



OPEN Optimized K-means algorithm for image segmentation based on improved dung beetle algorithm

Ning Li¹, Yan Luo¹✉, Zhiqiang Feng¹, Hu Qu² & Zixuan Liu³

To improve the quality and computational efficiency of image segmentation, and to overcome the limitations of the traditional K-means algorithm—such as sensitivity to initial cluster centers and susceptibility to local optima—this study proposes an Improved Dung Beetle Optimization (IDBO) algorithm and its application to K-means-based image segmentation. First, Latin Hypercube Sampling (LHS) is employed to initialize the population, enhancing diversity and uniformity in the search space and preventing premature convergence in early iterations. Second, a hybrid position updating strategy, combining a nonlinear decision factor with a competition mechanism, dynamically balances global exploration and local exploitation, improving adaptability across different search stages. Third, the Cauchy inverse cumulative distribution operator and tangent flight operator are integrated to perform dynamic perturbation and fine-tuning on optimal individuals, strengthening local exploitation and enhancing the ability to escape local optima. Comprehensive experiments on standard benchmark functions demonstrate that IDBO outperforms the original DBO and other comparative algorithms in convergence speed, optimization accuracy, and stability. The algorithm is further applied to optimize K-means clustering for image segmentation. Quantitative metrics, including Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR), confirm that IDBO-based segmentation achieves higher accuracy, better edge preservation, and improved texture fidelity. Additionally, an ablation study isolates the contributions of each enhancement strategy, demonstrating their complementary effects and validating the superiority of the integrated IDBO framework. These results highlight the potential of combining intelligent optimization and clustering algorithms to develop adaptive, high-performance image segmentation techniques.

Keywords Dung beetle optimization algorithm, K-Means, Latin hypercube sampling, Nonlinear decision factor, Competitive mechanism, Corsi inverse cumulative distribution, Image segmentation

Image segmentation is one of the core research areas in the field of computer vision and image processing^{1,2}. Its goal is to partition an image into several non-overlapping and internally homogeneous regions based on the similarity of pixel characteristics such as grayscale, spatial texture, and geometric shape^{3,4}. Within each region, pixels share similar attributes, while differences between regions are maximized. High-quality image segmentation not only enhances the accuracy of subsequent tasks such as image recognition and analysis but also has extensive applications in medical image analysis, object detection, intelligent transportation, and remote sensing image processing^{5,6}.

Among the many image segmentation methods, the K-means clustering algorithm^{7,8} is widely used due to its simplicity, computational efficiency, and ease of implementation. The algorithm segments an image by minimizing the sum of squared distances between pixels and their respective cluster centers⁹. However, traditional K-means has notable limitations¹⁰: its performance is highly sensitive to the selection of initial cluster centers, and different initializations may yield substantially different clustering outcomes. As a result, K-means is prone to converging to local optima, which can negatively impact segmentation accuracy and stability.

To overcome these drawbacks, researchers have integrated K-means with intelligent optimization algorithms to enhance the selection of initial cluster centers and improve global search capabilities¹¹. For example, Das et al.¹² combined the Levy–Cauchy Arithmetic Optimization Algorithm (LCAOA) with K-means to improve

¹Guangxi Technological College of Machinery and Electricity, Nanning 530007, Guangxi, China. ²College of Safety Engineering, China University of Mining and Technology, Xuzhou 221116, Jiangsu, China. ³Guangxi Vocational & Technical College of Manufacturing and Engineering Nanning, Nanning 530105, Guangxi, China. ✉email: 17304264947@163.com

initialization and segmentation accuracy; however, its performance remained limited when processing noisy images in different color spaces. Song et al.¹³ introduced Gaussian functions and adaptive scale operators to improve algorithm robustness, achieving superior performance on noisy images, though results were less satisfactory on low-contrast or simple-texture images. Liang et al.¹⁴ pointed out that clustering-based segmentation methods still depend heavily on initial cluster centers and are unsuitable for images with simple backgrounds or minimal color variation.

To further improve global optimization performance, Xue et al.¹⁵ proposed the Dung Beetle Optimization (DBO) algorithm, which mimics the dung beetle's behaviors of foraging, rolling, and nesting to achieve high search efficiency and fast convergence. Nevertheless, the original DBO still suffers from premature convergence and limited optimization accuracy. To address these issues, several researchers have proposed various improvements. For instance, some studies introduced a nonlinear weighted golden sine strategy into the rolling behavior to improve convergence speed and precision¹⁶, while others integrated the follower position updating mechanism from the Sparrow Search Algorithm into DBO, enhancing global optimization performance and successfully applying the improved algorithm to automotive collision simulations¹⁷. Mehmood et al.¹⁸ enhanced the Aquila Optimizer by incorporating multiple chaotic maps into its exploitation phase, demonstrating superior performance on benchmark functions and effective parameter estimation for electro-hydraulic systems, while highlighting improved robustness under noise and potential sensitivity to parameter tuning. Mehmood et al.¹⁸ proposed multiple chaotic variants of the Young's double slit experiment (YDSE) optimizer by integrating ten chaotic maps through different mechanisms, demonstrating that the Gauss-map-based variant significantly outperforms several state-of-the-art optimizers on benchmark functions and in modeling electrically stimulated muscle systems.

In summary, existing improvements have enhanced DBO's stability and applicability to a certain extent; however, challenges remain in solving high-dimensional and complex optimization problems, where convergence accuracy and speed are still insufficient, and the risk of local optima persists. To address these limitations, this paper proposes an Improved Dung Beetle Optimization (IDBO) algorithm, aiming to further strengthen both global exploration and local exploitation capabilities. The proposed IDBO is then applied to optimize parameters of the K-means algorithm for image segmentation. Comprehensive simulation experiments on 12 benchmark test functions, the Wilcoxon Rank-Sum Test, and the CEC2021 benchmark suite verify the proposed algorithm's superior performance in terms of convergence speed, optimization accuracy, and robustness. Furthermore, the IDBO-optimized K-means algorithm is applied to various image segmentation tasks, demonstrating its effectiveness and superiority in improving segmentation accuracy and stability. This research provides a new approach for enhancing the performance of traditional clustering algorithms and establishes a theoretical and practical foundation for applying intelligent optimization techniques in image segmentation and other computer vision applications.

Theoretical foundation

Principle of K-means algorithm

K-means algorithm for image segmentation, firstly, the target image is abstracted into a set of data sample points with d -dimensional vectors²⁰, which can be denoted as $M = (M_1, M_2, \dots, M_d)$, and the initial clustering center is selected from the set of data sample points K , i.e., $C = (C_1, C_2, \dots, C_K)$, and secondly, $M_i (i = 1, 2, \dots, d)$ is assigned to the K clusters according to the Euclidean minimum distance, and the objective function is denoted as²¹:

$$D = \sum_{i=1}^K \sum_{M \in C_j} \|M_j - C_i\|^2 \quad (1)$$

The smaller D is, the higher the similarity of the data within the cluster and the better the clustering. k-means is expressed as²²:

$$C_i = \frac{1}{n} \sum_{j=1}^n M_j \quad (2)$$

where n_i is the number of data points in the i -th cluster. Through continuous iteration, when the center of the K clusters no longer have any change or not much change or meet the iteration conditions, the clustering results can be obtained.

Dung beetle optimization algorithm

Ball rolling and dancing behavior

The Dung Beetle Optimization (DBO) algorithm was inspired by the natural behaviors of dung beetles, including ball-rolling, breeding, foraging, and stealing, and was designed with four distinct update rules to guide the search for the global optimum²³.

During the ball-rolling process, dung beetles rely on celestial cues such as the sun's position or wind direction to maintain a straight-line trajectory while rolling their dung balls. To simulate this rolling behavior, dung beetles in the algorithm move across the entire search space along a specific direction. In this process, the position of the rolling beetle is continuously updated, and the mathematical model describing the rolling behavior can be expressed as²⁴:

$$\begin{aligned}x_i(t+1) &= x_i(t) + \alpha \times k \times x_i(t-1) + b \times \Delta x \\ \Delta x &= |x_i(t) - X^w|\end{aligned}\quad (3)$$

where t denotes the current iteration number, $x_i(t)$ denotes the position information of the i -th dung beetle at the t -th iteration, $k \in (0, 0.2]$ denotes the constant of the deflection coefficient, b denotes the constant belonging to $(0, 1)$, α is a natural coefficient, and the input probability value λ , if $\lambda < \text{rand}(1)$ then α is 1, and vice versa α is -1. X^w denotes the global worst position, which Δx is used to simulate the variation of light intensity.

When a dung beetle encounters an obstacle that prevents it from moving forward, it needs to reorient itself by dancing to a new route. The tangent function is introduced to simulate the dung beetle determining a new rolling direction by dancing. The dung beetle dancing behavior position is updated as follows:

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

where $\theta \in [0, \pi]$, denotes the coefficient of deviation, when $\theta \in [0, \pi/2, \pi]$, $\tan \theta$ is meaningless, the position of the dung beetle does not change.

Reproductive behavior

In nature, dung balls are rolled to a safe place and hidden by dung beetles. To provide a safe environment for their offspring, choosing a suitable spawning site is very important for dung beetles. Inspired by this, a boundary selection strategy is proposed to model the area where female dung beetles lay their eggs, which is defined as²⁵:

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

where X^* denotes the current local optimal position, Lb^* and Ub^* denotes the lower and upper bounds of the spawning region, where $R = 1 - t/T_{\max}$, T_{\max} denotes the maximum number of iterations, and Lb and Ub denotes the lower and upper bounds of the search space, respectively.

The boundary range of the spawning area is dynamically adjusted with the number of iterations, therefore, the breeding sphere position is dynamically changed during the iteration process as shown in Eq. (6).

$$B_i(t+1) = X^* + b_1 \times (B_i(t) - Lb^*) + b_2 \times (B_i(t) - Ub^*) \quad (6)$$

where: $B_i(t)$ is the position information of the i -th breeding ball at the a -th iteration, and b_1 and b_2 denote two independent random vectors of size $1 \times D$.

Foraging behavior

The foraging process of dung beetles in nature was simulated and the optimal foraging area was established to guide the dung beetles to forage. The boundary of the optimal foraging area is defined as²⁶:

$$\begin{aligned}Lb^b &= \max(X^b \times (1 - R), Lb) \\ Ub^b &= \max(X^b \times (1 + R), Ub)\end{aligned}\quad (7)$$

where X^b is the global best position, Lb^b and Ub^b denote the lower and upper bounds of the best foraging area. Therefore, the position of the small dung beetle is updated as follows:

$$x_i(t+1) = x_i(t) + C_1(x_i(t) - Lb^b) + C_2(x_i(t) - Ub^b) \quad (8)$$

where $x_i(t)$ denotes the position information of the i -th small dung beetle in the t -th iteration, C_1 denotes a random number obeying normal distribution, and C_2 denotes a random vector belonging to.

Theft

Some dung beetles will steal dung balls from other dung beetles, and the location update can be described as:

$$x_i(t+1) = X^b + S \times g \times (|x_i(t) - X^*| + |x_i(t) - X^b|) \quad (9)$$

where $x_i(t)$ denotes the position information of the i -th stealing dung beetle at the t -th iteration, g is a random vector with normal distribution of size $1 \times D$, and denotes a constant.

Improving the Dung beetle optimization algorithm

Latin hypercube sampling

Population initialization directly affects the algorithm's optimization accuracy and convergence speed. The pseudo-random function used by the standard dung beetle optimization algorithm to initialize the population has some limitations in practice, especially when the variable space is large and complex, it may not be able to generate a sufficiently uniform initial population globally any more, causing the algorithm to over-concentrate in some regions or ignore other potentially favorable regions, which affects the overall optimization efficiency.

Latin Hypercube Sampling (LHS) is a stratified sampling method whose main feature is the high degree of homogenization of the extracted samples²⁷, which is more effective than the random sampling method, and

therefore it is widely used in the initialization population problem of intelligent algorithms. Assuming that N samples are drawn from a D -dimensional vector space, the steps of Latin hypercube sampling are as follows²⁸:

- a) Determine the vector space D and the number of samples to be drawn N , and let a hypercube variable have dimension D and variable $x^j \in [lb, ub], j = 1, 2, \dots, D$.
- b) Divide the domain of definition $x^j [lb, ub] Nlb = x_1^j < x_2^j < \dots < x_i^j < \dots < x_N^j = ub$.

of the variable into homogeneous interval, i.e.,
So the original hypercube is divided into N^D small hypercube.

- c) Produces $N \times D$ matrix A of A which each column is a fully permuted combination of N equalized intervals.
- d) Each row of matrix A corresponds to a selected small hypercube, and N samples can be drawn by generating a random sample within each selected small hypercube²⁹.

Figure 1 presents the comparative results used to verify the effectiveness of the Latin Hypercube Sampling (LHS) initialization strategy. The LHS initialization method is compared with random initialization³⁰, Tent chaotic mapping initialization³¹, and PWLCM chaotic mapping initialization³². The comparison results clearly demonstrate that the LHS-based initialization generates a more uniform population distribution across the search space than the other methods. The accompanying histogram provides an intuitive visualization of the differences among the initialization strategies, and the statistical results further confirm that LHS initialization achieves a well-balanced and evenly distributed sampling within the search space.

Hybrid location update strategy

Nonlinear decision factor Whether or not the algorithm can reasonably balance the algorithm’s global search and local exploitation ability will directly affect the algorithm’s ability to find the best³³. In the DBO algorithm, the dung beetle generates an initial solution randomly in the solution space, which guides the dung beetle’s position update. Rolling dung beetles can help the population to converge to the foraging area in the early stage and accelerate the convergence of the algorithm. The position update is as follows:

$$\begin{cases} x_i(t + 1) = x_i(t) + \alpha \times k \times x_i(t - 1) + b \times \Delta x, \text{rand} < R \\ x_i(t + 1) = x_i(t) + \tan(\theta) |x_i(t) - x_i(t - 1)|, \text{otherwise} \end{cases} \tag{10}$$

The constant R is a decision factor that controls the algorithm’s global search capability and local exploitation capability. The constant decision factor is not suitable for solving complex and high-dimensional real engineering problems. Therefore, in this paper, a nonlinear decision factor is used so that the algorithm decreases slowly in the early stage to increase the ball-rolling dung beetle search probability and help the algorithm converge quickly. In the late stage of the algorithm, it decreases rapidly to increase the probability of dung beetle searching to help the algorithm jump out of the local optimum. The nonlinear decision factor is formulated as follows³⁴:

$$R_v = R_{v0} \times \left[1 - \left(\frac{e^{L/Iter} - 1}{e - 1} \right)^k \right] \tag{11}$$

where R_{v0} is the initial value of the decision factor, which is taken as 1 in this study, k is the model control factor, the value of k controls the decay speed of the model, and the larger the value of k , the slower the decay speed of the model is. After many experiments to verify, when k take 10, the algorithm works best.

