**2.00×10^{2}**
	Std	1.30×10^{1}	5.20×10^{0}	0.00×10^{0}	0.00×10^{0}	0.00×10^{0}
F26	Best	1.01×10^{2}	1.00×10^{2}	1.01×10^{2}	1.01×10^{2}	2.00×10^{2}
	Worst	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Mean	**1.07×10^{2}**	1.24×10^{2}	1.77×10^{2}	1.50×10^{2}	2.00×10^{2}
	Std	2.52×10^{1}	4.28×10^{1}	4.64×10^{1}	5.04×10^{1}	0.00×10^{0}
F27	Best	1.35×10^{3}	2.00×10^{2}	2.00×10^{2}	1.41×10^{3}	2.00×10^{2}
	Worst	1.97×10^{3}	1.86×10^{3}	2.00×10^{2}	1.94×10^{3}	2.00×10^{2}
	Mean	1.66×10^{3}	1.00×10^{3}	**2.00×10^{2}**	1.65×10^{3}	**2.00×10^{2}**
	Std	1.47×10^{2}	7.22×10^{2}	0.00×10^{0}	1.34×10^{2}	0.00×10^{0}
F28	Best	1.76×10^{3}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	5.20×10^{3}	4.57×10^{3}	2.00×10^{2}	4.49×10^{3}	2.00×10^{2}
	Mean	3.03×10^{3}	1.78×10^{3}	**2.00×10^{2}**	2.38×10^{3}	**2.00×10^{2}**
	Std	8.16×10^{2}	1.52×10^{3}	0.00×10^{0}	1.13×10^{3}	0.00×10^{0}
F29	Best	1.05×10^{5}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	1.42×10^{8}	1.66×10^{8}	2.00×10^{2}	1.08×10^{8}	2.00×10^{2}
	Mean	6.91×10^{7}	5.71×10^{7}	**2.00×10^{2}**	5.65×10^{6}	**2.00×10^{2}**
	Std	3.52×10^{7}	4.28×10^{7}	0.00×10^{0}	2.18×10^{7}	0.00×10^{0}
F30	Best	2.37×10^{4}	2.00×10^{2}	2.00×10^{2}	2.03×10^{4}	2.00×10^{2}
	Worst	2.32×10^{6}	1.05×10^{6}	7.46×10^{4}	1.29×10^{6}	2.00×10^{2}
	Mean	2.89×10^{5}	3.24×10^{5}	8.26×10^{3}	2.47×10^{5}	**2.00×10^{2}**
	Std	4.32×10^{5}	2.54×10^{5}	2.07×10^{4}	2.86×10^{5}	0.00×10^{0}
	+/≈/–	26/3/1	23/6/1	12/15/3	19/9/2	
	ARV	4.1833	3.5000	2.4333	3.3000	**1.5833**
	Rank	5	4	2	3	**1**

5.2.2. Analysis of CEC2017 statistical results

Further analysis of the EDBO algorithm is conducted on the CEC2017 benchmark functions in 30D and 50D. The experimental environment and parameter settings remain consistent with those in the previous section. In addition, Wilcoxon's rank-sum test and Friedman test are employed to further analyze the performance differences between the algorithms.

The simulation results of the 30-dimensional test functions are shown in Table 6. In terms of solution accuracy, EDBO achieves the best results in 18 of the 29 functions, including F1, F3, F4, F10–F12, F14–F19, F22, F24, and F27–F30. Similarly, QHDBO provides promising solutions for F6–F9, F21, F23, and F25. IDBO performs better in F5, F13, F20, and F26. On the other hand, for the 50-dimensional functions, the results presented in Table 7 indicate that EDBO achieves superior optimization performance compared to the other DBO algorithms. Specifically, EDBO performs best in 20 functions, including F1, F4, F10–F16, F18, F19, and F22–F30. Meanwhile, QHDBO performs well in F5–F7 and F11. IDBO surpasses other algorithms in F17 and F21, but lacks competitiveness in F12, F23, F24, and F27. TDBO produces high-quality solutions in F3, F8, F9, and F20.

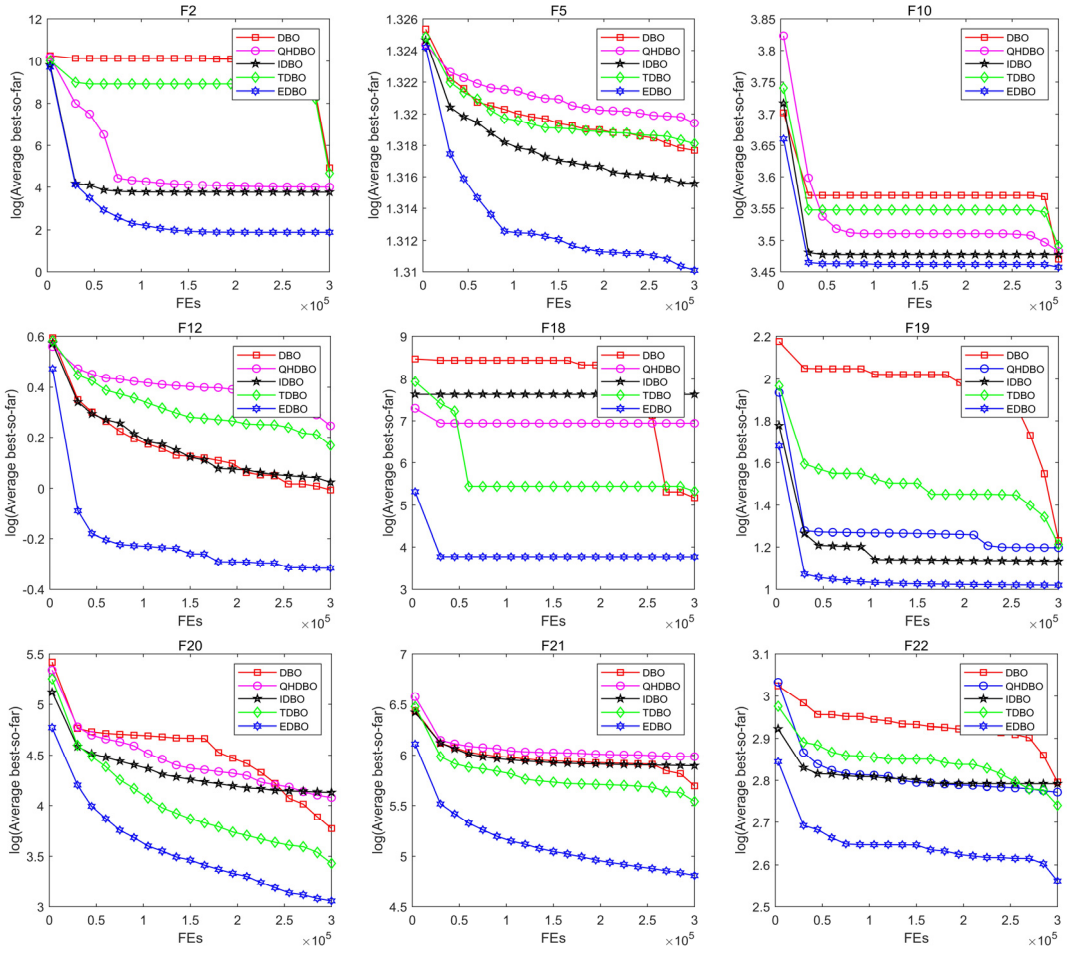


Figure 6. Convergence curves of DBO, QHDBO, IDBO, TDBO, and EDBO on CEC 2014 at 30 dimensions.

Table 6. Experimental results of DBO, QHDBO, IDBO, TDBO, and EDBO on CEC2017 at 30 dimensions.

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
F1	Best	9.69×10^{0}	1.15×10^{3}	4.07×10^{1}	2.16×10^{0}	1.01×10^{0}
	Worst	7.38×10^{5}	8.64×10^{4}	3.30×10^{6}	1.18×10^{6}	2.01×10^{4}
	Mean	1.13×10^{5}	1.80×10^{4}	1.21×10^{5}	1.29×10^{5}	$\mathbf{6.79\times10^{3}}$
	Std	1.90×10^{5}	2.14×10^{4}	6.00×10^{5}	3.02×10^{5}	5.75×10^{3}
F3	Best	1.88×10^{3}	1.60×10^{4}	2.56×10^{2}	7.58×10^{2}	8.98×10^{0}
	Worst	2.69×10^{4}	5.44×10^{4}	1.56×10^{4}	2.14×10^{4}	3.75×10^{3}
	Mean	9.82×10^{3}	3.37×10^{4}	3.04×10^{3}	6.56×10^{3}	$\mathbf{1.13\times10^{3}}$
	Std	6.34×10^{3}	9.17×10^{3}	2.90×10^{3}	5.21×10^{3}	1.14×10^{3}
F4	Best	5.06×10^{1}	6.86×10^{1}	1.26×10^{-1}	2.70×10^{1}	9.62×10^{-2}
	Worst	2.01×10^{2}	1.39×10^{2}	2.42×10^{2}	2.32×10^{2}	1.14×10^{2}
	Mean	1.09×10^{2}	9.76×10^{1}	1.05×10^{2}	1.09×10^{2}	$\mathbf{5.48\times10^{1}}$
	Std	3.25×10^{1}	1.84×10^{1}	4.71×10^{1}	3.77×10^{1}	3.68×10^{1}
F5	Best	9.18×10^{1}	6.88×10^{1}	7.07×10^{1}	7.45×10^{1}	9.45×10^{1}

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Worst	2.70×10^{2}	2.35×10^{2}	1.67×10^{2}	2.62×10^{2}	2.44×10^{2}
	Mean	2.00×10^{2}	1.28×10^{2}	$\mathbf{1.26\times10^{2}}$	1.51×10^{2}	1.69×10^{2}
	Std	4.56×10^{1}	5.03×10^{1}	2.40×10^{1}	3.93×10^{1}	3.78×10^{1}
F6	Best	3.12×10^{0}	2.71×10^{-1}	5.62×10^{0}	2.37×10^{0}	7.81×10^{0}
	Worst	6.05×10^{1}	2.29×10^{1}	3.52×10^{1}	5.17×10^{1}	4.82×10^{1}
	Mean	3.15×10^{1}	$\mathbf{5.39\times10^{0}}$	1.79×10^{1}	1.90×10^{1}	2.11×10^{1}
	Std	1.49×10^{1}	5.88×10^{0}	8.80×10^{0}	1.24×10^{1}	1.05×10^{1}
F7	Best	8.95×10^{1}	8.12×10^{1}	1.08×10^{2}	1.02×10^{2}	1.47×10^{2}
	Worst	3.66×10^{2}	2.51×10^{2}	3.75×10^{2}	3.47×10^{2}	4.19×10^{2}
	Mean	1.88×10^{2}	$\mathbf{1.61\times10^{2}}$	2.16×10^{2}	2.12×10^{2}	2.32×10^{2}
	Std	6.42×10^{1}	5.22×10^{1}	6.96×10^{1}	4.73×10^{1}	6.30×10^{1}
F8	Best	7.20×10^{1}	5.17×10^{1}	5.77×10^{1}	6.17×10^{1}	8.66×10^{1}
	Worst	2.66×10^{2}	2.34×10^{2}	1.88×10^{2}	1.94×10^{2}	2.55×10^{2}
	Mean	1.81×10^{2}	$\mathbf{1.17\times10^{2}}$	1.21×10^{2}	1.20×10^{2}	1.52×10^{2}
	Std	5.47×10^{1}	4.89×10^{1}	3.16×10^{1}	3.68×10^{1}	4.03×10^{1}
F9	Best	5.34×10^{2}	1.54×10^{1}	3.12×10^{2}	1.04×10^{2}	5.39×10^{2}
	Worst	5.56×10^{3}	1.83×10^{3}	4.46×10^{3}	5.36×10^{3}	4.81×10^{3}
	Mean	2.62×10^{3}	$\mathbf{2.82\times10^{2}}$	1.69×10^{3}	1.73×10^{3}	2.54×10^{3}
	Std	1.48×10^{3}	4.03×10^{2}	9.03×10^{2}	1.19×10^{3}	1.16×10^{3}
F10	Best	2.48×10^{3}	3.65×10^{3}	2.80×10^{3}	2.74×10^{3}	2.24×10^{3}
	Worst	6.12×10^{3}	7.24×10^{3}	6.38×10^{3}	6.25×10^{3}	5.76×10^{3}
	Mean	4.36×10^{3}	5.03×10^{3}	4.28×10^{3}	4.21×10^{3}	$\mathbf{4.00\times10^{3}}$
	Std	8.64×10^{2}	8.42×10^{2}	8.11×10^{2}	8.29×10^{2}	7.81×10^{2}
F11	Best	2.07×10^{2}	1.55×10^{2}	1.22×10^{2}	1.33×10^{2}	1.31×10^{2}
	Worst	5.54×10^{2}	5.13×10^{3}	5.38×10^{2}	4.54×10^{2}	3.98×10^{2}
	Mean	3.48×10^{2}	6.20×10^{2}	2.85×10^{2}	2.84×10^{2}	$\mathbf{2.45\times10^{2}}$
	Std	8.61×10^{1}	9.14×10^{2}	1.11×10^{2}	9.93×10^{1}	7.13×10^{1}
F12	Best	3.64×10^{4}	9.38×10^{4}	2.26×10^{4}	1.49×10^{5}	1.45×10^{4}
	Worst	1.05×10^{8}	3.84×10^{7}	2.37×10^{7}	8.70×10^{7}	1.10×10^{5}
	Mean	1.32×10^{7}	3.73×10^{6}	6.48×10^{6}	1.54×10^{7}	$\mathbf{5.22\times10^{4}}$
	Std	2.02×10^{7}	7.30×10^{6}	7.06×10^{6}	2.30×10^{7}	2.06×10^{4}
F13	Best	1.10×10^{4}	5.71×10^{3}	2.14×10^{4}	2.57×10^{4}	9.89×10^{3}
	Worst	2.85×10^{6}	7.19×10^{7}	3.52×10^{6}	3.13×10^{6}	4.66×10^{6}
	Mean	5.71×10^{5}	2.64×10^{6}	$\mathbf{4.79\times10^{5}}$	6.00×10^{5}	4.95×10^{5}
	Std	9.22×10^{5}	1.31×10^{7}	8.12×10^{5}	9.98×10^{5}	1.37×10^{6}
F14	Best	5.51×10^{2}	1.19×10^{3}	1.67×10^{3}	5.55×10^{2}	3.10×10^{3}
	Worst	1.35×10^{5}	4.63×10^{5}	2.51×10^{5}	2.66×10^{5}	3.57×10^{4}
	Mean	3.72×10^{4}	1.17×10^{5}	3.90×10^{4}	4.34×10^{4}	$\mathbf{1.65\times10^{4}}$
	Std	3.54×10^{4}	1.32×10^{5}	5.36×10^{4}	6.24×10^{4}	9.00×10^{3}
F15	Best	6.61×10^{3}	1.35×10^{3}	2.06×10^{3}	4.64×10^{2}	5.33×10^{2}
	Worst	2.35×10^{5}	6.92×10^{4}	1.58×10^{5}	1.85×10^{5}	4.27×10^{4}
	Mean	7.14×10^{4}	3.08×10^{4}	2.83×10^{4}	4.66×10^{4}	$\mathbf{1.35\times10^{4}}$

