Scientific Reports

https://doi.org/10.1038/s41598-026-36149-2

Article in Press

An enhanced connected banking system optimizer incorporating triple mechanism for solving global optimization problems

Received: 16 November 2025

Accepted: 9 January 2026

Published online: 07 February 2026

Cite this article as: Qian D., Cai X., Feng L. *et al.* An enhanced connected banking system optimizer incorporating triple mechanism for solving global optimization problems. *Sci Rep* (2026). https://doi.org/10.1038/s41598-026-36149-2

Dake Qian, Xinyu Cai, Leidong Feng & Yun Ye

An Enhanced Connected Banking System Optimizer Incorporating Triple Mechanism for Solving Global Optimization Problems

Dake Qian [1, +], Xinyu Cai [2, +], Leidong Feng [2,*] and and Yun Ye [3,4, *]

[1] School of Economics Jiaxing University, Jiaxing 314001, China;
[2] College of Business, Jiaxing University, Jiaxing 314001, China;
[3] Department of Civil and Environmental Engineering, Imperial College London, London SW7 2AZ, UK;
[4] Faculty of Maritime and Transportation, Ningbo University, Ningbo 315211, China
[+] These authors contributed equally to this work.
* Correspondence: ldfeng@zjxu.edu.cn (L.F.); yun.ye25@imperial.ac.uk (Y.Y.)

Abstract: Connected Banking System Optimizer (CBSO) is a recently proposed meta-heuristic inspired by inter-bank financial transactions. It models inter-bank transaction behaviors across four sequential stages, collectively balancing exploration and exploitation. When confronted with complex landscapes, however, CBSO exposes three critical weaknesses: limited global-search capacity, an abrupt phase switch that disrupts the exploitation-exploration balance, and a pronounced tendency toward premature stagnation. These shortcomings become more conspicuous as problem complexity rises, undermining the algorithm's ability to locate the true optimum. To overcome these deficiencies, this paper presents an enhanced variant—ECBSO—which incorporates three complementary mechanisms: dominant group guidance strategy, guided learning strategy, and hybrid elite strategy. The ECBSO algorithm is comprehensively evaluated on the CEC 2017 benchmark suite and on real-world constrained engineering problems, outperforming CBSO, ISGTOA, EMTLBO, LSHADE, APSM-jSO, GLS-MPA, ESLPSO, ACGRIME, RDGMVO in all comparisons. Statistically, ECBSO secures first place across every test case, delivering Friedman ranks of 2.069, 2.138, 2.690, and 2.759, thereby confirming its superior convergence accuracy, search reliability, and optimization precision across diverse landscapes.

Keywords: Connected banking system optimizer; meta-heuristic algorithm; CEC 2017 test suite

1. Introduction

1.1. Research Background

Over the past decade, artificial intelligence has progressed at an unprecedented pace, with optimization algorithms serving as a representative illustration [1]. Every optimization formulation can be decomposed into three indispensable elements: the objective function requiring optimization, the constraints that delimit the feasible region, and the decision variables whose numerical values must be ascertained [2]. Nevertheless, real-world optimization problems are often intricate because they simultaneously involve many variables, multiple objectives, numerous constraints, nonlinear relationships, multimodal landscapes, and non-differentiable points [3]. From a computational-complexity perspective, these problems are usually classified as NP-hard, so the discovery of exact solutions demands computational resources that grow beyond polynomial bounds and quickly become impractical. These difficulties motivate the design of efficient and robust solution strategies. Because of their intrinsic complexity, these problems are rarely solved effectively by traditional analytical or deterministic techniques [4]. Consequently, researchers have devised more advanced and efficient alternatives. In broad terms, optimization approaches can be divided into two groups: traditional optimization methods and metaheuristic algorithms. Traditional optimization methods rely on rigorous mathematical procedures—linear programming, nonlinear programming, branch-and-bound, and dynamic programming—to locate the globally optimal solution; yet their practical deployment is often restricted by the excessive computational resources they require, particularly for complex instances [5]. Their first weakness is an uncompromising dependence on idealized assumptions—linearity, convexity, and continuous differentiability—whose absence derails convergence to the true global optimum. When the objective surface is jagged or riddled with multiple local attractors, the search collapses into suboptimal basins. A second, equally critical deficiency is computational scalability: on NP-hard instances, exact procedures trigger an exponential surge in runtime and memory footprint. The combined effect renders traditional techniques ill-suited for large-scale, high-dimensional, and strongly nonlinear optimization challenges [6]. Metaheuristic algorithms constitute a family of stochastic optimizers whose designs are motivated by observations in nature, physics, or human decision-making. In contrast to classical techniques, they operate without strict prerequisites such as gradient expressions or convex structures [7]. Each iteration employs probabilistic operators to probe the search space and to refine the incumbent solution, yielding monotonic quality gains. An explicit synergy between global exploration—wide-ranging sampling of candidate configurations—and local exploitation—intensive refinement of high-potential regions—curtails premature stagnation in sub-optimal basins. The resulting robustness and adaptability allow these algorithms to handle rugged, high-dimensional, and multimodal objectives with comparative ease, a versatility evidenced by successful deployments in unmanned aerial vehicle path planning [8,9], real-time power dispatch in microgrids [10,11], parameter estimation of solar photovoltaic models [12], image segmentation [13,14], hyper-parameter optimization [15,16], microgrid operation model optimization [17], workshop scheduling [18], and feature selection [19,20]. These cross-disciplinary success cases collectively corroborate the universal utility of this class of algorithms in complex, dynamic, and high-dimensional environments.

1.2. Contribution of the Work

The primary contributions of this paper are:

(1) An enhanced variant of CBSO, termed ECBSO, is proposed, integrating three improvement mechanisms: dominant group guidance strategy, guided learning strategy, and hybrid elite strategy.

(2) Dominant group guidance strategy leverages effective information from the dominant group to guide the population search, enhancing global exploration capability.

(3) Guided learning strategy dynamically assesses the current algorithmic demand, maintaining an effective exploitation–exploration balance while strengthening adaptability to complex landscapes.

(4) Hybrid elite strategy prevents premature convergence by efficiently utilizing multiple elite agents through the incorporation of an equilibrium optimizer.

(5) Comprehensive evaluation using the CEC 2017 test suite demonstrates ECBSO's superiority in optimization performance, robustness, and scalability over other algorithms, and statistical validation confirms its consistent superiority across diverse optimization challenges.

1.3. Section Arrangement

A literature review is conducted in Section 2. Section 3 formulates the mathematical model of the CBSO algorithm. Section 4 presents the details of the proposed enhancement mechanisms. Sections 5 provide a comprehensive evaluation of the ECBSO algorithm on the CEC 2017 test suite. Also, Section 5 discusses and analyzes all experimental results. Finally, Section 6 summarizes the contributions of this work and outlines directions for future research.

2. Literature Review

Based on algorithmic lineage and design rationale, mainstream metaheuristics can be taxonomized into four paradigms: evolutionary drivers, swarm synergies, physical metaphors, and human behaviors. Evolution-based algorithms follow directions by imitating biological evolution principles. These include iterative algorithms to improve the quality of a solution from one generation to the next using common mechanisms like selection, crossover, and mutation. For instance, Differential Evolution (DE) [21] optimizes the solutions in every generation by retaining high-quality individuals and discarding poor ones to realize a gradual convergence in the best solutions. The Genetic Algorithm (GA) [22] is the best-known evolutionary algorithm that simulates natural selection according to the theory proposed by Darwin and uses genetic manipulations to address optimization problems. Other examples include Evolutionary Strategy (ES) [23], Genetic Programming (GP) [24], and Evolutionary Mating Algorithm (EMA) [25]. Swarm-based algorithms belong to this family, drawing inspiration from the self-organizing dynamics observed in natural swarms. Their unifying characteristic is a nature-centered architecture that replicates biological phenomena—social interactions, foraging strategies, and cooperative decision-making—by enabling a population of agents to traverse the solution space in a collective manner. For example, Ant Colony Optimization (ACO) [26] replicates the pheromone-laying and trail-following behavior of foraging ants to guide the search toward promising regions of the solution space. Similarly, Particle Swarm Optimization (PSO) [27] harnesses the exchange of positional information among individuals to steer the entire swarm toward the global optimum. Other examples include Tuna Swarm Optimization (TSO) [28], Superb Fairy-wren Optimization Algorithm (SFOA) [29], Chinese Pangolin Optimizer (CPO) [30], and Crayfish Optimization Algorithm (COA) [31]. Physics-based algorithms apply concepts of physics to model and solve complex optimization problems, including thermodynamic equilibration, mechanical motion, electromagnetic field evolution, quantum transitions, and optical refraction. For example, Simulated Annealing (SA) [32] emulates the controlled cooling of a heated metal, accepting occasional uphill moves during the thermal descent to escape local minima and settle into the global optimum. Snow Ablation Optimizer (SAO) [33] mimics the gradual melting and sublimation of snow cover, dynamically adjusting ablation rates to maintain an effective balance between wide-ranging exploration and focused exploitation. The RIME algorithm [34] draws on the physical processes that create soft-rime and hard-rime ice formations, using differential accretion rates to expand the search breadth while refining promising regions. Other examples include Polar Lights Optimizer (PLO) [35], Henry Gas Solubility Optimization (HGSO) [36], and Light

Spectrum Optimizer (LSO) [37]. Human-based algorithms translate individual or collective human behaviors, activities, and social systems into computational search strategies. Group Teaching Optimization Algorithm (GTOA) [38] recreates the dialogue between instructors and learners in a classroom, using pedagogical exchanges to guide candidate solutions toward optimality. Other examples include Football Team Training Algorithm (FTTA) [39], Enterprise Development Optimization (EDO) [40], Catch Fish Optimization Algorithm (CFOA) [41].

Although metaheuristics have demonstrated strong competitiveness on complex optimization tasks, their efficacy is often curtailed by premature convergence or sluggish convergence rates. The No-Free-Lunch theorem formally establishes that no single algorithm can maintain universal superiority across all problem domains [42]. Consequently, researchers systematically enhance basic metaheuristics to secure more reliable performance across diverse problem classes. Predominant enhancement routes include refined population initialization, adaptive parameter control, improved search operators, and hybrid algorithmic frameworks. Han et al. used sinusoidal chaotic mapping to generate the initial population and designed a new convergence factor to successfully improve the performance of the hippo optimization algorithm [43]. Hector proposes an unstructured framework that incorporates the particle swarm algorithm into the Cheetah optimizer to enhance its global search [44]. Celik et al. introduced a distance-fitness learning (DFL) mechanism that simultaneously trains each search agent from the best, worst and an ideal distance-fitness candidate, enabling the reptile search algorithm to explore more distant regions while retaining exploitation pressure toward promising zones; this noticeably enlarges the global-search capacity and improves solution quality [45]. Parul et al. used the Levy flight strategy to improve their search during the foraging phase and reduce the probability of falling into a local optimum [46]. Tegani et al. integrated multiple enhancement mechanisms to boost both population diversity and convergence speed of the Teaching–Learning-Based Optimization algorithm, and they validated the resulting variant on a wide range of benchmark suites, reporting significantly superior results compared with the original TLBO and several recent meta-heuristics [47]. Sanjib et al. enhanced the exploration capability of the backtracking search algorithm by integrating a power-mutation operator with an adaptive perturbation strength and a joint-mutation operator that performs a random jump whenever the best solution stagnates; this combination significantly enlarges the search scope while preserving exploitation pressure, yielding a well-balanced trade-off between exploration and exploitation [48]. Xiao et al. merged the Aquila optimizer exploration ratings and the African vultures optimization algorithm exploration ratings and proposed composite opposition-based learning and fitness-distance balance strategies in order to generate better solutions [49]. Ahmed et al. proposed a chaotic white shark optimizer that extends the search reins capability by using different chaotic operators in place of the original random sequence [50]. Cao et al. proposed an improved particle swarm optimization based on a grouping strategy and enhanced its runtime efficiency through a distributed framework [51]. Lu et al. employed a quantum random-number generator to enrich initial-population diversity and introduced a quantum-tunnelling mechanism to help the whale-optimization algorithm escape local optima [52].

The Connected Banking System Optimizer (CBSO) is a human-based metaheuristic inspired by inter-bank transaction behaviors [53]. It models the process in four sequential stages. By dynamically balancing exploration and exploitation, this high-level behavioral model enables CBSO to navigate complex optimization landscapes effectively. Although the original CBSO algorithm has shown promising results, it still suffers from three main weaknesses. Limited inter-population information exchange: the algorithm does not provide sufficient interaction among banks, narrowing its ability to discover unexplored regions of the solution space. Rigid phase switching: CBSO relies on a predetermined, manual division of search stages to alternate between exploration and exploitation, so it cannot dynamically diagnose the current search state and therefore wastes computational effort. Premature convergence: the search operators consider only the influence of a randomly chosen bank and the best-performing bank, which often leads the population to stagnate in local optima. To overcome these drawbacks, this paper proposes an Enhanced CBSO (ECBSO) that integrates three complementary mechanisms: dominant group guidance strategy, guided learning strategy, and hybrid elite strategy.