Competition mechanism The Empire Competition Algorithm was proposed in 2007, which contains important mechanisms such as assimilation, revolution, and flipping³⁵. Inspired by the Empire Competition Algorithm, a competition mechanism is introduced between the ball-rolling dung beetles and the stealers. At each iteration, the stealers will randomly select a ball-rolling dung beetle and move one distance in the direction of its position to execute “preparation for stealing”, which is similar to the “assimilation” mechanism in the Competition Algorithm. Similar to the “assimilation” mechanism in the competitive algorithm, the stealer has the probability to be attracted by other dung beetles when moving, resulting in a direction jump, which is updated in the following way:

$$(T_i^\tau)_{new} = T_i^\tau + p \times q \times \frac{T_i^\tau - R_{Rand}^\tau}{\|T_i^\tau - R_{Rand}^\tau\|} \tag{12}$$

where R_{Rand}^τ is the position of the τ -th iteration of the ball-rolling dung beetle of a randomly selected one, and p is a random number between 0 and $2 \times \|T_i^\tau - R_{Rand}^\tau\|$; q is the directional hopping coefficient, and the expression is:

$$q = \begin{cases} \tan \varphi, 0 < \xi \leq 0.3 \\ 0, 0.3 < \xi < 1 \end{cases} \tag{13}$$

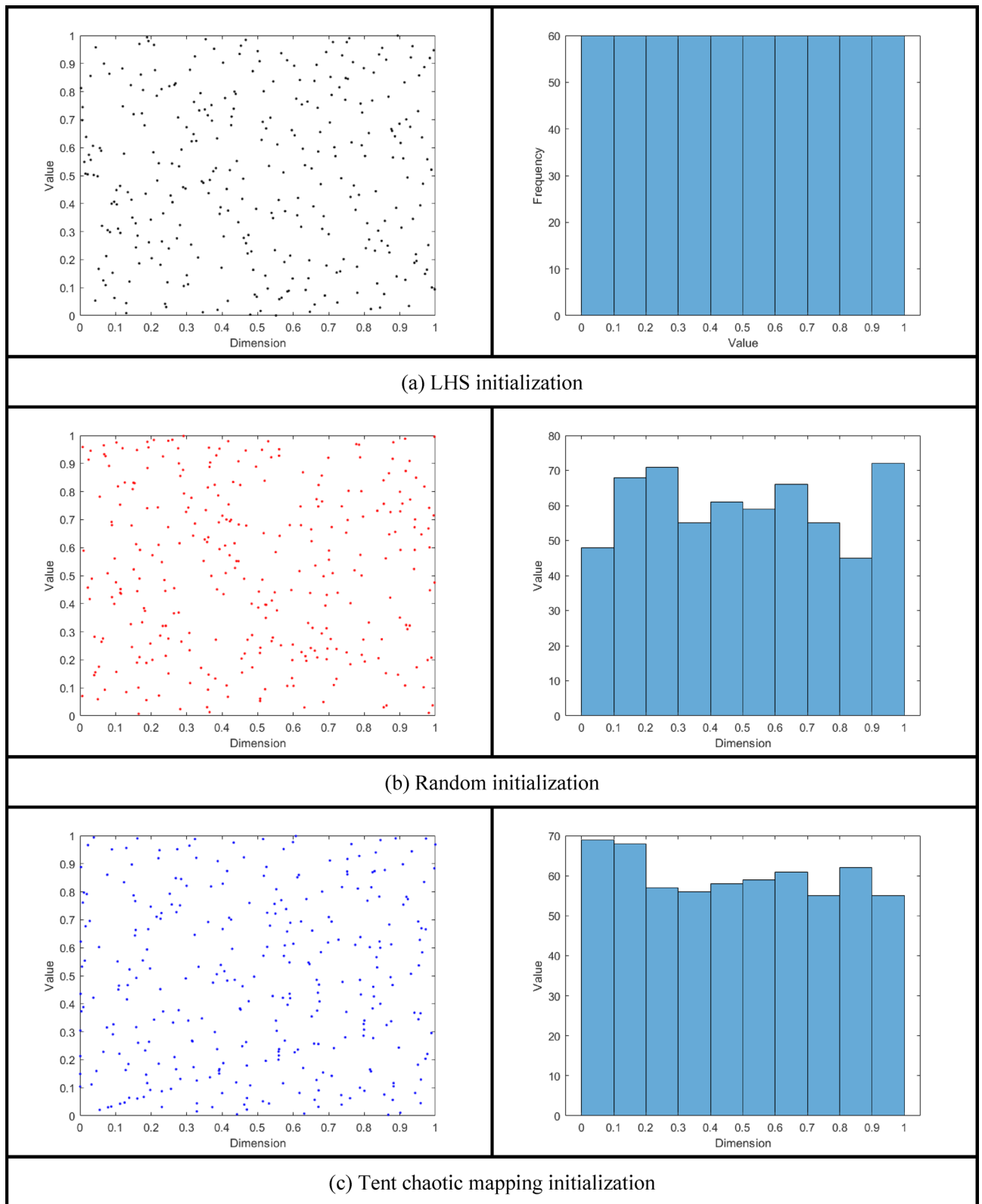


Fig. 1. Distribution of initialization. (a) LHS initialization, (b) Random initialization, (c) Tent chaotic mapping initialization, (d) PWLCM chaotic mapping initialization.

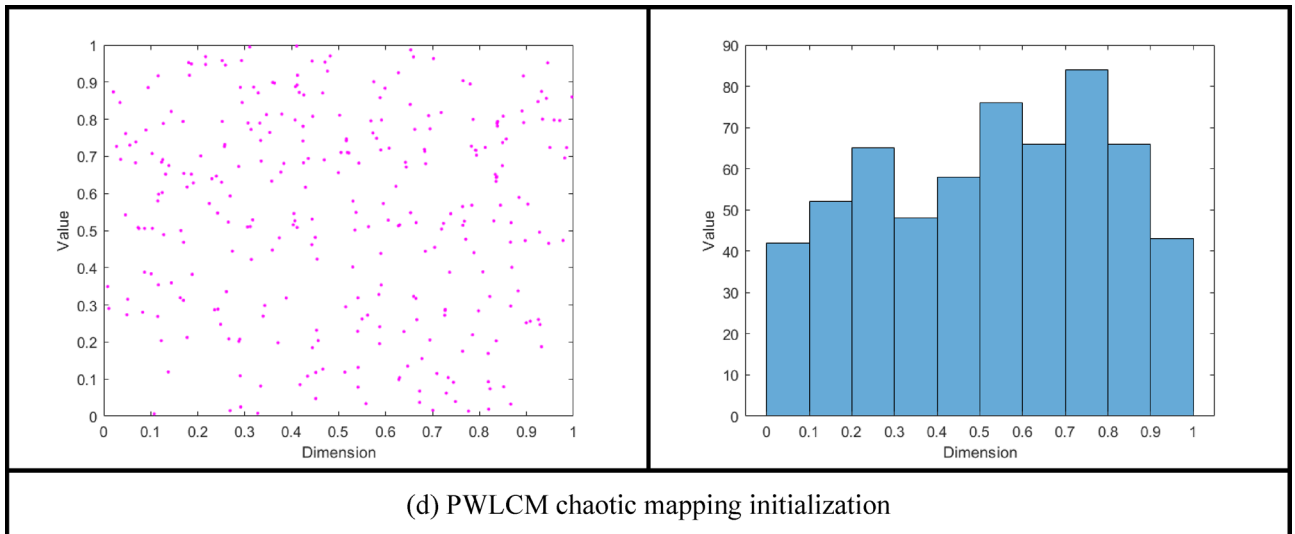


Fig. 1. (continued)

where φ is a random number between $-\frac{\pi}{4}$ and $\frac{\pi}{4}$; ξ is a random number between 0 and 1.

When the stealer succeeds in stealing the dung beetle’s dung ball, the identities of the stealer and the dung beetle are swapped, similar to the “inversion” mechanism in competitive algorithms, according to the following rules:

$$\begin{cases} \{switch((T_i^r)_{new}, R_{rand}^r), \text{ if } f((T_i^r)_{new}) < f(R_{rand}^r) \\ empty, \text{ else} \end{cases} \quad (14)$$

where *switch* indicates that a position swap operation is performed; *empty* indicates that no operation is performed.

By introducing the competition mechanism, the position updating strategy of the stealer, i.e., utilizing the position information of the ball-rolling dung beetle, strengthens the information interaction, improves the homogenization phenomenon among individuals in the process of convergence, and further optimizes the population structure, which not only accelerates the algorithm’s convergence speed, but also enhances the algorithm’s global search capability and avoids falling into the local optimum.

Cauchy inverse cumulative distribution operator combined with tangent flight operator

The selection of the Cauchy inverse cumulative distribution operator and the tangent flight operator is based on the inherent limitations of the original DBO’s local exploitation phase. For the foraging behavior of young dung beetles, the need to balance “precision approaching the optimal solution” and “avoiding local stagnation” requires an operator with adaptive step-size adjustment capability. The Cauchy inverse cumulative distribution operator is selected for its heavy-tailed distribution characteristic, which enables dynamic step-size reduction during the approach to the optimal position, ensuring both convergence efficiency and exploration breadth. Meanwhile, the tangent flight operator is chosen due to its strong random direction perturbation ability, which can break the spatial clustering of individuals in the late iteration stage. This combination of operators is designed to address the trade-off between local exploitation accuracy and global exploration capability, a key challenge in metaheuristic algorithms for complex optimization problems.

When the small dung beetle searches toward the global optimum, its position update is more random, which makes the algorithm’s convergence speed and convergence accuracy decrease, resulting in the inability to obtain the global optimum solution. To further improve the ability of the algorithm to jump out of the local optimum, the Cauchy inverse cumulative distribution operator is introduced into the dung beetle’s foraging behavior. The position is updated as:

$$x_i(t + 1) = x_i(t) + (X^b - x_i(t)) \left(0.01 \tan \left(\pi \left(p - \frac{1}{2} \right) \right) \right) \quad (15)$$

The foraging behavior primarily guides the young dung beetles to move toward the global optimal position. However, since the optimal position contains limited food resources, dung beetles located farther away cannot achieve rapid foraging. To address this issue, the Cauchy inverse cumulative distribution operator is introduced. This operator enables the dung beetles to gradually reduce the distance between themselves and the food source, thereby shrinking the overall step size and allowing the population to reach the optimal fitness value more efficiently.

In the later stages of iteration, dung beetle individuals tend to rapidly assimilate and cluster near the optimal position, which may cause the population to become trapped in local optima. To overcome this, the tangential

flight operator is adopted as a proportional factor to control the step size. This operator helps balance the exploration and exploitation processes, enhances the convergence performance of the DBO algorithm, and effectively avoids accuracy degradation.

The local random walk is performed based on the product of the tangential flight operator and the distance between dung beetles and the food source, which proves highly effective for spatial exploration and prevents the algorithm from falling into local optima²³. The corresponding position update formula is expressed as:

$$H^b(t) = X^b(t) * \tan\left(v \frac{\pi}{2}\right) \tag{16}$$

where $X^b(t)$ denotes the optimal position of the individual in the t -th iteration, $H^b(t)$ denotes the position after a tangent flight perturbation to the optimal position $X^b(t)$, v is a random number uniformly distributed within $[0, 1]$, and d is the dimension of the function²⁴.

Figure 2 shows the IDBO algorithm flowchart.

IDBO optimized K-means for image segmentation

In performing image segmentation, the target image is first treated as a d -dimensional space vector with a data sample point set of M , and m data points are randomly selected from it as the initial clustering center. Then the remaining data points in the set M are assigned to the class m . Let m_i be the i -th data point in set M , C_j is the j -th clustering center, and assuming the minimum when $\|M_i - C_j\|$, assign data point M_i to class j . The fitness function of IDBO is the minimum loss function of K-means algorithm, denoted as:

$$f_i = \sum_{i=1}^d \sum_{j=1}^m \|M_i - C_j\| \tag{17}$$

Specifically, IDBO optimized K-means for image segmentation is performed as follows:

- Step1: Set the initial parameters of IDBO algorithm; take the minimum loss function as the fitness function of the algorithm and calculate the fitness;
- Step2: Generate the clustering center;
- Step3: Cluster the pixels according to the Euclidean clustering;
- Step4: Take the pixel mean of each category as the new clustering center;
- Step5: Determine whether the convergence condition is satisfied, i.e., whether the clustering center is changed or not; if satisfied, output the result; otherwise return to step2.

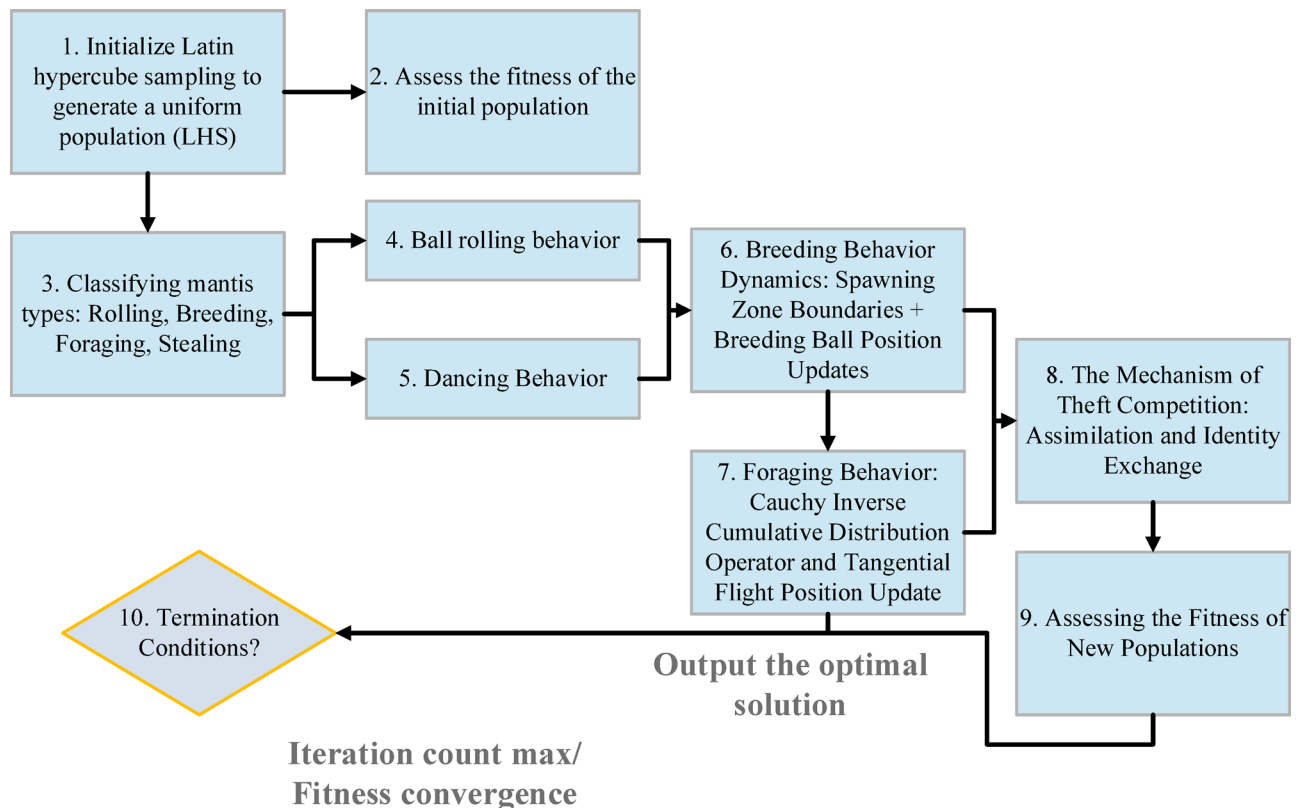


Fig. 2. IDBO Algorithm Flowchart.