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Std	5.90×10^{4}	1.93×10^{4}	3.08×10^{4}	4.63×10^{4}	1.36×10^{4}
F16	Best	5.51×10^{2}	2.73×10^{2}	2.23×10^{2}	4.31×10^{2}	2.49×10^{2}
	Worst	2.03×10^{3}	1.86×10^{3}	1.64×10^{3}	1.74×10^{3}	1.64×10^{3}
	Mean	1.29×10^{3}	1.26×10^{3}	1.10×10^{3}	1.25×10^{3}	$\mathbf{9.87\times10^{2}}$
	Std	3.73×10^{2}	3.47×10^{2}	3.27×10^{2}	2.76×10^{2}	3.71×10^{2}
F17	Best	2.43×10^{2}	2.01×10^{2}	8.01×10^{1}	1.74×10^{2}	2.13×10^{2}
	Worst	1.11×10^{3}	1.15×10^{3}	8.64×10^{2}	9.18×10^{2}	7.85×10^{2}
	Mean	5.98×10^{2}	5.87×10^{2}	4.83×10^{2}	5.47×10^{2}	$\mathbf{4.30\times10^{2}}$
	Std	2.21×10^{2}	2.62×10^{2}	2.08×10^{2}	1.99×10^{2}	1.42×10^{2}
F18	Best	7.93×10^{4}	1.69×10^{5}	5.66×10^{4}	2.07×10^{4}	3.62×10^{4}
	Worst	5.34×10^{6}	5.90×10^{6}	2.64×10^{6}	3.37×10^{6}	8.51×10^{5}
	Mean	7.80×10^{5}	1.22×10^{6}	5.27×10^{5}	5.61×10^{5}	$\mathbf{2.01\times10^{5}}$
	Std	1.07×10^{6}	1.18×10^{6}	5.96×10^{5}	6.76×10^{5}	1.67×10^{5}
F19	Best	1.79×10^{3}	7.19×10^{2}	9.63×10^{1}	4.65×10^{2}	1.40×10^{2}
	Worst	1.08×10^{6}	1.51×10^{6}	1.64×10^{6}	3.33×10^{6}	6.12×10^{4}
	Mean	2.00×10^{5}	8.70×10^{4}	2.00×10^{5}	2.62×10^{5}	$\mathbf{7.04\times10^{3}}$
	Std	2.97×10^{5}	2.78×10^{5}	3.53×10^{5}	6.40×10^{5}	1.24×10^{4}
F20	Best	2.11×10^{2}	1.55×10^{2}	1.75×10^{2}	2.11×10^{2}	1.71×10^{2}
	Worst	1.04×10^{3}	9.02×10^{2}	7.88×10^{2}	7.24×10^{2}	8.79×10^{2}
	Mean	5.45×10^{2}	4.90×10^{2}	$\mathbf{4.26\times10^{2}}$	4.41×10^{2}	4.47×10^{2}
	Std	2.15×10^{2}	1.91×10^{2}	1.61×10^{2}	1.32×10^{2}	1.87×10^{2}
F21	Best	1.80×10^{2}	2.45×10^{2}	2.66×10^{2}	2.56×10^{2}	2.95×10^{2}
	Worst	4.71×10^{2}	4.79×10^{2}	4.35×10^{2}	4.80×10^{2}	4.88×10^{2}
	Mean	3.81×10^{2}	$\mathbf{3.44\times10^{2}}$	3.51×10^{2}	3.50×10^{2}	3.67×10^{2}
	Std	6.84×10^{1}	5.49×10^{1}	4.41×10^{1}	5.85×10^{1}	5.69×10^{1}
F22	Best	1.00×10^{2}	1.00×10^{2}	1.00×10^{2}	1.00×10^{2}	1.00×10^{2}
	Worst	4.22×10^{3}	6.41×10^{3}	2.22×10^{3}	4.86×10^{3}	6.06×10^{3}
	Mean	1.12×10^{3}	1.68×10^{3}	2.68×10^{2}	5.60×10^{2}	$\mathbf{1.00\times10^{2}}$
	Std	1.32×10^{3}	2.33×10^{3}	4.90×10^{2}	1.23×10^{3}	7.36×10^{-1}
F23	Best	4.77×10^{2}	4.19×10^{2}	4.40×10^{2}	4.95×10^{2}	4.63×10^{2}
	Worst	7.05×10^{2}	5.85×10^{2}	7.19×10^{2}	7.08×10^{2}	6.01×10^{2}
	Mean	5.95×10^{2}	$\mathbf{4.85\times10^{2}}$	6.02×10^{2}	5.77×10^{2}	5.17×10^{2}
	Std	6.12×10^{1}	4.37×10^{1}	7.09×10^{1}	5.19×10^{1}	4.07×10^{1}
F24	Best	5.41×10^{2}	4.94×10^{2}	4.81×10^{2}	4.99×10^{2}	4.90×10^{2}
	Worst	7.86×10^{2}	7.36×10^{2}	8.43×10^{2}	7.89×10^{2}	7.10×10^{2}
	Mean	6.41×10^{2}	6.06×10^{2}	6.49×10^{2}	6.31×10^{2}	$\mathbf{5.89\times10^{2}}$
	Std	6.13×10^{1}	5.48×10^{1}	8.96×10^{1}	7.51×10^{1}	5.61×10^{1}
F25	Best	3.86×10^{2}	3.83×10^{2}	3.84×10^{2}	3.84×10^{2}	3.84×10^{2}
	Worst	4.79×10^{2}	3.97×10^{2}	4.72×10^{2}	5.13×10^{2}	4.43×10^{2}
	Mean	4.10×10^{2}	$\mathbf{3.87\times10^{2}}$	4.00×10^{2}	4.02×10^{2}	3.90×10^{2}
	Std	2.74×10^{1}	4.30×10^{0}	2.26×10^{1}	2.88×10^{1}	1.12×10^{1}
F26	Best	6.04×10^{2}	7.06×10^{2}	2.18×10^{2}	2.00×10^{2}	2.00×10^{2}

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Worst	4.71×10^{3}	4.28×10^{3}	3.87×10^{3}	4.17×10^{3}	3.90×10^{3}
	Mean	3.20×10^{3}	2.50×10^{3}	$\mathbf{2.49\times10^{3}}$	2.70×10^{3}	2.54×10^{3}
	Std	8.63×10^{2}	7.05×10^{2}	1.09×10^{3}	9.83×10^{2}	1.43×10^{3}
F27	Best	5.02×10^{2}	5.19×10^{2}	5.04×10^{2}	4.99×10^{2}	5.03×10^{2}
	Worst	6.76×10^{2}	6.51×10^{2}	7.54×10^{2}	7.36×10^{2}	5.52×10^{2}
	Mean	5.53×10^{2}	5.69×10^{2}	5.72×10^{2}	5.62×10^{2}	$\mathbf{5.24\times10^{2}}$
	Std	3.94×10^{1}	3.67×10^{1}	5.03×10^{1}	4.51×10^{1}	1.01×10^{1}
F28	Best	4.27×10^{2}	3.93×10^{2}	4.05×10^{2}	3.70×10^{2}	3.00×10^{2}
	Worst	6.56×10^{2}	3.71×10^{3}	6.37×10^{2}	6.36×10^{2}	4.56×10^{2}
	Mean	5.12×10^{2}	5.63×10^{2}	4.63×10^{2}	4.57×10^{2}	$\mathbf{3.45\times10^{2}}$
	Std	6.77×10^{1}	5.97×10^{2}	4.76×10^{1}	4.83×10^{1}	6.39×10^{1}
F29	Best	6.82×10^{2}	7.72×10^{2}	6.00×10^{2}	6.27×10^{2}	6.33×10^{2}
	Worst	1.94×10^{3}	2.32×10^{3}	1.43×10^{3}	2.02×10^{3}	1.35×10^{3}
	Mean	1.19×10^{3}	1.25×10^{3}	1.05×10^{3}	1.07×10^{3}	$\mathbf{1.01\times10^{3}}$
	Std	3.08×10^{2}	3.09×10^{2}	2.13×10^{2}	2.95×10^{2}	2.02×10^{2}
F30	Best	2.42×10^{3}	1.02×10^{4}	1.30×10^{4}	1.23×10^{4}	2.24×10^{3}
	Worst	9.59×10^{6}	8.39×10^{6}	7.26×10^{6}	8.68×10^{6}	3.99×10^{4}
	Mean	1.13×10^{6}	5.35×10^{5}	9.29×10^{5}	1.62×10^{6}	$\mathbf{9.99\times10^{3}}$
	Std	1.93×10^{6}	1.56×10^{6}	1.71×10^{6}	2.68×10^{6}	6.73×10^{3}
	+/≈/–	20/8/1	16/5/8	15/10/4	15/10/4	
	ARV	4.1034	3.0000	2.7069	3.2931	**1.8966**
	Rank	5	3	2	4	**1**

From the average ranking, for the 30-dimensional test functions, EDBO ranks first with a score of 1.8966, followed by IDBO, QHDBO, TDBO, and DBO. In the 50-dimensional case, EDBO still ranks first with a score of 1.8793, followed by IDBO, TDBO, QHDBO, and DBO. On the other hand, the results of the Wilcoxon's rank-sum test indicate that EDBO has a significant advantage over the other algorithms. Specifically, for the 30-dimensional test functions, EDBO significantly is superior to DBO, QHDBO, IDBO, and TDBO in 20, 16, 15, and 15 functions, respectively, while performing worse in 1, 8, 4, and 4 functions. For the 50-dimensional test functions, EDBO outperforms DBO, QHDBO, IAEFA, and TDBO in at least 18 functions. Therefore, EDBO is identified as the best-performing algorithm among these DBO variants. In conclusion, EDBO demonstrates reliability and effectiveness in solving complex benchmark problems.

Table 7. Experimental results of DBO, QHDBO, IDBO, TDBO, and EDBO on CEC2017 at 50 dimensions.

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
F1	Best	3.96×10^{3}	1.44×10^{5}	8.85×10^{2}	5.88×10^{0}	7.49×10^{0}
	Worst	5.57×10^{7}	1.89×10^{6}	4.20×10^{7}	3.80×10^{7}	3.17×10^{4}
	Mean	1.11×10^{7}	5.41×10^{5}	1.03×10^{7}	9.77×10^{6}	$\mathbf{5.19\times10^{3}}$
	Std	1.46×10^{7}	4.75×10^{5}	1.15×10^{7}	1.07×10^{7}	7.31×10^{3}
F3	Best	6.89×10^{4}	1.42×10^{5}	8.46×10^{4}	5.66×10^{4}	6.53×10^{4}

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Worst	1.54×10^{5}	3.13×10^{5}	1.60×10^{5}	1.66×10^{5}	1.85×10^{5}
	Mean	1.23×10^{5}	1.99×10^{5}	1.21×10^{5}	$\mathbf{1.15\times10^{5}}$	1.35×10^{5}
	Std	2.31×10^{4}	4.00×10^{4}	2.17×10^{4}	2.49×10^{4}	3.27×10^{4}
F4	Best	1.74×10^{2}	6.94×10^{1}	8.39×10^{1}	7.57×10^{1}	2.16×10^{1}
	Worst	5.70×10^{2}	3.02×10^{2}	4.27×10^{2}	3.01×10^{2}	2.04×10^{2}
	Mean	2.62×10^{2}	1.83×10^{2}	2.14×10^{2}	1.98×10^{2}	$\mathbf{7.22\times10^{1}}$
	Std	8.93×10^{1}	4.99×10^{1}	8.38×10^{1}	5.80×10^{1}	4.84×10^{1}
F5	Best	1.61×10^{2}	1.40×10^{2}	2.19×10^{2}	1.57×10^{2}	1.97×10^{2}
	Worst	6.46×10^{2}	3.71×10^{2}	3.94×10^{2}	3.73×10^{2}	4.36×10^{2}
	Mean	4.22×10^{2}	$\mathbf{2.64\times10^{2}}$	2.98×10^{2}	2.68×10^{2}	3.33×10^{2}
	Std	1.10×10^{2}	6.15×10^{1}	4.59×10^{1}	4.88×10^{1}	5.02×10^{1}
F6	Best	1.91×10^{1}	4.60×10^{0}	2.71×10^{1}	1.97×10^{1}	2.33×10^{1}
	Worst	7.67×10^{1}	6.24×10^{1}	6.18×10^{1}	5.97×10^{1}	7.07×10^{1}
	Mean	5.44×10^{1}	$\mathbf{2.04\times10^{1}}$	4.08×10^{1}	3.52×10^{1}	5.04×10^{1}
	Std	1.32×10^{1}	1.36×10^{1}	8.76×10^{0}	9.63×10^{0}	9.12×10^{0}
F7	Best	2.51×10^{2}	2.51×10^{2}	2.88×10^{2}	2.32×10^{2}	3.78×10^{2}
	Worst	8.64×10^{2}	5.83×10^{2}	7.77×10^{2}	8.31×10^{2}	1.02×10^{3}
	Mean	5.17×10^{2}	$\mathbf{3.76\times10^{2}}$	5.14×10^{2}	5.45×10^{2}	5.48×10^{2}
	Std	1.26×10^{2}	8.18×10^{1}	1.20×10^{2}	1.35×10^{2}	1.47×10^{2}
F8	Best	1.90×10^{2}	1.68×10^{2}	1.15×10^{2}	1.67×10^{2}	2.35×10^{2}
	Worst	5.93×10^{2}	5.56×10^{2}	4.02×10^{2}	3.79×10^{2}	4.23×10^{2}
	Mean	4.19×10^{2}	3.34×10^{2}	2.92×10^{2}	$\mathbf{2.86\times10^{2}}$	3.43×10^{2}
	Std	9.04×10^{1}	1.04×10^{2}	5.94×10^{1}	5.08×10^{1}	5.22×10^{1}
F9	Best	2.61×10^{3}	9.25×10^{2}	1.70×10^{3}	2.30×10^{3}	3.18×10^{3}
	Worst	3.16×10^{4}	1.92×10^{4}	1.48×10^{4}	1.09×10^{4}	1.47×10^{4}
	Mean	1.50×10^{4}	5.54×10^{3}	5.85×10^{3}	$\mathbf{5.28\times10^{3}}$	9.42×10^{3}
	Std	7.03×10^{3}	4.30×10^{3}	3.41×10^{3}	2.47×10^{3}	2.58×10^{3}
F10	Best	5.93×10^{3}	7.13×10^{3}	5.60×10^{3}	5.37×10^{3}	4.55×10^{3}
	Worst	1.11×10^{4}	1.39×10^{4}	1.29×10^{4}	1.26×10^{4}	8.03×10^{3}
	Mean	8.33×10^{3}	1.02×10^{4}	7.79×10^{3}	8.20×10^{3}	$\mathbf{6.68\times10^{3}}$
	Std	1.20×10^{3}	1.80×10^{3}	1.74×10^{3}	1.70×10^{3}	7.97×10^{2}
F11	Best	2.96×10^{2}	1.71×10^{2}	2.55×10^{2}	1.91×10^{2}	2.47×10^{2}
	Worst	1.03×10^{3}	6.35×10^{2}	8.38×10^{2}	7.46×10^{2}	6.36×10^{2}
	Mean	6.51×10^{2}	$\mathbf{3.88\times10^{2}}$	5.07×10^{2}	4.46×10^{2}	$\mathbf{3.88\times10^{2}}$
	Std	1.73×10^{2}	1.16×10^{2}	1.50×10^{2}	1.44×10^{2}	1.12×10^{2}
F12	Best	4.31×10^{6}	5.39×10^{5}	7.13×10^{6}	5.15×10^{5}	1.13×10^{5}
	Worst	3.02×10^{8}	3.34×10^{8}	3.38×10^{8}	5.51×10^{8}	3.66×10^{6}
	Mean	1.38×10^{8}	5.56×10^{7}	1.15×10^{8}	1.04×10^{8}	$\mathbf{1.55\times10^{6}}$
	Std	8.61×10^{7}	8.09×10^{7}	8.53×10^{7}	1.21×10^{8}	9.49×10^{5}
F13	Best	8.68×10^{4}	2.90×10^{4}	4.92×10^{4}	4.66×10^{4}	3.49×10^{3}
	Worst	8.19×10^{6}	2.46×10^{8}	7.21×10^{6}	9.16×10^{6}	9.96×10^{4}
	Mean	1.50×10^{6}	9.03×10^{6}	1.33×10^{6}	1.66×10^{6}	$\mathbf{2.40\times10^{4}}$

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Std	1.75×10^{6}	4.49×10^{7}	1.65×10^{6}	2.05×10^{6}	2.07×10^{4}
F14	Best	2.89×10^{4}	2.38×10^{4}	7.83×10^{3}	1.54×10^{4}	5.62×10^{3}
	Worst	3.24×10^{6}	1.20×10^{7}	1.42×10^{6}	2.28×10^{6}	1.87×10^{5}
	Mean	8.14×10^{5}	1.24×10^{6}	3.94×10^{5}	6.46×10^{5}	$\mathbf{5.86\times10^{4}}$
	Std	9.46×10^{5}	2.19×10^{6}	4.38×10^{5}	6.11×10^{5}	5.69×10^{4}
F15	Best	2.94×10^{4}	1.06×10^{4}	3.11×10^{4}	1.54×10^{4}	1.17×10^{3}
	Worst	3.52×10^{6}	7.13×10^{7}	1.92×10^{5}	3.55×10^{6}	4.27×10^{4}
	Mean	2.58×10^{5}	2.45×10^{6}	9.84×10^{4}	2.27×10^{5}	$\mathbf{2.35\times10^{4}}$
	Std	6.24×10^{5}	1.30×10^{7}	4.41×10^{4}	6.31×10^{5}	7.76×10^{3}
F16	Best	1.62×10^{3}	1.39×10^{3}	1.71×10^{3}	1.71×10^{3}	1.52×10^{3}
	Worst	3.16×10^{3}	3.68×10^{3}	3.16×10^{3}	3.50×10^{3}	2.86×10^{3}
	Mean	2.58×10^{5}	2.45×10^{6}	9.84×10^{4}	2.27×10^{5}	$\mathbf{2.35\times10^{4}}$
	Std	6.24×10^{5}	1.30×10^{7}	4.41×10^{4}	6.31×10^{5}	7.76×10^{3}
F17	Best	1.35×10^{3}	1.00×10^{3}	9.64×10^{2}	1.01×10^{3}	1.08×10^{3}
	Worst	2.57×10^{3}	2.40×10^{3}	2.24×10^{3}	2.63×10^{3}	2.42×10^{3}
	Mean	2.03×10^{3}	1.89×10^{3}	$\mathbf{1.71\times10^{3}}$	1.83×10^{3}	1.74×10^{3}
	Std	4.03×10^{2}	3.34×10^{2}	3.23×10^{2}	4.35×10^{2}	4.16×10^{2}
F18	Best	1.35×10^{5}	3.37×10^{5}	1.06×10^{5}	1.72×10^{5}	7.73×10^{4}
	Worst	8.93×10^{6}	9.13×10^{6}	1.14×10^{7}	1.68×10^{7}	1.56×10^{6}
	Mean	2.22×10^{6}	3.95×10^{6}	1.51×10^{6}	4.33×10^{6}	$\mathbf{3.89\times10^{5}}$
	Std	2.46×10^{6}	2.62×10^{6}	2.59×10^{6}	4.96×10^{6}	1.82×10^{5}
F19	Best	3.25×10^{3}	1.02×10^{3}	8.54×10^{3}	1.13×10^{4}	4.07×10^{2}
	Worst	5.46×10^{6}	6.98×10^{6}	4.78×10^{6}	8.35×10^{6}	4.39×10^{4}
	Mean	1.30×10^{6}	3.24×10^{5}	7.94×10^{5}	1.58×10^{6}	$\mathbf{2.18\times10^{4}}$
	Std	1.41×10^{6}	1.27×10^{6}	1.21×10^{6}	2.33×10^{6}	1.66×10^{4}
F20	Best	7.86×10^{2}	6.87×10^{2}	5.08×10^{2}	3.93×10^{2}	6.27×10^{2}
	Worst	2.13×10^{3}	2.34×10^{3}	1.94×10^{3}	1.80×10^{3}	1.75×10^{3}
	Mean	1.52×10^{3}	1.45×10^{3}	1.18×10^{3}	$\mathbf{1.02\times10^{3}}$	1.21×10^{3}
	Std	3.46×10^{2}	4.37×10^{2}	3.34×10^{2}	3.68×10^{2}	3.11×10^{2}
F21	Best	3.95×10^{2}	3.45×10^{2}	3.71×10^{2}	4.02×10^{2}	3.88×10^{2}
	Worst	8.20×10^{2}	7.42×10^{2}	6.53×10^{2}	7.28×10^{2}	6.57×10^{2}
	Mean	6.48×10^{2}	5.47×10^{2}	$\mathbf{4.91\times10^{2}}$	5.09×10^{2}	5.21×10^{2}
	Std	1.05×10^{2}	1.14×10^{2}	7.78×10^{1}	8.41×10^{1}	6.79×10^{1}
F22	Best	6.58×10^{3}	7.58×10^{3}	1.34×10^{2}	1.00×10^{2}	5.67×10^{3}
	Worst	1.09×10^{4}	1.38×10^{4}	1.02×10^{4}	1.37×10^{4}	8.29×10^{3}
	Mean	8.78×10^{3}	1.12×10^{4}	7.82×10^{3}	8.41×10^{3}	$\mathbf{7.38\times10^{3}}$
	Std	1.15×10^{3}	1.71×10^{3}	2.32×10^{3}	2.28×10^{3}	5.56×10^{2}
F23	Best	7.35×10^{2}	5.93×10^{2}	7.70×10^{2}	6.33×10^{2}	6.82×10^{2}
	Worst	1.35×10^{3}	1.10×10^{3}	1.23×10^{3}	1.25×10^{3}	1.02×10^{3}
	Mean	1.02×10^{3}	7.79×10^{2}	1.03×10^{3}	8.83×10^{2}	$\mathbf{8.21\times10^{2}}$
	Std	1.24×10^{2}	1.09×10^{2}	1.38×10^{2}	1.32×10^{2}	8.73×10^{1}
F24	Best	7.43×10^{2}	7.38×10^{2}	7.31×10^{2}	7.69×10^{2}	6.93×10^{2}

