Journal of Industrial and Management Optimization

JIMO, 22 (4): 2092–2118.
DOI: 10.3934/jimo.2026077
Received: 14 October 2025
Revised: 16 March 2026
Accepted: 17 March 2026
Published: 27 March 2026

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

Research article

A multi-strategy improved black-winged kite algorithm and its engineering applications

Hongluan Zhao[1,*], Zekai Jiao[2,*], Xiaoling Wang[1], Guixian Liu[1], Chenxu Yang[1] and Yang Gao[3]

[1] School of Science,Tianjin Chengjian University, Tianjin, China
[2] School of Computer and Information Engineering, Tianjin Chengjian University, Tianjin, China
[3] Institute of Science and Technology Information Research of Xizang Autonomous Region, Lasa, China

***Correspondence:** Email: hongluanzhao@163.com; 702096242@qq.com.

Abstract: The black-winged kite algorithm (BKA) draws inspiration from the predatory behavior of black-winged kite population. To address its shortcomings, including insufficient exploitation capability, declining population diversity, and susceptibility to local optima, a multi-strategy improved black-winged kite algorithm (MSBKA) is proposed. First, chaotic mapping is introduced in the initialization phase to generate sequences that broadly cover the solution space. Second, a dynamic probability mechanism is employed during the exploitation phase to enhance population diversity and prevent premature convergence. MSBKA incorporates Lévy flight to expand global search capabilities, utilizes differential mutation to strengthen local exploitation, and adopts a reflection boundary-handling strategy to resolve boundary violation issues. Furthermore, the golden sine strategy is integrated, which offers significant advantages in accelerating convergence and avoiding local optima. The algorithm is evaluated on 12 benchmark functions, demonstrating that it achieves a better balance among convergence, robustness, and global search capability. Finally, its applications to engineering problems validate the practical utility of the proposed method.

Keywords: black-winged kite algorithm; chaotic mapping; dynamic probability; lévy flight; differential mutation; golden sine; engineering problems
Mathematics Subject Classification: 68Q25, 90C30

1. Introduction

The essence of optimization problems lies in identifying optimal decision-making schemes through mathematical modeling, with widespread applications in engineering design, medical research, scientific exploration, economic management, and other fields. Efficiently solving such problems can significantly advance progress in relevant industries. However, with the exponential growth in the complexity of engineering systems, traditional optimization methods face bottlenecks when handling high-dimensional, nonlinear, and highly constrained problems. Issues such as premature convergence and degraded solution quality limit their effectiveness [1].

In contrast, metaheuristic algorithms offer several advantages, including domain independence, gradient-free operation, and ease of implementation. Their strengths lie in their global optimization capability, simplicity, and broad applicability. These algorithms are particularly well-suited for handling complex engineering optimization problems characterized by multimodality, discontinuity, and non-differentiability, which are often difficult for traditional numerical methods to address [2]. For example, swarm intelligence-based optimization has been successfully employed for short-term wind power prediction [3] and power load forecasting, by optimizing model parameters [4] and improving prediction robustness in complex energy systems. Similar optimization-assisted learning frameworks have also been reported in prognostics and health management, such as remaining useful life prediction of aero-engines [5]. As a result, metaheuristic optimization has become a vital research direction in modern optimization theory and applications.

The theories of swarm intelligence optimization have achieved substantial academic development. By simulating the self-organizing behaviors of biological collectives, such as bird flock foraging or fish swarm, researchers design mathematical models that capture these behaviors to establish population evolution mechanisms. Through information exchange and collaboration among individuals, the resulting intelligent search process effectively balances the global exploration and local exploitation in the solution space while maintaining the population diversity.

Being a frontier of computational intelligence, swarm intelligence optimization algorithms represent the cooperative evolutionary mechanisms in biological population, such as particle swarm optimization (PSO) [6], grey wolf optimization (GWO) [7], whale optimization algorithm (WOA) [8], sparrow search algorithm (SSA) [9], harris hawks optimization (HHO) [10], salp swarm algorithm (SSA) [11], butterfly optimization algorithm (BOA) [12], moth-flame optimization algorithm (MFO) [13], pelican optimization algorithm (POA) [14], and musk ox optimizer [15] and so on.

These algorithms are adaptable to large-scale parallel computing and have the robustness for non-convex and non-differentiable problems. However, they are confronted with three main challenges: (1) Premature convergence resulting from imbalances in exploration–exploitation mechanisms. (2) Algorithmic instability caused by parameter sensitivity. (3) The inherent conflict between population diversity and convergence speed. This issue becomes more pronounced in complex engineering systems, such as the multidisciplinary design optimization of semi-submersible platforms [16]. In such scenarios, metaheuristic algorithms are often adopted to coordinate structural and dynamic performance parameters for global optimal solutions.

To address these limitations, various improvement frameworks have emerged. In terms of parameter optimization, Cui et al. [17] integrated a nonlinear convergence factor into the dynamic equation of GWO, enabling phase-controlled global/local search and incorporating Lévy flight to escape local optima. Gao [18] introduced chaotic system initialization and opposition-based learning

in POA, and proposed the adaptive t distribution mutation operators to mitigate premature convergence through probabilistic perturbation. Jiang et al. [19] developed a multi-strategy cooperative framework within the dung beetle optimizer to accelerate convergence and effectively escape local optima. Additionally, Wang et al. [20] enhanced SSA by embedding chaos during initialization and designing adaptive weight allocation based on probability density, along with proposing a dual disturbance optimization strategy that employs dynamic Laplace operators.

The black-winged kite algorithm (BKA) [21], proposed in 2024, establishes a dual-mode optimization model inspired by the predation-migration behavior of the black-winged kite. Its main innovation lies in constructing a collaborative architecture whose core mechanisms include: (1) a model for transitioning between migration and attack phases, (2) a dynamic leader selection strategy, and (3) a streamlined control structure with only three parameters. By integrating bionic principles with cognitive computing theory, BKA enhances its ability to traverse nonlinear search spaces, demonstrating strong global optimization capability and adaptability. Nevertheless, BKA still faces typical convergence-related issues, such as a slow convergence rate, susceptibility to local optima, and an overly simplistic population topology.

To address these limitations, we propose a multi-strategy improved black-winged kite algorithm (MSBKA). The main innovations include: (1) chaotic mapping in the initialization phase, (2) dynamic adjustment of attack probability, (3) Lévy flight for global exploration, (4) differential mutation for local exploitation, (5) reflection boundary handling, and (6) integration of the golden sine strategy during migration.

The remainder is structured as follows. First, the principles of the original BKA are introduced. Then, we detail the improved strategies followed by simulation results and engineering applications to demonstrate the effectiveness of MSBKA.

2. Black-winged kite algorithm

The black-winged kite is characterized by its protective color camouflage formed by blue-gray back feathers and white belly feathers. As a predator, it feeds on small mammals, reptiles, birds, and insects, and is noted for its exceptional hovering ability. Key ecological traits of this species include migratory behavior, involving long-distance navigation, and complex predatory strategies. Inspired by their hunting strategies and migratory patterns, BKA is developed as a biomimetic optimization model that simulates its distinctive phases of migration and attack.

2.1. Population initialization

In BKA, a set of random solutions is first generated during the initialization. Each black-winged kite (BK) position is represented by the following matrix:

$$BK = \begin{bmatrix} BK_{1,1} & BK_{1,2} & L & L & BK_{1,dim} \\ BK_{2,1} & BK_{2,2} & L & L & BK_{2,dim} \\ M & M & M & M & M \\ M & M & M & M & M \\ BK_{pop,1} & BK_{pop,2} & L & L & BK_{pop,dim} \end{bmatrix}. \tag{1}$$

Among them, pop denotes the number of candidate solutions, dim the dimension of the given problem, and $BK_{i,j}$ the j^{th} dimension value of the i^{th} Black-winged kite. The positions of all individuals are uniformly initialized using the formula

$$X_i = BK_{lb} + rand(BK_{ub} - BK_{lb}), \tag{2}$$

where i is an integer, $1 \le i \le pop$, BK_{lb} and BK_{ub} respectively denote the lower and upper bounds of the i^{th} black-winged kite in the j^{th} dimension, and $rand$ is a uniformly distributed random number within $[0,1]$.

During the initialization, BKA selects the individual with the best fitness value as the leader X_L, which is regarded as the current optimal position. Taking minimization as an example, the initial leader can be expressed as

$$f_{best} = \min(f(X_i)), \tag{3}$$

$$X_L = X(find(f_{best} == f(X_i))). \tag{4}$$

2.2. Attack behavior

The modeling of predatory behavior in the algorithm reflects a deep integration of biomechanics and optimization theory. Its core mechanisms include an adaptive wing–tail posture adjustment equation based on fluid dynamics principles, which maximizes aerodynamic efficiency through real-time wind speed perception. During the hover–reconnaissance phases, the algorithm demonstrates a global exploration strategy, where this biomimetic mapping translates the organism's three-dimensional spatial decision-making ability into dynamic search strategies for high-dimensional optimization problems. The mathematical model is as follows:

$$X_{t+1}^{i,j} = \begin{cases} X_t^{i,j} + n \times (1 + \sin(r)) \times X_t^{i,j}, & p < r, \\ X_t^{i,j} + n \times (2r - 1) \times X_t^{i,j}, & else. \end{cases} \tag{5}$$

$X_t^{i,j}$ and $X_{t+1}^{i,j}$ respectively represent the position of the i^{th} black-winged kite of the j^{th} dimension in the t^{th} and $(t+1)^{th}$ iteration steps. r is a random number uniformly distributed in $[0,1]$. $p = 0.9$. Set T as the total number of iterations and t the number of the current iteration. The disturbance coefficient n is determined as follows:

$$n = 0.05 \times e^{-2 \times (\frac{t}{T})^2}. \tag{6}$$

2.3. Migration behavior

To adapt to seasonal changes, many bird species migrate during winter to secure better resource availability. The modeling of black-winged kite migration integrates recent research combining animal

migration ecology and group decision-making theory. Its theoretical foundation is built upon leadership evolution theory to achieve collective decision optimization. The algorithm innovatively establishes a dual-population interaction and verification mechanism. When the fitness value of the current solution is inferior to the fitness value of a randomly selected individual within the population, a leadership efficacy evaluation strategy is triggered.