When IDBO reaches the maximum number of iterations, the optimization search results are used to initialize the clustering center of the next process K-means algorithm, so as to overcome the defects of K-means algorithm affected by the initialization of the clustering center, and further improve the performance of K-means algorithm.

IDBO Algorithm Flow:

Inputs: population size N , maximum number of iterations T_{\max} , and other parameters;

Output: optimal solution $X^b(t)$, optimal fitness value $f(X^b(t))$.

- 1) Randomly initialize the population and calculate its fitness value
 - 2) While $t \leq T$
 - 3) If $\delta < ST$, update the position of the ball dung beetle according to Eq. (3), and vice versa, update the position of the ball dung beetle according to Eqs. (12)(14)
 - 4) Update the positions of the breeding ball, the small dung beetle, and the stealing dung beetle according to Equations (6), (8), and (9)
 - 5) Perform boundary determination for each dung beetle, calculate the stress value for each dung beetle, and find the optimal solution
 - 6) Update the current optimal solution according to Equations (15) and (16) to produce a new solution
 - 7) end
 - 8) $t = t + 1$
 - 9) end while
 - 10) return $X^b(t)$ 与 $f(X^b(t))$
-

Time complexity

Time complexity reflects the change in the execution time of an algorithm as the data size grows, and is usually measured as the growth rate as a function of the input size. Assume that the algorithm population size N , the number of iterations T , and the dimension D . Let the time complexity of the population initialization phase be $O(N)$, the time complexity of updating the population position be $O(TND)$, and the time complexity of the local search phase be $O(TND)$. The time complexity of the standard DBO algorithm is $O(N) + O(TN) + O(TND) = O(TND)$.

This study carries out an improvement of DBO by introducing Latin hypercubic sampling initialization has not improved the updating formula in the global search phase, at which point the complexity is still $O(N)$. Introducing a hybrid position update strategy in the global search phase³⁶ only changes the allocation of populations and the way the algorithm is executed without changing the algorithm itself the complexity is still $O(TN)$. The introduction of the Kersey inverse cumulative distribution operator that fuses tangent flights to execute the greedy principle increases the $O(T)$ computation changes under the same order of magnitude³⁷, and the complexity of the rest of the link is the same as the DBO algorithm, so the time complexity of this stage is $O(TND)$. So the total time complexity of IDBO is $O(N) + O(TN) + O(TND) = O(TND)$.

In summary, the total time complexity of image segmentation for IDBO optimized K-means is still $O(N) + O(TN) + O(TND) = O(TND)$.

Numerical experiments

Model parameter settings and baseline functions

To evaluate the performance of the proposed Improved Dung Beetle Optimization (IDBO) algorithm, a series of five comparative simulation experiments were conducted against several well-established optimization algorithms, including the original DBO, GWO, BWO, PSO, MODBO, and MSDBO. The experimental design is as follows:

Experiment 1: Analyze the effectiveness of the three improvement strategies integrated into the IDBO algorithm.

Experiment 2: Compare the optimization performance of IDBO with six other intelligent optimization algorithms under different dimensions.

Experiment 3: Perform the Wilcoxon rank-sum test to evaluate the statistical significance of the performance differences between IDBO and the other six swarm intelligence algorithms.

Experiment 4: Utilize CEC2021 benchmark functions to further verify the stability and robustness of the IDBO algorithm when handling complex optimization tasks.

Experiment 5: Compare the performance and effectiveness of IDBO and other algorithms when applied to K-means-based image segmentation optimization.

All experiments were conducted under a 64-bit Windows 10 operating system with an Intel(R) Core(TM) i5-7200U CPU at 2.70 GHz. To validate the performance improvement of DBO achieved by the proposed strategies, 12 benchmark test functions were selected³⁸. Among them, F1–F7 are unimodal functions used to test the local exploitation capability of the algorithms, while F8–F12 are multimodal functions used to evaluate their global exploration ability.

To ensure fairness and consistency in performance evaluation, all algorithms were configured with a population size of 30, a maximum of 500 iterations, and dimensional settings of 30, 100, and 500^{39,40}. Each experiment was independently executed 30 times, and the results were evaluated using three indicators: the mean (mean), standard deviation (std), and best value (best)¹⁴. Specifically, the best and mean values indicate the optimization accuracy and capability, while the standard deviation (std) reflects the stability of the algorithm⁴¹.

In addition to the original DBO, three widely used optimization algorithms and two recently developed algorithms were selected for comparison. The parameters of each algorithm were configured according to their respective references, as summarized in Table 1.

Performance effectiveness analysis of progressive strategies

To verify the effectiveness of the proposed improvement strategies in enhancing the performance of the Dung Beetle Optimization (DBO) algorithm, three variant algorithms were designed for comparison:

IDBO1: DBO improved only with the Latin Hypercube Sampling (LHS) strategy.

IDBO2: DBO improved only with the hybrid update strategy combining nonlinear decision factors and a competitive mechanism.

IDBO3: DBO improved only with the Cauchy inverse cumulative distribution operator and the tangent flight operator.

The experimental parameter settings were consistent with those described in Sect. 3.1.

As shown in Table 2, both IDBO-3 and the fully improved IDBO algorithm achieved theoretical optimal values for the best, mean, and standard deviation metrics on functions F1–F3, indicating that the integration of the Cauchy inverse cumulative distribution and tangent flight operators effectively enhances the optimization performance of the algorithm. For function F6, IDBO1 achieved the best and mean values among all variants, though its standard deviation was not optimal, suggesting a limitation in stability for this particular strategy. Overall, IDBO3 outperformed the original DBO algorithm, demonstrating that incorporating the Cauchy and tangent flight mechanisms helps the algorithm escape local optima, effectively balancing global exploration and local exploitation, and improving robustness⁴².

Furthermore, IDBO1 and IDBO2 achieved significantly better performance than the original DBO algorithm across all three evaluation metrics on functions F1–F4, confirming that both the LHS initialization and hybrid update strategy substantially enhance the optimization accuracy and search efficiency of the algorithm^{43,44}.

In summary, the comprehensive comparison across 12 benchmark functions shows that the fully improved IDBO algorithm, which integrates all proposed strategies, consistently outperforms the other four variants, thereby fully validating the effectiveness and superiority of the proposed improvement mechanisms.

Figure 3 shows the convergence curves of the five algorithms. It can be observed that IDBO1 exhibits a convergence trend similar to the original DBO; however, its convergence accuracy is improved, and the average optimization performance is more stable. Compared with DBO, the IDBO2 algorithm demonstrates further improvement in both average convergence accuracy and standard deviation, indicating that it effectively

algorithms	parameters	parameter value
IDBO	R	0.5
	k	10
	b	0.3
	p, φ, ξ, v	random number
PSO	r_1, r_2	random number
	c_1, c_2	2
	ω	$\omega_{\max} = 0.9, \omega_{\min} = 0.2$
DBO	b	0.3
	α	-1 or 1
	R	Decrease linearly from 1
GWO	r_1, r_2	random number
	a	Decrease linearly from 2
BWO	r_1, r_2, r_3, r_4	random number
MODBO	r_1, r_2	random number
	ω	$\omega_{\max} = 1.25, \omega_{\min} = 0.25$
MSDBO	ST	0.7
	β_1	Gradually decreasing from 1 to 0
	β_2	Gradual increase from 0 to 1

Table 1. Parameter variable settings.

Fun No.	Algorithm	Best	Mean	Std	Fun No.	Algorithm	Best	Mean	Std
F1	DBO	5.57E-161	2.39E-117	1.23E-116	F7	DBO	9.11E-05	1.79E-03	2.23E-03
	IDBO-1	1.37E-154	1.37E-103	7.50E-103		IDBO-1	2.00E-05	1.83E-04	1.44E-04
	IDBO-2	3.74E-153	8.45E-107	4.63E-106		IDBO-2	6.06E-05	1.47E-03	9.16E-04
	IDBO-3	0.00E+00	0.00E+00	0.00E+00		IDBO-3	6.85E-06	1.72E-04	1.96E-04
	IDBO	0.00E+00	0.00E+00	0.00E+00		IDBO	2.80E-06	1.53E-04	1.43E-04
F2	DBO	9.90E-91	3.72E-59	1.58E-58	F8	DBO	-1.25E+04	-8.49E+03	1.80E+03
	IDBO-1	4.89E-83	2.87E-60	1.30E-59		IDBO-1	-1.25E+04	-8.63E+03	2.07E+03
	IDBO-2	4.12E-85	4.03E-59	2.20E-58		IDBO-2	-1.22E+04	-9.33E+03	1.81E+03
	IDBO-3	0.00E+00	0.00E+00	0.00E+00		IDBO-3	-1.26E+04	-1.06E+04	1.87E+03
	IDBO	0.00E+00	0.00E+00	0.00E+00		IDBO	-1.36E+04	-1.09E+04	1.85E+03
F3	DBO	3.26E-143	1.08E-46	5.90E-46	F9	DBO	0.00E+00	2.09E+00	9.17E+00
	IDBO-1	7.87E-145	5.78E-75	3.16E-74		IDBO-1	0.00E+00	1.18E+00	3.95E+00
	IDBO-2	2.41E-131	1.32E-52	7.24E-52		IDBO-2	0.00E+00	4.19E+00	2.25E+01
	IDBO-3	0.00E+00	0.00E+00	0.00E+00		IDBO-3	0.00E+00	0.00E+00	0.00E+00
	IDBO	0.00E+00	0.00E+00	0.00E+00		IDBO	0.00E+00	0.00E+00	0.00E+00
F4	DBO	3.54E-84	1.81E-52	9.80E-52	F10	DBO	0.00E+00	2.09E+00	9.17E+00
	IDBO-1	9.56E-84	3.30E-55	1.49E-54		IDBO-1	0.00E+00	1.18E+00	3.95E+00
	IDBO-2	9.79E-79	4.12E-52	2.26E-51		IDBO-2	0.00E+00	4.19E+00	2.25E+01
	IDBO-3	0.00E+00	0.00E+00	0.00E+00		IDBO-3	0.00E+00	0.00E+00	0.00E+00
	IDBO	0.00E+00	0.00E+00	0.00E+00		IDBO	0.00E+00	0.00E+00	0.00E+00
F5	DBO	2.55E+01	2.58E+01	2.71E-01	F11	DBO	0.00E+00	2.09E+00	9.17E+00
	IDBO-1	2.54E+01	2.57E+01	1.69E-01		IDBO-1	0.00E+00	1.18E+00	3.95E+00
	IDBO-2	2.51E+01	2.57E+01	2.29E-01		IDBO-2	0.00E+00	4.19E+00	2.25E+01
	IDBO-3	2.37E+01	2.44E+01	6.30E-01		IDBO-3	0.00E+00	0.00E+00	0.00E+00
	IDBO	2.36E+01	2.46E+01	5.83E-01		IDBO	0.00E+00	0.00E+00	0.00E+00
F6	DBO	2.15E-05	1.73E-01	2.52E-01	F12	DBO	8.92E-08	9.12E-04	2.27E-03
	IDBO-1	5.53E-06	9.77E-03	5.16E-02		IDBO-1	1.75E-07	1.14E-04	4.45E-04
	IDBO-2	4.24E-06	8.06E-03	2.89E-02		IDBO-2	1.25E-07	6.68E-04	1.86E-03
	IDBO-3	3.58E-07	2.92E-01	3.75E-01		IDBO-3	3.27E-08	1.30E-03	4.29E-03
	IDBO	9.99E-08	1.03E-03	1.39E-03		IDBO	1.50E-08	3.46E-05	8.35E-05

Table 2. Comparative test results of different improvement strategies.

balances global exploration and local exploitation. For IDBO3, it is evident that the incorporation of the Cauchy inverse cumulative distribution operator and the tangent flight operator significantly enhances the algorithm's ability to escape from local optima⁴⁵, resulting in faster convergence toward the global optimum and improved overall search performance.

Performance comparison experiments with other swarm intelligence algorithms

To further verify the superior performance of the IDBO algorithm, it was compared with GWO, BWO, PSO, DBO, MODBO, and MSDBO on 12 benchmark test functions, with dimensions set to 30, 100, and 500, and other parameters configured as in Sect. 3.1. The comparison results are shown in Table 3.

For functions F1–F7, IDBO achieved the best performance across all evaluation metrics compared to the other algorithms. Under the 30-dimensional condition for F3, IDBO reached the optimal value of 0, with a mean value of 0, outperforming DBO. For functions F8–F12, the convergence accuracy of IDBO was generally able to reach the theoretical optimum of 0.

Under 100-dimensional conditions, all algorithms showed some performance degradation, but IDBO was least affected by the increase in dimensionality, maintaining a significant accuracy advantage. In particular, for F1–F4, F6, and F8, all evaluation metrics achieved the theoretical minimum, with no regression in accuracy. Compared to other swarm intelligence algorithms, the average convergence accuracy of IDBO remained superior, demonstrating its strong optimization capability. Additionally, the standard deviation of IDBO was smaller than that of the other metaheuristic algorithms, indicating better stability and ability to escape local optima. Among the other algorithms, only GWO was able to reach the theoretical value for F8, while their optimization performance on the remaining benchmark functions was relatively poor.

Under the high-dimensional condition of 500 dimensions, IDBO outperformed all other algorithms in all precision metrics. Its mean values were significantly better than those of the other algorithms, verifying IDBO's excellent optimization accuracy, and its smaller std values confirmed the algorithm's robustness.