Continued on next page

Function	Metric	DBO	QHDBO	IDBO	TDBO	EDBO
	Worst	1.24×10^{3}	9.78×10^{2}	1.32×10^{3}	1.31×10^{3}	1.13×10^{3}
	Mean	1.04×10^{3}	8.79×10^{2}	1.00×10^{3}	9.71×10^{2}	$\mathbf{8.77\times10^{2}}$
	Std	1.13×10^{2}	6.77×10^{1}	1.36×10^{2}	1.41×10^{2}	1.12×10^{2}
F25	Best	4.84×10^{2}	5.21×10^{2}	4.61×10^{2}	4.85×10^{2}	4.68×10^{2}
	Worst	7.30×10^{2}	6.30×10^{2}	6.40×10^{2}	6.15×10^{2}	6.07×10^{2}
	Mean	6.07×10^{2}	5.60×10^{2}	5.70×10^{2}	5.64×10^{2}	$\mathbf{5.37\times10^{2}}$
	Std	5.32×10^{1}	2.77×10^{1}	4.11×10^{1}	3.77×10^{1}	3.83×10^{1}
F26	Best	8.83×10^{2}	3.00×10^{2}	3.47×10^{2}	4.30×10^{2}	3.00×10^{2}
	Worst	8.03×10^{3}	6.52×10^{3}	6.60×10^{3}	8.13×10^{3}	8.76×10^{3}
	Mean	5.13×10^{3}	3.68×10^{3}	2.56×10^{3}	4.76×10^{3}	$\mathbf{2.52\times10^{3}}$
	Std	2.14×10^{3}	1.93×10^{3}	2.41×10^{3}	2.00×10^{3}	2.69×10^{3}
F27	Best	6.98×10^{2}	6.73×10^{2}	6.26×10^{2}	6.57×10^{2}	5.84×10^{2}
	Worst	1.10×10^{3}	1.53×10^{3}	1.57×10^{3}	1.24×10^{3}	1.03×10^{3}
	Mean	9.22×10^{2}	1.02×10^{3}	1.04×10^{3}	9.50×10^{2}	$\mathbf{7.39\times10^{2}}$
	Std	1.15×10^{2}	1.80×10^{2}	2.22×10^{2}	1.31×10^{2}	1.07×10^{2}
F28	Best	5.21×10^{2}	5.02×10^{2}	4.99×10^{2}	4.73×10^{2}	4.59×10^{2}
	Worst	7.02×10^{3}	7.83×10^{3}	6.35×10^{2}	6.12×10^{3}	5.33×10^{2}
	Mean	1.70×10^{3}	1.76×10^{3}	5.69×10^{2}	9.16×10^{2}	$\mathbf{4.92\times10^{2}}$
	Std	1.87×10^{3}	2.54×10^{3}	3.83×10^{1}	1.34×10^{3}	2.25×10^{1}
F29	Best	1.24×10^{3}	1.12×10^{3}	1.43×10^{3}	1.20×10^{3}	9.12×10^{2}
	Worst	5.22×10^{3}	4.89×10^{3}	3.43×10^{3}	4.31×10^{3}	2.66×10^{3}
	Mean	2.73×10^{3}	2.29×10^{3}	2.17×10^{3}	2.41×10^{3}	$\mathbf{1.81\times10^{3}}$
	Std	9.92×10^{2}	7.56×10^{2}	5.65×10^{2}	6.99×10^{2}	4.06×10^{2}
F30	Best	5.01×10^{6}	1.36×10^{6}	1.54×10^{6}	2.74×10^{6}	1.00×10^{6}
	Worst	7.95×10^{7}	2.19×10^{7}	2.30×10^{7}	7.76×10^{7}	4.52×10^{6}
	Mean	2.07×10^{7}	4.50×10^{6}	8.41×10^{6}	1.84×10^{7}	$\mathbf{1.79\times10^{6}}$
	Std	1.63×10^{7}	3.82×10^{6}	5.56×10^{6}	1.83×10^{7}	8.20×10^{5}
	+/≈/–	24/4/1	18/6/5	18/5/6	18/4/7	
	ARV	4.3103	3.1207	2.7241	2.9655	**1.8793**
	Rank	5	4	2	3	**1**

Under the 30-dimensional setting, twelve CEC2017 test functions were selected for statistical analysis using box plots. The experimental results presented in Figure 7 indicate that EDBO delivers competitive performance on the majority of the test functions, which fully demonstrates the algorithm's stability and robustness in the optimization process. To provide a clear comparison of EDBO and other benchmark algorithms, radar charts based on the "Mean" rankings from Tables 4–7 are presented in Figure 8. As illustrated in Figure 8, EDBO consistently surpasses the competing algorithms across all dimensions of each test set.

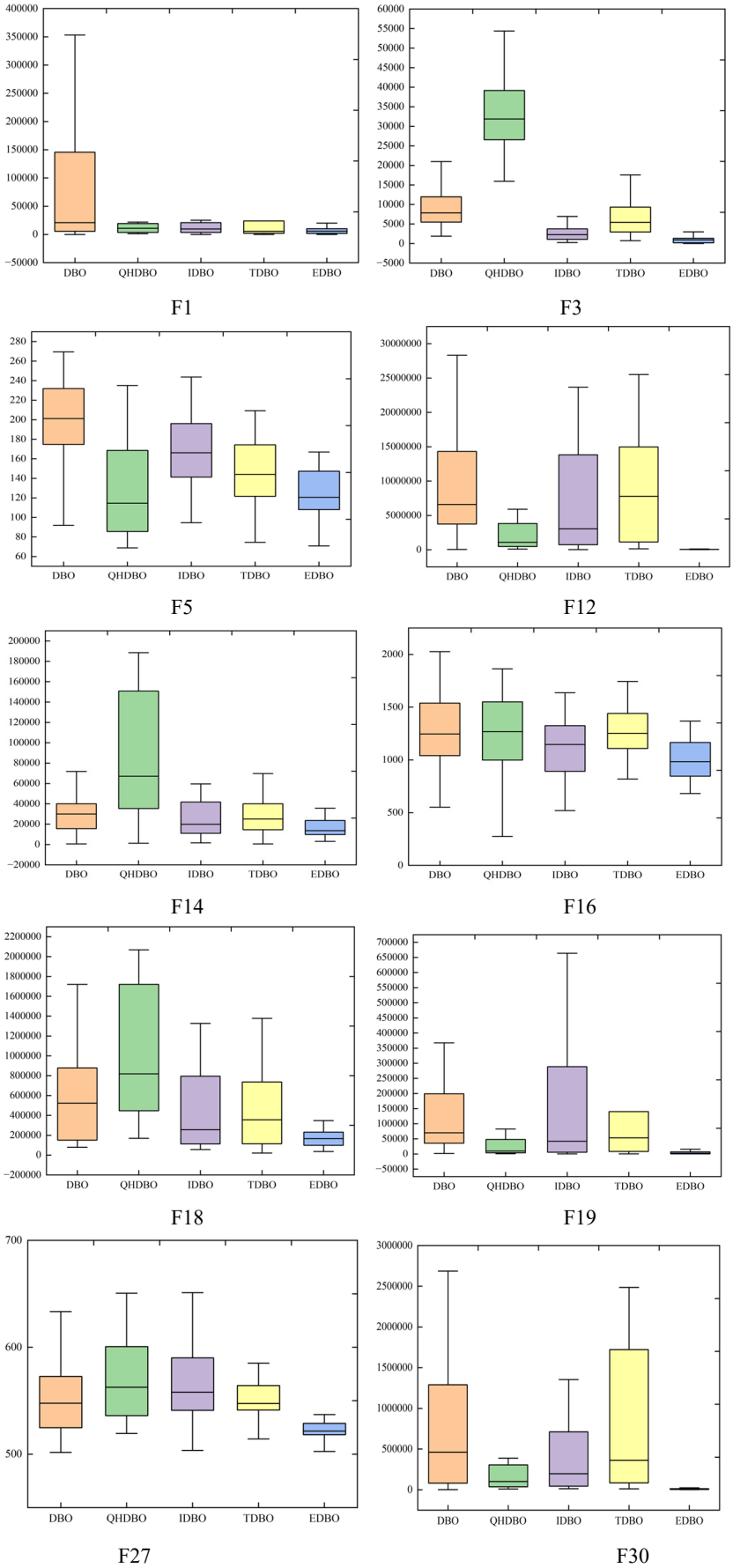


Figure 7. Box plots of DBO, QHDBO, IDBO, TDBO, and EDBO on CEC2017 at 30 dimensions.

5.3. Comparison with other evolutionary algorithms

This section presents a comparison between the EDBO algorithm and seven widely recognized evolutionary algorithms, including WOA, GWO, butterfly optimization algorithm (BOA) [51], dandelion optimizer (DO) [52], arithmetic optimization algorithm (AOA) [53], gravitational search algorithm (GSA) [54], and SCA. In the experiment, the CEC2014 and CEC2017 benchmark functions in 30 dimensions are used again, with the parameter configurations consistent with those in the corresponding literature. To ensure fairness, all algorithms are independently executed 30 times, with the population size and Max_Fes set to 50 and $10000 \times D$, respectively. Furthermore, the Wilcoxon's rank-sum test is conducted with the same parameters as in the previous section. Additionally, to provide a more comprehensive comparison of the performance differences among the algorithms, Friedman's test is performed, and the results are presented at the bottom of the tables.

The experimental results are shown in Tables 8 and 9. From Table 8, it can be observed that no algorithm provides the best result for every problem. In terms of solution accuracy, EDBO performs excellently on F1–F4, F10, F15, F17, F19, and F21–F30. WOA performs well on F14, GWO achieves reasonable results on F6, F9, F11, F16, F24, and F26, and BOA provides better solutions on F23–F25 and F29. DO performs excellently on F5, F8, F20, and F26, AOA excels on F23, F24, and F25, GSA exceeds on F7, F12, F13, F18, F24, and F25, while SCA achieves reasonable results on F24. Based on the statistical analysis conducted using Wilcoxon's rank-sum test, EDBO surpasses WOA, GWO, BOA, DO, AOA, GSA, and SCA on 25, 17, 27, 18, 26, 19, and 30 test functions, respectively. Furthermore, based on the performance ranking results from the Friedman test, EDBO ranks first with a value of 2.1167, followed by DO with a ranking value of 3.1667, while BOA has the highest average rank, indicating its relatively poor performance in handling these benchmark problems. Therefore, compared with these selected evolutionary algorithms, EDBO is undoubtedly the most competitive algorithm.

From Table 9, it is evident that no algorithm is able to efficiently adapt to all optimization problems. Specifically, GWO performs best on F10, F16, F17, F21, F23, and F24, DO provides promising solutions for F3, F14, F18, and F29, and GSA offers reasonable results on F1, F7–F9, F11, F13, F15, F19, and F26. EDBO achieves promising solutions on F4–F6, F12, F14, F20, F22, F25, and F27–F30, while ranking second on F1, F3, F13, F16–F19, F23, and F24. According to the statistical results of the Wilcoxon's rank-sum test, EDBO dominates WOA, GWO, BOA, DO, AOA, GSA, and SCA on 28, 15, 28, 14, 27, 18, and 29 test functions, respectively. Clearly, EDBO demonstrates better convergence performance when handling test problems with varying characteristics. Furthermore, based on the results from the Friedman test, EDBO yields better than other popular algorithms, ranking first, followed by DO, GWO, GSA, WOA, SCA, AOA, and BOA. In conclusion, the experimental results on these two test sets demonstrate that EDBO exhibits highly competitive overall performance.

From the overall experimental results, EDBO achieved relatively ideal outcomes in solving unimodal functions, hybrid functions, and composite functions. By comparing the variance of the test results with competing algorithms, it can be observed that EDBO demonstrates better robustness on different types of test functions through parameter selection, strategy improvements, and the introduction of an advanced adaptive mechanism. However, in some simple multimodal functions, EDBO exhibited instability, indicating that there is still room for further improvement in its ability to avoid local extrema.

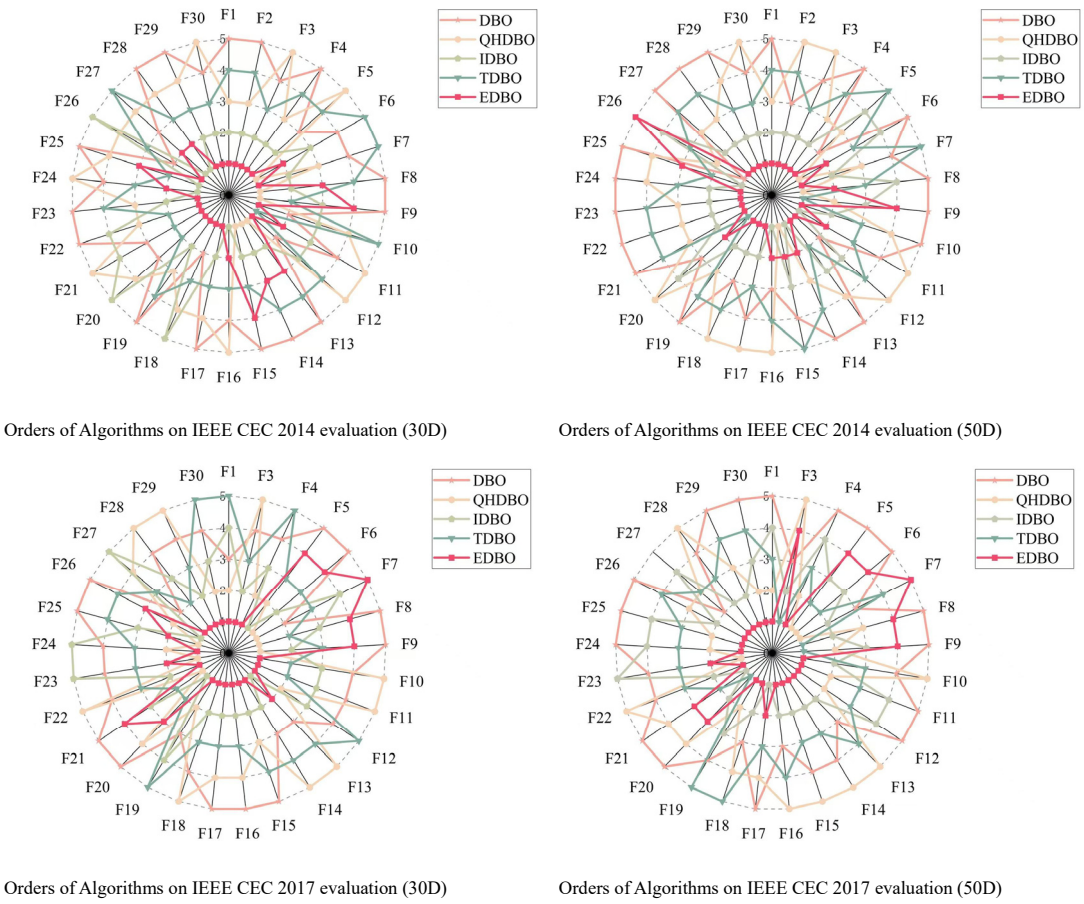


Figure 8. Rankings based on each algorithm's "Mean" on each test function.