Let L_t^j denote the leadership score of the j^{th} dimensional leader in the t^{th} iteration. The variables $X_t^{i,j}$ and $X_{t+1}^{i,j}$ represent the positions of the i^{th} individual in the j^{th} dimension during the t^{th} and $(t+1)^{th}$ iterations, respectively.

The model for the migration behavior of the black-winged kite is formulated as follows:

$$X_{t+1}^{i,j} = \begin{cases} X_t^{i,j} + C(0,1) \times (X_t^{i,j} - L_t^j), & f(X_i) < f(X_{ri}), \\ X_t^{i,j} + C(0,1) \times (L_t^j - m \times X_t^{i,j}), & else. \end{cases} \tag{7}$$

$$m = 2 \times \sin(r + \frac{\pi}{2}). \tag{8}$$

The introduction of standard Cauchy mutation $C(0,1)$ is aimed at enhancing population diversity, improving the global search capability, and expanding its search space. By incorporating a Cauchy operator, the algorithm can more effectively utilize the variation characteristics of the function, thereby refining the global optimal individual and accelerating convergence to the global optimum. The probability density function takes the standard form, expressed as follows:

$$f(x, \delta, \mu) = \frac{1}{\pi} \frac{1}{x^2 + 1}, -\infty < x < \infty. \tag{9}$$

3. Multi-strategy improved algorithm

To address the common limitations of swarm intelligence algorithms, numerous studies have proposed the instructive improvement frameworks. Current optimization approaches mainly evolve three means: (1) enhancing the uniformity and diversity of population spatial distribution during initialization to mitigate premature convergence, (2) designing adaptive inertia weight mechanisms to balance global exploration and local exploitation, and (3) constructing hybrid parallel architectures that integrate multiple methodologies to leverage complementary advantages and achieve synergistic optimization, thereby improving overall algorithm performance [22].

BKA also suffers from the deficiencies, including inadequate exploitation capability, declining population diversity, and susceptibility to local optima. However, research on improving BKA remains limited. To enhance its performance and achieve a better balance between exploration and exploitation, this study proposes multiple improvement strategies, resulting in MSBKA.

First, a chaotic mapping strategy is introduced during initialization to enhance the exploitation capability. Second, strategies, such as Lévy flight for global search and differential mutation for local refinement, are integrated into the attack phase. Furthermore, to optimize the migration behavior, this study incorporates the golden sine method, which samples the information from high-performing population segments to guide the direction, thereby accelerating the convergence speed.

3.1. Population initialization using tent chaotic mapping

The chaotic local search strategy enhances exploitation capability by exploring neighborhood searches around feasible solutions. Furthermore, chaotic maps exhibit favorable randomness and ergodicity, further improving the effectiveness of local search.

Common chaotic maps have logistic, tent, and sine maps, with distinct characteristics. Compared with other chaotic methods, tent map is not easy to fall into a short cycle with the appropriate parameters, and traverses the solution space faster, which makes it more advantageous in the initialization phase. Meanwhile, it achieves lower computational cost and faster execution than the sine map, which is particularly important for large-scale population initialization [23]. Therefore, this paper employs the tent chaotic map during the initialization phase. Its formulation is as follows:

$$T_{i+1} = \begin{cases} \lambda T_i , & 0 \le T_i < 0.5, \\ \lambda(1-T_i), & 0.5 \le T_i \le 1, \end{cases} \tag{10}$$

where i represents the initial population size and $1 \le i \le pop$. The initial value $T_1 \in (0,1)$ is randomly generated. Set $\lambda = 2$, which allows the tent map to achieve fully developed chaos with uniform distribution, commonly used for population initialization. T_i and T_{i+1} denote the values of the i^{th} and $(i+1)^{th}$ iterations, respectively. The population position can be initialized as follows:

$$X_i = BK_{lb} + T_i \ (BK_{ub} - BK_{lb}). \tag{11}$$

3.2. Optimization of attack behavior

3.2.1. Dynamic probability adjustment

In the original algorithm, the attack probability $p = 0.9$ remains constant, leading to slow convergence and limited adaptability. It is noteworthy that maintaining an excessively high attack probability may hinder convergence due to over-exploration, whereas low values can lead to premature stagnation in local optima. To address this, we introduce a dynamically adjusted attack probability that balances global exploration and local exploitation.

Let $p = 0.9 \times (1 - t/T)$. As the iterations proceed, the probability decreases, causing individuals to gradually shift from random attacks to in-depth exploitation around the potential high-quality solutions. This strategy mirrors the natural behavioral shift of predators from "extensive hunting" to "targeted pursuit", thereby effectively enhancing search efficiency.

3.2.2. Integration of lévy flight and differential mutation

Lévy flight, modeled on the foraging behavior observed in species such as albatrosses and honeybees, has demonstrated remarkable effectiveness particularly for high-dimensional nonlinear optimization problems [24]. Research proves that its characteristic heavy-tailed step distribution facilitates occasional long-range jumps, substantially enhancing the ability to escape local optima, while step sizes are progressively reduced in later phases to stabilize convergence and avoid oscillatory behavior. A dynamically adjusted random step size is implemented as follows:

$$LevyStep = \frac{u}{|v|^{\frac{1}{\beta}}}. \tag{12}$$

$u \sim \mathrm{N}(0, \sigma^2)$ and $\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}} \right)^{1/\beta}$. $v \sim \mathrm{N}(0,1)$. $\beta = 1.5$. Γ is a gamma function.

Then when $p < r$, Lévy flight is introduced in global search. Let $X_{t+1}^{i,j} = X_t^{i,j} + n \times LevyStep \times (L_t^j - X_t^{i,j})$. L_t^j denotes the position of the leader (i.e., the individual with the best fitness value) in the j^{th} dimension at the t^{th} iteration.

The disturbance coefficient is set to adaptive attenuation form $n = 0.1 \times e^{-(\frac{t}{T})^2}$ to slow down the attenuation rate of disturbance, so that the algorithm retains exploration abilities in the medium. The improved attack strategy replaces the original self-scaling perturbation with Lévy-driven directional exploration toward the leader, enabling both guided exploitation and long-range exploration.

When $p \geq r$, use the differential variation in local development. r_1, r_2, r_3 are randomly selected and $r_1 \neq r_2 \neq r_3$. A new solution is generated by perturbing the randomly vector X_{r1} using the difference vector $X_{r2} - X_{r3}$, which helps preserve population diversity and prevents from relying excessively on the current optimal solution. In addition, selecting a random as the base vector instead of the current leader reduces the risk of falling into local optimal traps. The update rule is defined as $X_{t+1}^{i,j} = X_{r1} + F \times (X_{r2} - X_{r3}) + 0.1 \times CauchyStep \times (L_t^j - X_t^{i,j})$.

The dynamic scaling factor F is defined as $F = 0.5 + 0.3 \times \sin(\frac{\pi t}{T})$,which provides a smooth, bounded, and non-monotonic adjustment mechanism throughout the optimization process. In the early stage of the search, when individuals are widely distributed in the solution space, a controlled baseline scaling effect ($F \approx 0.5$) is adopted to support stable exploration. As the search progresses to the middle stage, the scaling factor gradually increases and reaches its maximum value, enhancing the utilization of difference vectors for effective refinement. In the later stage, the scaling factor smoothly decreases back to its baseline level, preserving sufficient perturbation to help the algorithm escape potential local optima, which better matches the natural evolutionary characteristics compared to linear or abrupt adjustment strategies. The parameters 0.5 and 0.3 are selected as representative coefficients that control the baseline intensity and variation amplitude of the scaling factor. The subsequent experimental section will verify the effectiveness of these two parameters.

On the basis of differential variation, a Cauchy random disturbance is further incorporated to enhance the robustness of the search process. The Cauchy step is generated as $\mathrm{CauchyStep} = \tan\left(\pi \times \left(U - 0.5\right)\right)$, where $U \sim \mathrm{U}(0,1)$ represents a uniformly distributed random number within $[0,1]$. A scaling factor γ is employed to control the amplitude of the Cauchy perturbation. In this work, $\gamma = 0.1$ is adopted to balance the search step and avoid excessive jumps.

The improved attack behavior model of i^{th} black-winged kite constructed as follows:

$$X_{t+1}^{i,j} = \begin{cases} X_t^{i,j} + n \times LevyStep \times (L_t^j - X_t^{i,j}), & p < r, \\ X_{r1} + F \times (X_{r2} - X_{r3}) + \gamma \times CauchyStep \times (L_t^j - X_t^{i,j}), & else. \end{cases} \tag{13}$$

3.2.3. Boundary handling mechanism

When a solution exceeds the feasible bounds, directly resetting it to lb or ub may lead to the aggregation of individuals near the boundary. Traditional methods effectively transform the boundary into an "absorbing wall". Once an individual reaches the boundary, its movement capability is lost, increasing the risk of the algorithm stagnating in local optima. A reflection boundary handling strategy is introduced, which symmetrically reflects infeasible positions back into the feasible domain while preserving their directional tendency, thus helping the algorithm avoid boundary traps.

$$\begin{cases} X_t^{i,j} = 2 \times lb - X_t^{i,j}, & X_t^{i,j} < lb, \\ X_t^{i,j} = 2 \times ub - X_t^{i,j}, & X_t^{i,j} > ub. \end{cases} \tag{14}$$

This mechanism enhances the boundary region of exploration, as reflected positions may access high-quality solution areas.