The first mechanism, the dominant-group guidance strategy, systematically samples the high-quality sub-population and extracts their collective search experience to steer the entire swarm toward promising regions, thereby amplifying global exploration. To reconcile exploration and exploitation while adapting to the algorithm's momentary needs, the guided learning strategy continuously monitors the dispersion of recent search trajectories. By inferring whether the swarm is currently under-exploring or over-exploiting, it issues corrective guidance signals that dynamically tilt the search behavior toward the required phase. Finally, the hybrid elite strategy couples an intensification module—implemented through an equilibrium optimizer—with an explicit diversity-preservation routine. This dual design strengthens convergence without allowing premature collapse, ensuring that exploitation gains do not come at the expense of population variety.

This study employs the IEEE CEC 2017 benchmark suite to evaluate the proposed ECBSO across a broad spectrum of dimensions and numerical landscapes. This paper identifies the optimal parameter settings for ECBSO through systematic sensitivity analyses and demonstrates the individual impact of each proposed strategy. The algorithm is then compared extensively against advanced improved algorithms to verify its performance. Statistical validation via the Wilcoxon rank-sum, Friedman, and Nemenyi tests confirms the superior performance of ECBSO. Additional experiments on real-world constrained engineering problems further demonstrate the algorithm's wide applicability and robustness.

3. Connected banking system optimizer

The CBSO algorithm is based on the transaction behaviors among different banks, and its mathematical model is constructed exactly from these behaviors. Its framework switches between exploration and exploitation phases during the optimization process to locate the optimal solution (or an approximate optimum). The mathematical model of the CBSO algorithm is introduced in detail next.

First, CBSO initializes the entire population inside the search space by assigning each bank a position that is randomly sampled within the lower and upper bounds of every decision variable, as formalized in Equation (1).

$$X_{i,j} = lb_j + rand \times \left(ub_j - lb_j \right), \quad i = 1, 2, ..., N, \ j = 1, 2, ..., D \tag{1}$$

In Equation (1), $X_{i,j}$ represents the position of the i-th agent along the j-th decision variable, constrained by the lower bound lb and upper bound ub that define the feasible search interval. $rand$ is a random variable drawn from the uniform distribution U(0, 1), N denotes the population size, and D indicates the problem dimensionality.

As a meta-heuristic, CBSO calibrates its search trajectory by allocating the first 20 % of iterations to intensive global exploration (Stage 1), the final 60 % to focused local exploitation (Stage 3), and the intervening 20 % to a blended transition phase that concurrently conducts global and local refinement (Stage 2).

During the exploration phase (Phase 1), the CBSO algorithm generates new positions via Equation 2.

$$X_i^{t+1} = X_i^t + R_B \times \left(rand\left(1, D\right) \times X_b - X_i^t \right) \tag{2}$$

In Equation 2, R_B is sampled from the standard normal distribution N(0,1), X_b denotes the globally best agent discovered so far, and t represents the current iteration counter.

In Phase 2 (the transition phase), CBSO refreshes the population by applying two complementary strategies: half of the agents are allocated to exploration using Equation 3, and the other half to exploitation governed by Equation 4. Critically, assignment to either subpopulation is randomly redetermined at every generation.

$$X_i^{t+1} = X_i^t + R_L \times \left(X_b - R_L \times X_{r1}^t \right) \tag{3}$$

$$X_i^{t+1} = X_b + CF \times R_B \times \left(rand \times X_b - X_{r2}^t \right) \tag{4}$$

In Equation 3, R_L is a D-dimensional vector sampled from a Lévy distribution, X_{r1}^t and X_{r2}^t are two distinct agents selected uniformly at random from the current population, and CF is an adaptive coefficient whose value is updated via Equation 5, where T denotes the maximum number of iterations.

$$CF = rand \times \left(1 - t / T \right) \tag{5}$$

During the exploitation phase, CBSO refines the search within promising regions according to Equation 6.

$$X_i^{t+1} = X_b + CF \times R_L \times \left(R_L \times X_{r1}^t - X_{r2}^t \right) \tag{6}$$

Beyond the three core search phases, the CBSO algorithm augments its trajectory with an additional enhancement phase governed by Equation 7 to further intensify the search.

$$X_i^{t+1} = \begin{cases} X_i^t + rand \times \left(CF \times lb + rand(1, D) \times \left(ub - lb \right) \right), rand \le \delta \\ X_i^t + rand \times rand \times \left(X_{r1}^t - X_{r2}^t \right), others \end{cases} \tag{7}$$

In Equation 7, the constant δ is fixed at 0.2 in accordance with the source literature. The complete pseudocode for the CBSO algorithm is presented in Algorithm 1.

Algorithm 1 Connected banking system optimizer (CBSO)

Input: *lb, ub, D, N, T*
Initialize population randomly according to Equation 1
While ($t < T$) do
For $i = 1 : N$
If $t \le \frac{T}{5}$
Update the position using Equation 2 // ***Phase 1***
Else if $\frac{T}{5} < t \le \frac{3T}{5}$
If $i < \frac{N}{2}$
Update the position using Equation 3 // ***Phase 2***
Else
Update the position using Equation 4 // ***Phase 2***
End if
Else
Update the position using Equation 6 // ***Phase 3***
End if
Update the position using Equation 7 // ***Phase 4***
End for
$t = t + 1$
End while
Output: The best solution X_b

4. Proposed ECBSO

When tackling complex-landscape optimization, the original CBSO exhibits three critical shortcomings: (i) weak global search arising from insufficient inter-bank information exchange, (ii) a rigid phase transition that prevents the algorithm from dynamically calibrating the exploitation–exploration ratio, and (iii) premature stagnation caused by over-reliance on the single best agent. To remove these bottlenecks, the present study proposes the following remedies.

4.1. Dominant group guidance strategy

In Stage 1 and Stage 2 of CBSO, each agent is relocated by an individually-assigned step size. This scheme ignores inter-agent communication and relies solely on information exchange between the agent itself and either the random or the best individual, severely impairing global exploration and eroding population diversity. When confronted with rugged landscapes, the absence of vigorous global search frequently traps the algorithm in local optima. To remedy this drawback, we introduce the dominant group guidance strategy (DGS) to supplant the existing exploration mechanism. ABC constructs a covariance matrix of the dominant group and samples new agents from the resulting multivariate Gaussian. The covariance matrix aligns sampling along directions of strong variable correlation, pruning ineffective search, while the Gaussian model spans the entire space, preserving a broad distribution in early iterations to promote global exploration and reduce the risk of premature entrapment in local peaks. For a D-dimensional random vector, the joint Gaussian probability density function is given in Equation (8).

$$G(X)_{\mu,C} = \sqrt{\frac{1}{(2\times\pi)^{D}\times\det(C)}}\times e^{\left(-(X-\mu)^{T}\times C^{-1}\times(X-\mu)/2\right)} \tag{8}$$

$$\mu = \frac{1}{0.5N} \times \sum_{i=1}^{0.5N} X_i^t, X_i^t \in G_d \tag{9}$$

$$C = \frac{1}{0.5N} \times \sum_{i=1}^{0.5N} \left(X_i^t - \mu\right) \times \left(X_i^t - \mu\right)^T, X_i^t \in G_d \tag{10}$$

Once the Gaussian distribution of the elite set is obtained, new agents are generated by sampling from $N(0, C)$, as shown in Equation 11.

$$X_i^{t+1} = \mu + g_i, g_i \sim N\left(0, C\right) \tag{11}$$

In DGS, the cardinality of the dominant cohort used to build the covariance matrix is critical: too few agents yield an incomplete description of the distribution, hindering the algorithm's grasp of the search-space structure; too many dilute the influence of high-quality individuals and blunt the guidance they provide. To address this, an external archive Ar is introduced to store the best agents from each generation; its size is capped by a first-in-first-out rule, and the optimal capacity is determined empirically during the tuning phase.

The DGS captures the evolutionary trend of the population with high fidelity, eliminating ineffective search moves. The guidance provided by the dominant cohort continuously injects high-quality offspring into the swarm. Meanwhile, the external archive filters out the lingering influence of outdated individuals, accelerating convergence without eroding diversity. Collectively, this mechanism strengthens ECBSO's global exploration capacity and enriches population diversity.

4.2. Guided learning strategy

The CBSO algorithm partitions the search into rigid phases by iteration count, a design that cannot adapt to the heterogeneous demands of different optimization problems. Because the required ratio of exploration to exploitation varies across tasks, a meta-heuristic must be able to diagnose the current search state in real time. The original CBSO lacks this diagnostic capability. To fill the gap, we introduce a guided learning strategy (GLS) that quantifies population dispersion by computing, for each agent, the standard deviation of its positions over the last few generations. When the dispersion measure indicates an exploration bias, ABC redirects the search toward exploitation; conversely, it steers the swarm back to exploration. The detailed formulation of GLS is presented as follows.

$$X_i^{t+1} = \begin{cases} X_b + \tan\left(rand \times \pi\right) \times \left(ub - lb\right) / V_0, if\ \ V_0 > a \\ rand \times \left(ub - lb\right), if\ \ V_0 \le a \end{cases} \tag{12}$$

$$V_0 = std\left(St\right) \times B \tag{13}$$

$$B = 200 / \left(lb - ub\right) \tag{14}$$

where $std\left(\cdot\right)$ denotes the function that computes the standard deviation. St serves as the learning memory, retaining the most recent S agents from historical iterations; the exact number of stored individuals will be determined experimentally. B normalizes V_0 to insulate it from variations in variable bounds. a is a threshold parameter that dictates whether exploitation or exploration is currently required, and its value will also be established during the experimental phase.

The GLS equips ECBSO with a robust mechanism for balancing exploration and exploitation. By continuously computing the population's displacement statistics across recent generations, GLS infers the current search demand in real time. Overall, this mechanism enhances ECBSO's adaptability, enabling it to switch between exploitation and exploration across diverse and complex landscapes.

4.3. Hybrid elite strategy

During CBSO's exploitation phase, the swarm updates positions by referencing a single best agent; although this expedites convergence, it also heightens the risk of entrapment in local optima. To counteract this drawback, we introduce the hybrid elite strategy (HES), which hybridizes an Equilibrium Optimizer (EO) into CBSO, striking a balance between rapid convergence and the avoidance of premature stagnation. The EO algorithm is mathematically modeled by the following equations.

$$X_i^{t+1} = X_{ep} + (X_i^t - X_{eq}) \times F + \frac{G}{rand}(1-F)$$	(15)
$$X_{ep} = \left(X_b, X_{second}, X_{third}, X_{fourth}, X_{eave}\right)$$	(16)
$$F = w_1 \times sign(rand - 0.5) \times \left(e^{-rand \times (1-t/T)^{(w_2 \times t/T)}} - 1\right)$$	(17)
$$G = \begin{cases} 0.5 \times r_1 \times \left(X_{ep} - rand \times X_i^t\right), r_2 > 0.5 \\ 0, r_2 \le 0.5 \end{cases}$$	(18)

where X_{ep} denotes an agent randomly selected from the elite pool, while X_{second}, X_{third}, X_{fourth} and X_{eave} represent, respectively, the second-, third-, and fourth-ranked agents by fitness and the centroid of the top four agents within the elite pool. w_1, w_2, r_1, and r_2 are independent random numbers uniformly distributed in [0, 1].

The hybrid scheme adopted in this paper is governed solely by each agent's fitness rank: every agent whose rank exceeds 0.5N is updated by the EO operator, whereas the remainder continue to be refined by the CBSO and DGS operators. Agents that rank highly benefit from the EO operator because the embedded elite pool provides both an escape route from local optima and an expanded search scope. Conversely, lower-ranked agents exploit the CBSO and DGS operators, which inject additional diversity and elevate overall population quality. Collectively, this hybridization equips ECBSO with a stronger capacity to evade local optima while preserving rapid convergence.

4.4. Implementation steps for ECBSO

By integrating the three proposed mechanisms into CBSO, we obtain the ECBSO algorithm. ECBSO comprises four sequential components: population initialization, the EO phase, the CBSO phase augmented with DGS, and the GLS phase. The flowchart of ECBSO is depicted in Figure 1, and its pseudocode is provided in Algorithm 2.