In summary, based on the mean values, IDBO demonstrates higher solution accuracy and faster convergence speed; based on the standard deviation, it exhibits superior global search ability and excellent optimization

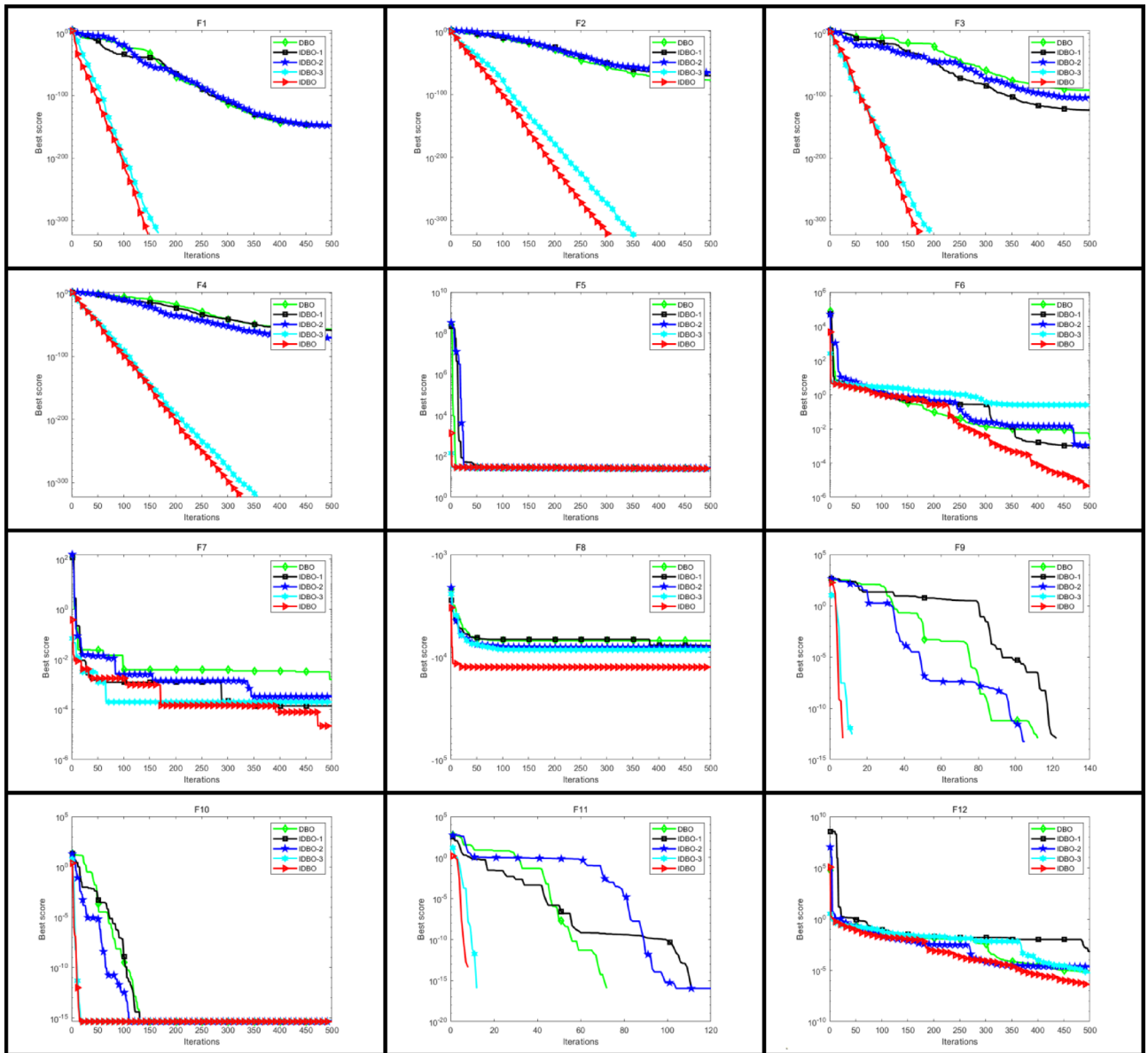


Fig. 3. Convergence curve of IDBO with different improvement strategies.

stability. IDBO can quickly converge to high precision on unimodal functions, while for multimodal functions, it avoids local optima and premature convergence, maintaining reliable global search performance.

Figure 4 shows the convergence curves of the IDBA algorithm with the other six algorithms under 12 benchmark test functions. It is clearly seen that the IDBO algorithm overwhelmingly has faster and higher convergence speed and accuracy under the 12 test functions, further verifying the algorithm’s excellent performance in solving the problem.

Wilcoxon rank sum test

To further compare the IDBO algorithm with other algorithms, Wilcoxon rank sum test was performed^{46,47}. The experiment was run independently 30 times for each algorithm, and the Wilcoxon rank sum test of the run results at the significant level of the value, if, it means that these two algorithms are significantly different, if, it means that these two algorithms are the same on the whole. “+” “=” “-” indicates that the performance of IDBO algorithm is better than, equal to and inferior to other algorithms respectively. NaN indicates that the results of these two algorithms are too similar to be significant.

Table 4 shows that the p-value of the rank sum test for most of the algorithms is less than 0.05, indicating that the difference in the performance of the IDBO algorithm over the other six algorithms is significant.

functions	algorithms	30D			100D			500D		
		mean	std	best	mean	std	best	mean	std	best
	DBO	7.58E-103	4.15E-102	2.53E-158	1.99E-108	1.09E-107	5.42E-153	3.68E-116	1.99E-115	2.18E-148
F1	GWO	1.18E-27	1.67E-27	3.15E-29	1.59E-12	1.15E-12	1.43E-02	1.74E-03	5.65E-04	8.99E-04
	BWO	3.99E-03	1.56E-02	-5.03E-02	1.73E-03	1.43E-02	-6.48E-02	5.80E-03	1.53E-02	-6.79E-02
	PSO	1.59E-01	3.67E+00	-1.27E+01	3.72E-01	6.62E+00	-3.96E+01	7.49E-01	6.13E+00	-9.97E+01
	MODBO	3.36E-132	1.82E-163	1.70E-131	1.30E-131	6.88E-131	1.80E-168	1.12E-111	6.13E-111	1.23E-153
	MSDBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	IDBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F2	DBO	5.06E-62	1.78E-61	6.41E-66	3.63E-51	1.99E-50	1.91E-87	2.62E-59	1.03E-58	1.64E-84
	GWO	3.91E-04	1.96E-03	-6.68E-03	4.13E-08	1.36E-08	2.09E-08	1.09E-02	1.68E-03	7.28E-03
	BWO	1.17E-16	7.17E-17	1.29E-17	2.03E05	1.73E-03	-7.95E-03	5.89E-04	1.69E-03	-7.50E-03
	PSO	9.03E-01	3.23E+00	-1.00E+01	1.45E+00	4.24E+00	-1.00E+01	1.79E+00	3.86E+00	-1.00E+01
	MODBO	5.60E-78	1.24E-65	1.22E-79	1.89E-80	9.12E-65	5.42E-80	8.59E-65	4.04E-64	2.84E-83
	MSDBO	2.36E-273	0.00E+00	1.30E-297	2.26E-269	0.00E+00	1.57E-294	6.56E-261	0.00E+00	2.41E-286
IDBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F3	DBO	2.25E-78	1.23E-77	4.65E-140	6.29E-38	3.45E-37	1.78E-150	3.96E-09	2.17E-08	5.92E-149
	GWO	5.97E-06	7.73E-06	2.29E-09	5.20E+02	4.49E+02	1.29E+01	3.29E+05	9.33E+04	2.02E+05
	BWO	4.64E-03	1.70E-01	-9.25E-01	1.04E-02	3.46E-01	-1.52E+00	4.50E-03	6.58E-01	-4.89E+00
	PSO	-3.13E-03	3.51E+01	-1.00E+02	4.21E-02	3.75E+01	-1.00E+02	-2.01E-02	5.27E+01	-1.00E+02
	MODBO	7.14E-86	3.90E-85	1.16E-139	2.19E-70	1.20E-69	2.46E-144	1.29E-47	7.07E-47	2.54E-137
	MSDBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
IDBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F4	DBO	1.90E-53	1.04E-52	4.29E-78	4.80E-48	2.63E-47	4.20E-84	1.64E-31	9.01E-31	4.36E-79
	GWO	8.54E-07	1.01E-06	9.75E-08	9.67E-07	9.82E-07	3.96E-08	6.59E+01	4.47E+00	5.73E+01
	BWO	2.98E-03	8.40E-03	-1.81E-02	1.04E-03	7.52E-03	-2.88E-02	2.22E-03	1.12E-02	-3.21E-02
	PSO	4.14E-01	5.12E+00	1.67+01	8.24E-02	5.66E+00	-1.61E+01	4.65E-01	1.34E+01	-4.83E+01
	MODBO	1.28E-56	7.03E-56	5.02E-100	3.99E-54	2.19E-53	6.23E-78	1.03E-54	3.15E-54	4.35E-74
	MSDBO	1.00E-264	0.00E+00	2.38E-284	5.79E-256	0.00E+00	1.92E-285	5.31E0255	0.00E+00	6.45E-273
IDBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	
F5	DBO	2.55E+01	2.78E+01	2.35E+01	6.53E+01	7.12E+01	5.89E+01	4.97E+02	2.26E-01	4.97E+02
	GWO	2.80E+01	4.48E+01	2.57E+01	6.26E+01	6.73E+01	5.92E+01	4.98E+02	2.60E-01	4.97E+02
	BWO	1.00E+02	1.80E-03	9.94E-01	1.01E+00	2.10E-03	9.96E-01	1.01E+00	1.70E-03	9.95E-01
	PSO	2.64E+03	2.01E+03	4.19E+02	2.75E+06	1.87E+07	3.28E+03	-1.08E+00	3.67E+00	-7.35E+00
	MODBO	2.84E+01	3.63E-01	2.78E+01	9.88E+01	7.41E-02	9.86E+01	4.98E+02	4.61E-02	4.99E+02
	MSDBO	2.71E+01	9.62E+00	2.50E-03	8.05E+01	8.05E+01	7.47E-01	4.95E+02	1.46E+00	4.91E+02
IDBO	2.50E+01	2.66E+01	2.32E+01	9.53E+01	7.12E+01	6.60E-01	4.85E-01	5.54E+01	1.91E-01	
F6	DBO	1.31E-02	3.17E-02	2.44E-05	5.29E+00	7.33E-01	2.85E+00	8.96E+01	2.82E+00	8.56E+01
	GWO	5.12E-01	4.74E-01	2.66E-01	2.12E+01	6.13E-01	7.32E+00	9.18E+01	1.76E+00	8.81E+01
	BWO	-4.58E-01	1.34E-01	-6.12E-01	-4.62E+00	1.24E-01	-6.96E+00	-6.54E-01	1.44E-01	-7.28E-01
	PSO	7.32E+01	6.13E+01	2.58E+01	1.19E+03	5.30E+02	4.73E+02	6.05E-01	5.33E+00	-1.89E+01
	MODBO	3.42E+00	4.65E-01	2.38E+00	1.71E+01	1.01E+00	1.44E+01	1.05E+02	2.81E+00	9.96E+01
	MSDBO	5.44E-01	8.66E-01	1.37E-06	8.97E+00	6.16E+00	1.80E+00	7.21E+01	1.51E+01	3.85EE+01
IDBO	3.52E-01	6.15E-01	3.21E-08	5.33E+00	3.72E+00	2.86E-01	4.96E-02	1.35E-01	1.63E+01	
F7	DBO	9.10E-04	6.26E-04	1.55E-04	1.44E-03	9.72E-04	2.55E-04	1.51E-03	1.27E-03	1.00E-4
	GWO	1.80E-03	8.50E-04	4.95E-04	7.08E-03	2.90E-03	2.85E-03	4.56E-02	1.34E-02	2.36E-02
	BWO	1.29E-02	1.57E-02	-4.36E-02	8.76E-03	8.55E-03	-2.79E-02	3.50E-03	6.45E-03	-2.18E-02
	PSO	2.99E-02	5.39E-02	-1.28E+00	1.58E-02	4.67E-01	-1.28E+00	5.34E-03	4.12E-01	-1.28E+00
	MODBO	2.07E-04	1.99E-04	5.03E-06	2.35E-04	2.28E-04	1.31E-06	1.51E-04	1.46E-04	5.76E-05
	MSDBO	3.53E-04	3.90E-04	1.86E-05	6.90E-04	4.87E-04	2.19E-05	8.31E-04	5.88E-04	3.68E-05
IDBO	1.50E-04	1.21E-04	1.84E-06	1.64E-04	1.83E-04	8.20E-06	1.87E-04	1.46E-04	2.02E-05	

