



Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems

Jun Wang¹ · Wen-chuan Wang¹ · Xiao-xue Hu¹ · Lin Qiu¹ · Hong-fei Zang¹

Accepted: 4 February 2024 / Published online: 23 March 2024
© The Author(s) 2024

Abstract

This paper innovatively proposes the Black Kite Algorithm (BKA), a meta-heuristic optimization algorithm inspired by the migratory and predatory behavior of the black kite. The BKA integrates the Cauchy mutation strategy and the Leader strategy to enhance the global search capability and the convergence speed of the algorithm. This novel combination achieves a good balance between exploring global solutions and utilizing local information. Against the standard test function sets of CEC-2022 and CEC-2017, as well as other complex functions, BKA attained the best performance in 66.7, 72.4 and 77.8% of the cases, respectively. The effectiveness of the algorithm is validated through detailed convergence analysis and statistical comparisons. Moreover, its application in solving five practical engineering design problems demonstrates its practical potential in addressing constrained challenges in the real world and indicates that it has significant competitive strength in comparison with existing optimization techniques. In summary, the BKA has proven its practical value and advantages in solving a variety of complex optimization problems due to its excellent performance. The source code of BKA is publicly available at <https://www.mathworks.com/matlabcentral/fileexchange/161401-black-winged-kite-algorithm-bka>.

Keywords Nature-inspired optimization · Black-winged kite algorithm · Meta-heuristic · Constrained problem

✉ Wen-chuan Wang
wangwen1621@163.com; wangwenchuan@ncwu.edu.cn

Jun Wang
wangjun20221121@163.com

Xiao-xue Hu
xiaoxue_2356@163.com

Lin Qiu
qiulin@ncwu.edu.cn

Hong-fei Zang
zhf6344@163.com

¹ College of Water Resources, North China University of Water Resources and Electric Power, Zhengzhou 450046, China

1 Introduction

In recent years, due to resource scarcity and increasing demand from people (Feng et al. 2024), improving production efficiency has become a research hotspot (Zhao et al. 2023a, b). As technology advances and problems become more complex, optimization tasks frequently exhibit multi-objective, large-scale, uncertain, and complicated traits to parse (Wan et al. 2023). In the real world, many problems have multiple optimization objectives and constraints, while traditional optimization algorithms (Inceyol and Cay 2022; Wang et al. 2022) are mainly designed for a single objective or a small number of objective issues (Atban et al. 2023; Hu et al. 2023; Wang et al. 2023a, b). Traditional algorithms may not be able to accurately find the optimal solution when faced with these challenging optimization tasks, or the solving procedure may be overly complicated and time-consuming. Secondly, the search space for some problems is vast, and traditional optimization algorithms find it challenging to efficiently search for the optimal solution in this situation. In addition, once the problem involves uncertainty and fuzziness (Berger and Bosetti 2020), traditional optimization algorithms cannot handle it well. This is because conventional optimization algorithms are mainly based on deterministic assumptions and constraints. At the same time, there are always uncertainties and randomness in areas such as venture capital (Xu et al. 2023a, b), supply chain management (Zaman et al. 2023), and resource scheduling (Al-Masri et al. 2023). Finally, traditional optimization algorithms typically rely on the analytical form of the problem, which requires the problem to be clearly defined and described in mathematical form (Kumar et al. 2023). In practical situations, it is often difficult to express the problem analytically, or the problem's objective function and constraint conditions are intricate (Wang et al. 2020). In summary, traditional optimization algorithms often cannot meet the needs and challenges of current optimization tasks.

In this context, meta-heuristic optimization algorithms (Fan and Zhou 2023) have rapidly developed due to their flexibility and gradient-free mechanisms. They have become essential tools for solving production efficiency improvement problems. The flexibility of meta-heuristic optimization algorithms enables them to adapt to diverse production environments and problem scenarios (Melman and Evsutin 2023). Meta-heuristic optimization algorithms can search and explore the problem space based on the characteristics of specific problems to find the best solution or a solution that comes close to the best one (Abdel-Basset et al. 2023a, b, c). Whether facing problems such as product design, production planning, resource allocation, or supply chain management, meta-heuristic optimization algorithms can flexibly adjust and optimize according to actual situations.