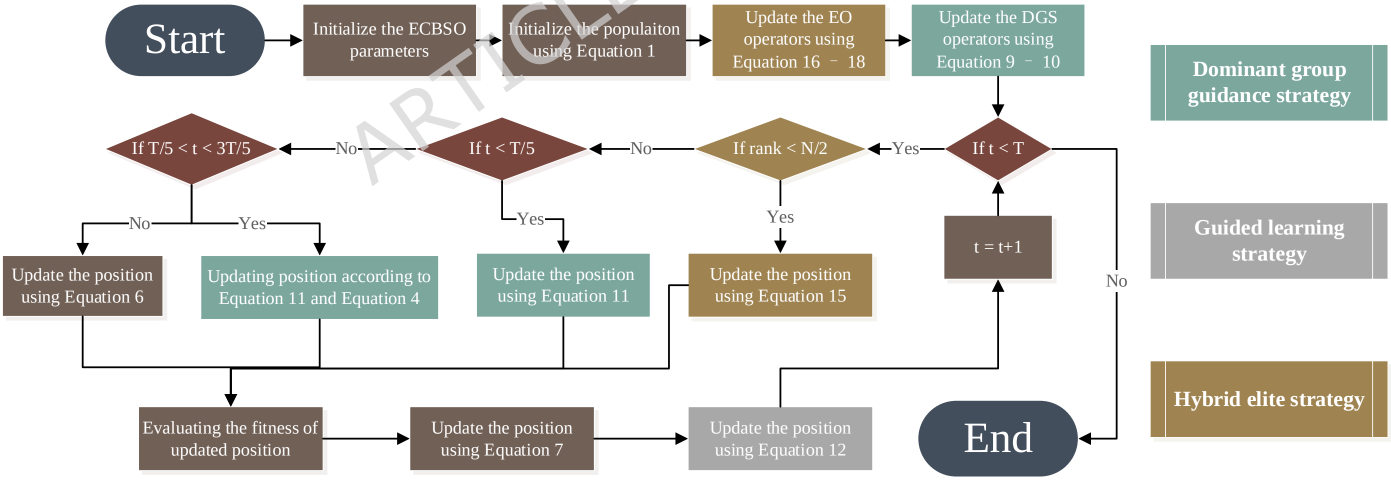


Figure 1. The flowchart of ECBSO algorithm

Algorithm 2 Enhanced connected banking system optimizer (ECBSO)
Input: *lb, ub, D, N, T*
Initialize population randomly according to Equation 1
While ($t < T$) do
Update the EO operators using Equation 16 – 18 // *****HES*****
Update the DGS operators using Equation 9 – 10 //*** **DGS*****
For i =1 : N

If $rank < \frac{N}{2}$
Update the position using Equation 15 // ***HES***
Else
If $t \le \frac{T}{5}$
Update the position using Equation 11 // ***DGS***
Else if $\frac{T}{5} < t \le \frac{3T}{5}$
If $i < \frac{N}{2}$
Update the position using Equation 11 // ***DGS***
Else
Update the position using Equation 4
End if
Else
Update the position using Equation 6
End if
End if
Update the position using Equation 7
Update the position using Equation 12 // ***GLS***
End for
$t = t + 1$
End while
Output: The best solution X_b

4.5. Analysis of ECBSO complexity

The computational complexity of the basic CBSO and proposed ECBAO is analyzed based on the main influencing factors: the population size N, the dimensionality of the problem D, and the total number of iterations T. The analysis considers the costs associated with initialization, fitness evaluation, and position updates throughout the optimization process.

For the CBSO algorithm, the initial population is generated randomly, requiring a complexity of $O(N \times D)$. At every iteration, each agent executes exactly one of the three main phases plus the supplementary stage, so the positional update incurs an $O(T \times 2N \times D)$ time complexity. Since each update is followed by a fitness evaluation and each agent is evaluated twice per iteration, the fitness-computation cost per generation is $O(2 \times N \times \log(N))$. As this process repeats for T iterations, the total time complexity of basic CBSO is $O(2 \times T \times N \times (D + \log(N)))$.

For the ECBSO algorithm, the population initialization remains unchanged, so its time complexity is identical to that of the original CBSO algorithm. DGS replaces the exploration components of Phases 1 and 2; hence, no additional time complexity is introduced. GLS is an additional update stage yet is invoked only when triggered; if it is executed T_1 times, its total time complexity is $O(T_1 \times N \times D)$. During every update, each agent randomly selects one operator from HES, DGS, or the original CBSO strategy, so HES does not introduce additional position updates. Consequently, the overall time complexity of ECBSO remains $O((2T + T_1) \times N \times (D + \log(N)))$.

Overall, the time complexity of ECBSO is higher than that of the basic CBSO, which implies that, for the same number of iterations, ECBSO consumes more function evaluations. To guarantee a fair comparison, we therefore adopt the maximum number of function evaluations as the unified termination criterion.

5. Results and analysis on benchmark functions

This section gives a thorough analysis of the ECBSO algorithm by describing the experimental setup as well as examining its performance in comparison to the other optimization techniques. The analysis consists of extensive experimentation on the CEC2017 benchmark functions. Convergence dynamics of ECBSO, as well as the distribution of performance depicted by boxplots, are also discussed in the section. Further, comparative experiments against advanced improved algorithms are also provided to demonstrate the efficacy of ECBSO in handling complicated optimization landscapes.

5.1. Experimental configuration and comparative algorithms

The widely-adopted IEEE CEC 2017 benchmark suite is employed to assess the search capability of ECBSO. This collection comprises 29 functions—three unimodal (F1–F3), seven multimodal (F4–F10), ten hybrid (F11–F20), and nine composite functions (F21–F29)—all defined over the domain [−100, 100]. Unimodal functions, despite their simplicity, are crucial for gauging an algorithm's convergence toward the global optimum. Multimodal functions, featuring numerous peaks and valleys, evaluate the capacity to locate the global optimum while escaping local attractors. Hybrid and composite functions, owing to their intricate structures, provide a comprehensive assessment of overall algorithmic performance. Details of the CEC 2017 suite are summarized in Table 1.

Table 1. Platform specifications utilized for experiments

Type	No.	Functions name	Min
Unimodal functions	F1	Shifted and Rotated Bent Cigar Function	100
	F3	Shifted and Rotated Zakharov Function	300
Multimodal functions	F4	Shifted and Rotated Rosenbrock's Function	400
	F5	Shifted and Rotated Rastrigin's Function	500
	F6	Shifted and Rotated Expanded Scaffer's F6 Function	600
	F7	Shifted and Rotated Lunacek Bi_Rastrigin Function	700
	F8	Shifted and Rotated Non-Continuous Rastrigin's Function	800
	F9	Shifted and Rotated Levy Function	900
	F10	Shifted and Rotated Schwefel's Function	1000
Hybrid functions	F11	Hybrid Function 1 (N=3)	1100
	F12	Hybrid Function 2 (N=3)	1200
	F13	Hybrid Function 3 (N=3)	1300
	F14	Hybrid Function 4 (N=4)	1400
	F15	Hybrid Function 5 (N=4)	1500
	F16	Hybrid Function 6 (N=4)	1600
	F17	Hybrid Function 6 (N=5)	1700
	F18	Hybrid Function 6 (N=5)	1800
	F19	Hybrid Function 6 (N=5)	1900
	F20	Hybrid Function 6 (N=6)	2000
Composition functions	F21	Composition Function 1 (N=3)	2100
	F22	Composition Function 2 (N=3)	2200
	F23	Composition Function 3 (N=4)	2300
	F24	Composition Function 4 (N=4)	2400
	F25	Composition Function 5 (N=5)	2500
	F26	Composition Function 6 (N=5)	2600
	F27	Composition Function 7 (N=6)	2700
	F28	Composition Function 8 (N=6)	2800
	F29	Composition Function 9 (N=3)	2900

F30	Composition Function 10 (N=3)	3000

Experiments were conducted on the CEC2017 benchmark at dimensions 10D, 30D, 50D, and 100D. The termination criterion was a maximum of 3000 × D function evaluations, and each run was repeated 30 times independently to ensure statistical reliability. Population sizes were set according to the recommendations in the source literature of the comparative algorithms. All performances were conducted on a computing system composed of a 2.50 GHz AMD R9 7945HX CPU, 32 GB RAM, and Windows 11 (64-bit), and all codes were written in MATLAB R2023a.

To fully demonstrate the superiority of the proposed ECBSO algorithm, this paper selected various types of improved algorithms as competitors, including: human-based improved algorithms: ISGTOA [54], EMTLBO [55]; evolutionary-based improved algorithms: LSHADE [56], APSM-jSO [57]; swarm-based improved algorithms: GLS-MPA [58], ESLPSO [59]; and physics-based improved algorithms: ACGRIME [60], RDGMVO [61]. To ensure that the competing algorithms perform at their best, their parameters were set to the values recommended in the original sources, as listed in Table 2. These comparative algorithms have demonstrated superior performance in their respective literature. ISGTOA is a newly proposed improved version of the human-based algorithm. EMTLBO is a variant of the classic human-based algorithm TLBO. LSHADE is a well-known variant of DE. APSM-jSO holds its own against advanced DE variants. GLS-MPA is a variant of the highly cited MPA algorithm. ESLPSO represents an enhanced version of the classic PSO. ACGRIME and RDGMVO are modified versions of physics-based algorithms.

Table 2. Parameter settings of ECBSO and other algorithms

Algorithm setting	
ECBSO	$N = 30D, \delta = 0.2, a = 10, S = 3000, Ar = 30D$
CBSO	$N = 150, \delta = 0.2$
ISGTOA	$N = 50, \lambda = 2$
EMTLBO	$N = 50, s = 0, p = 0$
LSHADE	$N = 18D, H = 6, F = 0.5, CR = 0.5$
APSM-jSO	$N = ceil\left(75 \times D^{2/3}\right), k = 3, F = 0.3, cr = 0.8, h = 6, a = 1.3$
GLS-MPA	$N = 30, FADs = 0.2, P = 0.5, Cmax = 250, \alpha = 30$
ESLPSO	$N = 30, CR = 0.5, F = 0.5, c = 0.1, M = N$
ACGRIME	$N = 30, a = 4, w = 5$
RDGMVO	$N = 30, w_{max} = 1, w_{min} = 0.2, p = 0.6$

5.2. Metrics for evaluating optimization performance

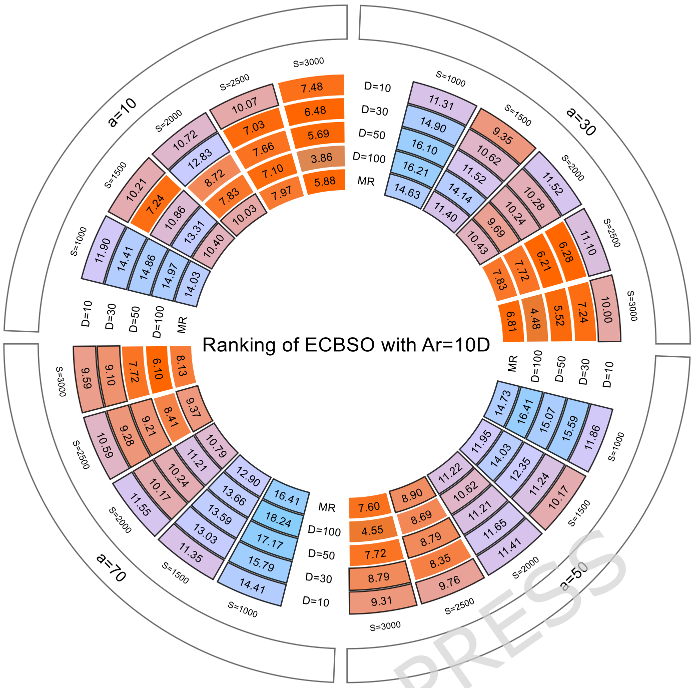
To impartially benchmark the ECBSO algorithm against its competitors, five complementary metrics are employed: the best value (Min), the average solution quality (Ave), the standard deviation (Std), the Friedman ranking test (FR), and the Wilcoxon rank-sum test (WRST). Together, these indicators quantify consistency, reliability, and overall superiority of the optimizer. Min gauges an algorithm's ability to locate the best possible solution, thereby revealing its maximum exploitation potential. Ave offers a pragmatic measure of overall correctness and performance across repeated runs under identical settings. Std quantifies the dispersion of fitness values around the mean, indicating the stability and consistency of the results. WRST performs pairwise comparisons between two algorithms across all test functions to establish statistical superiority. FR assesses whether significant global differences exist among all algorithms in the experimental set. When the Friedman test indicates such differences, the Nemenyi post-hoc procedure is applied, and critical-difference diagrams are used to pinpoint specific algorithmic distinctions. All statistical tests adopt a significance level of $\alpha = 0.05$ to ensure reliable conclusions. Together, these metrics provide a solid foundation for a comprehensive comparison of ECBSO against its competitors and guarantee that every finding regarding its optimization performance is statistically consistent and reflects its true potential.