Continued

functions	algorithms	30D			100D			500D		
		mean	std	best	mean	std	best	mean	std	best
F8	DBO	-7.99E+03	1.31E+03	-1.16E+04	-2.78E+04	6.51E+03	-3.76E+04	-1.35E+05	4.12E+04	-1.99E+05
	GWO	-6.05E+03	6.45E+02	-7.58E+03	-1.69E+04	1.70E+03	-2.25E+04	-5.70E+04	9.05E+03	-6.65E+04
	BWO	4.20E+02	1.24E+00	4.17E+02	4.21E+03	1.42E+00	4.18E+02	4.20E+02	1.73E+00	4.17E+02
	PSO	6.44E+01	4.22E+02	-5.00E+02	1.34E+02	3.77E+02	-5.00E+02	7.32E+01	3.58E+02	-5.00E+02
	MODBO	-8.36E+03	1.62E+03	-1.22E+04	-2.16E+04	3.55E+03	-3.02E+04	-1.01E+05	2.25E+04	-1.82+05
	MSDBO	-1.23E+04	7.14E+02	-1.25E+04	-3.85E+04	5.62E+03	-3.87E+04	-1.57E+05	2.53E+04	-2.09E+05
	IDBO	-1.25E+04	1.28E-02	-1.26E+04	-4.18E+04	3.34E+01	-4.18E+04	-2.05E+05	1.25E+04	-2.31E+05
F9	DBO	1.19E+00	4.90E+00	0.00E+00	2.14E+00	9.32E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	GWO	2.13E+00	2.91E+00	5.68E-14	1.15E+01	1.59E+01	9.63E-11	7.31E+01	2.21E+01	3.60E+01
	BWO	5.44E-05	6.26E-04	-2.34E-03	6.97E-05	5.89E-04	-2.49E-03	-6.38E-05	6.11E-04	-4.42E-03
	PSO	2.30E-01	1.89E+00	-5.12E+00	1.80E-01	1.35E+00	-5.12E+00	1.35E-01	1.43E+00	-5.12E+00
	MODBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	MSDBO	1.99E+00	7.57E+00	0.00E+00	6.68E+00	2.54E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	IDBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F10	DBO	8.88E-16	0.00E+00	8.88E-16	1.01E-15	6.49E-16	8.88E-16	8.88E-16	9.01E-15	8.88E-16
	GWO	9.68E-14	1.62E-14	7.55E-14	1.09E-07	3.65E-08	4.32E-08	1.79E-03	3.80E-04	1.01E-03
	BWO	4.20E-05	4.43E-03	-1.80E-02	-3.67E-04	4.12E-03	-2.35E-02	7.97E-04	3.83E-03	-3.36E-02
	PSO	7.67E-01	4.32E+00	-6.12E+00	8.19E-01	6.12E+00	-1.39+01	3.95E+00	1.27E+01	-3.2E+01
	MODBO	1.01E-15	6.49E-16	8.88E-16	1.84E-15	1.60E-15	8.88E-16	8.88E-16	1.35E-15	8.88E-16
	MSDBO	8.88E-16	0.00E+00	8.88E-16	8.88E-16	0.00E+00	8.88E-16	8.88E-16	0.00E+00	8.88E-16
	IDBO	8.88E-16	0.00E+00	8.88E-16	8.88E-16	0.00E+00	8.88E-16	8.88E-16	0.00E+00	8.88E-16
F11	DBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	GWO	2.06E-03	6.42E-03	0.00E+00	5.03E-03	1.03E-02	2.89E-02	1.77E-02	3.76E-02	9.79E-05
	BWO	7.10E-03	6.46E-02	-2.64E-01	2.23E-02	7.37E-02	-8.87E-01	-6.42E-03	6.83E-02	-4.30E-01
	PSO	9.31E-01	2.31E+00	-9.64E+01	9.50E_01	3.03E+01	-2.45E+02	1.41E+00	3.63E+01	-6.00E+02
	MODBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	MSDBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	IDBO	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
F12	DBO	9.53E-04	6.78E-04	2.51E-07	5.95E-02	1.58E-02	3.09E-02	6.13E-01	5.64E-02	5.36E-01
	GWO	4.02E-02	2.48E-02	1.83E-02	2.92E-01	6.55E-02	2.01E-01	7.59E-01	7.27E-02	6.92E-01
	BWO	-1.01E+00	7.17E-02	-1.13E+00	-9.98E-01	5.05E-02	-1.11E+00	-9.87E-01	5.02E-02	-1.12E+00
	PSO	-0.48E+00	3.8E+00	-5.48E+00	6.23E-01	3.98E+00	-1.12E+01	5.94E-01	5.68E+00	-1.27E+01
	MODBO	3.50E-01	1.15E-01	1.65E-01	6.40E-01	1.01E-01	4.67E-01	8.21E-01	7.10E-02	6.72E-01
	MSDBO	6.16E-02	1.69E-01	9.95E-08	3.12E-01	2.85E-01	5.89E-05	4.09E-01	1.58E-01	1.28E-01
	IDBO	8.87E-04	3.40E-04	4.33E-10	1.18E-02	1.23E-02	1.42E-04	3.73E-01	1.74E-02	2.80E-03

Table 3. Algorithm comparison statistical results.

Experimental analysis of CEC2021 test functions

In order to further verify the feasibility and robustness of IDBO algorithm, it is tested by CEC2021 objective function optimization. Ten functions of the CEC2021 benchmark test function are selected for testing, where dimension, population size, maximum number of iterations. The experimental results are shown in Table 5. From the results in Table 5, it can be seen that the IDBO algorithm shows significant advantages in overall performance. Whether in terms of convergence speed and accuracy on single-peak functions or global optimization ability on multi-peak functions and complex combinatorial functions, IDBO achieves better results than the other six comparative algorithms. Especially in high-dimensional complex optimization problems, IDBO can effectively balance the global exploration and local development processes, and shows strong stability, convergence and robustness. In conclusion, the experimental results fully verify the feasibility and superior performance of IDBO algorithm in dealing with different types of optimization problems^{2,6,15,25}.

IDBO optimized K-means for image segmentation experiments

In order to verify the feasibility and effectiveness of IDBO optimized K-means algorithm for image segmentation, the classical images from the standard test dataset are selected for experimental verification. The parameters are the same as the experiments in Sect. 3.1, the population size is 30, and the maximum number of iterations of the population is 500. In the part of K-means algorithm, the initial number of cluster centers of the segmentation graph is made to be 4, and each image is tested independently for 30 times. The experimental segmentation results use the mean square error (σ_{MSE}) and the peak signal-to-noise ratio (P_{PSNR}) to measure the quality

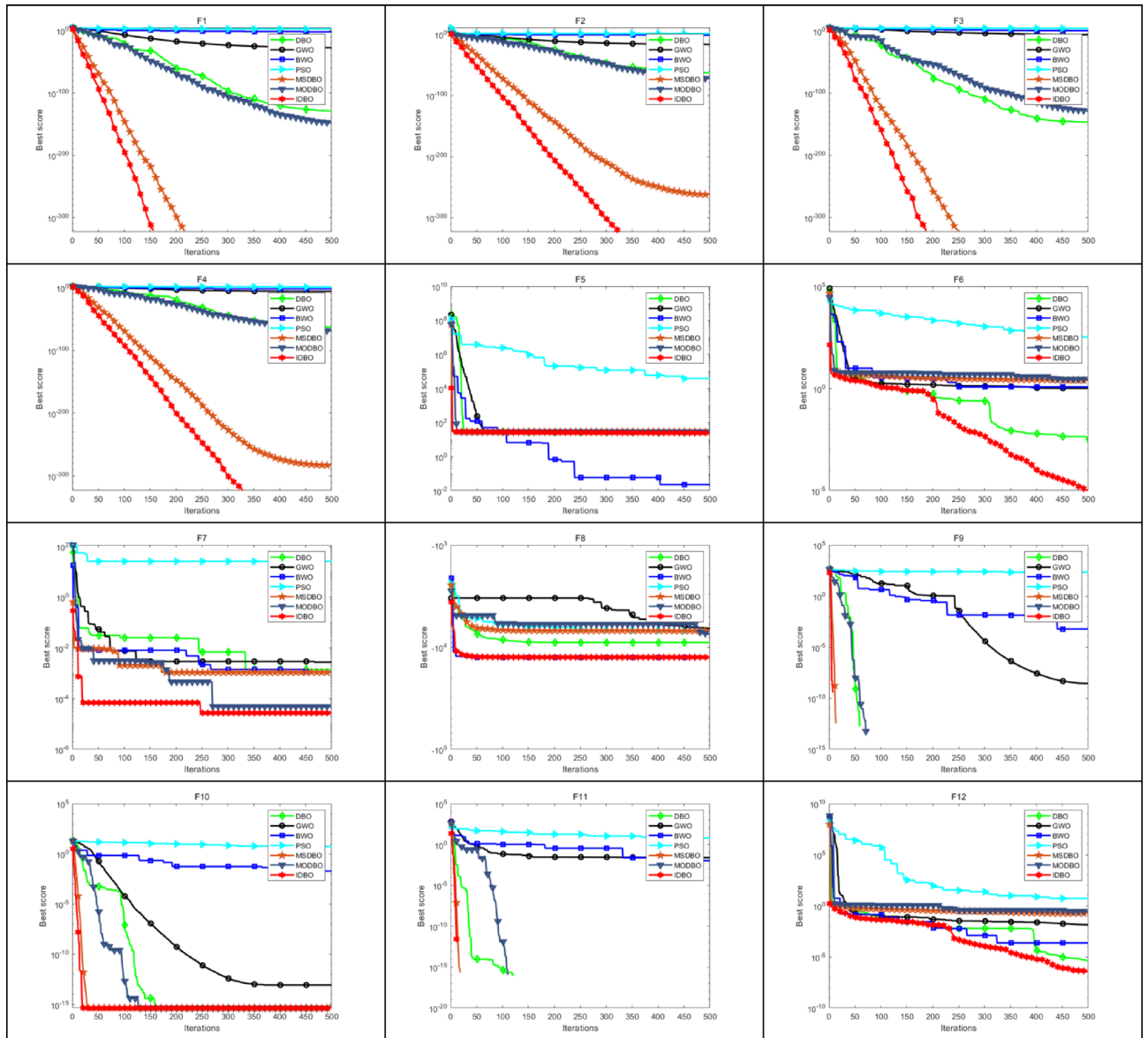


Fig. 4. Convergence curves of different algorithms.

	PSO	DBO	GWO	BWO	MODBO	MSDBO
F1	1.21E-12 +	1.21E-12 +	1.21E-12 +	1.21E-12 +	1.21E-12 +	1.21E-12 +
F2	1.21E-12 +	2.95E-11 +	2.95E-11 +	1.21E-12 +	1.21E-12 +	2.95E-11 +
F3	1.21E-12 +	1.21E-12 +	1.21E-12 +	1.21E-12 +	1.21E-12 +	1.21E-12 +
F4	3.02E-11 +	1.21E-12 +	3.02E-11 +	3.02E-11 +	1.21E-12 +	3.02E-11 +
F5	3.02E-11 +	1.54E-08 +	3.02E-11 +	3.02E-11 +	3.02E-11 +	3.02E-11 +
F6	3.02E-11 +	3.35E-08 +	1.96E-10 +	3.02E-11 +	3.02E-11 +	3.02E-11 +
F7	3.02E-11 +	6.07E-11 +	3.02E-11 +	6.12E-10 +	4.12E-06 +	7.73E-01 -
F8	1.21E-12 +	6.28E-06 +	3.02E-11 +	2.15E-10 +	1.21E-11 +	3.02E-11 +
F9	1.21E-12 +	8.15E-02 -	4.50E-12 +	NaN =	9.50E-06 +	8.56E-04 +
F10	1.21E-12 +	NaN =	1.21E-12 +	NaN =	4.08E-05 +	4.12E-06 +
F11	1.21E-12 +	NaN =	6.62E-04 +	NaN =	6.05E-07 +	3.02E-11 +
F12	3.02E-11 +	1.72E-12 +	3.02E-11 +	3.02E-11 +	3.02E-11 +	3.02E-11 +
+/-	12/0/0	9/2/1	12/0/0	9/3/0	12/0/0	11/0/1

Table 4. Wilcoxon rank sum test.

Fun No.	Index	PSO	DBO	GWO	BWO	MODBO	MSDBO	IDBO
CEC1	mean	1.58E+03	2.62E+04	1.72E+03	4.52E+03	2.42E+03	2.44E+03	2.06E+03
	std	3.48E+03	9.97E+04	3.55E+03	6.31E+03	2.71E+03	2.83E+03	2.70E+03
CEC2	mean	2.15E+03	2.11E+03	5.01E+03	5.25E+03	1.98E+03	1.86E+03	1.34E+03
	std	3.25E+02	3.14E+02	6.48E+02	6.33E+02	1.36E+02	1.40E+02	2.70E+02
CEC3	mean	6.71E+02	7.50E+02	6.24E+03	5.25E+03	4.15E+03	3.98E+03	3.03E+02
	std	7.58E+00	1.81E+01	7.25E+02	7.03E+02	3.73E+02	4.17E+02	3.21E+03
CEC4	mean	1.90E+03	1.90E+03	7.13E+03	8.94E+03	1.86E+03	1.77E+03	1.90E+03
	std	3.18E+03	1.35E+03	9.08E+02	9.71E+02	1.33E+03	1.26E+03	2.13E+02
CEC5	mean	5.34E+03	7.33E+03	2.18E+03	1.54E+03	6.12E+03	6.07E+03	3.82E+03
	std	2.79E+03	1.49E+04	4.04E+02	1.33E+03	1.35E+02	1.33E+02	1.24E+02
CEC6	mean	1.75E+03	1.37E+03	3.44E+03	2.86E+03	1.35E+03	1.29E+03	1.72E+03
	std	1.93E+02	2.68E+02	4.72E+02	2.52E+02	2.34E+02	2.33E+02	1.68E+02
CEC7	mean	3.72E+03	5.17E+03	2.45E+03	2.72E+03	5.02E+03	4.93E+03	2.32E+03
	std	1.90E+03	8.12E+03	3.71E+02	2.86E+02	7.41E+02	6.86E+02	1.74E+02
CEC8	mean	2.58E+03	2.70E+03	3.58E+03	2.57E+03	2.58E+03	2.47E+03	2.29E+03
	std	3.61E+02	3.23E+00	5.26E+02	3.02E+02	2.98E+02	2.87E+02	2.37E+02
CEC9	mean	2.42E+03	2.35E+03	3.38E+03	2.24E+03	2.20E+03	2.14E+03	2.38E+03
	std	1.54E+02	3.15E+02	4.37E+02	2.98E+03	2.88E+02	2.90E+02	1.27E+02
CEC10	mean	2.90E+03	2.90E+03	2.93E+03	2.90E+03	2.79E+03	2.59E+03	2.73E+02
	std	2.15E+01	1.51E+01	5.11E+02	3.53E+02	1.47E+02	1.53E+02	1.32E+02

Table 5. Test results on CEC2021 benchmark test function.

of the segmented image. The smaller σ_{MSE} is, the better the quality of the segmented image; the larger P_{PSNR} is, the more similar the segmented image is to the original image^{48,49}.

$$\sigma_{MSE} = \frac{1}{M \times N} \left[\sum_{x=1}^M \sum_{y=1}^N (f(x, y) - \hat{f}(x, y))^2 \right] \quad (18)$$

$$P_{PSNR} = 10 \times \lg \left(\frac{255^2}{\sigma_{MSE}} \right) \quad (19)$$

$f(x, y)$ and $\hat{f}(x, y)$ in Eq. (18) are the images corresponding to before and after image segmentation, respectively.

As shown in Fig. 5, there are obvious differences in the segmentation results of different algorithms: GWO-K, BWO-K and PSO-K have serious “exposure” phenomena in the process of segmentation, i.e., some areas are too bright and the edges are blurred, which leads to unclear boundaries between the target and the background; MSDBO-K improves this problem to a certain extent, but there is still a situation that the shadow and the water surface area are misclassified as the same category, which shows insufficient adaptation to complex lighting and texture features. Although MSDBO-K has improved this problem to a certain extent, there are still cases in which the shadow and the water surface area are misclassified as the same category, which shows the lack of adaptability to complex lighting and texture features. In contrast, the IDBO-K algorithm has significant advantages in image detail preservation and region boundary recognition, and its segmentation results can accurately extract the head and neck contours of the swan, and effectively eliminate the background interference, so as to realize the clear separation between the swan body and the water surface and maintain the structural integrity of the swan. This shows that the IDBO-K algorithm has stronger feature expression ability and robustness in the task of target segmentation under complex background.