Table 8. Experimental results of WOA, GWO, BOA, DO, AOA, GSA, SCA, and EDBO on CEC2014 at 30 dimensions.

Function	Metric	WOA	GWO	BOA	DO	AOA	GSA	SCA	EDBO
F1	Best	7.07×10^{6}	1.75×10^{7}	1.01×10^{9}	4.63×10^{5}	4.53×10^{8}	1.25×10^{6}	1.31×10^{8}	1.25×10^{5}
	Worst	5.05×10^{7}	3.48×10^{7}	2.31×10^{9}	5.28×10^{6}	1.12×10^{9}	2.97×10^{6}	4.17×10^{8}	1.15×10^{6}
	Mean	2.65×10^{7}	2.35×10^{7}	1.63×10^{9}	3.09×10^{6}	7.48×10^{8}	2.06×10^{6}	2.17×10^{8}	**4.43×10^{5}**
	Std	1.21×10^{7}	9.77×10^{6}	3.10×10^{8}	1.50×10^{6}	2.16×10^{8}	5.53×10^{5}	5.73×10^{7}	2.55×10^{5}
F2	Best	1.79×10^{5}	1.27×10^{9}	6.23×10^{10}	6.87×10^{2}	3.30×10^{10}	3.04×10^{3}	1.14×10^{10}	3.14×10^{-9}
	Worst	4.37×10^{7}	3.69×10^{9}	8.96×10^{10}	3.45×10^{4}	5.94×10^{10}	1.69×10^{4}	2.30×10^{10}	2.65×10^{-2}
	Mean	5.10×10^{6}	2.16×10^{9}	7.43×10^{10}	1.21×10^{4}	4.85×10^{10}	9.11×10^{3}	1.74×10^{10}	**2.58×10^{-3}**
	Std	1.15×10^{7}	1.33×10^{9}	6.38×10^{9}	1.00×10^{4}	7.27×10^{9}	3.16×10^{3}	2.42×10^{9}	6.81×10^{-3}
F3	Best	1.23×10^{4}	2.07×10^{4}	6.16×10^{4}	2.43×10^{1}	4.52×10^{4}	7.97×10^{3}	3.00×10^{4}	2.14×10^{-3}
	Worst	1.04×10^{5}	4.52×10^{4}	8.38×10^{4}	1.28×10^{3}	8.48×10^{4}	3.40×10^{4}	5.08×10^{4}	1.03×10^{2}
	Mean	4.09×10^{4}	3.32×10^{4}	7.51×10^{4}	2.98×10^{2}	6.67×10^{4}	1.73×10^{4}	4.00×10^{4}	**5.00×10^{0}**
	Std	2.53×10^{4}	1.23×10^{4}	5.71×10^{3}	3.39×10^{2}	9.82×10^{3}	7.27×10^{3}	5.59×10^{3}	5.57×10^{0}
F4	Best	8.58×10^{1}	3.02×10^{2}	1.07×10^{4}	1.04×10^{1}	2.09×10^{3}	5.51×10^{1}	5.88×10^{2}	2.68×10^{-1}
	Worst	3.00×10^{2}	4.14×10^{2}	1.71×10^{4}	1.54×10^{2}	9.63×10^{3}	9.63×10^{1}	2.20×10^{3}	1.36×10^{2}

Continued on next page

Function	Metric	WOA	GWO	BOA	DO	AOA	GSA	SCA	EDBO
	Mean	1.86×10^{2}	3.43×10^{2}	1.48×10^{4}	8.99×10^{1}	4.17×10^{3}	7.41×10^{1}	9.54×10^{2}	$\mathbf{2.15\times10^{1}}$
	Std	5.81×10^{1}	6.18×10^{1}	1.58×10^{3}	3.20×10^{1}	1.42×10^{3}	7.02×10^{0}	2.92×10^{2}	3.12×10^{1}
F5	Best	2.00×10^{1}	2.09×10^{1}	2.09×10^{1}	2.00×10^{1}	2.06×10^{1}	2.00×10^{1}	2.07×10^{1}	2.00×10^{1}
	Worst	2.06×10^{1}	2.10×10^{1}	2.11×10^{1}	2.02×10^{1}	2.10×10^{1}	$2.00\times10^{1}1$	2.10×10^{1}	2.10×10^{1}
	Mean	2.02×10^{1}	2.10×10^{1}	2.10×10^{1}	$\mathbf{2.01\times10^{1}}$	2.08×10^{1}	2.00×10^{1}	2.09×10^{1}	2.03×10^{1}
	Std	1.66×10^{-1}	2.57×10^{-2}	4.93×10^{-2}	5.43×10^{-2}	8.60×10^{-2}	5.64×10^{-4}	6.31×10^{-2}	3.61×10^{-1}
F6	Best	2.61×10^{1}	7.93×10^{0}	3.40×10^{1}	1.48×10^{1}	3.07×10^{1}	1.37×10^{1}	2.98×10^{1}	1.23×10^{1}
	Worst	4.30×10^{1}	1.44×10^{1}	4.19×10^{1}	3.04×10^{1}	3.80×10^{1}	2.16×10^{1}	3.88×10^{1}	2.77×10^{1}
	Mean	3.50×10^{1}	$\mathbf{1.01\times10^{1}}$	3.83×10^{1}	2.45×10^{1}	3.45×10^{1}	1.77×10^{1}	3.40×10^{1}	2.05×10^{1}
	Std	3.74×10^{1}	3.75×10^{0}	1.92×10^{0}	4.41×10^{0}	2.10×10^{0}	2.36×10^{0}	2.21×10^{0}	4.04×10^{0}
F7	Best	4.50×10^{-1}	5.43×10^{0}	5.36×10^{2}	5.09×10^{-3}	2.78×10^{2}	0.00×10^{0}	9.36×10^{1}	2.27×10^{-13}
	Worst	1.27×10^{0}	1.82×10^{1}	8.40×10^{2}	1.30×10^{-1}	5.49×10^{2}	1.14×10^{-13}	1.95×10^{2}	5.65×10^{-2}
	Mean	9.53×10^{-1}	1.01×10^{1}	7.25×10^{2}	5.04×10^{-2}	4.14×10^{2}	$\mathbf{6.44\times10^{-14}}$	1.46×10^{2}	1.40×10^{-2}
	Std	1.80×10^{-1}	7.04×10^{0}	7.99×10^{1}	3.45×10^{-2}	6.76×10^{1}	5.73×10^{-14}	2.87×10^{1}	1.56×10^{-2}
F8	Best	1.21×10^{2}	8.16×10^{1}	1.57×10^{2}	1.81×10^{1}	2.18×10^{2}	1.05×10^{2}	2.20×10^{2}	5.17×10^{1}
	Worst	2.55×10^{2}	9.72×10^{1}	2.61×10^{2}	6.23×10^{1}	3.15×10^{2}	1.68×10^{2}	2.69×10^{2}	1.60×10^{2}
	Mean	1.82×10^{2}	9.09×10^{1}	2.09×10^{2}	$\mathbf{3.70\times10^{1}}$	2.67×10^{2}	1.40×10^{2}	2.48×10^{2}	9.62×10^{1}
	Std	3.19×10^{1}	8.20×10^{0}	2.52×10^{1}	1.00×10^{1}	2.44×10^{1}	1.49×10^{1}	1.21×10^{1}	2.79×10^{1}
F9	Best	1.49×10^{2}	7.58×10^{1}	2.36×10^{2}	8.26×10^{1}	2.15×10^{2}	1.14×10^{2}	2.22×10^{2}	1.04×10^{2}
	Worst	3.49×10^{2}	1.13×10^{2}	3.75×10^{2}	2.62×10^{2}	3.03×10^{2}	2.01×10^{2}	3.14×10^{2}	2.68×10^{2}
	Mean	2.16×10^{2}	$\mathbf{9.28\times10^{1}}$	3.09×10^{2}	1.65×10^{2}	2.66×10^{2}	1.58×10^{2}	2.75×10^{2}	1.81×10^{2}
	Std	4.82×10^{1}	1.87×10^{1}	4.03×10^{1}	4.26×10^{1}	1.83×10^{1}	1.95×10^{1}	1.71×10^{1}	3.29×10^{1}
F10	Best	2.71×10^{3}	2.28×10^{3}	3.35×10^{3}	2.55×10^{2}	4.35×10^{3}	2.45×10^{3}	4.76×10^{3}	1.28×10^{3}
	Worst	4.71×10^{3}	3.29×10^{3}	5.01×10^{3}	1.38×10^{3}	6.04×10^{3}	3.95×10^{3}	6.93×10^{3}	4.36×10^{3}
	Mean	3.80×10^{3}	2.68×10^{3}	4.41×10^{3}	7.17×10^{2}	5.11×10^{3}	3.20×10^{3}	6.09×10^{3}	$\mathbf{2.46\times10^{3}}$
	Std	5.55×10^{2}	5.39×10^{2}	3.60×10^{2}	2.59×10^{2}	4.44×10^{2}	4.05×10^{2}	5.50×10^{2}	7.82×10^{2}
F11	Best	3.20×10^{3}	2.50×10^{3}	4.17×10^{3}	2.19×10^{3}	4.06×10^{3}	2.99×10^{3}	6.55×10^{3}	2.42×10^{3}
	Worst	6.48×10^{3}	3.02×10^{3}	5.41×10^{3}	4.49×10^{3}	6.83×10^{3}	4.69×10^{3}	7.50×10^{3}	4.78×10^{3}
	Mean	4.97×10^{3}	$\mathbf{2.76\times10^{3}}$	4.94×10^{3}	3.49×10^{3}	5.85×10^{3}	3.84×10^{3}	7.05×10^{3}	3.81×10^{3}
	Std	9.66×10^{2}	2.60×10^{2}	3.54×10^{2}	5.65×10^{2}	6.68×10^{2}	5.16×10^{2}	2.75×10^{2}	6.28×10^{2}
F12	Best	7.25×10^{-1}	2.47×10^{-1}	1.24×10^{0}	2.35×10^{-1}	8.79×10^{-1}	1.28×10^{-6}	1.76×10^{0}	1.82×10^{-1}
	Worst	2.84×10^{0}	2.24×10^{0}	2.42×10^{0}	9.58×10^{-1}	1.99×10^{0}	1.51×10^{-3}	3.00×10^{0}	1.21×10^{0}
	Mean	1.70×10^{0}	1.38×10^{0}	1.99×10^{0}	5.39×10^{-1}	1.34×10^{0}	$\mathbf{4.66\times10^{-4}}$	2.46×10^{0}	4.95×10^{-1}
	Std	5.08×10^{-1}	1.02×10^{0}	3.04×10^{-1}	2.15×10^{-1}	3.08×10^{-1}	4.48×10^{-4}	3.05×10^{-1}	2.06×10^{-1}
F13	Best	3.10×10^{-1}	3.61×10^{-1}	7.05×10^{0}	4.19×10^{-1}	4.51×10^{0}	1.14×10^{-1}	2.19×10^{0}	3.04×10^{-1}
	Worst	7.66×10^{-1}	4.09×10^{-1}	9.65×10^{0}	7.76×10^{-1}	6.55×10^{0}	2.84×10^{-1}	3.67×10^{0}	9.13×10^{-1}
	Mean	5.32×10^{-1}	3.78×10^{-1}	8.63×10^{0}	5.82×10^{-1}	5.33×10^{0}	$\mathbf{1.78\times10^{-1}}$	2.99×10^{0}	5.67×10^{-1}
	Std	1.25×10^{-1}	2.74×10^{-2}	6.73×10^{-1}	8.88×10^{-2}	4.62×10^{-1}	3.22×10^{-2}	3.79×10^{-1}	1.47×10^{-1}
F14	Best	1.94×10^{-1}	2.29×10^{-1}	1.90×10^{2}	2.04×10^{-1}	1.15×10^{2}	1.88×10^{-1}	3.27×10^{1}	2.19×10^{-1}
	Worst	3.75×10^{-1}	3.89×10^{0}	3.35×10^{2}	9.82×10^{-1}	2.09×10^{2}	4.04×10^{-1}	5.96×10^{1}	1.19×10^{0}
	Mean	$\mathbf{2.77\times10^{-1}}$	1.48×10^{0}	2.78×10^{2}	3.38×10^{-1}	1.47×10^{2}	3.02×10^{-1}	4.50×10^{1}	5.41×10^{-1}
	Std	4.55×10^{-2}	2.08×10^{0}	3.26×10^{1}	2.13×10^{-1}	2.26×10^{1}	5.47×10^{-2}	7.69×10^{0}	3.28×10^{-1}

Continued on next page

Function	Metric	WOA	GWO	BOA	DO	AOA	GSA	SCA	EDBO
F15	Best	2.30×10^{1}	1.02×10^{1}	1.54×10^{5}	1.37×10^{1}	1.35×10^{4}	1.95×10^{0}	3.32×10^{2}	6.10×10^{0}
	Worst	1.18×10^{2}	3.84×10^{1}	5.03×10^{5}	5.02×10^{1}	1.41×10^{5}	3.81×10^{0}	1.69×10^{4}	2.27×10^{1}
	Mean	7.36×10^{1}	2.49×10^{1}	2.96×10^{5}	2.39×10^{1}	6.23×10^{4}	2.79×10^{0}	3.77×10^{3}	1.50×10^{1}
	Std	2.30×10^{1}	1.41×10^{1}	9.17×10^{4}	7.33×10^{0}	3.27×10^{4}	4.18×10^{-1}	4.09×10^{3}	4.35×10^{0}
F16	Best	1.09×10^{1}	1.06×10^{1}	1.30×10^{1}	1.10×10^{1}	1.12×10^{1}	1.31×10^{1}	1.21×10^{1}	9.43×10^{0}
	Worst	1.33×10^{1}	1.09×10^{1}	1.38×10^{1}	1.27×10^{1}	1.31×10^{1}	1.42×10^{1}	1.30×10^{1}	1.31×10^{1}
	Mean	1.25×10^{1}	**1.07×10^{1}**	1.35×10^{1}	1.20×10^{1}	1.24×10^{1}	1.37×10^{1}	1.27×10^{1}	1.17×10^{1}
	Std	6.00×10^{-1}	1.82×10^{-1}	1.91×10^{-1}	4.63×10^{-1}	4.11×10^{-1}	2.74×10^{-1}	1.96×10^{-1}	7.43×10^{-1}
F17	Best	2.81×10^{5}	9.90×10^{5}	2.86×10^{7}	3.78×10^{4}	1.22×10^{6}	1.13×10^{5}	1.87×10^{6}	3.20×10^{4}
	Worst	1.04×10^{7}	3.25×10^{6}	2.58×10^{8}	9.26×10^{5}	1.01×10^{8}	3.39×10^{5}	1.87×10^{7}	5.92×10^{5}
	Mean	3.81×10^{6}	1.85×10^{6}	1.59×10^{8}	2.46×10^{5}	3.35×10^{7}	2.13×10^{5}	5.96×10^{6}	**1.23×10^{5}**
	Std	2.41×10^{6}	1.22×10^{6}	6.35×10^{7}	1.98×10^{5}	2.17×10^{7}	6.54×10^{4}	3.60×10^{6}	6.90×10^{4}
F18	Best	4.33×10^{2}	1.50×10^{4}	2.52×10^{9}	3.82×10^{2}	3.32×10^{3}	2.15×10^{2}	4.38×10^{7}	3.67×10^{2}
	Worst	1.51×10^{4}	5.99×10^{4}	8.62×10^{9}	2.06×10^{4}	2.49×10^{9}	1.38×10^{3}	4.10×10^{8}	2.63×10^{4}
	Mean	4.69×10^{3}	3.63×10^{4}	5.88×10^{9}	5.02×10^{3}	5.14×10^{8}	**4.68×10^{2}**	1.70×10^{8}	5.84×10^{3}
	Std	4.46×10^{3}	2.25×10^{4}	1.69×10^{9}	5.41×10^{3}	7.13×10^{8}	2.50×10^{2}	8.71×10^{7}	7.06×10^{3}
F19	Best	1.59×10^{1}	1.48×10^{1}	2.43×10^{2}	1.06×10^{1}	1.28×10^{2}	8.51×10^{0}	4.52×10^{1}	7.73×10^{0}
	Worst	1.55×10^{2}	4.07×10^{1}	6.55×10^{2}	9.53×10^{1}	3.95×10^{2}	7.02×10^{1}	1.40×10^{2}	1.71×10^{1}
	Mean	5.67×10^{1}	2.48×10^{1}	4.79×10^{2}	2.44×10^{1}	2.34×10^{2}	1.52×10^{1}	8.95×10^{1}	**1.04×10^{1}**
	Std	4.46×10^{1}	1.39×10^{1}	9.09×10^{1}	2.48×10^{1}	6.40×10^{1}	1.49×10^{1}	2.38×10^{1}	2.27×10^{0}
F20	Best	3.15×10^{3}	5.70×10^{3}	3.38×10^{4}	2.03×10^{2}	1.30×10^{4}	2.06×10^{4}	3.79×10^{3}	2.90×10^{2}
	Worst	5.37×10^{4}	1.97×10^{4}	3.47×10^{5}	9.04×10^{2}	1.73×10^{5}	3.94×10^{4}	2.36×10^{4}	4.70×10^{3}
	Mean	2.37×10^{4}	1.27×10^{4}	1.12×10^{5}	**3.82×10^{2}**	7.31×10^{4}	2.89×10^{4}	1.43×10^{4}	1.14×10^{3}
	Std	1.44×10^{4}	7.00×10^{3}	6.74×10^{4}	1.40×10^{2}	4.20×10^{4}	5.24×10^{3}	4.85×10^{3}	7.98×10^{2}
F21	Best	1.22×10^{5}	3.75×10^{4}	3.88×10^{6}	2.96×10^{4}	4.36×10^{5}	6.47×10^{4}	3.43×10^{5}	2.41×10^{3}
	Worst	3.17×10^{6}	2.16×10^{5}	7.88×10^{7}	4.66×10^{5}	2.87×10^{7}	3.13×10^{5}	2.84×10^{6}	3.56×10^{5}
	Mean	8.47×10^{5}	1.46×10^{5}	3.66×10^{7}	1.23×10^{5}	8.10×10^{6}	1.45×10^{5}	1.37×10^{6}	**6.35×10^{4}**
	Std	6.79×10^{5}	9.51×10^{4}	2.18×10^{7}	1.06×10^{5}	7.74×10^{6}	5.61×10^{4}	6.56×10^{5}	7.55×10^{4}
F22	Best	4.21×10^{2}	3.03×10^{2}	8.14×10^{2}	2.13×10^{2}	6.50×10^{2}	4.03×10^{2}	5.32×10^{2}	1.31×10^{2}
	Worst	1.29×10^{3}	5.12×10^{2}	1.54×10^{5}	8.05×10^{2}	3.05×10^{3}	1.30×10^{3}	1.03×10^{3}	6.76×10^{2}
	Mean	7.55×10^{2}	4.20×10^{2}	1.65×10^{4}	4.93×10^{2}	1.30×10^{3}	9.30×10^{2}	8.05×10^{2}	**3.64×10^{2}**
	Std	2.25×10^{2}	1.06×10^{2}	2.80×10^{4}	1.55×10^{2}	5.39×10^{2}	2.11×10^{2}	1.34×10^{2}	1.63×10^{2}
F23	Best	3.22×10^{2}	3.27×10^{2}	2.00×10^{2}	3.15×10^{2}	2.00×10^{2}	2.00×10^{2}	3.48×10^{2}	2.00×10^{2}
	Worst	3.45×10^{2}	3.37×10^{2}	2.00×10^{2}	3.15×10^{2}	2.00×10^{2}	3.15×10^{2}	4.32×10^{2}	2.00×10^{2}
	Mean	3.31×10^{2}	3.33×10^{2}	**2.00×10^{2}**	3.15×10^{2}	**2.00×10^{2}**	2.46×10^{2}	3.69×10^{2}	**2.00×10^{2}**
	Std	5.11×10^{0}	5.35×10^{0}	0.00×10^{0}	3.68×10^{-3}	0.00×10^{0}	5.74×10^{1}	1.70×10^{1}	0.00×10^{0}
F24	Best	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	2.12×10^{2}	2.00×10^{2}	2.00×10^{2}	2.24×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Mean	2.04×10^{2}	**2.00×10^{2}**	**2.00×10^{2}**	2.01×10^{2}	**2.00×10^{2}**	**2.00×10^{2}**	**2.00×10^{2}**	**2.00×10^{2}**
	Std	3.22×10^{0}	8.60×10^{-4}	2.16×10^{-7}	4.30×10^{0}	0.00×10^{0}	1.05×10^{-2}	1.33×10^{-1}	0.00×10^{0}
F25	Best	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}	2.00×10^{2}
	Worst	2.81×10^{2}	2.15×10^{2}	2.00×10^{2}	2.24×10^{2}	2.00×10^{2}	2.06×10^{2}	2.42×10^{2}	2.00×10^{2}