3.3. Optimize the migration behavior

During the migration phase, the algorithm needs to efficiently locate possible solution regions to avoid unfocused search. We integrate the golden sine strategy, combining mathematical regularity with bio-inspired intelligence. By the golden ratio into the position update, the algorithm dynamically adjusts its movement step sizes and directions, formulated as

$$X_{t+1}^{i,j} = \begin{cases} X_t^{i,j} + R_2 \sin R_1 \times C(0,1) \times (x_1 X_t^{i,j} - x_2 L_t^j), & f(X_i) < f(X_{ri}), \\ X_t^{i,j} + C(0,1) \times (L_t^j - m \times X_t^{i,j}), & else. \end{cases} \tag{15}$$

wherein $f(X_i)$ is the fitness value of the current population and $f(X_{ri})$ is the fitness value of random population. R_1 is a random number in $(0, 2\pi)$. R_2 is a random number in $(0, \pi)$. x_1 and x_2 are the golden coefficients, $x_1 = -\pi + (1-\delta) \times 2\pi$, $x_2 = -\pi + \delta \times 2\pi$ and $\delta = (\sqrt{5}-1)/2$. m is as Formula (8).

The sine function can realize global disturbance and maintain the diversity. Through the disturbance term generated by the sine function, individuals can generate periodic shifts near the current solution, out of local optimization. The range $[-1,1]$ limits the disturbance amplitude, avoiding the search instability caused by too large or small random jump steps. The synergy between the golden section and the sine function thus enables complementary advantages, enhancing both search efficiency and robustness, especially for multimodal functions and high-dimensional space.

3.4. Time complexity

Let M denote the population size, D the problem dimension, and T the maximum number of iterations. In MSBKA, tent-map initialization generates M solutions in a D-dimensional space, which costs $O(MD)$, followed by M fitness evaluations with cost $O(MC_f)$, where C_f is the complexity of computing one fitness value. During each iteration, for every individual, MSBKA performs vector-based updates in both attacking and migration phases, leading to $O(D)$ computational cost. It evaluates the fitness of new solutions (twice per iteration in the proposed implementation), resulting

in $O(C_f)$ cost. Therefore, the per-iteration complexity is $O(M(D+C_f))$, and the overall time complexity of MSBKA is $O(TM(D+C_f))$. For typical benchmark functions where $C_f = O(D)$, the overall complexity reduces to $O(TMD)$, which is the same asymptotic order as the original BKA, while MSBKA may introduce a larger constant factor due to additional update strategies and fitness evaluations.

3.5. Algorithm steps and pseudocode

Through the analysis and improvement of each stage, the steps for the improved algorithm are stated as follows (taking the minimal objective function as an example).

Step1 Generate the initial positions X_i ($1 \le i \le pop$) subject to uniform distribution of the population within the search space using tent chaotic mapping as Formula (10), (11).

Step2 Calculate the objective function value for each individual $f(X_i)$.

Step3 Sort the fitness values in ascending order and select the current optimal individual as the leader L_t^j $(1 \le j \le D)$ whose fitness value is f_{best}.

Step4 Calculate the dynamic probability $p = 0.9 \times (1 - t/T)$. Update the location using Formula (13).

Step5 Use reflection boundary processing if necessary as Formula (14).

Step6 Calculate the new fitness $f(X_i)$ as Formula (15).

Step7 Update and record the global optimal solution f_{best}.

Step8 Determine whether the termination condition is met. If it is met, the search is completed and the algorithm terminates. Otherwise, execute Step 3 in a loop.

Algorithm: multi-strategy improved black-winged kite algorithm

Input: The population size *pop*, maximum number of iterations T, and variable dimension D.

Output: The best quasi-optimal solution obtained by MSBKA for a given optimization problem.

/*Initialization phase*/

1. Tent mapping initialization of the position of black-winged kites and evaluation of the objective function.
2. Calculate the fitness value of each black-winged kite.
3. **while**(t<T)

/* Attacking behavior */

4. **Dynamic attack probability:** $p \leftarrow 0.9 \times (1 - t/T)$
5. **Generate a random number** $r \sim U(0,1)$
6. **if** $p < r$
7. $X_{t+1}^{i,j} = X_t^{i,j} + n \times LevyStep \times (L_t^j - X_t^{i,j})$
8. **else if**
9. $X_{t+1}^{i,j} = X_{r1} + F \times (X_{r2} - X_{r3}) + \gamma \times CauchyStep \times (L_t^j - X_t^{i,j})$
10. **end if**
11. **reflection boundary processing**
12. Update the fitness value of each black-winged kite.

/* Migration behavior */

13. **if** $f(X_i) < f(X_{ri})$
14. $X_{t+1}^{i,j} = X_t^{i,j} + R_2 \sin R_1 \times C(0,1) \times (x_1 X_t^{i,j} - x_2 L_t^j)$
15. **else if**
16. $X_{t+1}^{i,j} = X_t^{i,j} + C(0,1) \times (L_t^j - m \times X_t^{i,j})$
17. **end if**
18. Update the fitness value of each black-winged kite.
19. Update and record the f_{best}
20. **end while**
21. **Return** f_{best}

Figure 1. Pseudocode of MSBKA.

To present the algorithm steps more clearly, the pseudo-code of MSBKA is presented in Figure 1. This pseudocode clearly describes the execution process of MSBKA. It provides steps and operations to solve specific problems and optimizes the results through iteration and adjustment.

4. Simulation and analysis

4.1. Experimental settings

All experiments are implemented in MATLAB R2024a and conducted on a computer equipped with an Intel Core Ultra 9 processor (3.20 GHz, 64-bit) and 32 GB RAM, ensuring a consistent hardware and software environment for all experiments.

To ensure fair and reproducible experimental evaluation, each algorithm is independently executed 30 times on every benchmark function. All experiments are conducted under an identical random seed schedule to eliminate stochastic variability and enable run-to-run aligned comparisons. Specifically, the MATLAB pseudo-random number generator is initialized using the Mersenne Twister algorithm as $\mathrm{rng}(r,'\mathrm{twister}'),\quad \mathrm{r}=1,2,\ldots,\mathrm{NumRuns}$, where each value of r corresponds to one independent run and is consistently applied across all experimental evaluations, ensuring identical stochastic conditions across all runs.

The detailed parameter configurations used in all experiments are summarized in Table 1, which provides complete information required to reproduce the reported results.

Table 1. Parameter table.

Parameter	Symbol	Formula/ Value / Range
Population size	pop	30
Maximum iterations	$T_{\max}$	500
Problem dimension	D	30
Number of independent runs	NumRuns	30
Random seed	r	$\mathrm{rng}(r,'\mathrm{twister}'),\quad \mathrm{r}=1,2,\ldots,\mathrm{NumRuns}$
Tent chaotic mapping parameter	λ	2
Dynamic probability	$p(t)$	$p(t)=0.9\times(1-t/T)$
Lévy flight scale parameter	β	1.5
Dynamic scaling factor	F	$F=0.5+0.3\times\sin(\frac{\pi t}{T})$
Disturbance coefficient	n	$n=0.1\times e^{-(\frac{t}{T})^2}$
Cauchy perturbation scale parameter	γ	0.1
Golden ratio	δ	$(\sqrt{5}-1)/2$

4.2. Test functions

To comprehensively evaluate the performance of MSBKA, 12 classical benchmark functions (F_1–F_{12}) with well-established standard definitions, as listed in Table 2, are selected.

Table 2. Test functions.

Type	Function	Name	Range	min
Unimodal function	$F_1(\mathbf{x}) = \sum_{i=1}^{dim} x_i^2$	Sphere	[−100,100]	0
	$F_2(\mathbf{x}) = \sum_{i=1}^{dim} \left(\sum_{j=1}^{i} x_j \right)^2$	Schwefel 1.2	[−100,100]	0
	$F_3(\mathbf{x}) = \max \lvert x_i \rvert \{ \lvert x_i \rvert, 1 \le x_i \le dim \}$	Schwefel 2.21	[−10,10]	0
	$F_4(\mathbf{x}) = \sum_{i=1}^{dim} \lvert x_i \rvert^{i+1}$	Sum power	[−1,1]	0
	$F_5(\mathbf{x}) = dim \cdot \sum_{i=1}^{dim} x_i^2$	Sum squares	[−10,10]	0
	$F_6(\mathbf{x}) = \sum_{i=1}^{dim} x_i^2 + \left(\sum_{i=1}^{dim} \frac{i}{2} x_i \right)^2 + \left(\sum_{i=1}^{dim} \frac{i}{2} x_i \right)^4$	Zakharov	[−5,10]	0
	$F_7(\mathbf{x}) = \sum_{i=1}^{dim} i \cdot x_i^4$	Quartic	[−1.28,1.28]	0
	$F_8(\mathbf{x}) = 10dim + \sum_{i=1}^{dim} \left[x_i^2 - 10\cos(2\pi x_i) \right]$	Rastrigin	[−5.12,5.12]	0
	$F_9(\mathbf{x}) = 10dim + \sum_{i=1}^{dim} \left[y_i^2 - 10\cos(2\pi y_i) \right]$ $y_i = \begin{cases} 0.5 \cdot \mathrm{sign}(x_i) & \text{if } \lvert x_i \rvert \ge 0.5 \\ x_i & \text{otherwise} \end{cases}$	NCRastrigin	[−5.12,5.12]	0
Multimodal function	$F_{10}(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{dim}\sum_{i=1}^{dim} x_i^2} \right) - \exp\left(\frac{1}{dim}\sum_{i=1}^{dim} \cos(2\pi x_i) \right) + 20 + e$	Ackley	[−50,50]	0
	$F_{11}(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{dim} x_i^2 - \prod_{i=1}^{dim} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	Griewank	[−600,600]	0
	$F_{12}(\mathbf{x}) = \sum_{i=1}^{dim} \left(\lvert x_i \cdot \sin(x_i) + 0.1x_i \rvert \right)$	Alpine	[−10,10]	0

The benchmark functions follow the commonly used standard formulations in the optimization literature and are consistent with the mainstream experimental settings adopted in Constrained

Evolutionary Computation (CEC)-related studies [27–29]. All benchmark functions are evaluated in their original forms without introducing offset or rotation operations.