As shown in Fig. 6, the IDBO-K algorithm demonstrates superior visual quality and structural fidelity in image segmentation. The segmented results clearly preserve the overall contours of the subject, with natural edge transitions and no obvious breaks or blurring. Simultaneously, portions of the background texture are well maintained, effectively preventing excessive smoothing or loss of texture information during segmentation. Compared to other algorithms, the results produced by IDBO-K exhibit the lowest color deviation and the most balanced brightness, providing a clear advantage in terms of overall visual consistency and regional distinction.

As shown in Fig. 7, there are notable differences in the performance of various algorithms on Rice image segmentation. Except for IDBO-K, the other algorithms all exhibit varying degrees of overexposure, characterized by excessive bright regions and blurred edge details, which weakens the contrast between the rice grains and the background and results in unclear segmentation boundaries. This indicates that these algorithms have limited stability when handling images with high-brightness textures and uneven grayscale distributions.

Although IDBO-K achieves a relatively better overall segmentation, preserving the clear contours of the rice grains, some limitations remain: certain background texture details are misclassified as part of the target, leading to reduced segmentation accuracy in local regions. This suggests that, for images with complex background textures or subtle grayscale variations, IDBO-K still requires further optimization in feature discrimination and class boundary determination. Nonetheless, overall, IDBO-K outperforms the other algorithms in exposure

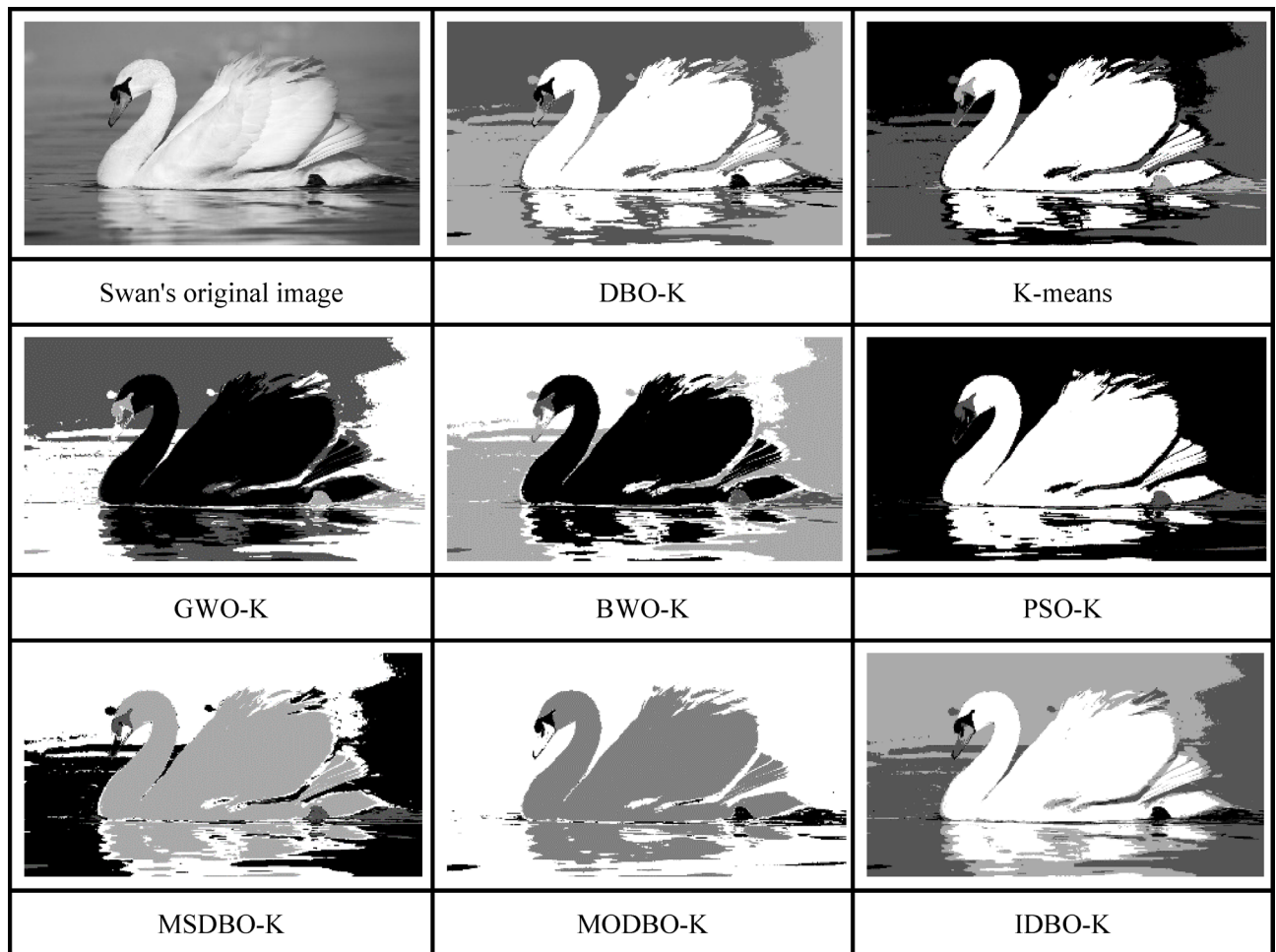


Fig. 5. Comparison of segmentation results before and after Swan image (available under the CC BY 4.0 license. The dataset can be accessed at <https://github.com/BIDS/BSDS500>.)

suppression and preservation of primary contours, demonstrating strong global optimization capability and detail retention performance.

As shown in Fig. 8, there are significant differences in the performance of various algorithms on the flower image segmentation task. GWO-K, PSO-K, and MSDBO-K exhibit varying degrees of overexposure, resulting in excessively bright regions, loss of detail, blurred petal edges, and unclear texture layers. While K-means and BWO-K show slight improvements in brightness control, they still perform poorly in texture detail segmentation, failing to effectively capture the fine structures of the petals, leading to relatively flat overall segmentation results.

In comparison, MODBO-K shows some improvement in overall segmentation quality, preserving the main contours and primary regions of the flower, with reduced overexposure, though it still falls short of achieving ideal brightness and contrast balance. IDBO-K, building on this, further optimizes segmentation quality, demonstrating better regional consistency and structural integrity. The segmented results clearly reflect the shape and layers of the petals, with more distinct separation between foreground and background.

However, compared to the original image, IDBO-K still shows some color reproduction limitations, indicating that further improvement is needed for handling color transitions and texture information under complex lighting conditions. Overall, IDBO-K outperforms the other algorithms in terms of overall visual quality and segmentation accuracy, demonstrating strong stability and generalization capability^{50,51}.

Four types of images are utilized to carry out the image segmentation comparison test at the same time, the segmentation results are objectively evaluated by 2 numerical indexes, namely mean square error and peak signal-to-noise ratio, and the evaluation and comparison data are shown in Table 6. When the K-means algorithm segmented the first three classical images, σ_{MSE} are all the largest, and P_{PSNR} are all the smallest; for the Swan, Lena, and Cameraman diagrams, the GWO-K, A and A obtained after segmentation of BWO-K and PSO-K are greatly improved compared to K-means algorithm, while IDBO-K obtains the smallest σ_{MSE} and the largest P_{PSNR} after segmentation of the three images, i.e., it shows that the algorithm can better adapt and optimize the clustering and segmentation process for the case of large changes in gray levels or complex backgrounds in gray scale images. However, when segmenting the last two classical images, the corresponding σ_{MSE} of IDBO-K increases compared to the first three images, and the corresponding P_{PSNR} of IDBO-K decreases compared to the first three images, which means that the algorithm's segmentation effect is not optimal

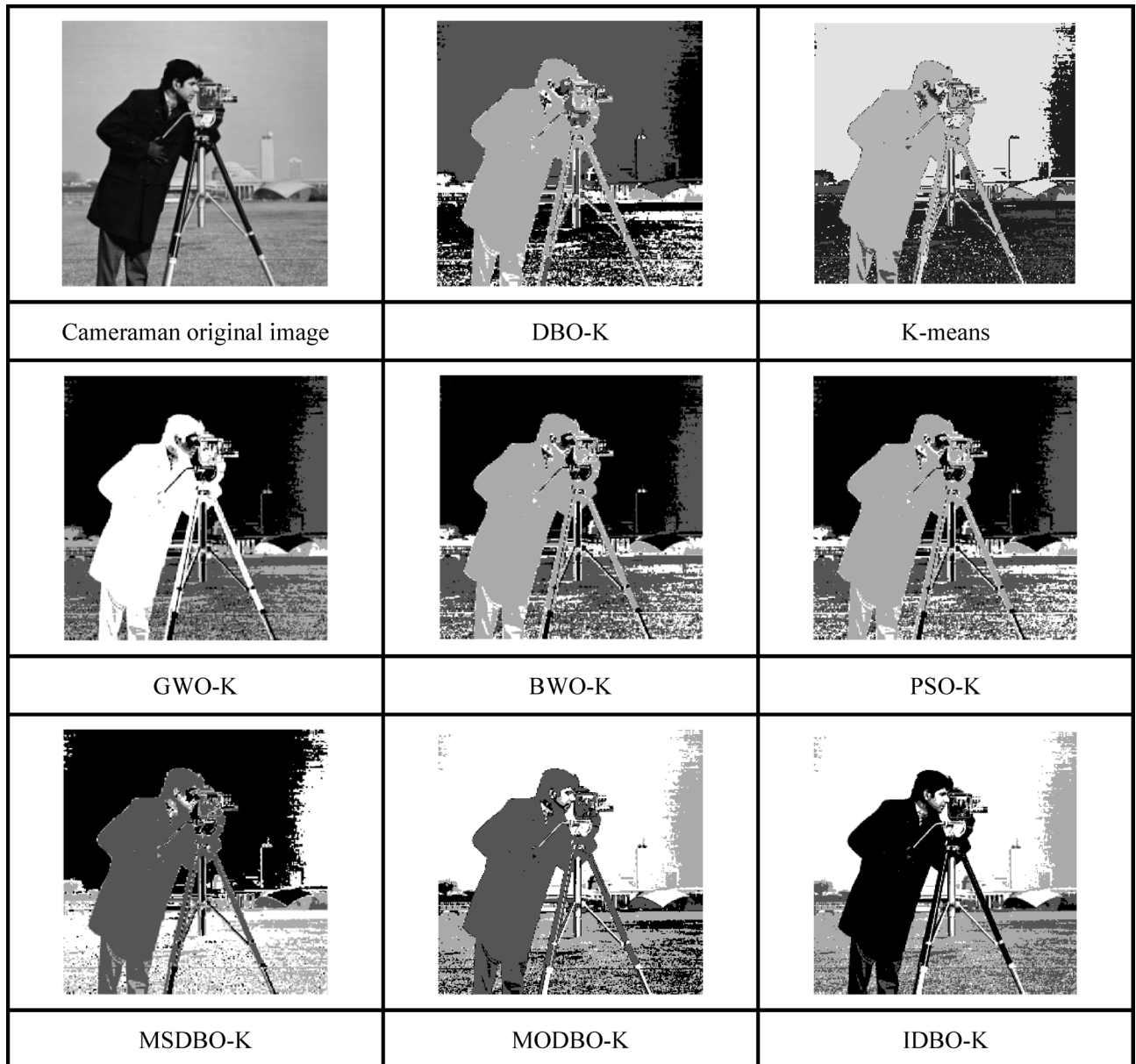


Fig. 6. Comparison of results before and after Cameraman image segmentation (available under the CC BY 4.0 license. The dataset can be accessed at <https://github.com/BIDS/BSDS500>.)

when dealing with the last two images that require very high precision. Therefore, IDBO-K is more effective in segmenting gray-scale images with large changes in gray levels or more complex backgrounds, which has certain feasibility and superiority^{24,36,38,43,52}.

Ablation experiment

Experiment design

To quantitatively verify the individual contributions of each enhancement strategy in the Improved Dung Beetle Optimization (IDBO) algorithm, this section conducts ablation experiments. The experiments are based on 12 benchmark functions (F1–F7 as unimodal functions, F8–F12 as multimodal functions) and the CEC2021 benchmark suite, with consistent parameter settings (population size = 30, maximum iterations = 500, dimensions = 30/100/500) as Sect. 3.1. Evaluation metrics include the best optimization value (Best), mean value (Mean), and standard deviation (Std) to comprehensively reflect optimization accuracy and stability.

To systematically evaluate the contribution of each proposed enhancement strategy, five algorithm variants were designed, with the original Dung Beetle Optimization (DBO) serving as the baseline. Specifically, DBO-Baseline represents the unmodified DBO algorithm; DBO-LHS incorporates only the Latin Hypercube Sampling (LHS) initialization; DBO-HPS integrates solely the hybrid position updating strategy, which combines a nonlinear decision factor with a competition mechanism; DBO-CTO employs only the local exploitation