Meanwhile, the meta-heuristic optimization algorithm also has the characteristic of no gradient mechanism (Liu and Xu 2023), which allows it to deal with problems without explicit gradient information or continuous derivatives. In many production environments, obtaining gradient information on the issues through analytical methods using traditional optimization methods is difficult. The meta-heuristic optimization algorithm utilizes local knowledge about the problem for optimization through heuristic search and random exploration. In addition to high-dimensional and nonlinear problems, this gradient-free optimization method is also appropriate for discrete and constraint-based problems (Boukroune et al. 2023).

1.1 Meta-heuristic methods

An optimization algorithm based on a heuristic search is called a meta-heuristic optimization algorithm (Wang et al. 2023a, b). They usually do not have any special requirements for the objective function but instead search by simulating intelligent behavior in nature (Chen et al. 2023) or other phenomena. They are more likely to find a globally optimal solution with a broader range of applications and a certain probability of escaping the local optimum. The characteristic of meta-heuristic optimization algorithms is their global solid search ability and robustness (Xu 2023a, b; Zhao et al. 2023a, b), which can find optimal solutions in large-scale, high-dimensional problems and quickly solve problems that do not exist or have not yet found polynomial time-solving algorithms. The classification diagram for the meta-heuristic optimization algorithm is shown in Fig. 1. Meta-heuristic algorithms, which combine random algorithms with local algorithms to solve challenging optimization problems, are inspired by random phenomena in nature (Bingi, et al. 2023). They can be broadly classified into the following four types based on their various sources of inspiration:

- (1) The algorithm is designed based on the behavioral characteristics of biological populations. These models simulate organisms' collective intelligence and collaborative strategies, enabling the rapid search of problem space and finding global optimal or better approximate solutions. Biologically inspired optimization models perform well in handling continuous and global search problems. Zamani et al. (2022) present a novel bio-inspired algorithm inspired by starlings' behaviors during their stunning murmuration named Starling Murmuration Optimizer (SMO) to solve complex and engineering optimization problems as the most appropriate application of metaheuristic algorithms. The SMO introduces a dynamic multi-flock construction and three new search strategies: separating, diving, and whirling. Sand Cat Swarm Optimization (Seyyedabbasi and Kiani 2023) is a meta-heuristic algorithm based on sand cats' natural behavior. This algorithm was influenced by sand cats' capacity to recognize low-frequency noise. Due