5.3. Parameter sensitivity analysis

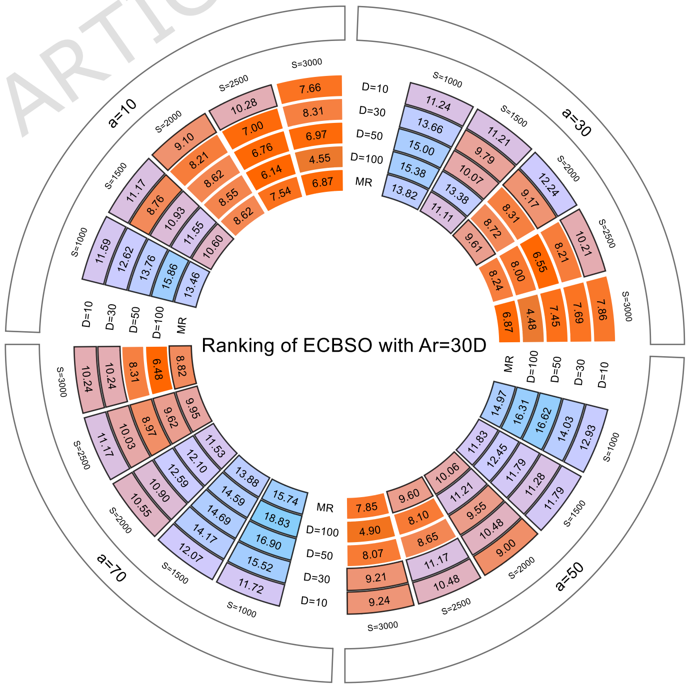
As a meta-heuristic variant, ECBSO's performance is inevitably sensitive to its parameter settings. Based on the design details in Section 3, DGS requires the capacity Ar of the external archive, whereas GLS needs the triggering threshold a and the memory size S. Ar governs the sample size used to build the covariance matrix, directly affecting the quality of offspring generated by DGS. Parameters a and S jointly determine how often GLS is activated and how exploitation and exploration are re-balanced on-the-fly. These three parameters therefore dominate the search behavior of ECBSO.

To identify their optimal values, a grid-based sensitivity study is conducted. The search grids are set as follows. Ar is varied from 10D to 70D in steps of 20D, a is swept from 10 to 70 in steps of 20, and S is incremented from 1000 to 3000 in steps of

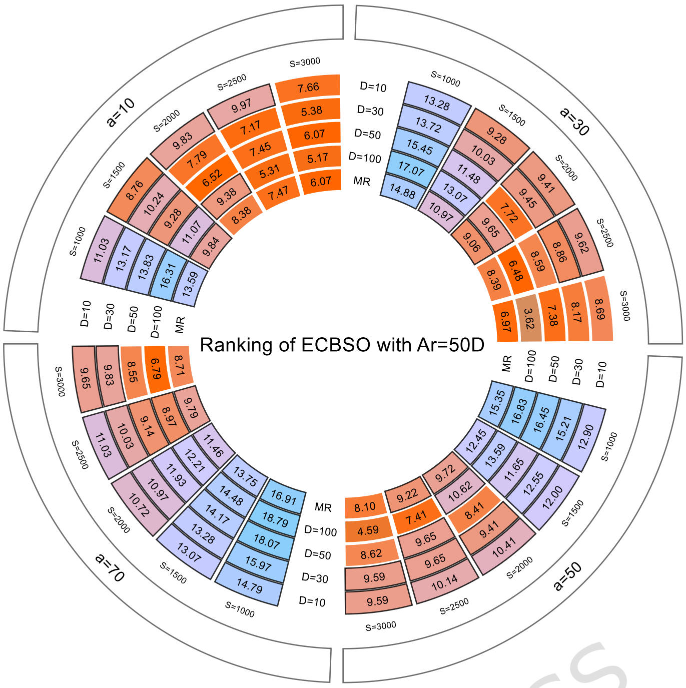
500, yielding a total of 4 × 4 × 5 = 80 distinct parameter configurations. To simplify the comparison, we first determine the best (a , S) pair under each Ar , and then compare the four resulting Ar -specific configurations to select the overall best setting. The Friedman ranks of ECBSO for each parameter combination across different Ar values are illustrated in Figure 2.



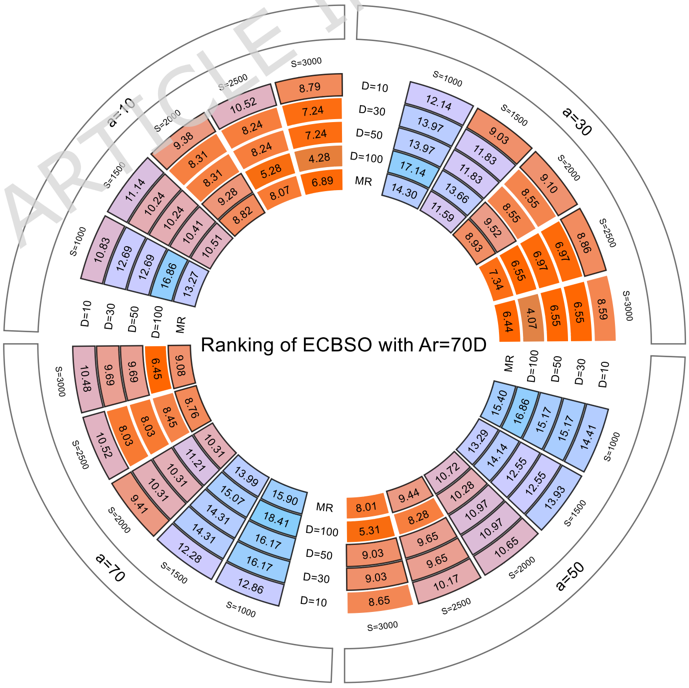
(a) Ar=10D



(b) Ar=30D



(c) Ar=50D



(d) Ar=100D

Figure 2. Friedman ranking of ECBSO with different Ar

Figure 2 reports the mean rank (MR) for each configuration. Across all values of a, larger S consistently yields better ranks, indicating that GLS need not be invoked frequently—excessive calls merely consume function evaluations without commensurate benefit. Conversely, for any fixed S, smaller a improves the rank, revealing that ECBSO currently benefits from a stronger exploratory bias. Specifically, when Ar = 10 D, the pairing (a = 10, S = 3000) attains the best Friedman rank of 5.879 and ranks first on 10 D, 50 D, and 100 D functions. When Ar = 30 D, both (a = 10, S = 3000) and (a = 30, S = 3000) achieve the top Friedman rank of 6.871, yet the latter exhibits a smaller rank standard deviation and is therefore preferred. When Ar = 50 D, (a = 10, S = 3000) secures the best Friedman rank of 6.609 and leads on 10 D, 30 D, and 50 D instances. Finally, for Ar = 70 D, the configuration (a = 30, S = 3000) delivers the best Friedman rank of 6.440 and ranks first across all four dimensionalities. Collectively, ECBSO performs best with a = 10 or 30 and S = 3000, regardless of the chosen Ar. This indicates that GLS does not need to be triggered frequently; infrequent activation biased toward exploration is sufficient. The underlying reason is that the baseline CBSO lacks a dedicated local-optimum escape mechanism, so GLS must provide occasional exploratory thrusts to help the swarm jump out of basins. Meanwhile, DGS already strengthens global exploration, further reducing the need for frequent GLS calls. In the next step, we compare the four best-performing parameter combinations identified above to finalize the optimal settings.

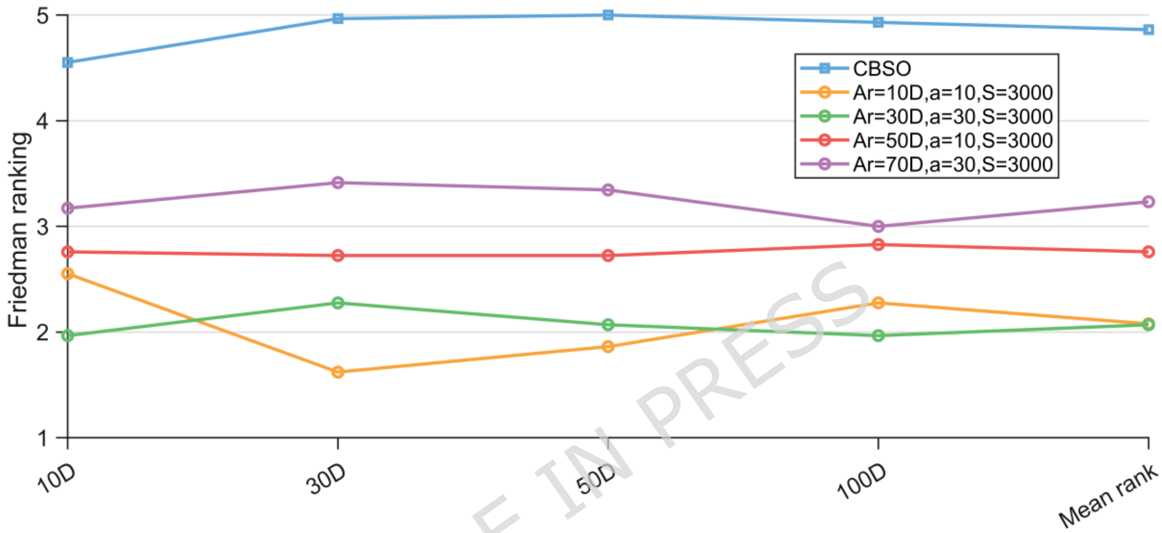


Figure 3. Friedman ranking of ECBSO with different parameters

Figure 3 presents the Friedman ranks for ECBSO with the four best parameter combinations. Evidently, the pairs (Ar = 30 D, a = 30, S = 3000) and (Ar = 10 D, a = 10, S = 3000) outperform the others, and they trade the top position across different dimensions. Specifically, the configuration (Ar = 30 D, a = 30, S = 3000) achieves a marginally superior rank of 2.069 compared with 2.078 for (Ar = 10 D, a = 10, S = 3000). Therefore, all subsequent experiments adopt the parameter set (Ar = 30 D, a = 30, S = 3000).

5.4. Ablation experiments analysis

This paper refactors CBSO through three mutually reinforcing mechanisms. To quantify the individual and joint contributions of each component, an ablation study is conducted. Starting from the basic CBSO, we derive six partial configurations by activating one or two mechanisms at a time: CBSO-D, CBSO-G, CBSO-H, CBSO-DG, CBSO-DH, and CBSO-GH (Table 3). In the table, "Y" marks the presence of a strategy, whereas "N" indicates its absence; for instance, CBSO-D employs DGS only, whereas CBSO-GH omits DGS. All variants, together with the full ECBSO and the original CBSO, are compared on the complete CEC 2017 test suite; detailed numerical results are consolidated in Tables A1–A4 of Appendix A. The subsequent analysis employs Friedman's test, Wilcoxon's rank-sum test, and the Nemenyi post-hoc procedure to establish statistical significance.