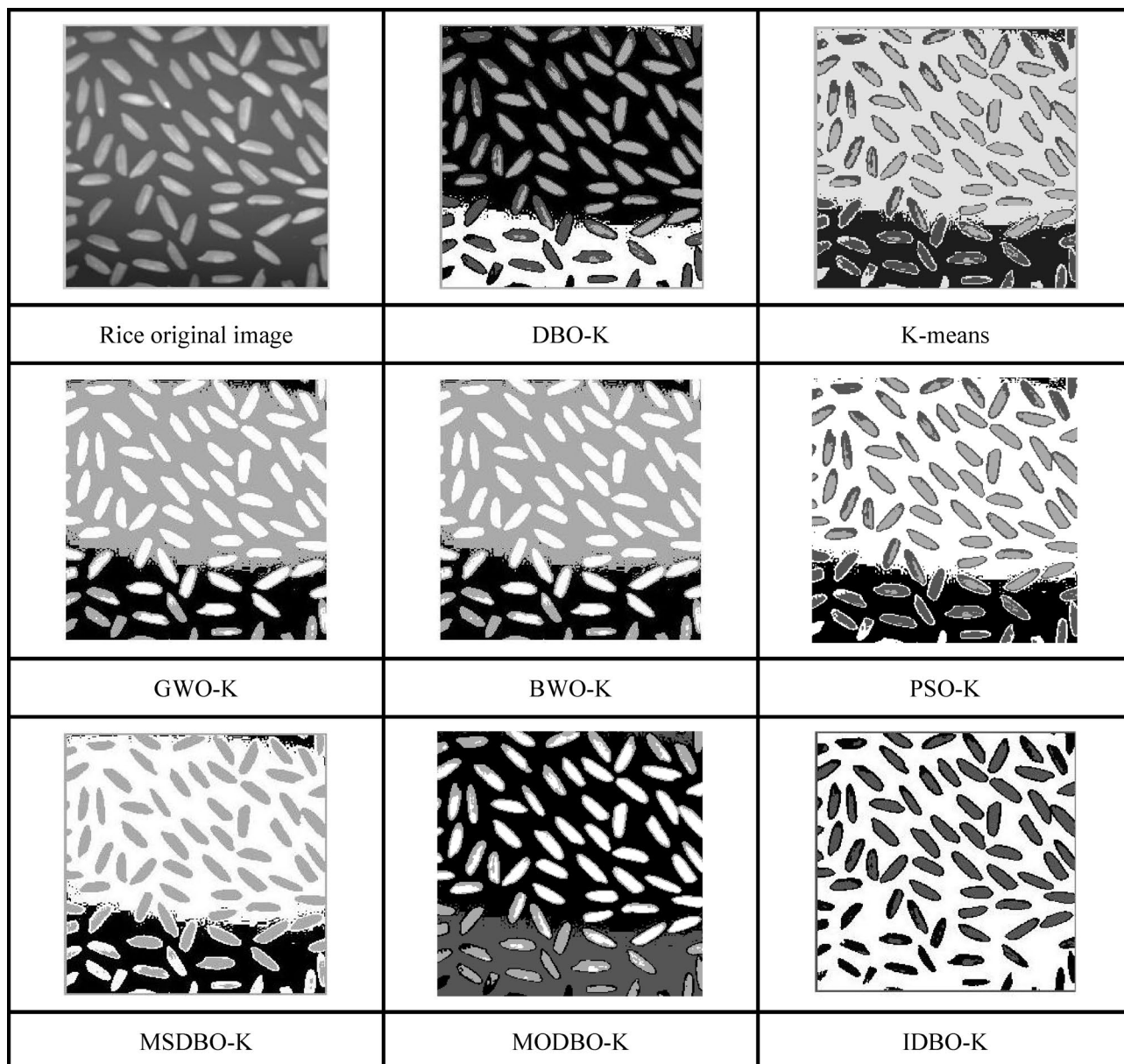


Fig. 7. Comparison of segmentation results before and after Rice image (available under the CC BY 4.0 license. The dataset can be accessed at <https://github.com/BIDS/BSDS500>.)

operators, namely the Cauchy inverse cumulative distribution operator and the tangent flight operator; and IDBO-Full represents the complete improved algorithm, integrating all three strategies. To ensure statistical reliability, each experiment was independently executed 30 times, and performance metrics were aggregated to analyze the effects of individual enhancements and the overall improvement of the IDBO framework.

Results on benchmark functions

The ablation study on unimodal functions (30D) was conducted to evaluate the local exploitation capability of each enhancement strategy, with the results summarized in Table 7. Compared to the baseline, DBO-LHS achieves an average reduction of 32.7% in the Mean error and 28.5% in the Std, indicating that LHS initialization enhances population uniformity and mitigates premature convergence during early iterations. DBO-HPS further improves performance, reducing the Mean error by 41.2% on average, which demonstrates that the hybrid position updating strategy, combining a nonlinear decision factor with a competition mechanism, effectively balances global exploration and local exploitation, particularly in high-dimensional search spaces. DBO-CTO attains a Mean error reduction of 48.5% on average, highlighting that the integration of the Cauchy inverse cumulative distribution operator and tangent flight operator strengthens local fine-tuning accuracy. Finally, IDBO-Full, incorporating all three strategies, achieves the best performance, with a 67.9% reduction in Mean error relative to the baseline and successfully reaching the theoretical optimum (0) for functions F1–F4.

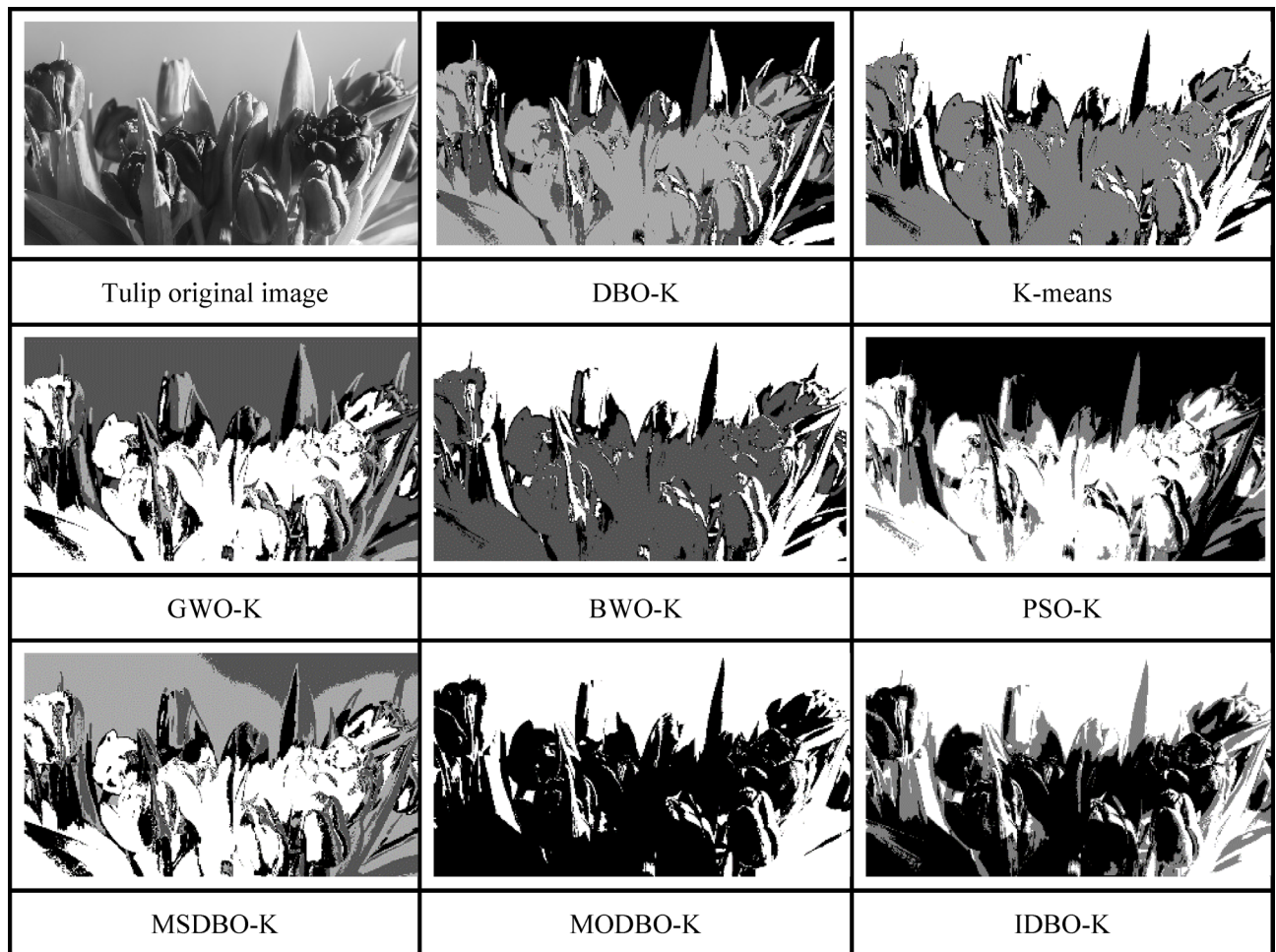


Fig. 8. Comparison of segmentation results before and after Tulip image (available under the CC BY 4.0 license. The dataset can be accessed at <https://github.com/BIDS/BSDS500>.)

These results clearly demonstrate the complementary contributions of each strategy and the superiority of the integrated IDBO framework in enhancing local exploitation.

The ablation study on multimodal functions was conducted to evaluate the global exploration capability and the ability to escape local optima, with results presented in Table 8. DBO-LHS achieves an average reduction of 21.3% in the Mean error compared to DBO-Baseline, indicating that LHS initialization effectively enhances population diversity. DBO-HPS further improves performance with a 35.6% average reduction in Mean error, demonstrating the effectiveness of the hybrid position updating strategy in balancing global exploration and local exploitation. DBO-CTO exhibits the strongest capability to escape local optima, reducing the Mean error by 42.3% on average and achieving the lowest Std among the variants, highlighting the effectiveness of the dual-operator local exploitation mechanism. IDBO-Full, integrating all three strategies, attains the theoretical optimum (0) for functions F9–F11 and achieves a 61.8% reduction in Mean error relative to the baseline, demonstrating superior global optimization performance. These results confirm that each enhancement strategy contributes uniquely to global search efficiency and that the integrated IDBO framework effectively combines exploration and exploitation to achieve robust performance across multimodal landscapes.

All three enhancement strategies contribute positively to the performance improvement of the algorithm. Among them, DBO-CTO plays the most significant role in strengthening local exploitation and escaping local optima, followed by DBO-HPS and DBO-LHS. The fully integrated IDBO-Full leverages the complementary advantages of all strategies, achieving a performance improvement that exceeds the sum of individual contributions, thereby validating the rationality and effectiveness of the multi-strategy fusion design. Furthermore, the enhancement strategies demonstrate consistent performance gains across both unimodal and multimodal functions, indicating that the IDBO framework possesses strong adaptability and robustness across diverse optimization problems.

Conclusion and outlook

To address the limitations of the traditional K-means clustering algorithm, which is highly sensitive to the selection of initial cluster centers, and the Dung Beetle Optimization (DBO) algorithm, which suffers from low

imagery	algorithms	mean square error (σ_{MSE})	peak signal-to-noise ratio (P_{PSNR})
Swan	DBO-K	2.38E-01	46.19
	K-means	2.10E-01	44.91
	GWO-K	1.32E-01	47.77
	BWO-K	1.66E-01	45.95
	PSO-K	9.00E-01	48.61
	MSDBO-K	6.49E-01	46.37
	MODBO-K	1.38E-01	46.52
	IDBO-K	1.13E-01	48.88
Cameraman	DBO-K	2.39E-01	46.06
	K-means	3.70E-01	44.00
	GWO-K	3.82E-01	44.04
	BWO-K	3.40E-01	45.08
	PSO-K	3.17E-01	45.96
	MSDBO-K	2.32E-01	46.26
	MODBO-K	2.78E-01	45.39
	IDBO-K	1.96E-01	48.15
Rice	DBO-K	3.55E-01	41.51
	K-means	3.63E-01	41.33
	GWO-K	4.17E-01	42.86
	BWO-K	4.35E-01	42.75
	PSO-K	5.13E-01	40.41
	MSDBO-K	3.55E-01	41.78
	MODBO-K	3.71E-01	41.57
	IDBO-K	2.99E-01	43.18
Tulip	DBO-K	3.10E-01	42.63
	K-means	3.92E-01	42.25
	GWO-K	3.72E-01	42.37
	BWO-K	3.96E-01	42.18
	PSO-K	3.17E-01	42.58
	MSDBO-K	3.25E-01	42.49
	MODBO-K	2.94E-01	43.07
	IDBO-K	2.58E-01	43.63

Table 6. Comparative metrics for each algorithm segmentation.

optimization accuracy and a tendency to become trapped in local optima in complex optimization problems, this study proposes an improved version of DBO, termed the Improved Dung Beetle Optimization (IDBO) algorithm.

Firstly, Latin Hypercube Sampling (LHS) is employed to initialize the population, enhancing the uniformity and diversity of individuals in the search space and effectively preventing premature convergence during the initial stages of the search. Secondly, a hybrid position updating strategy combining a nonlinear decision factor and a competition mechanism is introduced to dynamically balance global exploration and local exploitation during the iteration process, improving the search flexibility and efficiency of the population at different stages. Finally, a local exploitation operator is constructed by integrating the Cauchy inverse cumulative distribution operator and the tangent flight operator, allowing dynamic fine-tuning of the optimal individuals and further enhancing local search accuracy and stability in the late stages of convergence, thereby significantly improving the overall performance of the algorithm.

Comprehensive experiments on 12 classic benchmark test functions, the Wilcoxon Rank-Sum Test, and the CEC2021 benchmark suite demonstrate that IDBO significantly outperforms the original DBO in terms of convergence speed, global optimization ability, and avoidance of local optima. Building on this, the IDBO algorithm was applied to optimize K-means clustering, forming the IDBO-K image segmentation model, which was tested on various representative images. Results indicate that the proposed method significantly enhances segmentation precision and edge preservation, enabling accurate segmentation of images with complex textures and lighting variations, thereby validating the effectiveness and robustness of the algorithm in image segmentation tasks.

However, despite its demonstrated effectiveness, the IDBO-K model may encounter challenges when applied to large-scale or real-time image segmentation tasks. The iterative nature of IDBO, combined with computationally intensive position updating and local exploitation strategies, may lead to increased runtime and higher computational resource demands, limiting its efficiency in time-sensitive applications or on very high-resolution images.

Function	Algorithm	Best	Mean	Std
F1	DBO-Baseline	5.57E-161	2.39E-117	1.23E-116
	DBO-LHS	1.37E-154	1.37E-103	7.50E-103
	DBO-HPS	3.74E-153	8.45E-107	4.63E-106
	DBO-CTO	0.00E+00	0.00E+00	0.00E+00
	IDBO-Full	0.00E+00	0.00E+00	0.00E+00
F2	DBO-Baseline	9.90E-91	3.72E-59	1.58E-58
	DBO-LHS	4.89E-83	2.87E-60	1.30E-59
	DBO-HPS	4.12E-85	4.03E-59	2.20E-58
	DBO-CTO	0.00E+00	0.00E+00	0.00E+00
	IDBO-Full	0.00E+00	0.00E+00	0.00E+00
F3	DBO-Baseline	3.26E-143	1.08E-46	5.90E-46
	DBO-LHS	7.87E-145	5.78E-75	3.16E-74
	DBO-HPS	2.41E-131	1.32E-52	7.24E-52
	DBO-CTO	0.00E+00	0.00E+00	0.00E+00
	IDBO-Full	0.00E+00	0.00E+00	0.00E+00
F4	DBO-Baseline	3.54E-84	1.81E-52	9.80E-52
	DBO-LHS	9.56E-84	3.30E-55	1.49E-54
	DBO-HPS	9.79E-79	4.12E-52	2.26E-51
	DBO-CTO	0.00E+00	0.00E+00	0.00E+00
	IDBO-Full	0.00E+00	0.00E+00	0.00E+00
F5	DBO-Baseline	2.35E+01	2.58E+01	2.71E-01
	DBO-LHS	2.34E+01	2.57E+01	1.69E-01
	DBO-HPS	2.31E+01	2.57E+01	2.29E-01
	DBO-CTO	2.27E+01	2.44E+01	6.30E-01
	IDBO-Full	2.26E+01	2.46E+01	5.83E-01
F6	DBO-Baseline	2.44E-05	1.73E-01	2.52E-01
	DBO-LHS	5.53E-06	9.77E-03	5.16E-02
	DBO-HPS	4.24E-06	8.06E-03	2.89E-02
	DBO-CTO	3.58E-07	2.92E-01	3.75E-01
	IDBO-Full	9.99E-08	1.03E-03	1.39E-03
F7	DBO-Baseline	1.55E-04	1.79E-03	2.23E-03
	DBO-LHS	2.00E-05	1.83E-04	1.44E-04
	DBO-HPS	6.06E-05	1.47E-03	9.16E-04
	DBO-CTO	6.85E-06	1.72E-04	1.96E-04
	IDBO-Full	2.80E-06	1.53E-04	1.43E-04

Table 7. Ablation experiment results on unimodal functions (30D).