Continued on next page

Function	Metric	WOA	GWO	BOA	DO	AOA	GSA	SCA	EDBO
	Mean	2.20×10^{2}	2.10×10^{2}	$\mathbf{2.00\times10^{2}}$	2.16×10^{2}	$\mathbf{2.00\times10^{2}}$	$\mathbf{2.00\times10^{2}}$	2.24×10^{2}	$\mathbf{2.00\times10^{2}}$
	Std	1.93×10^{1}	8.36×10^{0}	0.00×10^{0}	4.87×10^{0}	0.00×10^{0}	1.15×10^{0}	1.05×10^{1}	0.00×10^{0}
F26	Best	1.00×10^{2}	1.00×10^{2}	1.05×10^{2}	1.00×10^{2}	1.05×10^{2}	1.14×10^{2}	1.01×10^{2}	1.00×10^{2}
	Worst	2.00×10^{2}	1.01×10^{2}	2.00×10^{2}	1.01×10^{2}	2.00×10^{2}	2.00×10^{2}	1.03×10^{2}	1.01×10^{2}
	Mean	1.07×10^{2}	$\mathbf{1.01\times10^{2}}$	1.55×10^{2}	$\mathbf{1.01\times10^{2}}$	1.62×10^{2}	1.92×10^{2}	1.03×10^{2}	$\mathbf{1.01\times10^{2}}$
	Std	2.53×10^{1}	2.31×10^{-1}	4.08×10^{1}	1.53×10^{-1}	4.71×10^{1}	2.12×10^{1}	5.65×10^{-1}	1.34×10^{-1}
F27	Best	4.10×10^{2}	6.45×10^{2}	4.50×10^{2}	4.03×10^{2}	2.00×10^{2}	8.73×10^{2}	4.73×10^{2}	2.00×10^{2}
	Worst	1.40×10^{3}	8.05×10^{2}	6.11×10^{2}	1.17×10^{3}	1.41×10^{3}	2.41×10^{3}	1.34×10^{3}	2.00×10^{2}
	Mean	1.09×10^{3}	7.22×10^{2}	5.04×10^{2}	8.98×10^{2}	3.12×10^{2}	1.71×10^{3}	8.66×10^{2}	$\mathbf{2.00\times10^{2}}$
	Std	3.42×10^{2}	8.02×10^{1}	4.27×10^{1}	2.87×10^{2}	3.18×10^{2}	3.84×10^{2}	3.48×10^{2}	0.00×10^{0}
F28	Best	2.00×10^{2}	8.11×10^{2}	3.79×10^{3}	9.79×10^{2}	2.00×10^{2}	9.61×10^{2}	1.64×10^{3}	2.00×10^{2}
	Worst	3.88×10^{3}	9.71×10^{2}	6.69×10^{3}	3.22×10^{3}	2.99×10^{3}	5.10×10^{3}	3.22×10^{3}	2.00×10^{2}
	Mean	2.24×10^{3}	8.80×10^{2}	5.54×10^{3}	1.70×10^{3}	2.93×10^{2}	2.80×10^{3}	2.08×10^{3}	$\mathbf{2.00\times10^{2}}$
	Std	8.57×10^{2}	8.22×10^{1}	7.11×10^{2}	5.78×10^{2}	5.09×10^{2}	9.39×10^{2}	3.59×10^{2}	0.00×10^{0}
F29	Best	1.08×10^{4}	1.01×10^{4}	2.00×10^{2}	3.13×10^{3}	2.00×10^{2}	2.00×10^{2}	3.76×10^{6}	2.00×10^{2}
	Worst	1.13×10^{7}	9.08×10^{6}	2.00×10^{2}	9.32×10^{3}	1.05×10^{8}	6.63×10^{3}	2.82×10^{7}	2.00×10^{2}
	Mean	6.91×10^{6}	3.13×10^{6}	$\mathbf{2.00\times10^{2}}$	5.59×10^{3}	3.16×10^{7}	6.80×10^{2}	1.12×10^{7}	$\mathbf{2.00\times10^{2}}$
	Std	4.28×10^{6}	5.15×10^{6}	0.00×10^{0}	1.62×10^{3}	4.15×10^{7}	1.51×10^{3}	6.52×10^{6}	0.00×10^{0}
F30	Best	1.20×10^{4}	1.13×10^{4}	2.00×10^{2}	2.55×10^{3}	2.00×10^{2}	7.15×10^{3}	7.33×10^{4}	2.00×10^{2}
	Worst	2.45×10^{5}	4.14×10^{4}	1.27×10^{5}	7.69×10^{3}	2.14×10^{6}	1.02×10^{4}	5.24×10^{5}	2.00×10^{2}
	Mean	7.06×10^{4}	2.78×10^{4}	7.78×10^{3}	4.47×10^{3}	1.26×10^{6}	8.42×10^{3}	2.21×10^{5}	$\mathbf{2.00\times10^{2}}$
	Std	5.13×10^{4}	1.52×10^{4}	2.91×10^{4}	1.24×10^{3}	6.52×10^{5}	7.93×10^{2}	1.03×10^{5}	0.00×10^{0}
	+/≈/–	25/4/1	17/9/4	27/3/0	18/7/5	26/4/0	19/3/8	30/0/0	
	ARV	5.0333	3.7333	6.5333	3.1667	6.0000	3.3333	6.0833	**2.1167**
	Rank	5	4	8	2	6	3	7	**1**

Table 9. Experimental results of WOA, GWO, BOA, DO, AOA, GSA, SCA, and EDBO on CEC2017 at 30 dimensions.

Function	Metric	WOA	GWO	BOA	DO	AOA	GSA	SCA	EDBO
F1	Best	4.23×10^{6}	1.57×10^{8}	2.88×10^{10}	6.57×10^{1}	2.94×10^{10}	1.67×10^{-1}	8.82×10^{9}	1.01×10^{0}
	Worst	1.13×10^{8}	8.54×10^{9}	6.04×10^{10}	3.13×10^{4}	4.81×10^{10}	5.04×10^{3}	1.70×10^{10}	2.01×10^{4}
	Mean	2.17×10^{7}	1.85×10^{9}	4.62×10^{10}	1.15×10^{5}	3.79×10^{10}	$\mathbf{2.05\times10^{3}}$	1.25×10^{10}	6.79×10^{3}
	Std	2.30×10^{7}	1.71×10^{9}	6.94×10^{9}	9.09×10^{3}	4.63×10^{9}	1.39×10^{3}	2.02×10^{9}	5.75×10^{3}
F3	Best	3.71×10^{4}	9.42×10^{3}	5.60×10^{4}	6.14×10^{-2}	4.08×10^{4}	6.21×10^{4}	2.62×10^{4}	8.98×10^{0}
	Worst	9.72×10^{4}	5.63×10^{4}	8.19×10^{4}	1.05×10^{0}	8.41×10^{4}	9.19×10^{4}	4.42×10^{4}	3.75×10^{3}
	Mean	6.17×10^{4}	3.28×10^{4}	6.95×10^{4}	$\mathbf{3.53\times10^{-1}}$	7.40×10^{4}	7.43×10^{4}	3.37×10^{4}	1.13×10^{3}
	Std	1.74×10^{4}	1.16×10^{4}	7.12×10^{3}	2.71×10^{-1}	9.59×10^{3}	8.19×10^{3}	5.56×10^{3}	1.14×10^{3}
F4	Best	1.52×10^{2}	1.01×10^{2}	1.13×10^{4}	2.49×10^{1}	3.33×10^{3}	8.50×10^{1}	1.06×10^{3}	9.62×10^{-2}
	Worst	1.83×10^{3}	8.54×10^{2}	2.47×10^{4}	1.18×10^{2}	1.12×10^{4}	1.53×10^{2}	2.39×10^{3}	1.14×10^{2}
	Mean	5.24×10^{2}	2.09×10^{2}	1.87×10^{4}	8.73×10^{1}	8.48×10^{3}	1.29×10^{2}	1.64×10^{3}	$\mathbf{5.48\times10^{1}}$
	Std	3.89×10^{2}	1.63×10^{2}	2.96×10^{3}	2.35×10^{1}	1.63×10^{3}	1.25×10^{1}	3.64×10^{2}	3.68×10^{1}

Continued on next page

Function	Metric	WOA	GWO	BOA	DO	AOA	GSA	SCA	EDBO
F5	Best	1.41×10^{2}	5.88×10^{1}	2.93×10^{2}	1.04×10^{2}	2.37×10^{2}	1.75×10^{2}	2.52×10^{2}	9.45×10^{1}
	Worst	3.50×10^{2}	1.99×10^{2}	4.15×10^{2}	2.62×10^{2}	3.83×10^{2}	2.66×10^{2}	3.19×10^{2}	2.44×10^{2}
	Mean	2.30×10^{2}	1.88×10^{2}	3.64×10^{2}	1.74×10^{2}	3.20×10^{2}	1.99×10^{2}	2.83×10^{2}	$\mathbf{1.69\times10^{2}}$
	Std	5.06×10^{1}	3.49×10^{1}	2.81×10^{1}	4.04×10^{1}	3.43×10^{1}	1.87×10^{1}	1.68×10^{1}	3.78×10^{1}
F6	Best	3.61×10^{1}	1.30×10^{0}	4.97×10^{1}	2.09×10^{1}	5.70×10^{1}	3.02×10^{1}	4.19×10^{1}	7.81×10^{0}
	Worst	8.28×10^{1}	1.76×10^{1}	8.11×10^{1}	4.64×10^{1}	7.86×10^{1}	4.94×10^{1}	6.46×10^{1}	4.82×10^{1}
	Mean	5.53×10^{1}	7.83×10^{0}	6.80×10^{1}	3.05×10^{1}	6.67×10^{1}	3.92×10^{1}	5.46×10^{1}	$\mathbf{2.11\times10^{1}}$
	Std	9.95×10^{0}	3.64×10^{0}	7.05×10^{0}	6.12×10^{0}	4.97×10^{0}	5.14×10^{0}	5.44×10^{0}	1.05×10^{1}
F7	Best	4.11×10^{2}	1.04×10^{2}	4.61×10^{2}	1.85×10^{2}	4.48×10^{2}	3.78×10^{1}	3.17×10^{2}	1.47×10^{2}
	Worst	6.45×10^{2}	2.74×10^{2}	6.83×10^{2}	4.00×10^{2}	6.80×10^{2}	6.65×10^{1}	4.68×10^{2}	4.19×10^{2}
	Mean	5.08×10^{2}	1.63×10^{2}	6.07×10^{2}	2.82×10^{2}	5.87×10^{2}	$\mathbf{4.88\times10^{1}}$	3.86×10^{2}	2.32×10^{2}
	Std	6.20×10^{1}	4.44×10^{1}	5.14×10^{1}	6.72×10^{1}	5.14×10^{1}	5.95×10^{0}	3.21×10^{1}	6.30×10^{1}
F8	Best	1.16×10^{2}	5.68×10^{1}	2.67×10^{2}	8.06×10^{1}	1.84×10^{2}	1.09×10^{2}	2.06×10^{2}	8.66×10^{1}
	Worst	2.62×10^{2}	1.16×10^{2}	3.35×10^{2}	2.46×10^{2}	3.02×10^{2}	2.14×10^{2}	2.91×10^{2}	2.55×10^{2}
	Mean	1.91×10^{2}	$\mathbf{8.16\times10^{1}}$	3.00×10^{2}	1.49×10^{2}	2.36×10^{2}	1.47×10^{2}	2.45×10^{2}	1.52×10^{2}
	Std	3.90×10^{1}	1.59×10^{1}	1.74×10^{1}	4.36×10^{1}	2.84×10^{1}	2.15×10^{1}	1.71×10^{1}	4.03×10^{1}
F9	Best	2.87×10^{3}	1.78×10^{2}	5.68×10^{3}	1.34×10^{3}	2.93×10^{3}	5.36×10^{0}	3.06×10^{3}	5.39×10^{2}
	Worst	6.50×10^{3}	2.28×10^{3}	9.74×10^{3}	9.01×10^{3}	7.71×10^{3}	1.24×10^{3}	6.63×10^{3}	4.81×10^{3}
	Mean	4.78×10^{3}	7.64×10^{2}	8.29×10^{3}	3.81×10^{3}	5.08×10^{3}	$\mathbf{4.54\times10^{2}}$	4.44×10^{3}	2.54×10^{3}
	Std	1.10×10^{3}	5.35×10^{2}	1.14×10^{3}	1.91×10^{3}	1.08×10^{3}	2.61×10^{2}	9.37×10^{2}	1.16×10^{3}
F10	Best	3.70×10^{3}	1.75×10^{3}	6.21×10^{3}	2.57×10^{3}	3.57×10^{3}	2.93×10^{3}	5.96×10^{3}	2.24×10^{3}
	Worst	7.51×10^{3}	4.36×10^{3}	8.03×10^{3}	5.27×10^{3}	6.71×10^{3}	4.95×10^{3}	7.51×10^{3}	4.00×10^{3}
	Mean	4.84×10^{3}	$\mathbf{2.98\times10^{3}}$	7.49×10^{3}	3.91×10^{3}	5.30×10^{3}	3.89×10^{3}	6.99×10^{3}	4.00×10^{3}
	Std	8.83×10^{2}	5.16×10^{2}	4.13×10^{2}	6.98×10^{2}	6.90×10^{2}	4.80×10^{2}	3.96×10^{2}	7.81×10^{2}
F11	Best	1.59×10^{2}	1.71×10^{2}	1.71×10^{3}	5.07×10^{1}	5.13×10^{2}	6.80×10^{1}	5.82×10^{2}	1.31×10^{2}
	Worst	3.01×10^{3}	2.72×10^{3}	7.13×10^{3}	2.14×10^{2}	4.80×10^{3}	1.46×10^{2}	1.45×10^{3}	3.98×10^{2}
	Mean	8.57×10^{2}	7.55×10^{2}	4.02×10^{3}	1.14×10^{2}	2.32×10^{3}	$\mathbf{8.85\times10^{1}}$	9.56×10^{2}	2.45×10^{2}
	Std	7.61×10^{2}	7.47×10^{2}	1.31×10^{3}	4.35×10^{1}	9.59×10^{2}	2.19×10^{1}	1.98×10^{2}	7.13×10^{1}
F12	Best	5.17×10^{6}	1.20×10^{6}	5.47×10^{9}	4.44×10^{5}	3.58×10^{9}	2.08×10^{5}	7.33×10^{8}	1.45×10^{4}
	Worst	1.03×10^{9}	1.87×10^{8}	1.40×10^{10}	5.09×10^{6}	1.18×10^{10}	9.53×10^{5}	2.70×10^{9}	1.10×10^{5}
	Mean	2.09×10^{8}	4.22×10^{7}	9.91×10^{9}	2.15×10^{6}	7.79×10^{9}	5.04×10^{5}	1.54×10^{9}	$\mathbf{5.22\times10^{4}}$
	Std	2.37×10^{8}	4.68×10^{7}	2.30×10^{9}	1.20×10^{6}	2.19×10^{9}	2.03×10^{5}	4.41×10^{8}	2.06×10^{4}
F13	Best	1.42×10^{4}	1.14×10^{4}	8.00×10^{8}	2.34×10^{4}	1.94×10^{4}	1.41×10^{4}	2.60×10^{8}	9.89×10^{3}
	Worst	1.61×10^{8}	1.16×10^{8}	1.47×10^{10}	2.28×10^{5}	1.56×10^{7}	4.81×10^{4}	1.02×10^{9}	4.66×10^{6}
	Mean	1.56×10^{7}	7.40×10^{6}	6.18×10^{9}	9.24×10^{5}	5.58×10^{5}	$\mathbf{2.85\times10^{4}}$	5.35×10^{8}	4.95×10^{5}
	Std	4.64×10^{7}	2.23×10^{7}	3.96×10^{9}	4.64×10^{4}	2.84×10^{6}	7.81×10^{3}	1.66×10^{8}	1.37×10^{6}
F14	Best	4.37×10^{3}	6.12×10^{2}	3.02×10^{4}	1.02×10^{3}	1.05×10^{3}	1.63×10^{4}	2.92×10^{4}	3.10×10^{3}
	Worst	1.62×10^{6}	2.14×10^{6}	2.93×10^{6}	5.17×10^{4}	1.68×10^{5}	1.68×10^{5}	3.17×10^{5}	3.57×10^{4}
	Mean	4.41×10^{5}	2.49×10^{5}	1.03×10^{6}	$\mathbf{1.65\times10^{4}}$	4.97×10^{4}	6.22×10^{4}	1.29×10^{5}	$\mathbf{1.65\times10^{4}}$
	Std	4.52×10^{5}	5.18×10^{5}	7.73×10^{5}	1.44×10^{4}	5.26×10^{4}	3.82×10^{4}	6.75×10^{4}	9.00×10^{3}
F15	Best	1.45×10^{4}	1.03×10^{4}	4.64×10^{6}	3.96×10^{3}	1.30×10^{4}	2.49×10^{3}	6.71×10^{5}	5.33×10^{2}
	Worst	1.20×10^{5}	5.74×10^{7}	1.27×10^{9}	7.93×10^{4}	5.44×10^{4}	9.23×10^{3}	2.41×10^{7}	4.27×10^{4}