Table 3. Details of ECBSO variants with different strategies

Algorithm	CBSO-D	CBSO-G	CBSO-H	CBSO-DG	CBSO-DH	CBSO-GH	ECBSO
DGS	Y	N	N	Y	Y	N	Y
GLS	N	Y	N	Y	N	Y	Y
HES	N	N	Y	N	Y	Y	Y

Table 4. Friedman test results obtained by ECBSO variants with different strategies

Test suite	Dimension	CBSO	CBSO-D	CBSO-G	CBSO-H	CBSO-DG	CBSO-DH	CBSO-GH	ECBSO	p-value

CEC 2017	10	6.379	4.207	5.828	5.276	3.621	3.724	4.414	2.552	2.41E-09
	30	7.862	4.103	6.414	5.241	3.207	2.655	4.448	2.069	1.09E-24
	50	7.966	4.448	6.621	4.793	3.655	2.690	4.241	1.586	3.35E-27
	100	8.000	5.103	6.655	4.241	4.069	2.621	3.724	1.586	3.32E-28
Mean rank		7.552	4.466	6.379	4.888	3.638	2.922	4.207	1.948	N/A
Overall rank		8	5	7	6	3	2	4	1	N/A

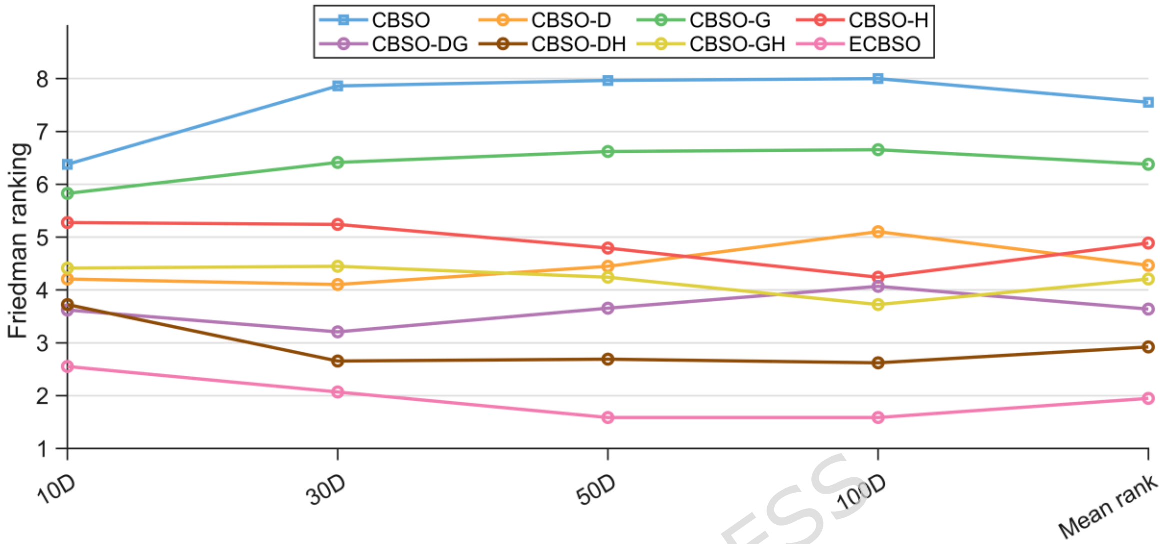


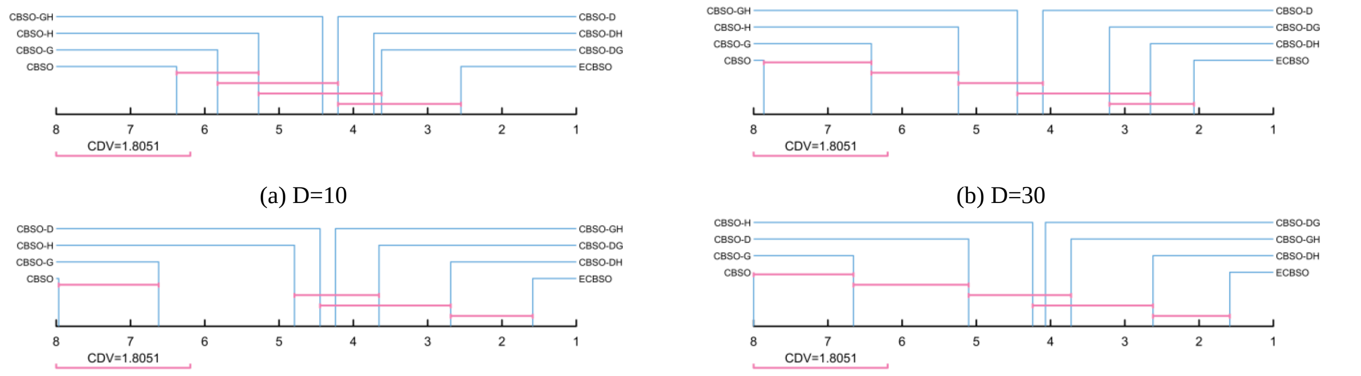
Figure 4. Friedman ranking of ECBSO variants with different strategies

Table 4 summarizes the Friedman ranks for ECBSO and its six ablated variants together with the original CBSO; the corresponding visual ranking is provided in Figure 4. The following conclusions can be drawn: ECBSO, which integrates all three enhancements, achieves the best overall performance, indicating that the strategies act synergistically to boost the search capability of CBSO. The three derived algorithms that combine two strategies (CBSO-DG, CBSO-DH, CBSO-GH) consistently outperform those equipped with only one strategy (CBSO-D, CBSO-G, CBSO-H), demonstrating that pairwise combinations already strengthen CBSO. Every single-strategy variant is superior to the baseline CBSO, confirming that each mechanism contributes positively to the algorithm's performance.

Table 5. Wilcoxon rank sum test results obtained by ECBSO variants with different strategies

vs. CBSO +/=/-	**CEC 2017 test suite**			
	10D	30D	50D	100D
CBSO-D	18/8/3	27/2/0	26/3/0	29/0/0
CBSO-G	8/20/1	24/5/0	24/5/0	27/2/0
CBSO-H	13/15/1	23/4/2	26/3/0	27/2/0
CBSO-DG	19/7/3	27/1/1	28/1/0	29/0/0
CBSO-DH	19/8/2	29/0/0	29/0/0	29/0/0
CBSO-GH	15/13/1	27/2/0	27/2/0	29/0/0
ECBSO	21/6/2	29/0/0	29/0/0	29/0/0

Table 5 summarizes the Wilcoxon rank-sum test results, where "+" denotes the number of functions on which ECBSO or its derived variants significantly outperform the baseline CBSO, "–" indicates the number of functions where they are significantly worse, and "=" counts the functions for which no significant difference is detected. The outcomes show that ECBSO and all two-strategy variants achieve significantly better results than CBSO on more than half of the test functions. Single-strategy variants also accumulate more "+" counts than "–" counts. These consistent patterns provide strong statistical evidence for the effectiveness of every proposed enhancement.



(a) D=10 (b) D=30

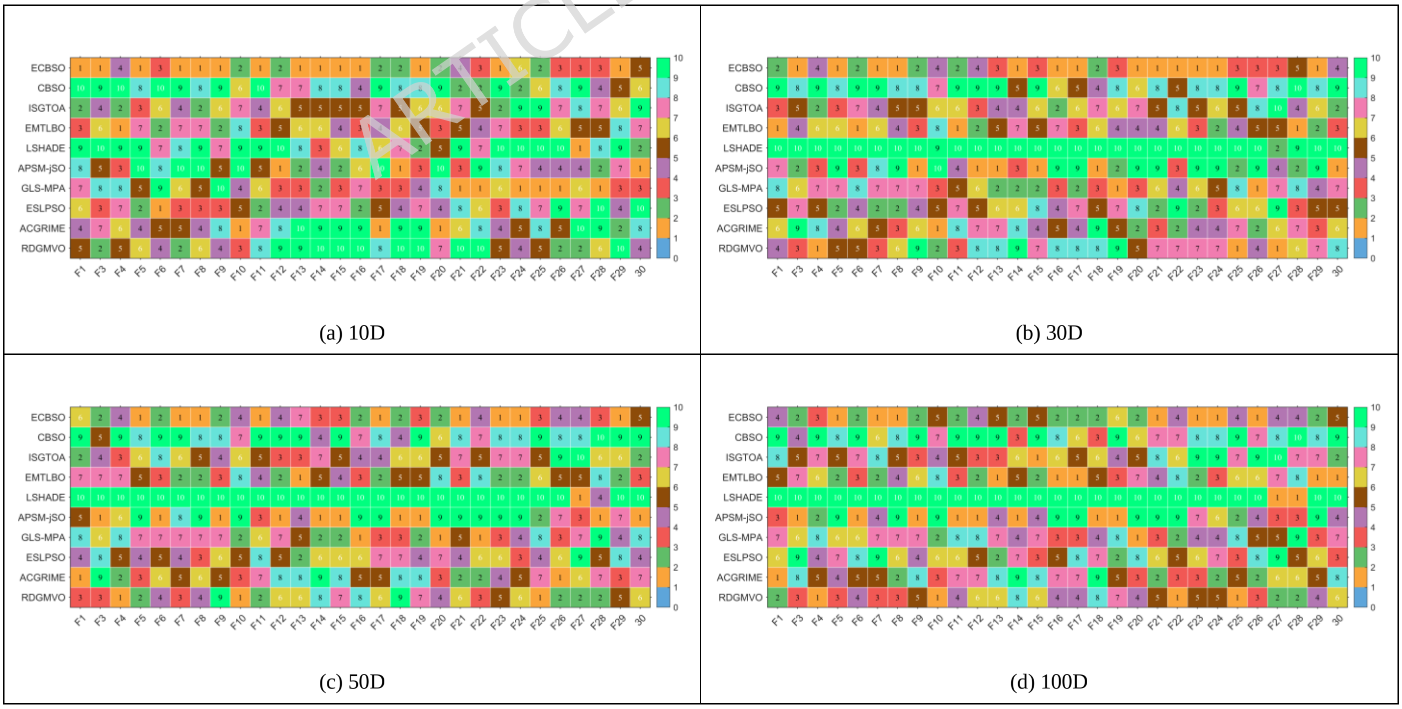
(c) D=50 (d) D=100

Figure 5. Nemenyi post hoc test of ECBSO variants with different strategies

The Nemenyi post-hoc test quantifies the magnitude of pairwise differences revealed by the Friedman test. By computing the critical-difference value (CDV) and connecting algorithms whose rank gap is smaller than the CDV, the test provides a visual measure of significance. Figure 5 shows the Nemenyi diagrams for ECBSO, its ablated variants, and the baseline CBSO across 10 D, 30 D, 50 D and 100 D. Two observations are immediate: ECBSO is never statistically separated from CBSO-DH in any dimension, indicating that the additional GLS module yields only a marginal improvement. CBSO-G is indistinguishable from plain CBSO under every setting, confirming that GLS alone contributes little to overall performance. Conversely, the remaining two-strategy variants (CBSO-DG and CBSO-DH) are significantly different from the full ECBSO in most cases, implying that removing either DGS or HES visibly degrades the algorithm.

Overall, all three enhancements are statistically beneficial, but their individual impact varies: DGS provides the largest contribution, HES ranks second, and GLS offers the smallest yet still measurable gain.

5.5. Comparison experiments analysis



(a) 10D (b) 30D

(c) 50D (d) 100D

Figure 6. The ranking heatmaps based on "Ave" of ECBSO and comparison algorithms

This section demonstrates the superiority of the proposed ECBSO algorithm by comparing it with other advanced variants. The complete experimental results are presented in Tables A5–A8 of the Appendix. First, the heat-map in Figure 6 visualizes the ranking of ECBSO and the competing algorithms on every function. ECBSO secures a top-three position on 25 functions in 10-D, 24 in 30-D, 20 in 50-D and 18 in 100-D, providing an initial indication of its consistently strong performance across dimensionalities.

Statistical validation is a cornerstone of the experimental analysis, offering objective evidence of superiority that goes beyond raw performance metrics. The inherent stochasticity of meta-heuristics demands rigorous tests to establish whether observed differences are systematic or merely random. We therefore apply two complementary non-parametric procedures: the Friedman test for global ranking and the Wilcoxon rank-sum test for pairwise comparisons. Acting as a non-parametric alternative to repeated-measures ANOVA, the Friedman test determines whether significant global differences exist among all algorithms across the benchmark set while respecting the paired nature of the results. Subsequently, the Wilcoxon rank-sum test supplies detailed pairwise comparisons between ECBSO and every competitor, assessing statistical significance without assuming normality. Finally, the Nemenyi post-hoc procedure quantifies the magnitude of the differences revealed by the Friedman test, completing a statistically coherent evaluation chain.