Future research will focus on integrating machine learning and deep learning techniques to further improve the DBO algorithm, aiming to enhance its adaptability, generalization, and computational efficiency in high-dimensional and large-scale scenarios. Specifically, optimizing parallelization strategies, reducing iteration costs, and incorporating predictive models could expand the potential applications of IDBO in real-time image processing, feature extraction, and pattern recognition while maintaining segmentation accuracy and robustness.

Function	Algorithm	Best	Mean	Std
F8	DBO-Baseline	-1.16E+04	-8.49E+03	1.80E+03
	DBO-LHS	-1.16E+04	-8.63E+03	2.07E+03
	DBO-HPS	-1.12E+04	-9.33E+03	1.81E+03
	DBO-CTO	-1.26E+04	-1.06E+04	1.87E+03
	IDBO-Full	-1.36E+04	-1.09E+04	1.85E+03
F9	DBO-Baseline	0.00E+00	2.09E+00	9.17E+00
	DBO-LHS	0.00E+00	1.18E+00	3.95E+00
	DBO-HPS	0.00E+00	4.19E+00	2.25E+01
	DBO-CTO	0.00E+00	0.00E+00	0.00E+00
	IDBO-Full	0.00E+00	0.00E+00	0.00E+00
F10	DBO-Baseline	0.00E+00	2.09E+00	9.17E+00
	DBO-LHS	0.00E+00	1.18E+00	3.95E+00
	DBO-HPS	0.00E+00	4.19E+00	2.25E+01
	DBO-CTO	0.00E+00	0.00E+00	0.00E+00
	IDBO-Full	0.00E+00	0.00E+00	0.00E+00
F11	DBO-Baseline	0.00E+00	2.09E+00	9.17E+00
	DBO-LHS	0.00E+00	1.18E+00	3.95E+00
	DBO-HPS	0.00E+00	4.19E+00	2.25E+01
	DBO-CTO	0.00E+00	0.00E+00	0.00E+00
	IDBO-Full	0.00E+00	0.00E+00	0.00E+00
F12	DBO-Baseline	2.51E-07	9.12E-04	2.27E-03
	DBO-LHS	1.75E-07	1.14E-04	4.45E-04
	DBO-HPS	1.25E-07	6.68E-04	1.86E-03
	DBO-CTO	3.27E-08	1.30E-03	4.29E-03
	IDBO-Full	1.50E-08	3.46E-05	8.35E-05

Table 8. Ablation experiment results on multimodal functions (30D).

Data availability

The data used in this study is derived from the BSD500 dataset, which is publicly available under the terms of the Creative Commons Attribution 4.0 International License (CC BY 4.0). The dataset can be accessed through the following GitHub repository: <https://github.com/BIDS/BSDS500>. This dataset includes a variety of high-resolution images and ground truth annotations for image segmentation tasks. The images are used in accordance with the CC BY 4.0 license, which allows for sharing, adaptation, and redistribution with appropriate attribution.

Received: 19 November 2025; Accepted: 29 January 2026

Published online: 02 April 2026

References

- Gao, Y. et al. Medical image segmentation: A comprehensive review of deep Learning-Based methods. *Tomography* **11**, 52 (2025).
- Wiley, V. & Lucas, T. Computer vision and image processing: a paper review. *Int. J. Artif. Intell. Res.* **2**, 29–36 (2018).
- Ashour, L. S. et al. Non-overlapping Patch-Based Pre-trained CNN for breast cancer classification. *Iraqi J. Comput. Sci. Math.* **6**, 29 (2025).
- Chen, E., Ting, H. N., Chuah, J. H. & Zhao, J. Segmentation of overlapping cells in cervical cytology images: A survey. *IEEE Access*. **12**, 114170–114189 (2024).
- Obuchowicz, R., Strzelecki, M. & Piórkowski, A. MDPI, Vol. 16 1870 (2024).
- Xu, Y. et al. Advances in medical image segmentation: A comprehensive review of traditional, deep learning and hybrid approaches. *Bioengineering* **11**, 1034 (2024).
- Li, Y., Song, X., Tu, Y. & Liu, M. G. A. P. B. A. S. Genetic algorithm-based privacy budget allocation strategy in differential privacy K-means clustering algorithm. *Computers Secur.* **139**, 103697 (2024).
- Liao, J., Wu, X., Wu, Y. & Shu, J. K-NNDP: K-means algorithm based on nearest neighbor density peak optimization and outlier removal. *Knowl. Based Syst.* **294**, 111742 (2024).
- Chai, X. et al. Image segmentation based on the optimized K-Means algorithm with the improved hybrid grey Wolf optimization: application in ore particle size detection. *Sensors* **25**, 2785 (2025).
- Zhao, J., Bao, Y., Li, D. & Guan, X. An improved K-Means algorithm based on contour similarity. *Mathematics* **12**, 2211 (2024).
- Premkumar, M. et al. Augmented weighted K-means grey Wolf optimizer: an enhanced metaheuristic algorithm for data clustering problems. *Sci. Rep.* **14**, 5434 (2024).
- Das, A., Namtirtha, A. & Dutta, A. Lévy–Cauchy arithmetic optimization algorithm combined with rough K-means for image segmentation. *Appl. Soft Comput.* **140**, 110268 (2023).
- Song, Y., Peng, G., Sun, D. & Xie, X. Active contours driven by Gaussian function and adaptive-scale local correentropy-based K-means clustering for fast image segmentation. *Sig. Process.* **174**, 107625 (2020).
- Belete, B. A., Gelmecha, D. J. & Singh, R. S. Enhancing colour image encryption through parameters optimization of memristive hyperchaotic system with CPSO algorithm and LSAIM. *Imaging Sci. J.* **73** (6), 681–701 (2025).

15. Xue, J. & Shen, B. Dung beetle optimizer: A new meta-heuristic algorithm for global optimization. *J. Supercomputing*. **79**, 7305–7336 (2023).
16. Lu, M., Wang, Y., Qiu, S., Li, N. & Shi, Y. in *Proceedings of the 2024 3rd International Symposium on Control Engineering and Robotics*. 498–504.
17. Yong, X., Bicong, S. & Yi, Z. Application of an improved pelican optimization algorithm based on comprehensive strategy in PV parameter identification. *Sci. Rep.* **15**, 27931 (2025).
18. Mehmood, K. et al. Design of chaos induced Aquila optimizer for parameter estimation of electro-hydraulic control system. *Comput. Model. Eng. Sci.* **143**, 1809–1841 (2025).
19. Mehmood, K. et al. Design of chaotic young's double Slit experiment optimization heuristics for identification of nonlinear muscle model with key term separation. *Chaos Solitons Fractals*. **189**, 115636 (2024).
20. Zheng, X., Lei, Q., Yao, R., Gong, Y. & Yin, Q. Image segmentation based on adaptive K-means algorithm. *EURASIP Journal on Image and Video Processing* 1–10 (2018).
21. Yao, H., Duan, Q., Li, D. & Wang, J. An improved K-means clustering algorithm for fish image segmentation. *Math. Comput. Model.* **58**, 790–798 (2013).
22. Ruban, I. et al. Methods of Uavs images segmentation based on k-means and a genetic algorithm. *Eastern-European J. Enterpr. Technologies* **118** (2022).
23. Mao, Z. et al. A multi-strategy enhanced Dung beetle algorithm for solving real-world engineering problems. *Artif. Intell. Rev.* **58**, 253 (2025).
24. Fang, R. et al. Dung beetle optimization algorithm based on improved Multi-Strategy fusion. *Electronics* **14**, 197 (2025).
25. Ma, Z., Liu, S. & Xu, L. Enhancing dung beetle optimization algorithm with hybrid multi-strategy and its engineering applications. *Neural Comput. Applic.* **37**, 19123–19175 (2025).
26. Liu, L. et al. A full-coverage path planning method for an orchard mower based on the Dung beetle optimization algorithm. *Agriculture* **14**, 865 (2024).
27. Huang, B., Yang, G., Lei, J. & Wang, X. A partitioned conditioned Latin hypercube sampling method considering Spatial heterogeneity in digital soil mapping. *Sci. Rep.* **15**, 12851 (2025).
28. Pandey, N. K. & Satyam, N. Role of finer particles in rheological characterization of debris flows: insights from Western Himalayas. *Indian Geotech. J.* **55**, 2308–2324 (2025).
29. Phromphan, P., Suvisuthikasame, J., Kaewmongkol, M., Chanpichitwanich, W. & Slesongsom, S. A new Latin hypercube sampling with maximum diversity factor for reliability-based design optimization of HLM. *Symmetry* **16**, 901 (2024).
30. Al-Daraiseh, A. et al. Cryptographic grade chaotic random number generator based on tent-map. *J. Sens. Actuator Networks*. **12**, 73 (2023).
31. Chen, Q. et al. Spontaneous coal combustion temperature prediction based on an improved grey Wolf optimizer-gated recurrent unit model. *Energy* **314**, 133980 (2025).
32. Motwakel, A. et al. Chaotic mapping Lion optimization Algorithm-Based node localization approach for wireless sensor networks. *Sensors* **23**, 8699 (2023).
33. Deng, W. et al. Multi-strategy quantum differential evolution algorithm with cooperative co-evolution and hybrid search for capacitated vehicle routing. *IEEE Trans. Intell. Transp. Syst.* **26**, 18460–18470 (2025).
34. Khoshaba, F. S., Kareem, S. W. & Hawezi, R. S. Swallow search algorithm (SWSO): A swarm intelligence optimization approach inspired by swallow bird behavior. *Computers* **14**, 345 (2025).
35. Munteanu, A. & Words Meaning and Vocabulary An introduction to modern English lexicology. *Words, Meaning and Vocabulary An introduction to modern English lexicology* (2025).
36. Givhan, C. A. Design and analysis of a dual antenna vector tracking software defined receiver for robust navigation and mitigation of spoofing threats. (2025).
37. Hemayat, S., Baharlou, S. M., Sergienko, A. & Ndao, A. Efficient Inverse Design of Plasmonic Patch Nanoantennas using Deep Learning. *arXiv preprint arXiv:2407.03607* (2024).
38. Etesami, R., Madadi, M. & Keynia, F. A new improved fruit fly optimization algorithm based on particle swarm optimization algorithm for function optimization problems. *Journal Mahani Math. Res. Center* **13** (2024).
39. Cuevas, E., González-Sánchez, O. A., Delgado-Castañeda, N. & Zaldivar, D. Rodríguez-Vazquez, A. A novel metaheuristic algorithm using structured population and virtual particles. *J. Supercomputing*. **81**, 1–90 (2025).
40. Thapliyal, S. & Kumar, N. Comprehensive performance metric for bio-inspired optimizers: a generation-wise population convergence approach toward global optimality. *Iran J. Comput. Sci.*, **8**, 843–891 (2025).
41. Jia, H., Wen, Q., Wang, Y. & Mirjalili, S. Catch fish optimization algorithm: a new human behavior algorithm for solving clustering problems. *Cluster Comput.* **27**, 13295–13332 (2024).
42. Wang, X., Wei, P. & Li, Y. Enhanced secretary bird optimization algorithm with multi-strategy fusion and Cauchy–Gaussian crossover. *Sci. Rep.* **15**, 23163 (2025).
43. Escobar-Cuevas, H., Cuevas, E., Avila, K. & Avalos, O. An advanced initialization technique for metaheuristic optimization: a fusion of Latin hypercube sampling and evolutionary behaviors. *Comput. Appl. Math.* **43**, 234 (2024).
44. Wang, Z. et al. Improved Latin hypercube sampling initialization-based Whale optimization algorithm for COVID-19 X-ray multi-threshold image segmentation. *Sci. Rep.* **14**, 13239 (2024).
45. Wu, L., Wu, J. & Wang, T. The improved grasshopper optimization algorithm with cauchy mutation strategy and random weight operator for solving optimization problems. *Evol. Intel.* **17**, 1751–1781 (2024).
46. Gu, W. & Wang, F. A multi-strategy improved Dung beetle optimisation algorithm and its application. *Cluster Comput.* **28**, 49 (2025).
47. Lyu, L., Jiang, H. & Yang, F. Improved Dung beetle optimizer algorithm with multi-strategy for global optimization and UAV 3D path planning. *IEEE Access*. **12**, 69240–69257 (2024).
48. Jia, H. et al. Improved artificial rabbits algorithm for global optimization and multi-level thresholding color image segmentation. *Artif. Intell. Rev.* **58**, 55 (2024).
49. Zhang, B. et al. Denoising Swin transformer and perceptual peak signal-to-noise ratio for low-dose CT image denoising. *Measurement* **227**, 114303 (2024).
50. OuYanga, K. et al. Escape.
51. Tu, K. & Cheng, J. Enhanced Dung beetle optimization algorithm and its application in 3D UAV path planning. *Electron. Res. Archive* **33**, 2618–2667 (2025).
52. Hu, Y., Zhu, L. & Zhao, H. Multistage threshold segmentation method based on improved electric eel foraging optimization. *Mathematics* **13**, 1212 (2025).

Author contributions

Author Contributions: Ning Li : Writing – review & editing, Validation, Supervision, Project administration, Methodology, Investigation, Data curation, Conceptualization. Yan Luo : Writing – review & editing, Software, Methodology, Formal analysis, Data curation, Conceptualization. Zhiqiang Feng : Writing – review & editing, Writing – original draft, Validation, Software, Funding acquisition, Data curation. Hu Qu : Formal analysis,

Methodology, Investigation, Data curation, Conceptualization. Zixuan Feng : Methodology, Validation, Software.

Funding

This work was supported by the National Natural Science Foundation of China (Grant Nos. 52261044 and 52074283) and the National Key Research and Development Program of China (Grant No. 2020YFA0711802).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

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

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

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

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2026