Continued on next page

Function	Metric	WOA	GWO	BOA	DO	AOA	GSA	SCA	EDBO
	Mean	4.98×10^{4}	2.06×10^{6}	2.37×10^{8}	2.50×10^{4}	2.02×10^{4}	**5.03×10^{3}**	5.57×10^{6}	1.35×10^{4}
	Std	3.05×10^{4}	1.05×10^{7}	2.68×10^{8}	2.14×10^{4}	9.90×10^{3}	1.75×10^{3}	6.32×10^{6}	1.36×10^{4}
F16	Best	1.22×10^{3}	3.07×10^{2}	2.63×10^{3}	4.13×10^{2}	1.42×10^{3}	9.27×10^{2}	1.78×10^{3}	2.49×10^{2}
	Worst	2.98×10^{3}	1.33×10^{3}	8.27×10^{3}	1.78×10^{3}	4.49×10^{3}	1.96×10^{3}	2.54×10^{3}	1.64×10^{3}
	Mean	1.95×10^{3}	**7.66×10^{2}**	4.32×10^{3}	2.92×10^{3}	2.52×10^{3}	1.46×10^{3}	2.18×10^{3}	9.87×10^{2}
	Std	5.09×10^{2}	3.06×10^{2}	1.36×10^{3}	2.87×10^{2}	7.44×10^{2}	3.07×10^{2}	2.02×10^{2}	3.71×10^{2}
F17	Best	1.44×10^{2}	6.79×10^{1}	9.71×10^{2}	2.01×10^{2}	5.16×10^{2}	6.14×10^{2}	3.78×10^{2}	2.13×10^{2}
	Worst	1.24×10^{3}	7.16×10^{2}	2.53×10^{4}	8.47×10^{2}	1.44×10^{3}	1.52×10^{3}	8.99×10^{2}	7.85×10^{2}
	Mean	6.56×10^{2}	**2.62×10^{2}**	4.56×10^{3}	5.12×10^{2}	1.11×10^{3}	9.95×10^{2}	6.55×10^{2}	4.30×10^{2}
	Std	2.54×10^{2}	1.52×10^{2}	5.32×10^{3}	1.69×10^{2}	2.83×10^{2}	2.36×10^{2}	1.42×10^{2}	1.42×10^{2}
F18	Best	1.22×10^{5}	4.52×10^{4}	1.39×10^{6}	2.99×10^{4}	1.21×10^{5}	6.59×10^{4}	3.34×10^{5}	3.62×10^{4}
	Worst	7.45×10^{6}	1.00×10^{7}	8.34×10^{7}	6.46×10^{5}	4.79×10^{6}	6.22×10^{5}	4.25×10^{6}	8.51×10^{5}
	Mean	2.00×10^{6}	1.25×10^{6}	1.94×10^{7}	**1.74×10^{5}**	1.10×10^{6}	2.11×10^{5}	1.76×10^{6}	2.01×10^{5}
	Std	2.02×10^{6}	2.31×10^{6}	2.08×10^{7}	1.29×10^{5}	1.14×10^{6}	1.18×10^{5}	9.62×10^{5}	1.67×10^{5}
F19	Best	3.22×10^{4}	3.07×10^{3}	1.08×10^{7}	1.29×10^{3}	8.01×10^{5}	1.52×10^{3}	4.01×10^{6}	1.40×10^{2}
	Worst	7.51×10^{6}	3.19×10^{7}	8.76×10^{8}	5.96×10^{4}	1.31×10^{6}	5.93×10^{3}	5.99×10^{7}	6.12×10^{4}
	Mean	2.11×10^{6}	2.30×10^{6}	2.51×10^{8}	1.75×10^{4}	1.05×10^{6}	**3.66×10^{3}**	2.03×10^{7}	7.04×10^{3}
	Std	2.01×10^{6}	6.56×10^{6}	2.70×10^{8}	2.00×10^{4}	1.15×10^{5}	1.21×10^{3}	1.38×10^{3}	1.24×10^{4}
F20	Best	3.31×10^{2}	1.67×10^{2}	3.75×10^{2}	1.41×10^{2}	3.65×10^{2}	7.34×10^{2}	3.92×10^{2}	1.71×10^{2}
	Worst	1.04×10^{3}	6.80×10^{2}	1.02×10^{3}	8.61×10^{2}	9.50×10^{2}	1.24×10^{3}	9.12×10^{2}	8.79×10^{2}
	Mean	6.00×10^{2}	5.34×10^{2}	8.36×10^{2}	4.91×10^{2}	6.92×10^{2}	1.02×10^{3}	5.62×10^{2}	**4.47×10^{2}**
	Std	2.01×10^{2}	1.32×10^{2}	1.17×10^{2}	1.99×10^{2}	1.50×10^{2}	1.53×10^{2}	1.11×10^{2}	1.87×10^{2}
F21	Best	3.54×10^{2}	2.47×10^{2}	1.50×10^{2}	2.82×10^{2}	3.84×10^{2}	3.64×10^{2}	4.33×10^{2}	2.95×10^{2}
	Worst	5.52×10^{2}	3.18×10^{2}	5.76×10^{2}	4.44×10^{2}	5.94×10^{2}	4.64×10^{2}	5.25×10^{2}	4.88×10^{2}
	Mean	4.35×10^{2}	**2.82×10^{2}**	3.16×10^{2}	3.44×10^{2}	5.13×10^{2}	4.11×10^{2}	4.64×10^{2}	3.67×10^{2}
	Std	4.79×10^{1}	1.84×10^{1}	8.86×10^{1}	3.44×10^{1}	5.21×10^{1}	2.69×10^{1}	2.25×10^{1}	5.69×10^{1}
F22	Best	1.19×10^{2}	1.62×10^{2}	7.83×10^{2}	1.00×10^{2}	4.23×10^{3}	1.00×10^{2}	1.26×10^{3}	1.00×10^{2}
	Worst	6.25×10^{3}	4.36×10^{3}	2.85×10^{3}	5.34×10^{3}	6.59×10^{3}	5.55×10^{3}	2.03×10^{3}	6.06×10^{3}
	Mean	3.10×10^{3}	2.24×10^{3}	1.83×10^{3}	3.49×10^{3}	5.94×10^{3}	2.18×10^{3}	1.62×10^{3}	**1.00×10^{2}**
	Std	2.15×10^{3}	1.62×10^{3}	5.51×10^{2}	1.65×10^{3}	5.71×10^{2}	2.44×10^{3}	2.03×10^{2}	7.36×10^{-1}
F23	Best	6.83×10^{2}	3.96×10^{2}	5.19×10^{2}	4.14×10^{2}	8.05×10^{2}	7.68×10^{2}	7.42×10^{2}	4.63×10^{2}
	Worst	1.06×10^{3}	5.48×10^{2}	1.15×10^{3}	7.00×10^{2}	1.47×10^{3}	1.35×10^{3}	9.66×10^{2}	6.01×10^{2}
	Mean	8.19×10^{2}	**4.53×10^{2}**	8.87×10^{2}	5.59×10^{2}	1.17×10^{3}	1.06×10^{3}	8.37×10^{2}	5.17×10^{2}
	Std	9.50×10^{1}	3.36×10^{1}	1.08×10^{2}	6.27×10^{1}	1.67×10^{2}	1.40×10^{2}	5.52×10^{1}	4.07×10^{1}
F24	Best	6.82×10^{2}	4.77×10^{2}	9.11×10^{2}	5.87×10^{2}	1.07×10^{3}	6.20×10^{2}	8.53×10^{2}	4.90×10^{2}
	Worst	1.20×10^{3}	6.61×10^{2}	1.75×10^{3}	8.53×10^{2}	1.77×10^{3}	8.31×10^{2}	1.03×10^{3}	7.10×10^{2}
	Mean	9.02×10^{2}	**5.36×10^{2}**	1.31×10^{3}	6.76×10^{2}	1.44×10^{3}	7.43×10^{2}	9.45×10^{2}	5.89×10^{2}
	Std	1.09×10^{2}	4.77×10^{1}	2.05×10^{2}	6.29×10^{1}	1.88×10^{2}	5.35×10^{1}	4.55×10^{1}	5.61×10^{1}
F25	Best	4.03×10^{2}	4.39×10^{2}	2.24×10^{3}	3.84×10^{2}	1.28×10^{3}	3.93×10^{2}	6.43×10^{2}	3.84×10^{2}
	Worst	6.38×10^{2}	7.07×10^{2}	4.00×10^{3}	4.40×10^{2}	2.72×10^{3}	4.40×10^{2}	9.04×10^{2}	4.43×10^{2}
	Mean	4.97×10^{2}	5.01×10^{2}	2.93×10^{3}	3.93×10^{2}	1.77×10^{3}	4.16×10^{2}	7.55×10^{2}	**3.90×10^{2}**
	Std	5.68×10^{1}	5.83×10^{1}	4.81×10^{2}	1.59×10^{1}	3.18×10^{2}	1.65×10^{1}	5.35×10^{1}	1.12×10^{1}

Continued on next page

Function	Metric	WOA	GWO	BOA	DO	AOA	GSA	SCA	EDBO
F26	Best	1.30×10^{3}	1.60×10^{3}	3.97×10^{3}	2.01×10^{2}	4.67×10^{3}	2.00×10^{2}	2.15×10^{3}	2.00×10^{2}
	Worst	6.37×10^{3}	2.81×10^{3}	9.26×10^{3}	5.08×10^{3}	8.30×10^{3}	4.75×10^{3}	5.59×10^{3}	3.90×10^{3}
	Mean	4.85×10^{3}	2.03×10^{3}	7.48×10^{3}	2.86×10^{3}	6.79×10^{3}	**1.58×10^{3}**	4.62×10^{3}	2.54×10^{3}
	Std	1.09×10^{3}	3.02×10^{2}	1.07×10^{3}	9.05×10^{2}	7.90×10^{2}	1.76×10^{3}	1.02×10^{3}	1.43×10^{3}
F27	Best	7.52×10^{2}	5.12×10^{2}	6.65×10^{2}	5.23×10^{2}	1.21×10^{3}	8.48×10^{2}	9.16×10^{2}	5.03×10^{2}
	Worst	1.35×10^{3}	6.08×10^{2}	1.38×10^{3}	7.14×10^{2}	2.22×10^{3}	1.65×10^{3}	1.18×10^{3}	5.52×10^{2}
	Mean	9.59×10^{2}	5.42×10^{2}	9.85×10^{2}	5.74×10^{2}	1.73×10^{3}	1.11×10^{3}	1.03×10^{3}	**5.24×10^{2}**
	Std	1.43×10^{2}	1.97×10^{1}	1.63×10^{2}	3.72×10^{1}	2.69×10^{2}	2.26×10^{2}	6.18×10^{1}	1.01×10^{1}
F28	Best	5.40×10^{2}	5.27×10^{2}	3.17×10^{3}	3.00×10^{2}	1.81×10^{3}	3.35×10^{2}	8.69×10^{2}	3.00×10^{2}
	Worst	1.24×10^{3}	1.73×10^{3}	5.75×10^{3}	4.61×10^{2}	4.03×10^{3}	5.22×10^{2}	1.68×10^{3}	4.56×10^{2}
	Mean	7.32×10^{2}	6.65×10^{2}	4.89×10^{3}	3.87×10^{2}	3.24×10^{3}	4.23×10^{2}	1.21×10^{3}	**3.45×10^{2}**
	Std	1.33×10^{2}	2.13×10^{2}	6.21×10^{2}	4.56×10^{1}	5.46×10^{2}	4.60×10^{1}	1.73×10^{2}	6.39×10^{1}
F29	Best	1.24×10^{3}	5.95×10^{2}	3.32×10^{3}	5.56×10^{2}	1.83×10^{3}	1.07×10^{3}	1.47×10^{3}	6.33×10^{2}
	Worst	3.05×10^{3}	1.10×10^{3}	1.52×10^{4}	1.55×10^{3}	4.03×10^{3}	1.86×10^{3}	2.59×10^{3}	1.35×10^{3}
	Mean	1.88×10^{3}	8.33×10^{2}	5.26×10^{3}	**1.01×10^{3}**	3.08×10^{3}	1.52×10^{3}	1.97×10^{3}	**1.01×10^{3}**
	Std	4.52×10^{2}	1.38×10^{2}	2.33×10^{3}	2.45×10^{2}	5.97×10^{2}	1.62×10^{2}	2.67×10^{2}	2.02×10^{2}
F30	Best	5.23×10^{5}	2.15×10^{5}	3.72×10^{7}	1.93×10^{4}	3.37×10^{6}	2.54×10^{4}	2.49×10^{7}	2.24×10^{3}
	Worst	2.68×10^{8}	2.16×10^{7}	3.11×10^{9}	2.63×10^{5}	4.52×10^{8}	6.12×10^{4}	1.86×10^{8}	3.99×10^{4}
	Mean	2.91×10^{7}	6.07×10^{6}	7.12×10^{8}	1.20×10^{5}	4.74×10^{7}	3.61×10^{4}	6.83×10^{7}	**9.99×10^{3}**
	Std	4.93×10^{7}	4.75×10^{6}	6.11×10^{8}	6.93×10^{4}	9.37×10^{7}	8.34×10^{3}	3.32×10^{7}	6.73×10^{3}
	+/≈/–	28/1/0	15/5/9	28/0/1	14/13/2	27/2/0	18/6/5	29/0/0	
	ARV	5.2069	3.0690	7.3103	3.0345	6.3793	3.3103	5.6552	**2.0345**
	Rank	5	3	8	2	7	4	6	**1**

6. UAV 3D path planning model

The comprehensive objective function is constructed based on the UAV's trajectory length cost, threat cost, altitude cost, and smoothness cost. The optimization process of this comprehensive objective function aims to find the optimal path for the UAV.