Table 6. Friedman test results obtained by ECBSO and comparison algorithms

Test suite	Dimension	ECBSO	CBSO	ISGTOA	EMTLBO	LSHADE	APSM-jSO	GLS-MPA	ESLPSO	ACGRIME	RDGMVO	p-value
CEC-2017	10	2.069	7.241	5.483	4.862	7.483	5.586	4.414	5.207	6.241	6.414	2.42E-11
	30	2.138	7.759	5.172	4.069	9.655	5.138	5.000	5.138	5.207	5.724	1.17E-20
	50	2.690	7.931	5.379	4.379	9.483	5.000	4.931	5.414	5.276	4.517	4.00E-18
	100	2.759	7.621	5.862	4.103	9.379	4.655	5.483	5.862	5.276	4.000	5.78E-18
Mean rank		2.414	7.638	5.474	4.353	9.000	5.095	4.957	5.405	5.500	5.164	N/A
Overall rank		1	9	7	2	10	4	3	6	8	5	N/A

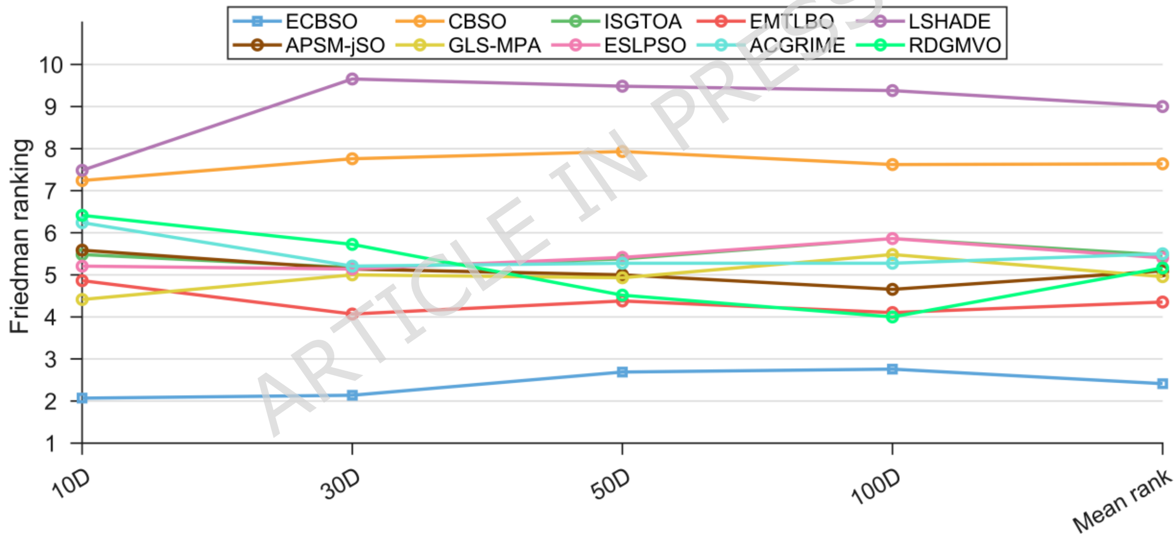


Figure 7. Friedman ranking of ECBSO and comparison algorithms

Table 6 reports the Friedman ranks of ECBSO and its competitors, with the corresponding scores visualised in Figure 7. ECBSO attains the best overall position with an average rank of 2.414, markedly outperforming the original CBSO (rank 7.638, 9th place). This large gap demonstrates that ECBSO consistently delivers superior results regardless of dimension or function type. Figure 7 also shows that all algorithms scale reasonably well—performance remains stable as dimension increases—yet the superiority of ECBSO is evident: CBSO and LSHADE are visibly separated from the top performer, while the remaining contenders cluster tightly in the middle of the ranking axis.

Table 7. Wilcoxon rank sum test results obtained by ECBSO and comparison algorithms

ECBSO vs.	CEC-2017 test suite			
+/=/-	10D	30D	50D	100D
CBSO	26/2/1	29/0/0	29/0/0	29/0/0
ISGTOA	17/11/1	21/6/2	22/3/4	21/4/4
EMTLBO	20/7/2	14/13/2	21/5/3	19/6/4
LSHADE	27/0/2	28/0/1	27/1/1	27/0/2
APSM-jSO	23/6/0	16/5/8	14/3/12	15/1/13
GLS-MPA	20/5/4	22/3/4	19/7/3	23/5/1
ESLPSO	17/11/1	21/8/0	23/3/3	19/5/5

ACGRIME	20/9/0	22/6/1	22/4/3	23/4/2
RDGMVO	22/4/3	21/5/3	17/6/6	21/1/7

The Wilcoxon rank-sum results, summarized in Table 7, furnish unequivocal statistical confirmation of ECBSO's pairwise superiority over every competitor. In every head-to-head comparison, the number of functions on which ECBSO significantly outperforms the rival ("+") exceeds those on which it is significantly outperformed ("–"). A detailed Wilcoxon rank-sum analysis is provided below.

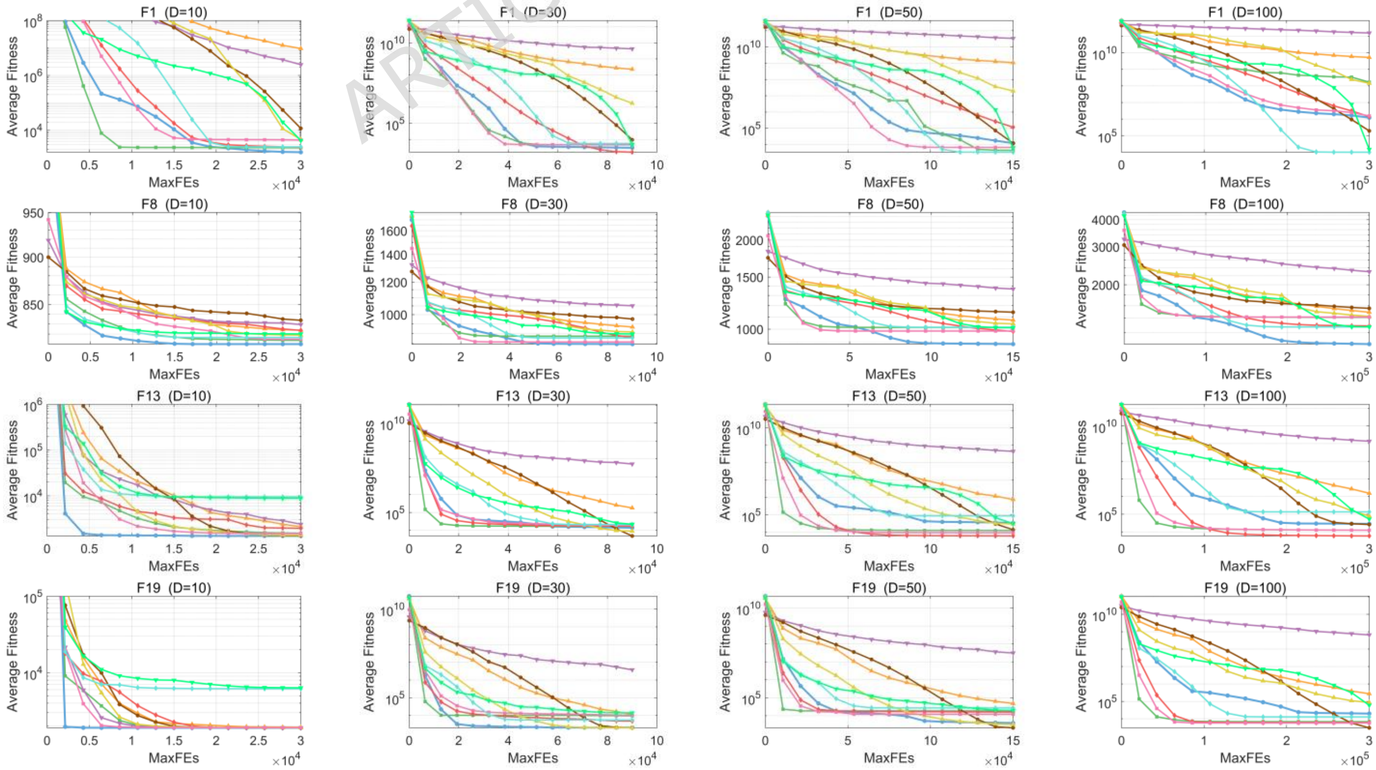
For 10D functions, ECBSO is superior/similar/inferior (+/=/-) to CBSO, ISGTOA, EMTLBO, LSHADE, APSM-jSO, GLS-MPA, ESLPSO, ACGRIME, RDGMVO on 26/2/1, 17/11/1, 20/7/2, 27/0/2, 23/6/0, 20/5/4, 17/11/1, 20/9/0, and 22/4/3 test functions. In other words, when compared against every competing algorithm, ECBSO achieves statistically significant superiority on at least 17 out of the 29 test functions.

For 30D functions, ECBSO is superior/similar/inferior (+/=/-) to CBSO, ISGTOA, EMTLBO, LSHADE, APSM-jSO, GLS-MPA, ESLPSO, ACGRIME, RDGMVO on 29/0/0, 21/6/2, 14/13/2, 28/0/1, 16/5/8, 22/3/4, 21/8/0, 22/6/1, and 21/5/3 test functions. In other words, when compared against every competing algorithm, ECBSO achieves statistically significant superiority on at least 14 out of the 29 test functions.

For 50D functions, ECBSO is superior/similar/inferior (+/=/-) to CBSO, ISGTOA, EMTLBO, LSHADE, APSM-jSO, GLS-MPA, ESLPSO, ACGRIME, RDGMVO on 29/0/0, 22/3/4, 21/5/3, 27/1/1, 14/3/12, 19/7/3, 23/3/3, 22/4/3, and 17/6/6 test functions. In other words, when compared against every competing algorithm, ECBSO achieves statistically significant superiority on at least 14 out of the 29 test functions.

For 100D functions, ECBSO is superior/similar/inferior (+/=/-) to CBSO, ISGTOA, EMTLBO, LSHADE, APSM-jSO, GLS-MPA, ESLPSO, ACGRIME, RDGMVO on 29/0/0, 21/4/4, 19/6/4, 27/0/2, 15/1/13, 23/5/1, 19/5/5, 23/4/2, and 21/1/7 test functions. In other words, when compared against every competing algorithm, ECBSO achieves statistically significant superiority on at least 15 out of the 29 test functions.

This overwhelming and consistent pattern demonstrates that the observed gains are not confined to specific problem types but constitute a systematic improvement across the entire test bed. ECBSO never exhibits a statistically significant disadvantage, providing strong evidence that the enhancements are genuine algorithmic advances rather than random fluctuations. The overwhelming preponderance of "+" outcomes establishes that these differences are significant at conventional confidence levels, cementing ECBSO's position as the most reliable and effective optimizer among all methods evaluated and furnishing a solid statistical foundation for its deployment across diverse optimization domains.



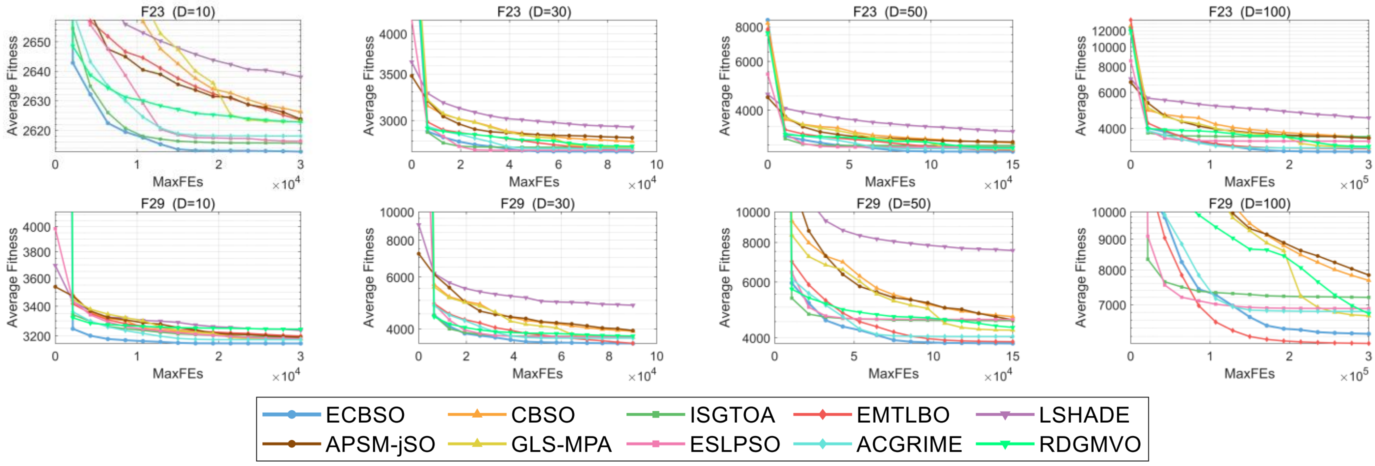
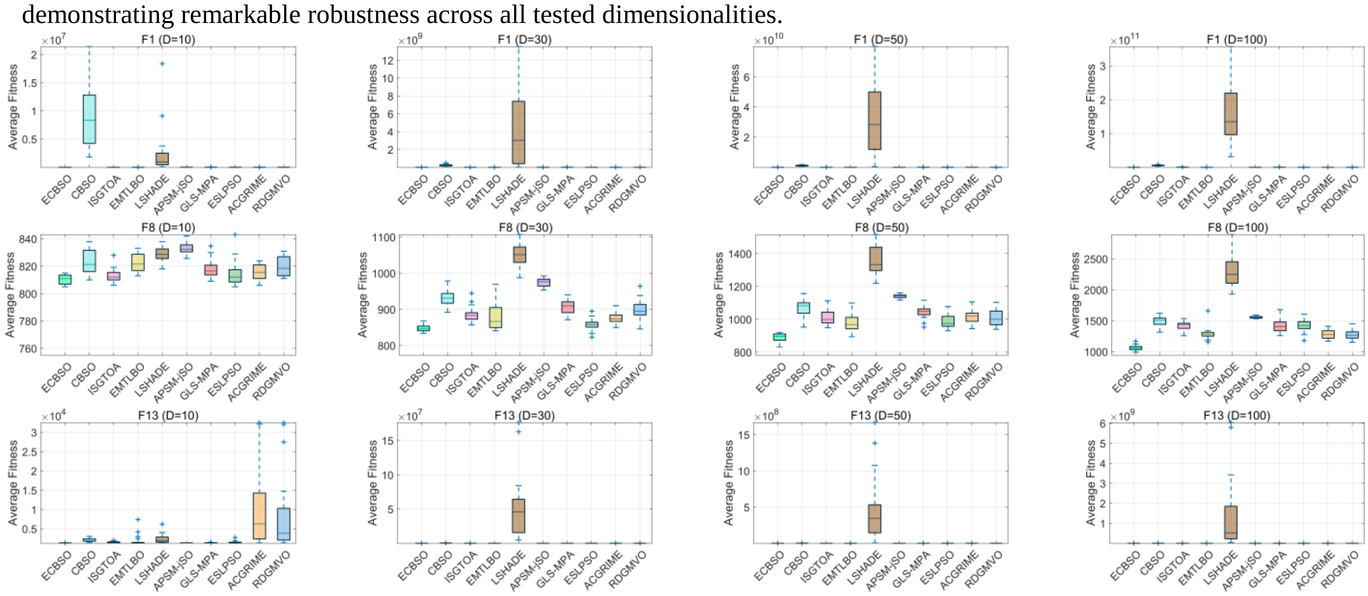