Unimodal functions (F_1-F_7): Functions with a single global optimum, designed to verify exploitation capability and convergence efficiency of the algorithm.

Multimodal functions (F_8-F_{12}): Functions with numerous local extremums, used to assess the exploration ability of the algorithm and resistance to premature convergence.

4.3. Robustness analysis

To validate the robustness and search capability of MSBKA, comprehensive experiments are conducted to analyze its performance in terms of search space exploration, search trajectory, and convergence behavior on benchmark functions. Due to space constraints, this section presents detailed results specifically for multimodal test functions (F_8–F_{12}) as the following Figures 2–6, which are particularly effective in illustrating the algorithm's ability to navigate complex optimization landscapes.

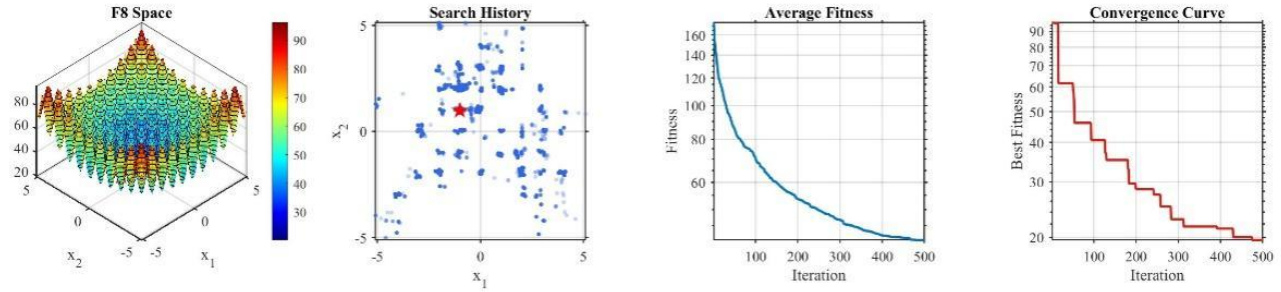


Figure 2. Search space, search history, average fitness, and convergence curve of MSBKA for F_8.

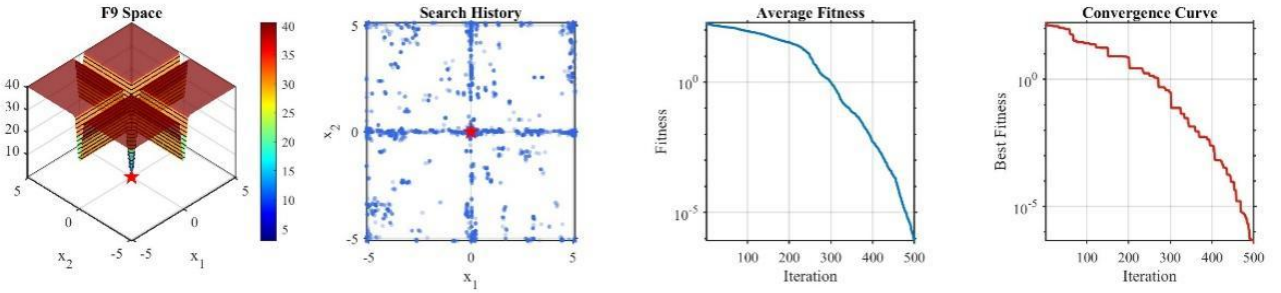


Figure 3. Search space, search history, average fitness, and convergence curve of MSBKA for F_9.

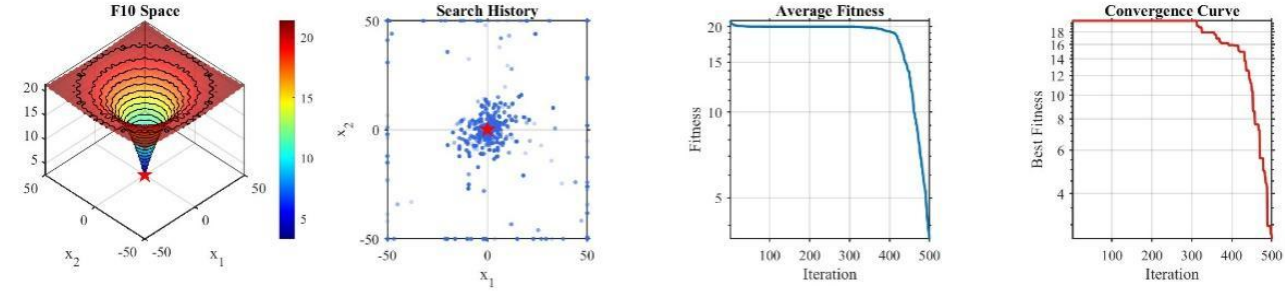


Figure 4. Search space, search history, average fitness, and convergence curve of MSBKA for F_{10}.

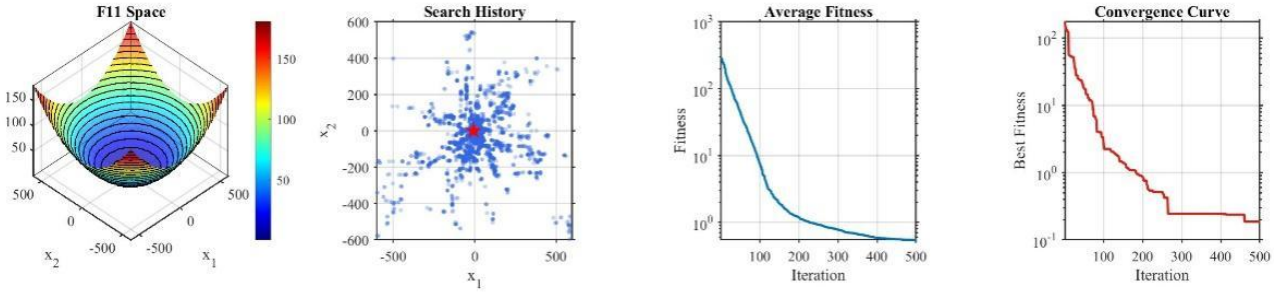


Figure 5. Search space, search history, average fitness, and convergence curve of MSBKA for F_{11}.

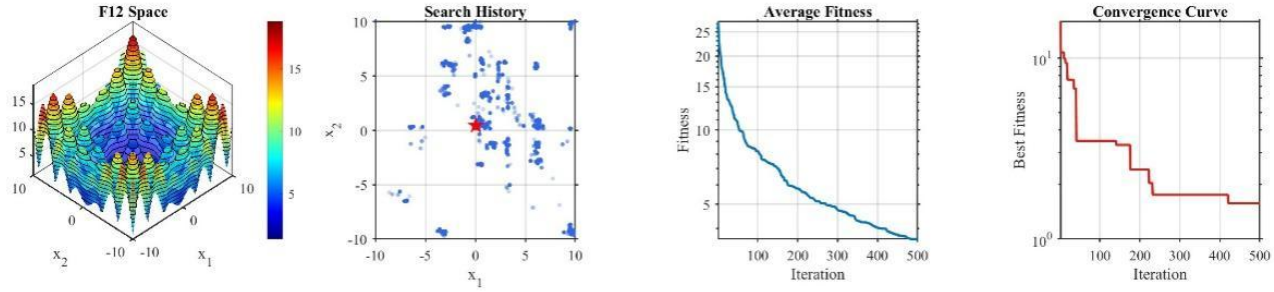


Figure 6. Search space, search history, average fitness, and convergence curve of MSBKA for F_{12}.

As observed from the Figures 2–6, the search trajectory of the algorithm presents a hash-like pattern and does not get trapped in local optima. This demonstrates the effectiveness and robustness of MSBKA in tackling complex problems. It further confirms that MSBKA possesses excellent global search capability and stability when handling multimodal functions.

4.4. Parameter sensitivity analysis

To rigorously justify the mechanism configurations of MSBKA and ensure methodological rigor, a comprehensive parameter sensitivity analysis is conducted on two core control variables: the baseline coefficient of the dynamic scaling factor ($F = 0.5 + 0.3 \times \sin(\frac{\pi t}{T})$) and the Cauchy perturbation scale parameter (γ). Representative unimodal F_1 and multimodal F_{12} benchmark functions are selected as test objects to evaluate the performance of each parameter across different candidate values. Each configuration is executed independently for 30 runs, with results recorded as mean ± standard deviation (Std) to assess both numerical accuracy and algorithm stability.

As illustrated in Figure 7(a) and Figure 7(b), the baseline coefficient of the dynamic scaling factor exerts a significant impact on the convergence precision. The empirical results demonstrate that, compared with 0.5 and 0.7, a baseline value of 0.3 yields the lowest average fitness on both F_1 and F_{12}. Figure 7(c) and Figure 7(d) exhibit the influence of the Cauchy scale parameter across three candidate values: 0.1, 0.3, and 0.5. The results clearly confirm 0.1 is the optimal for this parameter.