6.1. Trajectory length cost

The trajectory length represents the distance traveled by the UAV from the starting point to the endpoint. A shorter trajectory reduces the risk of UAV losing control caused by frequent attitude changes and helps shorten the flight time. The formula for calculating the trajectory length is shown in Eq (17).

$$F_1(X_i) = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \tag{17}$$

where n is the number of trajectory points, and (x_i, y_i, z_i) represents the coordinates of the i-th trajectory point.

6.2. Threat cost

Introducing the flight threat cost ensures the safety of the UAV. As shown in Figure 9, the obstacle radius is R_k, D is the UAV's safe flight distance threshold, and d_k is the perpendicular distance from the UAV's trajectory to the obstacle. To ensure safe flight, $d_k$must be greater than D. The threat cost calculation formula is shown in Eq (18), and the expression for $T_k(\overrightarrow{W_{iJ}W_{i,J+1}})$ is given in Eq (19).

$$F_2(X_i) = \sum_{j=1}^{n-1}\sum_{k=1}^{k} T_k\left(\overrightarrow{W_{iJ}W_{i,J+1}}\right) \tag{18}$$

$$T_k(\overrightarrow{W_{iJ}W_{i,J+1}})=\begin{cases} 0,\ (d_k > D + R_k) \\ \gamma_c\big((D + R_k) - d_k\big),\ (R_k < d_k < D + R_k) \\ \infty,\ (d_k < R_k) \end{cases} \tag{19}$$

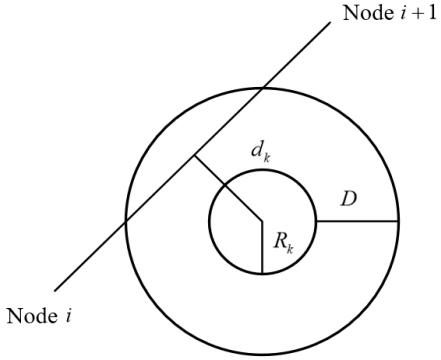


Figure 9. Obstacle collision area diagram.

6.3. Flight altitude cost

The flight altitude of the UAV should be within a certain range. Flying too low can ma ke the UAV susceptible to terrain obstacles, while flying too high can lead to excessive ener gy consumption. The calculation formula for the flight altitude is given in Eq (20).

$$H_i = \begin{cases} \left|h_{ij} - \frac{h_{max}-h_{min}}{2}\right|,\ h_{min} < h_i \le h_{max} \\ \infty,\quad other \end{cases} \tag{20}$$

The flight altitude cost function is shown in Eq (21).

$$F_3(X_i) = \sum_{j=1}^{n} H_i \tag{21}$$

6.4. Path smoothness cost

The flight path of the UAV should minimize sharp turns and large altitude changes as much as possible. Both conditions must comply with the actual turn angle constraints of the UAV; otherwise,

the path planning model will fail to generate a feasible flight path. Eqs (22) and (23) are the calculation formulas for the deflection angle φ_i and pitch angle ϕ_i, respectively. Meanwhile, the path smoothness cost function is shown in Eq (24).

$$\varphi_i = \arctan\left(\frac{\|l_i l_{i+1}\|}{l_i l_{i+1}}\right) \tag{22}$$

$$\phi_i = \arctan\left(\frac{z_{i+1}-z_i}{\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2}}\right) \tag{23}$$

$$F_4 = \sum_{i-1}^{n-2} \varphi_i + \sum_{i=1}^{n-1} (\varphi_i - \varphi_{i-1}) \tag{24}$$

6.5. Comprehensive path cost

This paper performs a weighted combination of the above four costs to construct a comprehensive objective function for multi-objective path planning, as shown in Eq (25).

$$F = b_1 F_1 + b_2 F_2 + b_3 F_3 + b_4 F_4 \tag{25}$$

where b_1, b_2, b_3, b_4 are the weight coefficients for the path length cost, threat cost, flight altitude cost, and path smoothness cost, respectively. The smaller the value of F, the higher the quality of the algorithm's path.

7. Simulation and analysis of UAV 3D path planning

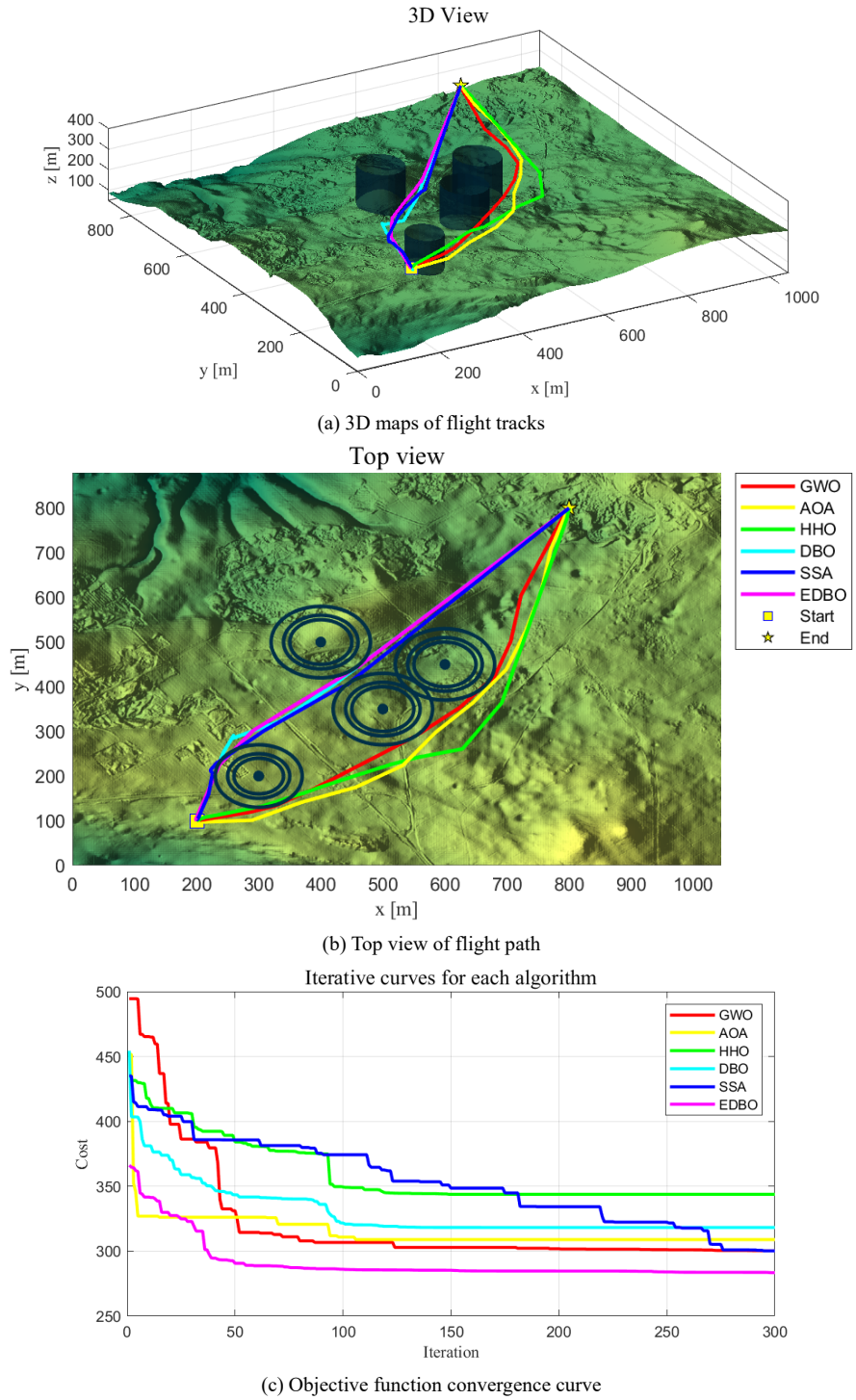
To assess the effectiveness and advantages of the EDBO algorithm in trajectory planning proposed in this study, UAV trajectory planning simulation experiments were conducted on multiple maps. Five representative advanced metaheuristic algorithms, namely, GWO, DBO, AOA, Harris Hawks optimizer (HHO) [55], and SSA, are selected for comparison. To ensure fairness, the population size for all algorithms is set to 200, with a maximum number of iterations set to 300. Each metaheuristic algorithm is independently run 30 times. The task is established in a 1000 m × 900 m × 400 m space. The start point is set at coordinates (200, 100, 150), and the endpoint is set at coordinates (800, 800, 150).

First, a low-complexity scenario (Scenario 1) is established with four obstacles. The 2D coordinates and radius parameters of each obstacle are (400, 500, 50), (300, 200, 40), (500, 350, 50), and (600, 400, 40). The trajectory maps and the convergence curves of the overall cost fitness for all algorithms are shown in Figure 10.

As seen in Figure 10(a),(b), the trajectory obtained by HHO is the longest and most winding. The trajectories obtained by GWO and AOA are similar to that of HHO, but neither achieved satisfactory results. The trajectories planned by GWO, DBO, and EDBO are similar, but the trajectory planned by EDBO is smoother, with a shorter flight distance and a further reduction in trajectory cost. As shown in Figure 10(c), EDBO converges quickly in the early stages and reaches the optimal value after about 60 iterations. In comparison with the final results, EDBO has the lowest overall cost, followed by GWO. It is also observed that EDBO is able to converge very quickly to the optimal value at the start of the iterations, and its performance is significantly improved compared to the original DBO. This indicates that the proposed search strategy effectively enhances the convergence rate and improves the

precision of convergence.

Next, a high-complexity scenario (Scenario 2) is established with seven obstacles. The 2D coordinates and radius parameters of each obstacle are (400, 500, 50), (600, 200, 40), (500, 350, 50), (350, 200, 40), (700, 550, 40), (550, 400, 45), and (650, 750, 50). The trajectory maps and the convergence curves of the overall cost fitness function for all algorithms are shown in Figure 11.

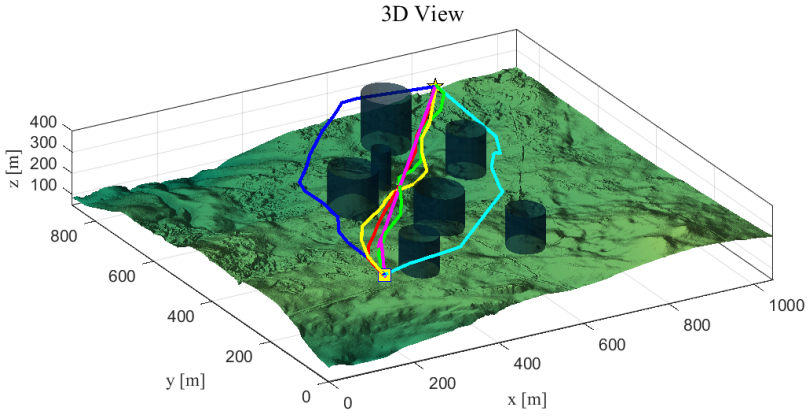


(a) 3D maps of flight tracks

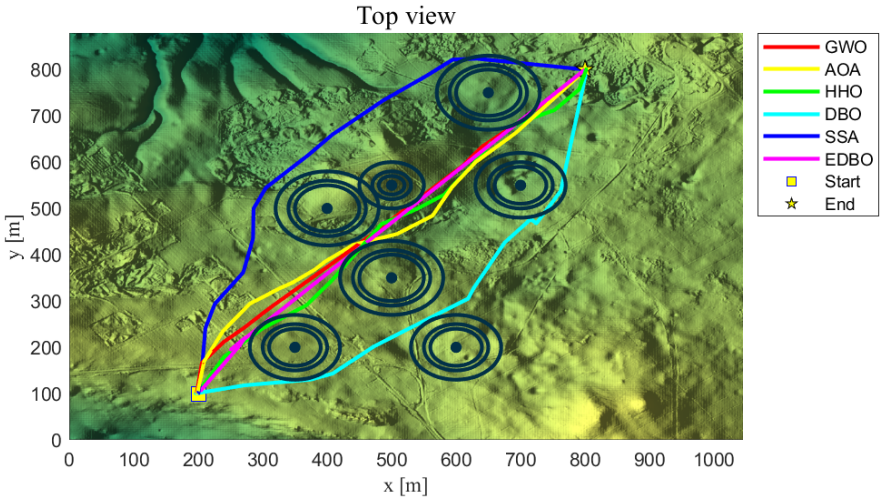
(b) Top view of flight path

(c) Objective function convergence curve

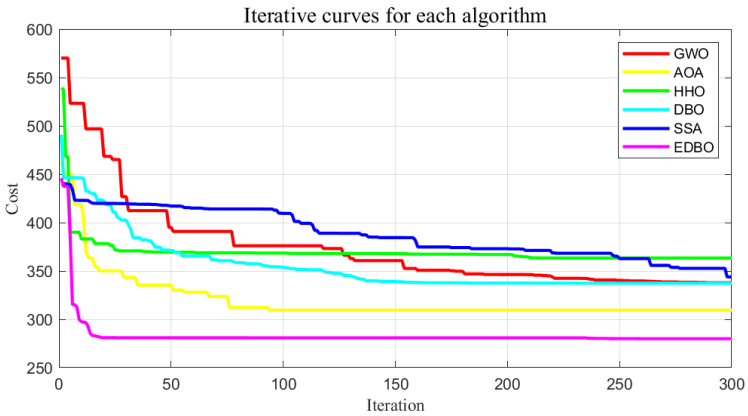
Figure 10. The trajectory planning comparison results and convergence curves in Scene 1.



(a) 3D maps of flight tracks



(b) Top view of flight path



(c) Objective function convergence curve

Figure 11. The trajectory planning comparison results and convergence curves in Scene 2.

Scenario 2 has more obstacles than Scenario 1, and the distribution of the obstacles is more complex, significantly increasing the difficulty of trajectory planning. By analyzing the trajectory planning results in Figures 11(a),(b), it can be observed that the trajectories planned by SSA and DBO are the longest, while the trajectories planned by GWO, AOA, and HHO are similar, with AOA achieving a lower smoothness cost. EDBO, utilizing appropriate turn rates and flight proximity rates, safely navigates near hazardous areas, and the planned 3D trajectory is more reasonable. As shown in

Figure 11(c), even with more obstacles, EDBO still performs the best, quickly converging in the early stages and achieving the highest accuracy.

Each experiment for the scenes is independently run 30 times, and the trajectory planning results are recorded in Table 10, with the best-ranked values in each metric highlighted in bold. The results show that the average overall cost of EDBO is reduced by 9.4% and 14.5% compared to the original DBO algorithm. To more intuitively demonstrate the advantages of the EDBO algorithm, Figures 12 and 13 respectively present box plots comparing EDBO with other benchmark algorithms under Scenario 1 and Scenario 2, clearly illustrating the performance differences across different scenarios. In conclusion, the EDBO algorithm demonstrates strong performance, achieving not only higher optimization accuracy but also greater adaptability in different scenarios, leading to optimal trajectories.

Table 10. Statistical results of the trajectory planning experiment.

Map	Algorithm	Worst	Best	Mean	Std
Scenario 1	GWO	366.071	286.208	301.597	23.590
	AOA	331.347	281.569	291.529	11.657
	HHO	386.081	343.961	355.995	19.287
	DBO	347.247	299.006	313.699	11.237
	SSA	339.943	291.548	306.861	15.876
	EDBO	**296.541**	**270.117**	**284.057**	**6.665**
Scenario 2	GWO	384.263	281.574	317.813	26.759
	AOA	331.778	289.457	309.418	14.571
	HHO	384.201	347.324	363.379	22.747
	DBO	357.562	309.457	327.290	19.363
	SSA	388.147	327.660	343.770	21.549
	EDBO	**287.158**	**269.581**	**279.946**	**4.759**

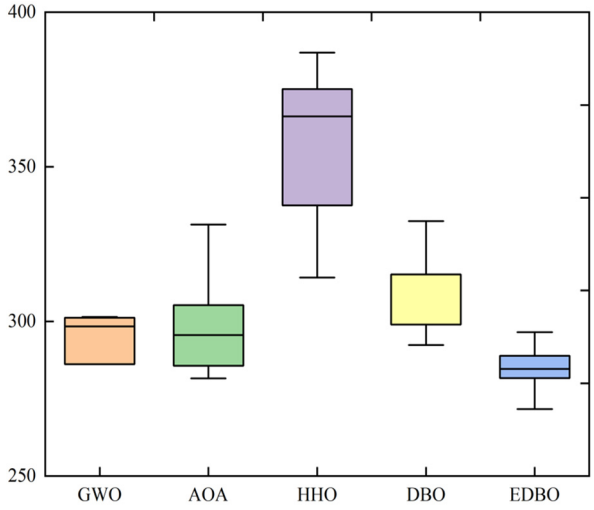


Figure 12. Box plot of Scenario 1.

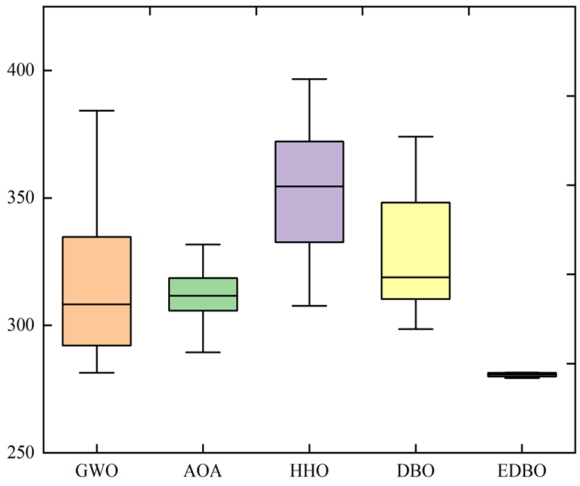


Figure 13. Box plot of Scenario 2.