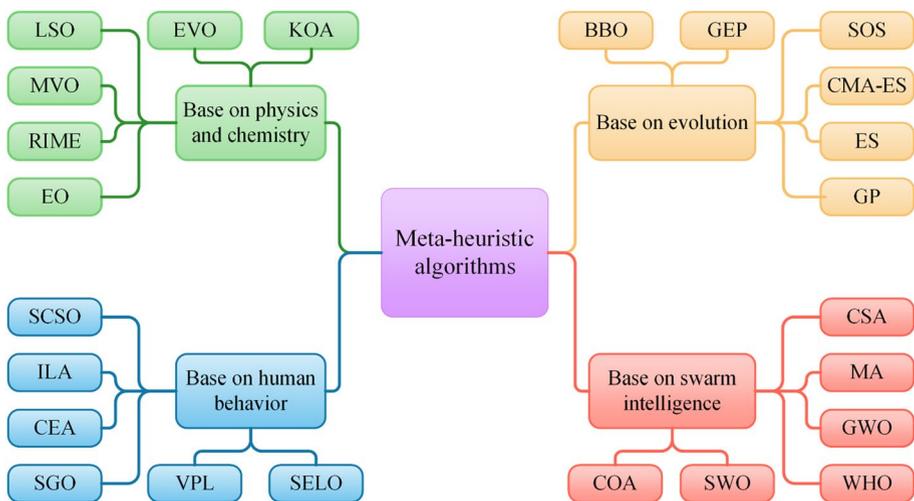


Fig. 1 Classification of metaheuristic algorithms

to its unique traits, the sand cat can find prey above and below ground. The Squirrel Search Algorithm (SSA) (Jain et al. 2019) is a single-objective optimization problem-solving heuristic algorithm based on the feeding habits of wild squirrels. This algorithm simulates the search strategy of squirrels when searching for food, gradually approaching the optimal solution by continuously adjusting the search position and range. To achieve the goal of optimization, Aquila Optimizer (AO) (Abualigah et al. 2021) primarily mimics eagles' behavior while capturing prey. It has strong optimization ability and fast convergence speed. The inspiration for the Sea Horse Optimizer (SHO) (Zhao et al. 2023a, b) comes from the hippocampus's movement, predation, and reproductive behavior in nature. The foraging and navigational habits of African vultures served as the basis for the African Vultures Optimization Algorithm (AVOA) (Abdollahzadeh et al. 2021). Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995) is a search algorithm developed based on group collaboration by simulating the foraging behavior of bird flocks. The Chameleon Swarm Algorithm (CSA) (Braik 2021) models the chameleons' dynamic foraging behavior in and around trees, deserts, and swamps. The Mayfly Algorithm (MA) (Zervoudakis and Tsafarakis 2020) is inspired by the mayflies' flight behavior and mating process. Wild horses' lives and behaviors inspired the Wild Horse Optimizer (WHO) (Naruei and Keynia 2022). Spider Wasp Optimizer (SWO) (Abdel-Basset et al. 2023b) is proposed based on female spider wasps' hunting, nesting, and mating behavior. The Coati Optimization Algorithm (COA) (Dehghani et al. 2022) is inspired by coatis. The grey wolf's social structure and hunting strategies served as the basis for the Grey Wolf Optimization (GWO) algorithm (Mirjalili et al. 2014). The Marine Predators Algorithm (MPA) (Faramarzi et al. 2020a, b) draws inspiration from the prey-hunting Brownian and Lévy movements of marine predators. The Ant Lion Optimizer (ALO) (Mirjalili 2015) is modeled after how ants navigate between their nests and food in their natural behavior. The humpback whales' bubble net hunting techniques and natural behavior served as the basis for the Whale Optimization Algorithm (WOA) (Mirjalili and Lewis 2016). The Dandelion Optimizer (DO) (Zhao et al. 2022) was proposed to simulate the process of dandelion seeds flying over long distances by wind. This algorithm considers two main factors, wind speed, and weather, and introduces Brownian motion and Levi flight to describe the seed's motion trajectory. Golden Jackal Optimization (GJO) (Chopra and Ansari 2022) is inspired by the cooperative hunting behavior of golden jackals in nature.

- (2) Algorithms abstracted from human behavior or social phenomena. These models have strong learning ability and adaptability and have demonstrated excellent performance in image recognition and natural language processing fields. The Volleyball Premier League (VPL) (Moghdani and Salimifard 2018) is inspired by the rivalry and interaction between various volleyball teams throughout the season. The social learning behavior of humans arranged in families in the social environment is the basis for the Social Evolution and Learning Optimization (SELO) (Kumar et al. 2018) algorithm. The inspiration for Social Group Optimization (SGO) (Satapathy and Naik 2016) comes from social group learning. The inspiration for the Cultural Revolution Algorithm (CEA) (Kuo and Lin 2013) comes from the process of social transformation. Hunter Prey Optimization (HPO) (Naruei et al. 2021) is inspired by the process of animal hunting. The inspiration for the IBI Logic Algorithm (Azizi et al.) (Mirrashid and Naderpour 2023) comes from thinking about brain logic.
- (3) Inspired by genetic evolution algorithms. These models can handle discrete and multi-objective optimization problems and have strong robustness and global search ability for complex issues. Gene Expression Programming (GEP) (Sharma 2015) aims to use

gene expression programming to simulate the mathematical expression relationship between data points in a set of data points based on the laws of genetic inheritance, the idea of natural selection, survival of the fittest, and elimination of the best. The population is constantly evolving to find the most suitable chromosome. The processes of how species move from one island to another, how new species appear, and how species go extinct are the inspirations for Biogeography-Based Optimization (BBO) (Simon 2008) and Covariance Matrix Adaptation Evolution Strategy (CMA-ES) (Hansen and Kern 2004). The inspiration for Symbiotic Organisms Search (SOS) (Cheng and Prayogo 2014) comes from symbiotic phenomena in biology. The inspiration for Evolution Strategies (ES) (Beyer and Schwefel 2002) comes from biological evolution. Genetic programming is inspired by natural selection (GP) (Koza 1992).