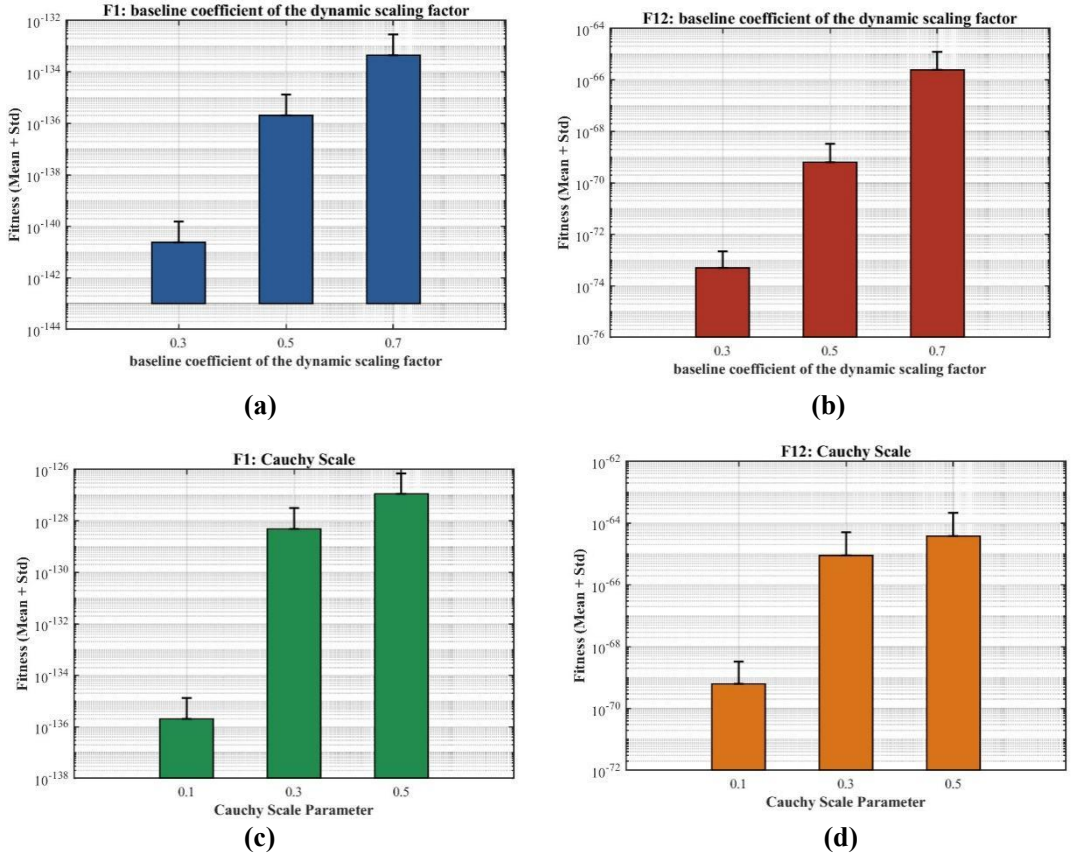


Figure 7. Sensitivity analysis of the proposed MSBKA on parameters.

4.5. Ablation study analysis

To evaluate the contribution of each individual strategy, an ablation study is conducted using several variants of BKA. The tested algorithms include the original BKA, BKA with tent chaotic initialization (BKA-Tent), BKA with improved attacking behavior (BKA-Attack), BKA with improved migration behavior (BKA-Migration), and MSBKA.

The performance of each algorithm is evaluated using the average (Avg) and standard deviation (Std) of the best fitness values obtained over 30 independent runs. The Avg value reflects the optimization accuracy, while the Std value measures the level of stability of the algorithms.

The ablation results are summarized in Table 3. Overall, a clear performance hierarchy can be observed: The original BKA provides a baseline level of optimization accuracy, the single-improvement strategy variants exhibit limited and problem-dependent performance gains, and the complete MSBKA consistently achieves the best results across almost all benchmark functions.

BKA-Tent exhibits relatively limited performance improvements and, in some cases, even underperforms BKA. This can be attributed to the fact that chaotic initialization enhances population diversity at the early stage but does not provide sustained guidance during subsequent iterations.

In contrast, BKA-Attack achieves more noticeable performance improvements on several functions. By enhancing the attack behavior, it strengthens local exploitation and accelerates convergence. However, without the support of complementary exploration and migration mechanisms, the search may become overly biased toward exploitation. As a result, although BKA-Attack outperforms BKA in many cases, its performance remains inferior to MSBKA.

BKA-Migration also demonstrates performance improvements compared to BKA, particularly on functions that benefit from directional search refinement. Nevertheless, the improvements are not uniformly significant across all functions. This is because migration-focused enhancements alone may not sufficiently regulate population diversity during earlier stages, causing their effectiveness to diminish on more complex or highly multimodal landscapes.

Table 3. Ablation study results.

Algorithm	Statistics	F_1	F_2	F_3	F_4	F_5	F_6
BKA	Avg	2.59e-80	3.38e-80	6.48e-42	5.52e-122	7.79e-81	8.75e-84
	Std	1.42e-79	1.85e-79	3.52e-41	2.70e-121	4.26e-80	4.79e-83
BKA-Tent	Avg	4.81e-78	2.55e-80	7.05e-43	1.24e-123	1.44e-78	1.46e-83
	Std	2.63e-77	1.38e-79	2.68e-42	6.24e-123	7.89e-78	8.02e-83
BKA-Attack	Avg	2.08e-120	4.28e-117	5.73e-62	3.76e-187	6.24e-121	1.30e-112
	Std	1.01e-119	1.72e-116	1.04e-61	0	3.03e-120	7.16e-112
BKA-Migration	Avg	1.71e-102	1.65e-104	2.00e-53	4.42e-122	5.15e-103	1.98e-107
	Std	9.39e-102	6.31e-104	1.04e-52	2.36e-121	2.81e-102	9.97e-107
MSBKA	Avg	9.63e-133	8.68e-132	1.57e-69	1.60e-194	2.89e-133	1.50e-132
	Std	5.27e-132	3.78e-131	3.91e-69	0	1.58e-132	7.85e-132

Algorithm	Statistics	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}
BKA	Avg	5.40e-163	0	0	4.44e-16	0	2.10e-43
	Std	3.14e-162	0	0	0	0	9.53e-43
BKA-Tent	Avg	3.38e-170	0	0	4.44e-16	0	4.02e-42
	Std	0	0	0	0	0	1.99e-41
BKA-Attack	Avg	2.62e-239	0	0	4.44e-16	0	3.73e-61
	Std	0	0	0	0	0	1.85e-60
BKA-Migration	Avg	5.20e-213	0	0	4.44e-16	0	1.41e-54
	Std	0	0	0	0	0	6.33e-54
MSBKA	Avg	2.24e-274	0	0	4.44e-16	0	9.58e-68
	Std	0	0	0	0	0	5.24e-67

4.6. Convergence analysis

The convergence behaviors of MSBKA and the comparison algorithms are to be analyzed to further illustrate the dynamic optimization characteristics of the proposed method. Six representative swarm intelligence algorithms are considered for comparison, including BKA, WOA, SCA, GWO, MPA, and BKA enhanced by Lévy flight and an adaptive random cosine oscillation mechanism (MIBKA) [30]. The control parameters are all set to the values the algorithm proposer suggested. Figure 8 presents the convergence curves of these algorithms on benchmark functions, where the vertical axis represents the fitness value, and the horizontal axis represents the number of iterations.

For unimodal functions F_1–F_7, MSBKA consistently demonstrates the fastest and most stable convergence performance. Although MIBKA shows considerable improvements, it still lags behind MSBKA. For F_8 and F_9, in the early iterations, MSBKA shows convergence performance comparable to several competing algorithms, indicating that it does not aggressively exploit at the expense of search diversity. As the iterations progress, MSBKA maintains continuous fitness improvement and gradually

outperforms other algorithms in the middle and late stages. On F_{10}, SCA converges faster in the early iterations, whereas MSBKA achieves comparable accuracy in the later iterations with more stable and robust convergence. On F_{11}, both MSBKA and MIBKA exhibit strong early convergence capability. However, MSBKA reaches lower fitness values with fewer iterations and maintains a more stable convergence trajectory. The original BKA converges more slowly and shows noticeable deceleration during the middle stages. SCA experiences mid-stage stagnation, while WOA, GWO, and MPA remain trapped at relatively high fitness levels, indicating limited ability to escape local optima. On F_{12}, MSBKA exhibits a sustained and nearly monotonic decrease in fitness across all iterations, reflecting strong convergence momentum and effective late-stage refinement. Although MIBKA and BKA continue to improve their solutions, their convergence rates and final solution quality remain inferior to MSBKA.