Figure 8. Convergence curves of ECBSO and comparison algorithms

Figure 8 illustrates the convergence behavior of ECBSO and the competing algorithms on six representative CEC-2017 functions: unimodal F1, multimodal F8, hybrid F13 & F19, and composite F23 & F29. When solving unimodal function F1, ECBSO attains the best final accuracy at 10 D; although its initial slope is not the steepest, the curve keeps a steady downward trend. This sustained refinement is credited to the covariance-guided sampling of DGS and the multi-elite refinement of HES. When solving multimodal function F8, ECBSO converges smoothly without being trapped in local optima, owing to the exploratory bias injected by GLS and the periodically refreshed search directions provided by the EO-based HES module. On hybrid and composite landscapes, ECBSO maintains a rapid early convergence slope, sustains a steady mid-term deepening search, and exhibits superior late-term exploitation. This behavior stems from the synergistic interplay of the three strategies: DGS provides wide initial sampling, GLS dynamically re-balances exploration versus exploitation, and HES refines multiple elite attractors simultaneously. Consequently, ECBSO continuously adapts its search dynamics to the problem structure and demonstrates the most reliable and predictable convergence trajectory among all tested optimizers.

Figure 9 uses box-plots to visualize the solution spread of ECBSO and the competing algorithms on six representative CEC-2017 functions (F1, F8, F13, F19, F23, F29). Across all dimensions, ECBSO produces the narrowest boxes and the fewest outliers, indicating a tight, stable distribution of final solutions. This consistency is attributed to the synergistic effect of the three proposed mechanisms: DGS employs a covariance model that aligns sampling with the principal axes of the elite set, strengthening global exploration; GLS continuously monitors population dispersion and dynamically re-weights exploration versus exploitation, preventing both premature convergence and over-exploitation; HES injects multiple elite attractors via the equilibrium optimizer, providing vigorous local refinement while maintaining diversity. Consequently, ECBSO preserves a uniformly superior and compact distribution regardless of landscape type—unimodal, multimodal, hybrid or composite—demonstrating remarkable robustness across all tested dimensionalities.



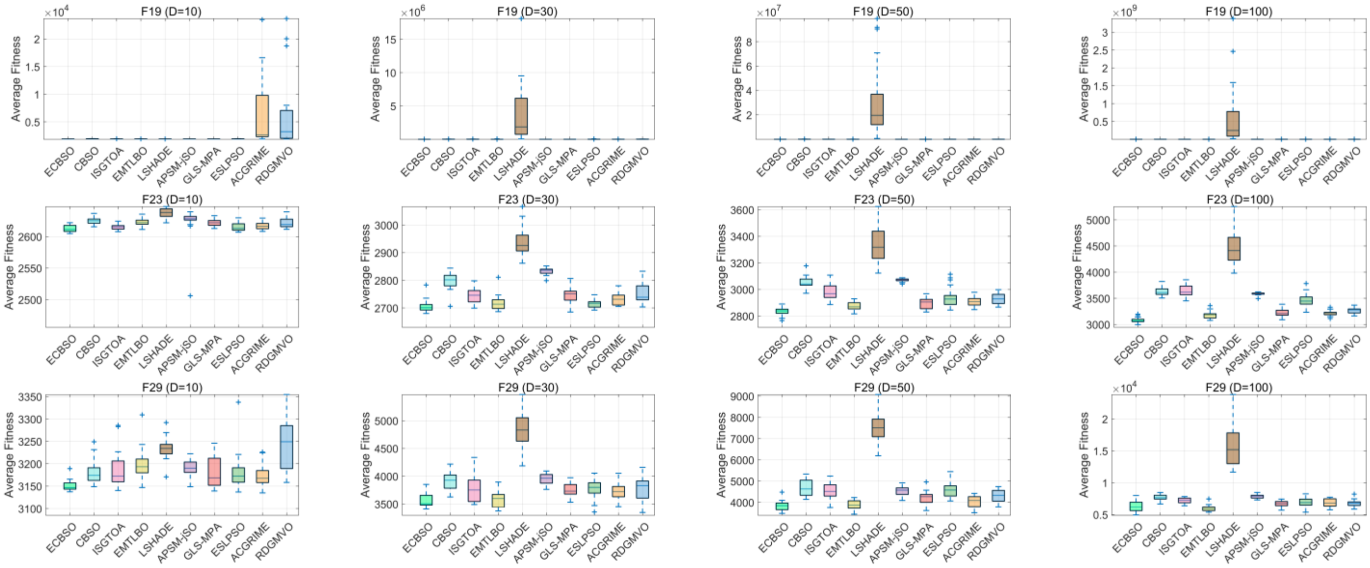


Figure 9. ECBSO and comparison algorithms

Comprehensive comparisons on the CEC-2017 suite show that ECBSO consistently outperforms all rival algorithms. Relative to the original CBSO, the three proposed modules jointly deliver a marked increase in both convergence accuracy and reliability, corroborating that the introduced strategies materially elevate the search capability of the baseline method.

5.6. Running time analysis

To ensure a fair and practical assessment, wall-clock running times measured on the same hardware are reported together with the theoretical complexity. Table 8 lists the average CPU time of ECBSO and its competitors on the CEC 2017 suite at 10 D, 30 D, 50 D and 100 D. Owing to the repeated covariance-matrix updates, ECBSO is the slowest variant; the extra minutes, however, buy a sizeable performance leap (cf. Friedman ranks in Table 6). The original CBSO, by contrast, remains among the fastest algorithms, while APSM-jSO exhibits a runtime close to that of ECBSO—consistent with their almost identical rank order. In short, ECBSO trades runtime for markedly better solution quality; this trade-off is acceptable for off-line design or planning tasks, but may preclude its use in latency-critical, real-time applications.

Table 8. Running time of ECBSO and comparison algorithms (/seconds)

Test suite	Dimension	ECBSO	CBSO	ISGTOA	EMTLBO	LSHADE	APSM-jSO	GLS-MPA	ESLPSO	ACGRIME	RDGMVO
CEC-2017	10	0.0383	0.0110	0.0174	0.0165	0.0425	0.0210	0.0114	0.0274	0.0384	0.0161
	30	0.2610	0.0724	0.0996	0.0812	0.1985	0.2083	0.0732	0.1717	0.3345	0.1172
	50	1.2931	0.4058	0.5396	0.4150	0.8102	1.2891	0.4089	0.8770	1.9978	0.6221
	100	11.7411	4.1473	5.7040	4.2512	6.6779	12.7568	4.1441	8.7575	22.1977	6.1617

5.7. Advantages and disadvantages of the proposed ECBSO algorithm

The proposed ECBSO algorithm demonstrates clear superiority over its rivals. The synergy of the three modules enables fast and stable generation of higher-quality solutions: DGS produces high-quality offspring and wider spatial coverage, markedly strengthening global exploration. GLS introduces an on-line exploitation–exploration indicator that adaptively re-balances global and local search tendencies. HES embeds the EO operator and maintains population diversity through an elite pool, effectively preventing stagnation while accelerating convergence. Collectively, these strategies endow ECBSO with both effectiveness and reliability across the entire test experiments.

Although ECBSO achieves consistent performance gains, several limitations remain to be addressed. First, the covariance matrix maintained by DGS introduces additional computational overhead, which may render the algorithm unsuitable for time-critical optimization tasks. Second, the reported optimal parameter set has been tuned exclusively on the CEC 2017 benchmark; its robustness across a broader range of problem classes is yet to be verified. Third, the update rule embedded in GLS is still rudimentary and requires further refinement to enhance its ability to navigate landscapes with highly convoluted attraction basins. Finally, while ECBSO performs well on standard benchmarks, its behavior may change when applied to highly specialized optimization problems that possess radically different constraint structures or objective characteristics that have not yet been examined.

6. Conclusion and future works

This study addresses the limitations of the original CBSO algorithm by introducing ECBSO, which integrates three synergistic enhancements: the Dominant-Group Sampling (DGS) strategy, the Guided Learning Strategy (GLS), and the Hybrid Elite Strategy (HES). Comprehensive comparisons on the CEC2017 benchmark against a variety of state-of-the-art algorithms verify the superior performance of ECBSO. The proposed algorithm consistently ranks first across different dimensionalities and landscape types, exhibiting the best convergence speed and robustness.

Although ECBSO has demonstrated commendable and superior performance, several limitations remain to be addressed in future work. The maintenance of the covariance matrix may compromise its timeliness in large-scale or real-time applications. The optimal parameter set identified on the CEC 2017 suite may not generalize to landscapes with radically different characteristics, and the exploration–exploitation operator of GLS could be further enhanced to self-adapt to both simple and complex problems. Future research will therefore focus on three directions: (1) extending the applicability of ECBSO to multi-objective optimization, real-time optimization, and other challenging scenarios. (2) refining its architecture through adaptive parameter control and the integration of cutting-edge techniques from artificial intelligence. (3) leveraging large language models to design more comprehensive search operators can achieve a balance between the breadth of exploration and the depth of exploitation. In short, we envision broader applications and deeper improvements.

Competing Interest

The authors declare that the authors have no competing interests as defined by Nature Research, or other interests that might be perceived to influence the results and/or discussion reported in this paper.

Data Availability Statement

The data is provided within the manuscript.

Code Availability Statement

The source codes of ECBSO are available at https://ww2.mathworks.cn/matlabcentral/fileexchange/182915-ecbso-an-enhanced-connected-banking-system-optimizer.

Conflicts of Interest

The authors declare no conflict of interest.

Author contributions

Dake Qian: conceptualization, methodology, writing, data testing, reviewing, software, supervision, formal analysis. **Xinyu Cai**: methodology, writing, data testing, reviewing, software. **Leidong Feng:** conceptualization, methodology, writing, reviewing. **Yun Ye:** conceptualization, visualization, reviewing, formal analysis, supervision, project administration.

Funding

This work was supported by the Zhejiang Provincial Natural Science Foundation of China (LQN25E080011), Ningbo Natural Science Foundation (2024J440), Major Special Project of Philosophy and Social Sciences Research by the Ministry of Education of the People's Republic of China: "All-Round Human Development and Common Prosperity" (2023JZDZ035), the Zhejiang Provincial Philosophy and Social Sciences Planning Special Project on 'Higher Education Basic Research Funding Reform' (Grant Number: 25NDJC153YBMS), Major Humanities and Social Sciences Research Projects in Zhejiang Higher Education Institutions (Grant Number: 2024QN018), the Special Research Project for Outstanding Young Talents in Humanities and Social Sciences (Grant Number: 06) and the Nanhu Outstanding Young Scholar Program (Grant Number: 01).

References

1. Jin, B., Cruz, L. & Goncalves, N. Face Depth Prediction by the Scene Depth. *2021 IEEE/ACIS 20TH INTERNATIONAL CONFERENCE ON COMPUTER AND INFORMATION SCIENCE (ICIS 2021-SUMMER)* 42–48 at https://doi.org/10.1109/ICIS51600.2021.9516598 (2021).