8. Conclusions

This study proposed an EDBO based on the limitations of the original DBO algorithm. First, the population is initialized using the Sobol sequence to improve population diversity and reduce the likelihood of the algorithm becoming trapped in local optima. Second, a broad global search is conducted using random angles and adaptive step sizes, while individual positions are fine-tuned based on probability ratios and linear transformations. This improves the algorithm's ability to explore the global search space and enhances its convergence precision. Then, an improved convergence factor is introduced to give the algorithm stronger global search ability in the early stages and enhance local search precision in the later stages. Finally, an adaptive T-distribution mutation strategy is developed to enhance the likelihood of the algorithm avoiding local optima.

To validate the performance of EDBO, benchmark functions from CEC2014 and CEC2017 were used to assess the improvements brought by the algorithm. In addition, three existing DBO variants and seven advanced evolutionary algorithms were selected for comparison. The results demonstrate that EDBO consistently outperforms these competitors, showing significant improvements in convergence speed, stability, and robustness. These results highlight EDBO's superior optimization performance, particularly in the context of complex real-world applications such as three-dimensional UAV path planning. This study validates EDBO's superior optimization efficacy, which not only demonstrates its operational viability in real-world systems but also highlights scalability across multidisciplinary domains. Furthermore, the framework's dynamic adaptability to constraint-rich environments positions it as a robust candidate for solving high-dimensional nonconvex optimization challenges, particularly in precision-critical engineering applications requiring rigorous computational efficiency. However, it should be noted that in UAV path planning tasks, UAVs must be able to respond rapidly to dynamic task changes, and may need to scale in number when necessary. This is a current limitation of the study that warrants further investigation into more feasible strategies to enhance the overall performance of the algorithm and address emerging real-world challenges.

At present, more advanced optimization algorithms, such as adaptive heuristic algorithms, generative constructive hyper-heuristic algorithms, and polyploidy algorithms, have been proposed by researchers and applied in fields such as online learning, scheduling, multi-objective optimization, transportation, medicine, and data classification [56–59]. These advanced algorithms possess stronger

problem-specific adaptability, flexibility, and scalability, making them indispensable in solving complex decision-making problems. In the future, we will focus on expanding the EDBO algorithm and exploring its integration with advanced optimization methods, such as generation-perturbation hyper-heuristics. We aim to apply the enhanced EDBO framework to various domains including UAV path planning, machine learning, healthcare, and energy management. Furthermore, we plan to refine the structure of EDBO to further enhance its performance. Future work will not only aim to push the boundaries of algorithmic optimization but also provide practical solutions to pressing problems faced across different industries.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the Scientific Research Foundation of Guilin University of Technology (No. RD2100002991).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. F. Li, H. Xu, F. Qiu, Modified artificial rabbits optimization combined with bottlenose dolphin optimizer in feature selection of network intrusion detection, *Electron. Res. Arch.*, **32** (2024), 1770–1880. https://doi.org/10.3934/era.2024081
2. B. A. Berg, Locating global minima in optimization problems by a random-cost approach, *Nature*, **361** (1993), 708–710. https://doi.org/10.1038/361708a0
3. M. Kmich, I. Harrade, H. Karmouni, M. Sayyouri, S. S. Askar, M. Abouhawwash, Image registration using the arithmetic optimization algorithm for robotic visual servoing, *Int. J. Comput. Intell. Syst.*, **18** (2025), 1. https://doi.org/10.1007/s44196-024-00612-7
4. M. A. Tahiri, H. Karmouni, A. Bencherqui, A. Daoui, M. Sayyouri, H. Qjidaa, et al., New color image encryption using hybrid optimization algorithm and Krawtchouk fractional transformations, *Visual Comput.*, **39** (2023), 6395–6420. https://doi.org/10.1007/s00371-022-02736-3
5. M. Kmich, H. Karmouni, I. Harrade, M. J. Ouazzani, H. Qjidaa, M. Sayyouri, Improved intensity-based image registration via archimedes optimization algorithm, in *2024 International Conference on Intelligent Systems and Computer Vision (ISCV)*, (2024), 1–5. https://doi.org/10.1109/ISCV60512.2024.10620082
6. Y. Zhang, Z. Wang, H. Wang, F. Blaabjerg, Artificial intelligence-aided thermal model considering cross-coupling effects, *IEEE Trans. Power Electron.*, **35** (2020), 9998–10002. https://doi.org/10.1109/TPEL.2020.2980240
7. H. Karmouni, M. Chouiekh, S. Motahhir, H. Qjidaa, M. O. Jamil, M. Sayyouri, Optimization and implementation of a photovoltaic pumping system using the sine-cosine algorithm, *Eng. Appl. Artif. Intell.*, **114** (2022), 105104. https://doi.org/10.1016/j.engappai.2022.105104

8. S. Wang, S. Li, H. Yu, A power generation accumulation-based adaptive chaotic differential evolution algorithm for wind turbine placement problems, *Electron. Res. Arch.*, **32** (2024), 4659–4683. https://doi.org/10.3934/era.2024212
9. A. Bencherqui, M. A. Tahiri, H. Karmouni, M. Alfidi, Y. El Afou, H. Qjidaa, et al., Chaos-enhanced archimede algorithm for global optimization of real-world engineering problems and signal feature extraction, *Processes*, **12** (2024), 406. https://doi.org/10.3390/pr12020406
10. R. Zhang, X. Li, H. Ren, Y. Ding, Y. Meng, Q. Xia, UAV Flight path planning based on multi-strategy improved white sharks optimization, *IEEE Access*, **11** (2023). https://doi.org/88462-88475.10.1109/ACCESS.2023.3304708
11. J. Luo, B. Shi, A hybrid whale optimization algorithm based on modified differential evolution for global optimization problems, *Appl. Intell.*, **49** (2019), 1982–2000. https://doi.org/10.1007/s10489-018-1362-4
12. Z. Wei, W. Gao, G. Li, Q. Zhang, A penalty-based differential evolution for multimodal optimization, *IEEE Trans. Cybern.*, **52** (2021). https://doi.org/6024-6033.10.1109/TCYB.2021.3117359
13. A. E. Ezugwu, A. K. Shukla, R. Nath, A. A. Akinyelu, J. O. Agushaka, H. Chiroma, et al., Metaheuristics: A comprehensive overview and classification along with bibliometric analysis, *Artif. Intell. Rev.*, **54** (2021), 4237–4316. https://doi.org/10.1007/s10462-020-09952-0
14. J. H. Holland, Genetic algorithms and the optimal allocation of trials, *SIAM J. Comput.*, **2** (1973), 88–105. https://doi.org/10.1137/0202009
15. J. Kennedy, R. Eberhart, Particle swarm optimization, in *Proceedings of ICNN'95-International Conference on Neural Networks*, **4** (1995), 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
16. M. Dorigo, V. Maniezzo, A. Colorni, Ant system: Optimization by a colony of cooperating agents, *IEEE Trans. Syst. Man Cybern. Part B*, **26** (1996), 29–41. https://doi.org/10.1109/3477.484436
17. R. Storn, K. Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, *J. Global Optim.*, **11** (1997), 341–359. https://doi.org/10.1023/A:1008202821328
18. R. Rajabioun, Cuckoo optimization algorithm, *Appl. Soft Comput.*, **11** (2011), 5508–5518. https://doi.org/10.1016/j.asoc.2011.05.008
19. M. K. Marichelvam, T. Prabaharan, X. S. Yang, A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems, *IEEE Trans. Evol. Comput.*, **18** (2014), 301–305. https://doi.org/10.1109/TEVC.2013.2240304
20. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Software*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
21. J. Cheng, Q. Lin, Y. Xiong, Sine cosine algorithm with peer learning for global numerical optimization, *Eng. Optim.*, **2024** (2024), 1–18. https://doi.org/10.1080/0305215X.2024.2340054
22. A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, *Comput. Struct.*, **169** (2016), 1–12. https://doi.org/10.1016/j.compstruc.2016.03.001
23. M. Seyedali, L. Andrew, The whale optimization algorithm, *Adv. Eng. Software*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
24. J. Xue, B. Shen, A novel swarm intelligence optimization approach: Sparrow search algorithm, *Syst. Sci. Control Eng.*, **8** (2020), 22–34. https://doi.org/10.1080/21642583.2019.1708830
25. J. Xue, B. Shen, Dung beetle optimizer: A new meta-heuristic algorithm for global optimization, *J. Supercomput.*, **79** (2023), 7305–7336. https://doi.org/10.1007/s11227-022-04959-6

26. B. Zheng, Y. Chen, C. Wang, A. A. Heidari, L. Liu, H. Chen, The moss growth optimization (MGO): Concepts and performance, *J. Comput. Design Eng.*, **11** (2024), 184–221. https://doi.org/10.1093/jcde/qwae080
27. D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE Trans. Evol. Computat.*, **1** (1997), 67–82. https://doi.org/10.1109/4235.585893
28. J. Xu, Z. Han, L. Yin, Z. Yan, Y. Yu, G. Ma, Multi-strategy-based artificial bee colony algorithm for AUV path planning with angle constraints, *Ocean Eng.*, **312** (2024), 119155. https://doi.org/10.1016/j.oceaneng.2024.119155
29. I. Attiya, M. Abd Elaziz, L. Abualigah, T. N. Nguyen, A. A. El-Latif, An improved hybrid swarm intelligence for scheduling IoT application tasks in the cloud, *IEEE Trans. Ind. Inf.*, **18** (2022), 6264–6272. https://doi.org/10.1109/TII.2022.3148288
30. Y. Shen, C. Zhang, F. S. Gharehchopogh, S. Mirjalili, An improved whale optimization algorithm based on multi-population evolution for global optimization and engineering design problems, *Exp. Syst. Appl.*, **215** (2023), 119296. https://doi,org/10.1016/j.eswa.2022.119269
31. F. Zhu, G. Li, H. Tang, Y. Li, X. Lv, X. Wang, Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems, *Exp. Syst. Appl.*, **236** (2024), 121219. https://doi.org/10.1016/j.eswa.2023.121219
32. J. He, L. Fu, Robot path planning based on improved dung beetle optimizer algorithm, *J. Braz. Soc. Mech. Sci. Eng.*, **46** (2024), 235. https://doi.org/10.1007/s40430-024-04768-3
33. Z. Wang, L. Huang, S. Yang, D. Li, D. He, S. Chan, A quasi-oppositional learning of updating quantum state and Q-learning based on the dung beetle algorithm for global optimization, *Alexandria Eng. J.*, **81** (2023), 469–488. https://doi.org/10.1016/j.aej.2023.09.042
34. Q. Li, H. Shi, W. Zhao, C. Ma, Enhanced dung beetle optimization algorithm for practical engineering optimization, *Mathematics*, **12** (2024), 1084. https://doi.org/10.3390/math12071084
35. M. Bukumira, M. Zivkovic, M. Antonijevic, L. Jovanovic, N. Bacanin, T. Zivkovic, The extreme gradient boosting method optimized by hybridized sine cosine metaheuristics for ship vessel classification, in *International Conference on Advances in Data-driven Computing and Intelligent Systems*, (2023), 255–270. https://doi.org/10.1007/978-981-99-9524-0_20
36. B. Radomirovic, L. Jovanovic, C. Stoean, M. Zivkovic, A. Njegus, N. Bacanin, Solar flare classification using modified metaheuristic optimized xgboost, in *2023 25th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, (2023), 287–292. https://doi.org/10.1109/SYNASC61333.2023.00049
37. C. Lopez-Franco, D. Diaz, J. Hernandez-Barragan, N. Arana-Daniel, M. Lopez-Franco, A metaheuristic optimization approach for trajectory tracking of robot manipulators, *Mathematics*, **10** (2022), 1051. https://doi.org/10.3390/math10071051
38. A. Bencherqui, M. A. Tahiri, H. Karmouni, M. Alfidi, S. Motahhir, M. Abouhawwash, et al., Optimal algorithm for color medical encryption and compression images based on DNA coding and a hyperchaotic system in the moments, *Eng. Sci. Technol. Int. J.*, **50** (2024), 101612. https://doi.org/10.1016/j.jestch.2023.101612
39. M. Kmich, N. El Ghouate, A. Bencharqui, H. Karmouni, M. Sayyouri, S. S. Askar, et al., Chaotic Puma optimizer algorithm for controlling wheeled mobile robots, *Eng. Sci. Technol. Int. J.*, **63** (2025), 101982. https://doi.org/10.1016/j.jestch.2025.101982
40. H. Karmouni, M. A. Tahiri, I. Dagal, H. Amakdouf, M. O. Jamil, H. Qjidaa, et al., Secure and optimized satellite image sharing based on chaotic eπ map and Racah moments, *Exp. Syst. Appl.*, **236** (2024), 121247. https://doi.org/10.1016/j.eswa.2023.121247

41. N. El Ghouate, A. Bencherqui, H. Mansouri, A. E. Maloufy, M. A. Tahiri, H. Karmouni, et al., Improving the Kepler optimization algorithm with chaotic maps: Comprehensive performance evaluation and engineering applications, *Artif. Intell. Rev.*, **57** (2024), 313. https://doi.org/10.1007/s10462-024-10857-5
42. Y. Lu, X. Lu, G. Yang, X. Xiong, Robust zero-watermarking algorithm for multi-medical images based on FFST-Schur and Tent mapping, *Biomed. Signal Process. Control*, **96** (2024) 106557. https://doi.org/10.1016/j.bspc.2024.106557
43. M. Ye, H. Zhou, H. Yang, B. Hu, X. Wang, Multi-strategy improved dung beetle optimization algorithm and its applications, *Biomimetics*, **9** (2024), 291. https://doi.org/10.3390/biomimetics9050291
44. M. Zhang, D. Long, T. Qin, J. Yang, A chaotic hybrid butterfly optimization algorithm with particle swarm optimization for high-dimensional optimization problems, *Symmetry*, **12** (2020), 1800. https://doi.org/10.3390/sym12111800
45. S. Mirjalili, A. Lewis, Adaptive gbest-guided gravitational search algorithm, *Neural Comput. Appl.*, **25** (2014), 1569–1584. https://doi.org/10.1007/s00521-014-1640-y
46. W. C. Wong, C. Y. Chung, K. W. Chan, H. Chen, Quasi-Monte Carlo based probabilistic small signal stability analysis for power systems with plug-in electric vehicle and wind power integration, *IEEE Trans. Power Syst.*, **28** (2013), 3335–3343. https://doi.org/10.1109/TPWRS.2013.2254505
47. C. Xu, Y. Mao, H. Chen, H. Tao, F. Liu, Skew t distribution-based nonlinear filter with asymmetric measurement noise using variational Bayesian inference, *Comput. Model. Eng. Sci.*, **131** (2022), 349–364. https://doi.org/10.32604/cmes.2021.019027
48. H. Zhang, Q. Huang, L. Ma, Z. Zhang, Sparrow search algorithm with adaptive t distribution for multi-objective low-carbon multimodal transportation planning problem with fuzzy demand and fuzzy time, *Exp. Syst. Appl.*, **238** (2024) 122042. https://doi.org/10.1016/j.eswa.2023.122042
49. J. Liang, B. Qu, P. N. Suganthan, *Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization*, Technical Report 201311, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University: Singapore, 2013.
50. G. Wu, R. Mallipeddi, P. Suganthan, Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization, *Nanyang Technol. Univ. Singapore Tech. Rep*, **2016** (2016), 1–18.
51. S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, *Soft Comput.*, **23** (2019), 715–734. https://doi.org/10.1007/s00500-018-3102-4
52. S. Zhao, T. Zhang, S. Ma, M. Chen, Dandelion optimizer: A nature-inspired metaheuristic algorithm for engineering applications, *Eng. Appl. Artif. Intell.*, **114** (2022), 105075. https://doi.org/10.1016/j.engappai.2022.105075
53. L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A. H. Gandomi, The arithmetic optimization algorithm, *Comput. Methods Appl. Mech. Eng.*, **376** (2021), 113609. https://doi.org/10.1016/j.cma.2020.113609
54. E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: A gravitational search algorithm, *Inf. Sci.*, **179** (2009), 2232–2248. https://doi.org/10.1016/j.ins.2009.03.004
55. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris Hawks optimization: Algorithm and applications, *Future Generation Comput. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028

56. E. Singh, N. Pillay, A study of ant-based pheromone spaces for generation constructive hyper-heuristics, *Swarm Evol. Comput.*, **72** (2022), 101095. https://doi.org/10.1016/j.swevo.2022.101095
57. B. Li, P. Afkhami, R. Khayamim, Z. Elmi, R. Moses, J. Sobanjo, et al., A holistic optimization-based approach for sustainable selection of level crossings for closure with safety, economic, and environmental considerations, *Reliab. Eng. Syst. Safety*, **249** (2024), 110197. https://doi.org/10.1016/j.ress.2024.110197
58. M. A. Dulebenets, An adaptive Polyploid memetic algorithm for scheduling trucks at a cross-docking terminal, *Inf. Sci.*, **565** (2021), 390–421. https://doi.org/10.1016/j.ins.2021.02.039
59. M. Chen, J. Xu, W. Zhang, Z. Li, A new customer-oriented multi-task scheduling model for cloud manufacturing considering available periods of services using an improved hyper-heuristic algorithm, *Exp. Syst. Appl.*, **269** (2025), 126419. https://doi.org/10.1016/j.eswa.2025.126419

AIMS Press