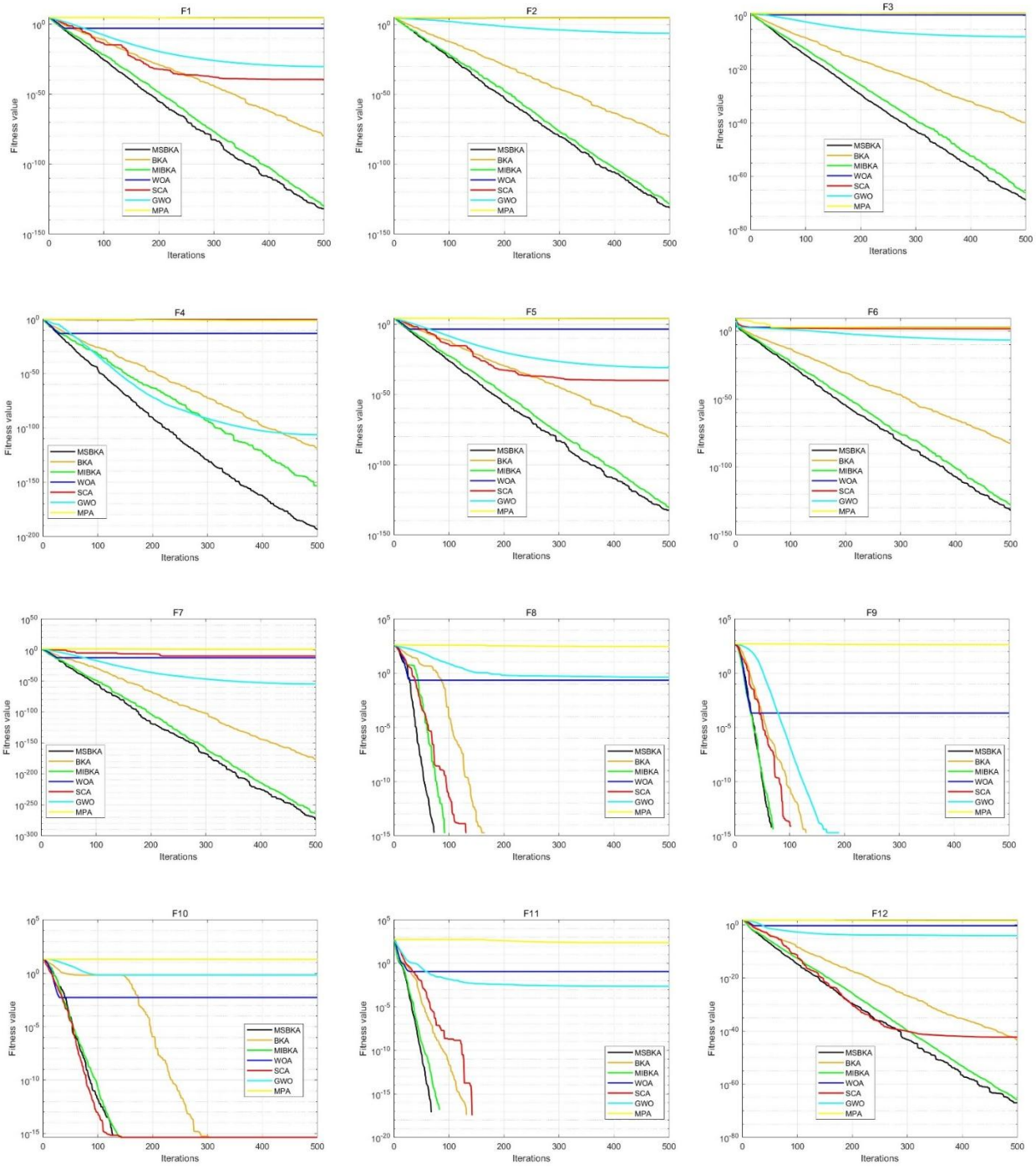


Figure 8. Convergence analysis of MSBKA and competitor algorithms in test functions.

4.7. Precision comparison and analysis

To assess the algorithm performance in solving standard test functions, a comparative analysis is performed with the average value, standard deviation, and ranking adopted as evaluation metrics. The ranking comprehensively measures the overall performance of the algorithms. The data in the below Table 4 denotes the optimal average value for each function after being tested with seven different algorithms.

Table 4. Comparison of test results of test functions.

Algorithm	Statistics	F_1	F_2	F_3	F_4	F_5	F_6
	Avg	9.63e-133	8.68e-132	1.57e-69	1.60e-194	2.89e-133	1.50e-132
MSBKA	Std	5.27e-132	3.78e-131	3.91e-69	0	1.58e-132	7.85e-132
	Rank	1	1	1	1	1	1
	Avg	7.08e-131	5.45e-129	8.76e-67	5.36e-154	2.12e-131	1.54e-128
MIBKA	Std	2.12e-130	2.77e-128	4.12e-66	2.92e-153	6.37e-131	8.12e-128
	Rank	2	2	2	2	2	2
	Avg	2.15e-80	7.56e-81	1.08e-40	1.06e-119	6.47e-81	1.70e-83
BKA	Std	1.18e-79	4.14e-80	5.92e-40	4.13e-119	3.54e-80	6.48e-83
	Rank	3	3	3	3	3	3
	Avg	1.15e-03	5.73e+04	2.05e+00	8.11e-14	3.46e-04	5.01e+02
WOA	Std	2.77e-03	1.17e+04	2.20e+00	2.58e-13	8.33e-04	1.06e+02
	Rank	6	6	5	5	6	7
	Avg	3.64e-40	6.20e+04	7.99e+00	2.34e-01	1.09e-40	4.15e+01
SCA	Std	1.99e-39	2.12e+04	5.91e-01	2.25e-01	5.97e-40	3.66e+01
	Rank	4	7	7	7	4	5
	Avg	4.91e-31	7.28e-07	1.57e-08	4.63e-107	1.47e-31	2.27e-07
GWO	Std	9.26e-31	3.21e-06	2.03e-08	1.60e-106	2.77e-31	4.79e-07
	Rank	5	4	4	4	5	4
	Avg	2.71e+04	4.84e+04	7.10e+00	5.54e-02	8.15e+03	3.20e+02
MPA	Std	7.13e+03	1.08e+04	7.91e-01	5.14e-02	2.13e+03	6.80e+01
	Rank	7	5	6	6	7	6

Algorithm	Statistics	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}
	Avg	2.24e-274	0	0	4.44e-16	0	9.58e-68
MSBKA	Std	0	0	0	0	0	5.24e-67
	Rank	1	1	1	1	1	1
	Avg	1.70e-264	0	0	4.44e-16	0	1.44e-66
MIBKA	Std	0	0	0	0	0	7.29e-66
	Rank	3	1	1	1	1	2
	Avg	2.65e-178	0	0	4.44e-16	0	3.42e-44
BKA	Std	0	0	0	0	0	1.86e-43
	Rank	3	1	1	1	1	3

Continued on next page

Algorithm	Statistics	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}
	Avg	3.36e-13	2.26e-01	2.10e-04	5.47e-03	1.15e-01	4.44e-01
WOA	Std	7.43e-13	1.13e+00	3.99e-04	4.41e-03	2.68e-01	2.40e+00
	Rank	5	5	6	5	6	6
	Avg	2.44e-10	0	0	4.44e-16	0	5.43e-43
SCA	Std	1.31e-09	0	0	0	0	1.90e-42
	Rank	6	1	1	1	1	4
	Avg	1.34e-55	4.25e-01	**0**	6.74e-01	2.32e-03	8.59e-05
GWO	Std	6.80e-55	1.49e+00	0	3.69e+00	5.46e-03	1.87e-04
	Rank	4	6	1	6	5	5
	Avg	2.07e+01	2.93e+02	4.26e+02	2.05e+01	2.45e+02	3.40e+01
MPA	Std	7.03e+00	2.58e+01	2.18e+01	1.19e-01	6.41e+01	4.86e+00
	Rank	7	7	7	7	7	7

Experimental data indicates that MSBKA outperforms all other algorithms on test functions F_1 to F_7, achieving Avg of 9.63×10^{-133}, 8.68×10^{-132}, 1.57×10^{-69}, 1.60×10^{-194}, 2.89×10^{-133}, 1.50×10^{-132} and 2.24×10^{-274}, respectively. For multimodal functions F_8, F_9 and F_{11}, the algorithm rapidly converges to the ideal optimal 0, demonstrating the best performance. Furthermore, the MSBKA algorithm also surpasses other algorithms on functions F_{10} and F_{12} and comparative analysis of Std further confirms the exceptionally high stability of the algorithm. In terms of overall ranking, it also maintains the first position consistently across all test functions.

4.8. Nonparametric statistical analysis

Table 5. Results of the friedman test ranking.

	MSBKA	MIBKA	BKA	WOA	SCA	GWO	MPA
F_1	1.07	1.93	3.03	6.00	3.97	5.00	7.00
F_2	1.03	1.97	3.00	6.23	6.23	4.00	5.53
F_3	1.13	1.87	3.00	5.03	6.87	4.00	6.10
F_4	1.00	2.00	3.00	5.13	6.60	4.00	6.27
F_5	1.03	1.97	3.37	5.97	3.63	5.03	7.00
F_6	1.07	1.93	3.00	6.90	5.00	4.00	6.10
F_7	1.03	1.97	3.23	5.93	4.00	4.83	7.00
F_8	2.73	2.73	2.73	5.90	2.73	4.17	7.00
F_9	3.00	3.00	3.00	6.00	3.00	3.00	7.00
F_{10}	2.50	2.50	2.50	5.97	2.50	5.03	7.00
F_{11}	2.90	2.90	2.90	5.83	2.90	3.57	7.00
F_{12}	1.03	1.97	3.37	5.97	3.63	5.03	7.00

To statistically evaluate the performance differences among MSBKA and the competing algorithms, the Friedman test and the Wilcoxon signed-rank test are employed. The Friedman test is employed to conduct an overall performance comparison among all algorithms. For each benchmark function, the algorithms are first ranked according to their performance in each independent run, where

the best-performing algorithm is assigned rank 1. In the case of identical performance values, average ranks are assigned to the corresponding algorithms.

The Wilcoxon signed-rank test is to perform pairwise performance comparisons between MSBKA and each competing algorithm. Specifically, the differences between paired samples from each independent run are calculated, and the test is applied to determine whether the median of these differences is statistically significantly different from zero. The significance level is set $\alpha = 0.05$.

Table 5 presents the Friedman ranking results of MSBKA and the competing algorithms on all benchmark functions. As can be observed, MSBKA achieves the lowest average rank on most functions, indicating its superior overall performance. Moreover, MSBKA obtains the best or near-best ranking on both unimodal and multimodal benchmarks, demonstrating its robustness and effectiveness across different optimization landscapes.

Table 6. Results of the Wilcoxon signed-rank test.

	vs MIBKA	vs BKA	vs WOA	vs SCA	vs GWO	vs MPA
F_1	4.86e-05(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)
F_2	4.72e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)
F_3	2.59e-05(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)
F_4	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)
F_5	4.86e-05(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)
F_6	1.97e-05(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)
F_7	6.98e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)
F_8	1(=)	1(=)	1.73e-06(+)	1(=)	2.97e-05(+)	1.73e-06(+)
F_9	1(=)	1(=)	1.73e-06(+)	1(=)	1(=)	1.73e-06(+)
F_{10}	1(=)	1(=)	1.73e-06(+)	1(=)	1.63e-06(+)	1.63e-06(+)
F_{11}	1(=)	1(=)	1.73e-06(+)	1(=)	3.12e-02(=)	1.73e-06(+)
F_{12}	2.84e-05(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)	1.73e-06(+)

Table 6 reports the p-values of the Wilcoxon signed-rank test, where the symbols "+", "–", and "=" indicate that MSBKA performs significantly better than, worse than, or statistically equivalent to the corresponding algorithms, respectively. As in Table 6, MSBKA significantly outperforms most. It is noteworthy that many p-values take the same extremely small value (e.g., 1.73×10^{-6}). It indicates that, for these functions, MSBKA consistently achieves better results than the compared algorithms, leading to identical Wilcoxon test statistics and consequently identical p-values. In contrast, for several benchmark functions (e.g., F_8–F_{11}), p-values equal to 1 are observed, which implies that the compared algorithms reach identical solutions in all runs and no statistically significant difference exists.