2. Dhiman, G., Garg, M., Nagar, A., Kumar, V. & Dehghani, M. A novel algorithm for global optimization: Rat Swarm Optimizer. *J. Ambient Intell. Humaniz. Comput.* (2021) doi:10.1007/s12652-020-02580-0.
3. Dai, M., Tang, D., Giret, A. & Salido, M. A. Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints. *Robot. Comput. Integr. Manuf.* (2019) doi:10.1016/j.rcim.2019.04.006.
4. Tang, A., Zhou, H., Han, T. & Xie, L. A modified manta ray foraging optimization for global optimization problems. *IEEE Access* (2021) doi:10.1109/ACCESS.2021.3113323.
5. Tang, A., Zhou, H., Han, T. & Xie, L. A Chaos Sparrow Search Algorithm with Logarithmic Spiral and Adaptive Step for Engineering Problems. *Comput. Model. Eng. Sci.* (2021) doi:10.32604/cmes.2021.017310.
6. Osaba, E. *et al.* A Tutorial On the design, experimentation and application of metaheuristic algorithms to real-World optimization problems. *Swarm Evol. Comput.* (2021) doi:10.1016/j.swevo.2021.100888.
7. Droste, S., Jansen, T. & Wegener, I. Upper and lower bounds for randomized search heuristics in black-box optimization. *Theory Comput. Syst.* (2006) doi:10.1007/s00224-004-1177-z.
8. Xie, L., Wei, Z., Ding, D., Zhang, Z. & Tang, A. Long and Short Term Maneuver Trajectory Prediction of UCAV Based on Deep Learning. *IEEE Access* (2021) doi:10.1109/ACCESS.2021.3060783.
9. Tan, M., Tang, A., Ding, D., Xie, L. & Huang, C. Autonomous Air Combat Maneuvering Decision Method of UCAV Based on LSHADE-TSO-MPC under Enemy Trajectory Prediction. *Electron.* (2022) doi:10.3390/electronics11203383.
10. Jain, A. K. & Gidwani, L. Dynamic economic load dispatch in microgrid using hybrid moth-flame optimization algorithm. *Electr. Eng.* (2024) doi:10.1007/s00202-023-02183-w.
11. Wang, Y. & Xiong, G. J. Metaheuristic optimization algorithms for multi-area economic dispatch of power systems: part II-a comparative study. *Artif. Intell. Rev.* **58**, (2025).
12. Çelik, E. *et al.* Reconfigured single- and double-diode models for improved modelling of solar cells/modules. *Sci. Rep.* **15**, (2025).
13. Hashim, F. A. *et al.* An enhanced exponential distribution optimizer and its application for multi-level medical image thresholding problems. *Alexandria Eng. J.* (2024) doi:10.1016/j.aej.2024.02.012.
14. Qiao, L., Liu, K., Xue, Y., Tang, W. & Salehnia, T. A multi-level thresholding image segmentation method using hybrid Arithmetic Optimization and Harris Hawks Optimizer algorithms. *Expert Syst. Appl.* (2024) doi:10.1016/j.eswa.2023.122316.
15. Wan, A. P. *et al.* Bayesian-Driven Optimization of MDCNN-LSTM-RSA: A New Model for Predicting Aeroengine RUL. *IEEE Trans. Reliab.* (2025) doi:10.1109/TR.2025.3574975.
16. Agrawal, U. K., Panda, N., Tejani, G. G. & Mousavirad, S. J. Improved salp swarm algorithm-driven deep CNN for brain tumor analysis. *Sci. Rep.* **15**, (2025).
17. Liu, H. *et al.* Blockchain-based optimization of operation and trading among multiple microgrids considering market fairness. *Int. J. Electr. POWER ENERGY Syst.* **166**, (2025).
18. Zhang, B., Wang, Z. X., Meng, L. L., Sang, H. Y. & Jiang, X. C. Multi-objective scheduling for surface mount technology workshop: automatic design of two-layer decomposition-based approach. *Int. J. Prod. Res.* **63**, 7570–7590 (2025).
19. Seyyedabbasi, A., Hu, G., Shehadeh, H. A., Wang, X. P. & Canatalay, P. J. V-shaped and S-shaped binary artificial protozoa optimizer (APO) algorithm for wrapper feature selection on biological data. *Clust. Comput. J. NETWORKS Softw. TOOLS Appl.* **28**, (2025).
20. Jia, H., Sun, K., Li, Y. & Cao, N. Improved marine predators algorithm for feature selection and SVM optimization. *KSII Trans. Internet Inf. Syst.* (2022) doi:10.3837/tiis.2022.04.003.
21. Opara, K. R. & Arabas, J. Differential Evolution: A survey of theoretical analyses. *Swarm Evol. Comput.* (2019) doi:10.1016/j.swevo.2018.06.010.
22. Holland, J. H. Genetic algorithms. *Sci. Am.* (1992) doi:10.1038/scientificamerican0792-66.
23. Beyer, H.-G. & Schwefel, H.-P. Evolution strategies – A comprehensive introduction. *Nat. Comput.* (2002) doi:10.1023/A:1015059928466.
24. Ahvanooey, M. T., Li, Q., Wu, M. & Wang, S. A survey of genetic programming and its applications. *KSII Trans. Internet Inf. Syst.* (2019) doi:10.3837/tiis.2019.04.002.

25. Sulaiman, M. H., Mustaffa, Z., Saari, M. M., Daniyal, H. & Mirjalili, S. Evolutionary mating algorithm. *NEURAL Comput. Appl.* **35**, 487–516 (2023).
26. Dorigo, M. & Di Caro, G. Ant colony optimization: A new meta-heuristic. in *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999* (1999). doi:10.1109/CEC.1999.782657.
27. Huang, W. & Xu, J. Particle Swarm Optimization. in *Springer Tracts in Civil Engineering* (2023). doi:10.1007/978-981-99-2213-0_2.
28. Xie, L. *et al.* Tuna Swarm Optimization: A Novel Swarm-Based Metaheuristic Algorithm for Global Optimization. *Comput. Intell. Neurosci.* (2021) doi:10.1155/2021/9210050.
29. Jia, H. M., Zhou, X. L., Zhang, J. R. & Mirjalili, S. Superb Fairy-wren Optimization Algorithm: a novel metaheuristic algorithm for solving feature selection problems. *Clust. Comput. J. NETWORKS Softw. TOOLS Appl.* **28**, (2025).
30. Guo, Z. Q., Liu, G. W. & Jiang, F. Chinese Pangolin Optimizer: a novel bio-inspired metaheuristic for solving optimization problems. *J. Supercomput.* **81**, (2025).
31. Jia, H., Rao, H., Wen, C. & Mirjalili, S. Crayfish optimization algorithm. *Artif. Intell. Rev.* (2023) doi:10.1007/s10462-023-10567-4.
32. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science (80-.).* (1983) doi:10.1126/science.220.4598.671.
33. Deng, L. & Liu, S. Snow ablation optimizer: A novel metaheuristic technique for numerical optimization and engineering design. *Expert Syst. Appl.* (2023) doi:10.1016/j.eswa.2023.120069.
34. Su, H. *et al.* RIME: A physics-based optimization. *Neurocomputing* (2023) doi:10.1016/j.neucom.2023.02.010.
35. Yuan, C. *et al.* Polar lights optimizer: Algorithm and applications in image segmentation and feature selection. *Neurocomputing* **607**, 128427 (2024).
36. Hashim, F. A., Houssein, E. H., Mabrouk, M. S., Al-Atabany, W. & Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Futur. Gener. Comput. Syst. Int. J. ESCIENCE* **101**, 646–667 (2019).
37. Abdel-Basset, M., Mohamed, R., Sallam, K. M. & Chakrabortty, R. K. Light Spectrum Optimizer: A Novel Physics-Inspired Metaheuristic Optimization Algorithm. *Mathematics* (2022) doi:10.3390/math10193466.
38. Zhang, Y. & Jin, Z. Group teaching optimization algorithm: A novel metaheuristic method for solving global optimization problems. *Expert Syst. Appl.* (2020) doi:10.1016/j.eswa.2020.113246.
39. Tian, Z. & Gai, M. Football team training algorithm: A novel sport-inspired meta-heuristic optimization algorithm for global optimization. *Expert Syst. Appl.* (2024) doi:10.1016/j.eswa.2023.123088.
40. Truong, D. N. & Chou, J. S. Metaheuristic algorithm inspired by enterprise development for global optimization and structural engineering problems with frequency constraints. *Eng. Struct.* **318**, (2024).
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. Wolpert, D. H. & Macready, W. G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* (1997) doi:10.1109/4235.585893.
43. Han, T., Wang, H. Y., Li, T. T., Liu, Q. Z. & Huang, Y. R. MHO: A Modified Hippopotamus Optimization Algorithm for Global Optimization and Engineering Design Problems. *Biomimetics* **10**, (2025).
44. Escobar-Cuevas, H., Cuevas, E., Lopez, J. & Perez-Cisneros, M. Integration of metaheuristic operators through unstructured evolutive game theory approach: a novel hybrid methodology. *Evol. Intell.* **18**, (2025).
45. Çelik, E. *et al.* Novel distance-fitness learning scheme for ameliorating metaheuristic optimization. *Eng. Sci. Technol. Int. JOURNAL-JESTECH* **65**, (2025).
46. Punia, P., Raj, A. & Kumar, P. Enhanced zebra optimization algorithm for reliability redundancy allocation and engineering optimization problems. *Clust. Comput. J. NETWORKS Softw. TOOLS Appl.* **28**, (2025).
47. Tejani, G. G., Sharma, S. K. & Mishra, S. Parallel sub class modified teaching learning based optimization. *Sci. Rep.* **15**, (2025).
48. Debnath, S. *et al.* Backtracking Search Algorithm with Mutation Strategy for Engineering Optimization. *Int. J. Comput. Intell. Syst.* **18**, (2025).

49. Xiao, Y. *et al.* IHAOAVOA: An improved hybrid aquila optimizer and African vultures optimization algorithm for global optimization problems. *Math. Biosci. Eng.* (2022) doi:10.3934/mbe.2022512.
50. El Maloufy, A. *et al.* Chaos-enhanced white shark optimization algorithms CWSO for global optimization. *ALEXANDRIA Eng. J.* **122**, 465–483 (2025).
51. Cao, B. *et al.* RFID Reader Anticollision Based on Distributed Parallel Particle Swarm Optimization. *IEEE INTERNET THINGS J.* **8**, 3099–3107 (2021).
52. Lu, Y. *et al.* A quantum-enhanced heuristic algorithm for optimizing aircraft landing problems in low-altitude intelligent transportation systems. *Sci. Rep.* **15**, (2025).
53. Nemati, M., Zandi, Y. & Sabouri, J. Application of a novel metaheuristic algorithm inspired by connected banking system in truss size and layout optimum design problems and optimization problems. *Sci. Rep.* **14**, (2024).
54. Zhang, Y. & Chi, A. Group teaching optimization algorithm with information sharing for numerical optimization and engineering optimization. *J. Intell. Manuf.* (2023) doi:10.1007/s10845-021-01872-2.
55. Jiang, Z. *et al.* An ensemble multi-swarm teaching–learning-based optimization algorithm for function optimization and image segmentation. *Appl. Soft Comput.* (2022) doi:10.1016/j.asoc.2022.109653.
56. Tanabe, R. & Fukunaga, A. S. Improving the search performance of SHADE using linear population size reduction. in *Proceedings of the 2014 IEEE Congress on Evolutionary Computation, CEC 2014* (2014). doi:10.1109/CEC.2014.6900380.
57. Li, Y. *et al.* APSM-jSO: A novel jSO variant with an adaptive parameter selection mechanism and a new external archive updating mechanism. *Swarm Evol. Comput.* (2023) doi:10.1016/j.swevo.2023.101283.
58. Jia, H. & Lu, C. Guided learning strategy: A novel update mechanism for metaheuristic algorithms design and improvement. *Knowledge-Based Syst.* (2024) doi:10.1016/j.knosys.2024.111402.
59. Deng, L. Y. & Liu, S. Y. Advancing photovoltaic system design: An enhanced social learning swarm optimizer with guaranteed stability. *Comput. Ind.* **164**, (2025).
60. Batis, M. *et al.* ACGRIME: adaptive chaotic Gaussian RIME optimizer for global optimization and feature selection. *Clust. Comput. J. NETWORKS Softw. TOOLS Appl.* **28**, (2025).
61. Zhu, W. *et al.* Optimizing microseismic monitoring: a fusion of Gaussian-Cauchy and adaptive weight strategies. *J. Comput. Des. Eng.* **11**, (2024).