- (4) Algorithms abstracted from physical properties or chemical reactions as inspiration. These models can jump between multiple local optimal solutions and find global optimal solutions by simulating the characteristics of physical phenomena and optimizing search strategies. The Kepler Optimization Algorithm (KOA) (Abdel-Basset et al. 2023a, b, c) is a physics-based meta-heuristic algorithm that predicts the position and motion of planets at any given time by Kepler's laws of planetary motion. Energy Valley Optimizer (EVO) (Azizi et al. 2023) is a brand-new meta-heuristic algorithm that draws inspiration from physical theory's various particle decay modes and stability laws. Light Spectrum Optimizer (LSO) (Abdel-Basset et al. 2022) is a new physics-inspired meta-heuristic algorithm that generates meteorological phenomena of colored rainbow spectra inspired by the dispersion of light at different angles when passing through raindrops. Rime Optimization Algorithm (RIME) (Su et al. 2023), which constructs a soft time search strategy and a hard time puncture mechanism, simulates ice's soft time and hard time growth processes and achieves exploration and development behavior in optimization methods. Multi-verse Optimization (MVO) (Mirjalili et al. 2016) is inspired by the fact that the universe has an expansion rate, utilizing the principle that white holes have higher and black holes have a lower expansion rate. The particles in the universe search through the principle of transferring from white spots to black holes through wormholes. The control volume mass balance model, used to estimate dynamic and equilibrium states, is the primary source of inspiration for the Equilibrium Optimizer (EO) (Faramarzi et al. 2020a, b).

1.2 Related work

In this section, we discussed some recent work.

Banaie-Dezfouli et al. (2023) introduce an improved binary GWO algorithm called the extreme value-based GWO (BE-GWO) algorithm. This algorithm proposes a new cosine transfer function (CTF) to convert continuous GWO into binary form. Then, it introduces an extreme value (Ex) search strategy to improve the efficiency of converting binary solutions. Nama et al. (2023) propose a new ensemble algorithm called e-mPSOBSA with the reformed Backtracking Search Algorithm (BSA) and PSO. Chakraborty et al. (2022) suggest an enhanced SOS algorithm called nwSOS to resolve higher dimensional optimization issues. Nama and Saha (2022) introduce an improved BSA (ImBSA) based on multi-group methods and modified control parameter settings to understand the collection of various mutation strategies. Nama (2021) proposes an improved form of SOS to establish an increasingly stable balance between discovery and activity cores. This technology uses three unique programs: adjusting benefit factors, changing parasitic stages, and searching