5. Solving the engineering problems

The above experimental results confirm that MSBKA demonstrates strong performance on benchmark functions. To further validate its capability in solving practical problems, MSBKA is applied to several engineering optimization problems, including the designs of pressure vessel, three-bar truss, tension/compression spring, corrugated bulkhead.

5.1. Solving the design of pressure vessel

The primary objective of the pressure vessel design is to minimize production costs while satisfying the specific design constraints by optimizing four variables: shell thickness (Ts), head thickness (Th), inner radius (R) and cylindrical section length excluding heads (L). The mathematical model is formulated as follows.

Consider variable $H=[h_1,h_2,h_3,h_4]=[Ts,\ Th,\ \mathrm{R},\ \mathrm{L}]$.

Minimize $$f(H) = 0.6224h_1h_3h_4 + 1.7781h_2h_3^2 + 3.1661h_1^2h_4 + 19.84h_1^2h_3. \tag{16}$$

Subject to $$l_1(H) = 0.0193h_3 - h_1 \le 0, \tag{17}$$

$$l_2(H) = 0.00954h_3 - h_2 \le 0, \tag{18}$$

$$l_3(H) = 1296000 - \pi h_3^2 h_4 - \frac{4}{3}\pi h_3^3 \le 0, \tag{19}$$

$$l_4(H) = -240 + h_4 \le 0, \tag{20}$$

$$0 \le h_j \le 100,\ j=1,2, \qquad 10 \le h_j \le 200,\ j=3,4. \tag{21}$$

MSBKA and other comparative algorithms are respectively applied to this optimization problem based on 30 independent runs of each algorithm. The maximum number of iterations for all algorithms is set to 500, and the population size 30. The optimization results are presented in Table 7.

Table 7. The best solutions to the pressure vessel design problem using the algorithms.

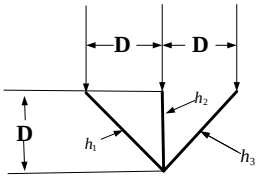
Algorithm	Optimal values for variables				Optimal cost
	Ts	Th	R	L	
MSBKA	0.775	0.383	40.320	200.000	5870.12
MIBKA	0.775	0.384	40.329	199.863	5870.89
BKA	0.775	0.383	40.320	200.000	5870.13
WOA	0.788	0.383	40.694	194.858	5933.99
SCA	0.967	0.470	49.472	102.889	6337.46
GWO	0.774	0.384	40.320	200.000	5870.60
MPA	0.787	0.496	40.912	199.894	6383.86

From Table 7, it can be concluded that MSBKA achieves the minimal production cost of 5870.12 when the shell thickness is 0.775cm, the head thickness 0.383cm, the inner radius 40.320cm, and the cross-sectional length 200.000cm. Compared with the other algorithms, MSBKA achieves a better optimal. It indicates that MSBKA has even better performance in the design of the pressure vessel.

5.2. Solving the design problem of three bar truss

The design of three bar truss aims to minimize the structural weight while maintaining a given total load capacity. To achieve this, three constraints must be satisfied, such as stress constraints, buckling constraints, and deflection constraints for each reinforcing bar.

By satisfying these three constraints, engineers can achieve a minimum-weight structural design without compromising the total load capacity. Such an optimized design not only makes efficient use of materials and reduces engineering cost, but ensures structural safety and performance reliability.



Using mathematical equations and geometric constraints to describe the truss structure, the mathematical model is established as follows.

Consider variable $$H = [h_1, h_2].$$

Minimize $$f(H) = (2\sqrt{2}h_1 + h_2) \times l. \tag{22}$$

Subject to $$l_1(H) = \frac{\sqrt{2}h_1 + h_2}{\sqrt{2}h_1^2 + 2h_1h_2} P - \sigma \le 0, \tag{23}$$

$$l_2(H) = \frac{h_2}{\sqrt{2}h_1^2 + 2h_1h_2} P - \sigma \le 0, \tag{24}$$

$$l_3(H) = \frac{1}{h_1 + \sqrt{2}h_2} P - \sigma \le 0, \tag{25}$$

$$l = 100cm, \quad P = 2KN/cm^2, \quad \sigma = 2KN/cm^2, \tag{26}$$

$$0 \le x_i \le 1, \quad i = 1, 2. \tag{27}$$

Table 8. The best solutions to solve the design problem of three bar truss using the algorithms.

Algorithm	Optimal values for variables		Optimal cost
	h_1	h_2	
MSBKA	0.789	0.408	263.89149110008623
MIBKA	0.789	0.408	263.89149110008628
BKA	0.789	0.408	263.89149110008628
WOA	0.789	0.408	263.89150229275839
SCA	0.789	0.408	263.89286367519787
GWO	0.789	0.408	263.89150573970915
MPA	0.789	0.408	263.89149787835851

Table 8 compares the solutions of seven algorithms. In this problem, MSBKA is compared with the other six algorithms, and it achieves the best solution performance.

5.3. Design issue with tension/compression springs

This engineering challenge aims to reduce the coil weight while meeting three criteria. These limitations ensure that the coil design meets certain engineering limitations and requirements. Taking the diameter of spring wire d , the average diameter of spring coil D , and the number of effective coils of spring N as variables, the following mathematical expression is used to explain this problem.

Consider variable $$H=[h_1,h_2,h_3]=[d,D,N].$$

Minimize $$f(H)=(h_3+2)\times h_2h_1^2. \tag{28}$$

Subject to $$l_1(H)=-\frac{h_2^3h_3}{71,785h_1^4}+1\le 0, \tag{29}$$

$$l_2(H)=\frac{4h_2^2-h_1h_2}{12,566(h_1^3h_2-h_1^4)}+\frac{1}{5,108h_1^2}-1\le 0, \tag{30}$$

$$l_3(H)=1-\frac{140.45h_1}{h_2{}^2h_3}\le 0, \tag{31}$$

$$l_4(H)=-1+\frac{h_1+h_2}{1.5}\le 0, \tag{32}$$

$$0.05\le h_1\le 2, 0.25\le h_2\le 1.3, 2\le h_3\le 15. \tag{33}$$

Table 9. The best solution to the design problem of tension/compression spring using the algorithms.

Algorithm	Optimal values for variables			Optimal cost
	d	D	N	
MSBKA	0.052	0.357	11.297	0.0126652332
MIBKA	0.051	0.348	11.790	0.0126687737
BKA	0.052	0.356	11.317	0.0126652396
WOA	0.051	0.341	12.249	0.0126729577
SCA	0.052	0.372	10.488	0.0127166697
GWO	0.051	0.345	12.008	0.0126745933
MPA	0.052	0.375	10.294	0.0127012456

MSBKA and other algorithms are used for comparison, and the results are set in Table 9. MSBKA can get the optimal value of 0.0126652332. This discovery provides engineers and decision-makers with reliable tools and methods to improve the design, planning, and decision-making process to achieve higher quality engineering solutions.

5.4. Design of trough bulkhead

The corrugated bulkhead of a chemical tanker is optimized with the objective to minimize its weight. The design variables include the width, depth, length, and plate thickness of the trough bulkhead. The mathematical model is formulated as follows.

Consider variable $H = [h_1, h_2, h_3, h_4].$

Minimize
$$f(H) = \frac{5.885 h_4 (h_1 + h_3)}{h_1 + \sqrt{|h_3^2 - h_2^2|}}. \tag{34}$$

Subject to
$$l_1(H) = -h_4 h_2 \left(0.4 h_1 + \frac{h_3}{6}\right) + 8.94\left(h_1 + \sqrt{|h_3^2 - h_2^2|}\right) \le 0, \tag{35}$$

$$l_2(H) = -h_4 h_2^2 \left(0.2 h_1 + \frac{h_3}{12}\right) + 2.2\left(8.94\left(h_1 + \sqrt{|h_3^2 - h_2^2|}\right)\right)^{\frac{4}{3}} \le 0, \tag{36}$$

$$l_3(H) = -h_4 + 0.0156 h_1 + 0.15 \le 0, \tag{37}$$

$$l_4(H) = -h_4 + 0.0156 h_3 + 0.15 \le 0, \tag{38}$$

$$l_5(H) = -h_4 + 1.05 \le 0, \tag{39}$$

$$l_6(H) = -h_3 + h_2 \le 0, \tag{40}$$

$$0 \le h_1, h_2, h_3 \le 100, 0 \le h_4 \le 5. \tag{41}$$

Table 10. The optimal solution to solve the design problem of slot-shaped bulkheads using the algorithms.

Algorithm	Optimal values for variables				Optimal cost
	h_1	h_2	h_3	h_4	
MSBKA	57.692	34.148	57.692	1.050	6.842953
MIBKA	57.654	34.145	57.674	1.050	6.843493
BKA	57.663	34.183	57.689	1.050	6.844932
WOA	56.384	34.154	57.689	1.050	6.851789
SCA	54.279	34.437	57.538	1.051	6.888043
GWO	57.544	34.147	57.666	1.050	6.844791
MPA	49.938	34.465	56.620	1.054	6.965725

As shown in Table 10, the MSBKA algorithm achieves a better optimal cost of 6.842953 better than the other six algorithms.

Overall, the results obtained from the four representative engineering design problems clearly demonstrate the effectiveness and practical applicability of MSBKA. Across all problems, MSBKA consistently achieves high-quality solutions with competitive or superior objective values compared with the other algorithms, while strictly satisfying the corresponding design constraints. These results indicate that the proposed multi-strategy improvement enables MSBKA to maintain a robust balance between exploration and exploitation when handling constrained optimization problems. Moreover, the stability of the obtained solutions across independent runs confirms the reliability of MSBKA for engineering design applications. Therefore, MSBKA can be regarded as a promising optimization tool for solving complex constrained engineering problems.