based on random weights. To achieve the best DE efficiency, Nama and Saha (2020) proposed a new version of the DE algorithm to control parameters and mutation operators, making appropriate adjustments to time-consuming control parameters. Nama (2022) offers a new quasi-reflective slime mold (QRSMA) method that combines the SMA algorithm with a reflective learning mechanism (QRBL) to improve the performance of SMA. Nama, Sharma et al. (2022a, b) proposed an improved BSA framework called gQR-BSA, which is based on quasi-reflection initialization, quantum Gaussian mutation, adaptive parameter execution, and quasi-reflection hopping to change the coordinate structure of BSA. This algorithm adopts adaptive parameter settings, Lagrange interpolation formulas, and a new local search strategy embedded in Levy flight search to enhance search capabilities and better balance exploration and development. Nadimi-Shahraki et al. (2023a, b) wrote a review of Whale optimization algorithms, systematically explaining the theoretical basis, improvement, and mixing of WOA algorithms. Sharma et al. (2022a, b) proposed a new variant of BOA, mLBOA, to improve its performance. Sahoo et al. (2023) propose an improved dynamic reverse learning-based MFO algorithm (m-DMFO) combined with an enhanced emotional reverse learning (DOL) strategy. Sharma et al. (2022a, b) propose a hybrid sine cosine butterfly optimization algorithm (m-SCBOA), which combines the improved butterfly optimization algorithm with the sine cosine algorithm to achieve excellent exploratory and developmental search capabilities. Chakraborty et al. (2023) have proposed a hybrid slime mold algorithm (SMA) to address the issues above and accelerate the exploration of natural slime molds. Nadimi-Shahraki et al. (2023b) have developed an enhanced moth flame optimization algorithm called MFO-SFR to solve global optimization problems. Zamani et al. (2021) propose a novel DE algorithm named Quantum-based Avian Navigation Optimizer Algorithm (QANA) inspired by the extraordinary precision navigation of migratory birds during long-distance aerial paths. In the QANA, the population is distributed by partitioning into multiple flocks to explore the search space effectively using proposed self-adaptive quantum orientation and quantum-based navigation consisting of two mutation strategies, DE/quantum/I and DE/quantum/II. Nama et al. (2022a, b) proposes a new integrated technology called e-SOSBSA to completely change the degree of intensification and diversification, thereby striving to eliminate the shortcomings of (Wolpert and Macready 1997)traditional SOS.

1.3 Motivation of the work

It should be noted that no algorithm can find comprehensive solutions for every problem. As the 'No Free Lunch' (NFL) theorem reasonably indicates, no meta-heuristic algorithm is superior in solving every optimization problem. In other words, a specific meta-heuristic algorithm may achieve excellent results on particular issues but may not perform as well on other types of problems. With the continuous progress of technology and the increasing complexity of problems, some traditional algorithms cannot effectively solve these problems. After reviewing relevant literature, we found that many algorithms have limitations, including insufficient search ability, difficulty in converging to the global optimal solution, etc. These shortcomings have had a certain impact on the performance of the algorithm. We have been prompted to propose an updated and more powerful algorithm to overcome these limitations of existing algorithms and seek more effective solutions. After careful consideration, we have introduced an intelligent optimization algorithm inspired by the black-winged kite. We chose black-winged kites as our source of inspiration because they exhibit high adaptability and intelligent behavior in attack and migration. This inspired us

to develop an algorithm to better cope with complex problems. Therefore, the above reasons have become the main driving force behind our research.

1.4 Contribution and innovation to the work

The contribution and innovation of this article are as follows:

- (1) The proposed Black Winged Kite Algorithm (BKA) lies in its unique biological heuristic features, which not only capture the flight and predatory behavior of black winged kites in nature, but also deeply simulate their high adaptability to environmental changes and target positions. The imitation of this biological mechanism provides the algorithm with robust dynamic search capabilities, enabling it to effectively cope with changing optimization environments.
- (2) In the black winged kite algorithm, we first introduced the Cauchy mutation strategy, which is a probability distribution strategy that helps the algorithm jump out of local optima and increases the probability of discovering better solutions in the global search space. This strategy improves the performance of the algorithm in discovering global optimal solutions and provides new solutions for high-dimensional complex optimization problems.
- (3) We have integrated a leadership strategy that mimics the leadership role of leaders in the kite community, ensuring that the algorithm can effectively utilize the current best solution and guide the search direction. This method not only helps to enhance the efficiency of the algorithm in utilizing the current search area, but also effectively balances the dynamics between exploration and utilization, ensuring that potential competitive new areas are not overlooked in the pursuit of optimal solutions.

The remainder of this research is structured as follows: The second section introduces the Black-winged kite's attack strategy and migration behavior (Wu et al. 2023) and develops a mathematical model based on them. The third section analyzes 59 benchmark functions and the test results. Five real-world engineering cases are presented in the fourth section, and the outcomes are examined. This article is summarized, and prospects are suggested in the fifth section.

2 The black-winged kite algorithm (BKA)

In this section, a naturally inspired algorithm called the BKA is proposed.