6. Conclusions

In this paper, an enhanced version of BKA is proposed to mitigate its limitations in slow convergence and susceptibility to local optima. At the initialization, tent chaotic mapping is introduced to enhance population diversity and accelerate convergence. During the attack phase, Lévy flight and differential mutation strategies are integrated to prevent premature convergence while balancing global exploration and local exploitation. In the migration phase, the golden sine strategy is employed to improve search efficiency. The performance of the proposed MSBKA is evaluated on 12 benchmark functions and compared with six other optimization algorithms. Experimental results indicate that MSBKA achieves notable improvements in unimodal and multimodal functions, demonstrating superior stability, faster convergence, and enhanced global search capability. Moreover, its practical applicability has been validated through several engineering problems. The next step of our work is to continue improving the algorithm and applying it to solve more practical and valuable problems.

Author contributions

Hongluan Zhao: Conceptualization, writing-original draft preparation, supervision, methodology; Zekai Jiao: Writing-original draft preparation, methodology; Xiaoling Wang: Writing-review, methodology; Guixian Liu: Writing-review, editing. Chenxu Yang: Editing and experimental simulation; Yang Gao: Experimental simulation. All authors have read and approved the final version of the manuscript for publication.

Use of Generative-AI tools declaration

The authors declare they haven’t used Artificial Intelligence tools in the creation of this article.

Acknowledgments

This work is supported by Tianjin Philosophy and Social Science Planning Project (TJTQ25-04), Tianjin Teaching Quality and Teaching Reform Research Program for Regular Institutions of Higher Education (B251006509), Teaching Reform of Tianjin Chengjian University (JG(Y)-ZD-2514, JG-ZD-25041), Undergraduate Innovation and Entrepreneurship Competition Program (202510792033), Xizang Natural Science Foundation (XZ202401ZR0135), Science and Technology Projects of Xizang Autonomous Region (XZ202501ZY0093).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. A. B. Krishna, S. Saxena, V. K. Kamboj, A novel statistical approach to numerical and multidisciplinary design optimization problems using pattern search inspired harris hawks optimizer, *Neural Comput. Appl.*, **33** (2021), 7031–7072. https://doi.org/10.1007/s00521-020-05475-5
2. R. Abbassi, A. Abbassi, A. A. Heidari, S. Mirjalili, An efficient salp swarm-inspired algorithm for parameters identification of photovoltaic cell models, *Energy Convers. Manag.*, **179** (2019), 362–372. https://doi.org/10.1016/j.enconman.2018.10.069
3. Y. L. Yuan, Q. K. Yang, J. J. Ren, K. P. Li. Z. X. Wang, Y. N. Li, et al., Short-term wind power prediction based on IBOA-AdaBoost-RVM, *J. King Saud Univ. Sci.*, **36** (2024), 103550. https://doi.org/10.1016/j.jksus.2024.103550
4. Y. L. Yuan, Q. K. Yang, J. J. Ren, X. K. Mu, Z. X. Wang, Q. L. Shen, et al., Short-term power load forecasting based on SKDR hybrid model, *Electr. Eng.*, **107** (2025), 5769–5785. https://doi.org/10.1007/s00202-024-02821-x
5. Y. L. Yuan, Q. K. Yang, G. H. Wang, J. J. Ren, Z. X. Wang, F. Qiu, et al., Combined improved tuna swarm optimization with graph convolutional neural network for remaining useful life of engine, *Qual. Reliab. Eng. Int.*, **41** (2025), 174–191. https://doi.org/10.1002/qre.3651
6. J. Kennedy, R. C. Eberhart, Particle swarm optimization, *Proc. Int. Conf. Neural Netw.*, 1995, 1942–1948. https://doi.org/10.1109/ICNN.1995.488968
7. M. H. Song, H. M. Sulaiman, R. M. Mohamed, An application of grey wolf optimizer for solving combined economic emission dispatch problems, *Int. Rev. Model. Simul.*, **7** (2014), 838–844. https://doi.org/10.15866/iremos.v7i5.2799
8. S. Mirjalili, A Lewis, The whale optimization algorithm, *Adv. Eng. Softw.*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
9. J. K. Xue, B. Shen, A novel swarm intelligence optimization approach: sparrow search algorithm, *Syst. Sci. Control Eng.*, **8** (2020), 22–34. https://doi.org/10.1080/21642583.2019.1708830
10. P. Du, J. Wang, Y. Hao, T. Niu, W. D. Yang, A novel hybrid model based on multi-objective Harris hawks optimization algorithm for daily PM2.5 and PM10 forecasting, *Appl. Soft Comput.*, **96** (2020), 106620. https://doi.org/10.1016/j.asoc.2020.106620
11. S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, *Adv. Eng. Softw.*, **114** (2017), 163–191. https://doi.org/10.1016/j.advengsoft.2017.07.002
12. S. Arora, S. Singh, Butterfly optimization algorithm: A novel approach for global optimization, *Soft Comput.*, **23** (2019), 715–734. https://doi.org/10.1007/s00500-018-3102-4
13. S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowl.-Based Syst.*, **89** (2015), 228–249. https://doi.org/10.1016/j.knosys.2015.07.006
14. P. Trojovsky, M. Dehghani, Pelican optimization algorithm: A novel nature-inspired algorithm for engineering applications, *Sensors*, **22** (2022), 855. https://doi.org/10.3390/s22030855

15. Y. L. Yuan, G. Y. Chong, J. J. Ren, W. Zhao, Y. N. Li, Z. X. Wang, et al., Musk ox optimizer (MO): a novel optimization algorithm and its application, *Cluster Comput.*, **28** (2025), 1041. https://doi.org/10.1007/s10586-025-05735-w
16. Y. L. Yuan, Q. L. Shen, W. H. Xi, S. Wang, J. J. Ren, J. Z. Yu, et al., Multidisciplinary design optimization of dynamic positioning system for semi-submersible platform, *Ocean Eng.*, **285** (2023), 115426. https://doi.org/10.1016/j.oceaneng.2023.115426
17. M. Cui, Q. B. Jin, Gray wolf optimization algorithm based on Levy flight strategy, *Comput. Digit. Eng.*, **50** (2022), 948–952, 958. https://doi.org/10.3969/j.issn.1672-9722.2022.05.006
18. M. Gao, X. W. Zeng, Pelican optimisation algorithm based on Circle mapping and adaptive t-distribution variation, *Comput. Mod.*, **9** (2024), 69–73. https://doi.org/10.3969/j.issn.1006-2475.2024.09.012
19. A. H. Jiang, W. H. Liu, Three-dimensional path planning of UAV based on improved dung beetle optimization algorithm, *Electron. Meas. Technol.*, **10** (2024), 1–9. https://doi.org/10.19651/j.cnki.emt.2416204
20. Y. H. Wang, Z. H. Wang, H. Fu, Research on transformer fault diagnosis based on multi-strategy improved sparrow algorithm and Bi-LSTM, *Chin. J. Sci. Instrum.*, **43** (2022), 87–97. https://doi.org/10.19650/j.cnki.cjsi.J2108366
21. J. Wang, W. C. Wang, X. X. Hu, L. Qiu, Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems, *Artif. Intell. Rev.*, **57** (2024), 1–53. https://doi.org/10.1007/s10462-024-10723-4
22. H. Faris, A. M. Al-Zoubi, A. A. Heidari, I. Aljarah, M. Mafarja, M. A. Hassonah, et al., An intelligent system for spam detection and identification of the most relevant features based on evolutionary random weight networks, *Inf. Fusion*, **48** (2019), 67–83. https://doi.org/10.1016/j.inffus.2018.08.002
23. J. Cai, Non-linear grey wolf optimization algorithm based on chaotic Tent mapping and elite Gauss perturbation, *Comput. Eng. Des.*, **43** (2022), 186–195.
24. M. Abdel-Basset, R. Mohamed, S. A. Azeem, Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion, *Knowl.-Based Syst.*, **268** (2023), 110454. https://doi.org/10.1016/j.knosys.2023.110454
25. Q. P. Jiang, F. Shao, W. S. Lin, K. Gu, G. Y. Jiang, H. F. Sun, Optimizing multistage discriminative dictionaries for blind image quality assessment, *IEEE Trans. Multimed.*, **20** (2018), 2035–2048. https://doi.org/10.1109/TMM.2017.2763321
26. T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, A. Cosar, A survey on new generation metaheuristic algorithms, *Comput. Ind. Eng.*, **137** (2019), 106040. https://doi.org/10.1016/j.cie.2019.106040
27. G. Wu, R. Mallipeddi, P. Suganthan, *Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization,* South Korea and Nanyang Technological University, Singapore, 2016.
28. D. Yazdani, J. Branke, M. N. Omidvar, C. H. Li, W. G. Luo, S. X. Yang, IEEE CEC 2022 competition on dynamic optimization problems generated by generalized moving peaks benchmark, *arXiv preprint arXiv: 2106.06174,* 2016. https://doi.org/10.48550/arXiv.2106.06174
29. W. Xie, P. Huang, Extreme estimation of wind pressure with unimodal and bimodal probability density function characteristics: a maximum entropy model based on fractional moments, *J. Wind Eng. Ind. Aerodyn.*, **214** (2021), 104663. https://doi.org/10.1016/j.jweia.2021.104663

30. K. X. Wang, Y. N. Tian, Y. X. Li, An improved Black-winged kite optimization algorithm for solving complex functions and engineering problems, *Comput. Technol. Dev.*, **35** (2025), 191–198. https://doi.org/10.20165/j.cnki.ISSN1673-629X.2025.0150
31. S. Singh, H. Singh, N. Mittal, J. K. Punj, L. Kumar, K. A. Fante, A hybrid swarm intelligent optimization algorithm for antenna design problems, *Sci. Rep.*, **15** (2025), 4444. https://doi.org/10.1038/s41598-025-88846-5
32. S. Mishra, A. G. Shaik, O. P. Mahela, Swarm intelligent search and rescue method for economic emission load dispatch of renewable integrated power system considering uncertainty, *Swarm Evol. Comput.*, **95** (2025), 101928. https://doi.org/10.1016/j.swevo.2025.101928

AIMS Press