2.1 Inspiration and behavior of black-winged kites

The black-winged kite is a small bird with a blue gray upper body and a white lower body. Their notable features include migration and predatory behavior (Ramli and Fauzi 2018). They feed on small mammals, reptiles, birds, and insects, possess strong hovering abilities, and can achieve extraordinary hunting success (Wu et al. 2023). Inspired by their hunting skills and migration habits, we established an algorithm model based on black-winged kites.

Algorithm: Black-winged kite algorithm

Input: The population size pop , maximum number of iterations T , and variable dimension dim

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

1. **Initialization phase**
2. Initialization of the position of Black-winged kites and evaluation of the objective function.
3. Calculate the fitness value of each Black-winged kite
4. **while** ($t < T$) **do**
5. /* **Attacking behavior** */
6. **if** $p < r$
7. $y_{t+1}^{i,j} = y_t^{i,j} + n(1 + \sin(r)) \times y_t^{i,j}$
8. **else if do**
9. $y_{t+1}^{i,j} = y_t^{i,j} + n \times (2r - 1) \times y_t^{i,j}$
10. **end if**
11. /* **Migration behavior** */
12. **if** $Fi < Fri$ **do**
13. $y_{t+1}^{i,j} = y_t^{i,j} + C(0,1) \times (y_t^{i,j} - L_t^j)$
14. **else if do**
15. $y_{t+1}^{i,j} = y_t^{i,j} + C(0,1) \times (L_t^j - m \times y_t^{i,j})$
16. **end if**
17. /***Select the best individual***/
18. **if** $y_{t+1}^{i,j} < L_{j,t}$
19. $Xbest = y_{t+1}^{i,j}$, $Fbest = f(y_{t+1}^{i,j})$
20. **else if do**
21. $Xbest = L_{j,t}$, $Fbest = f(L_{j,t})$
22. **end if**
23. **end while**
24. **Return** $Xbest$ and $Fbest$

Fig. 2 Pseudocode of BKA

2.2 Mathematical model and algorithm

The development of the BKA algorithm as a simple and effective meta-heuristic optimization method is illustrated in this section. We modeled the migration and attack stages of the proposed BKA based on the Black-winged kite's attack strategy and migration behavior. In Fig. 2, the pseudo-code of BKA is presented. This pseudocode clearly describes the execution process of the BKA algorithm. It provides steps and operations to solve specific problems and optimizes the results through iteration and adjustment.

2.2.1 Initialization phase

In BKA, creating a set of random solutions is the first step in initializing the population. The following matrix can be used to represent the location of every Black-winged kite (BK):



Fig. 3 **a** Black-winged kite hovering in the air, **b** The Black-winged kite rushed towards its prey at great speed

$$BK = \begin{bmatrix} BK_{1,1} & BK_{1,2} & \dots & \dots & BK_{1,dim} \\ BK_{2,1} & BK_{2,2} & \dots & \dots & BK_{2,dim} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ BK_{pop,1} & BK_{pop,2} & \dots & \dots & BK_{pop,dim} \end{bmatrix}, \tag{1}$$

where *pop* is the number of potential solutions, *dim* is the size of the given problem’s dimension, and BK_{ij} is the j^{th} dimension of the i^{th} Black-winged kite. We are distributing the position of each Black-winged kite uniformly.

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

where *i* is an integer between 1 and *pop*, where BK_{lb} and BK_{ub} are the lower and upper bounds of i^{th} Black-winged kites in the j^{th} dimension, respectively, and the *rand* is a value chosen at random between [0, 1].

In the initialization process, BKA selects the individual with the best fitness value as the leader X_L in the initial population, which is considered the optimal location of the Black-winged kites. Here is the mathematical representation of the initial leader X_L using the minimum value as an example.

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

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

2.2.2 Attacking behavior

As a predator of small grassland mammals and insects, black-winged kites adjust their wings and tail angles according to wind speed during flight, hover quietly to observe prey, and then quickly dive and attack. This strategy includes different attack behaviors for global exploration and search. Figure 3a shows a scene of a black-winged kite hovering in the air, spreading its wings and maintaining balance. Figure 3b shows the scene of the

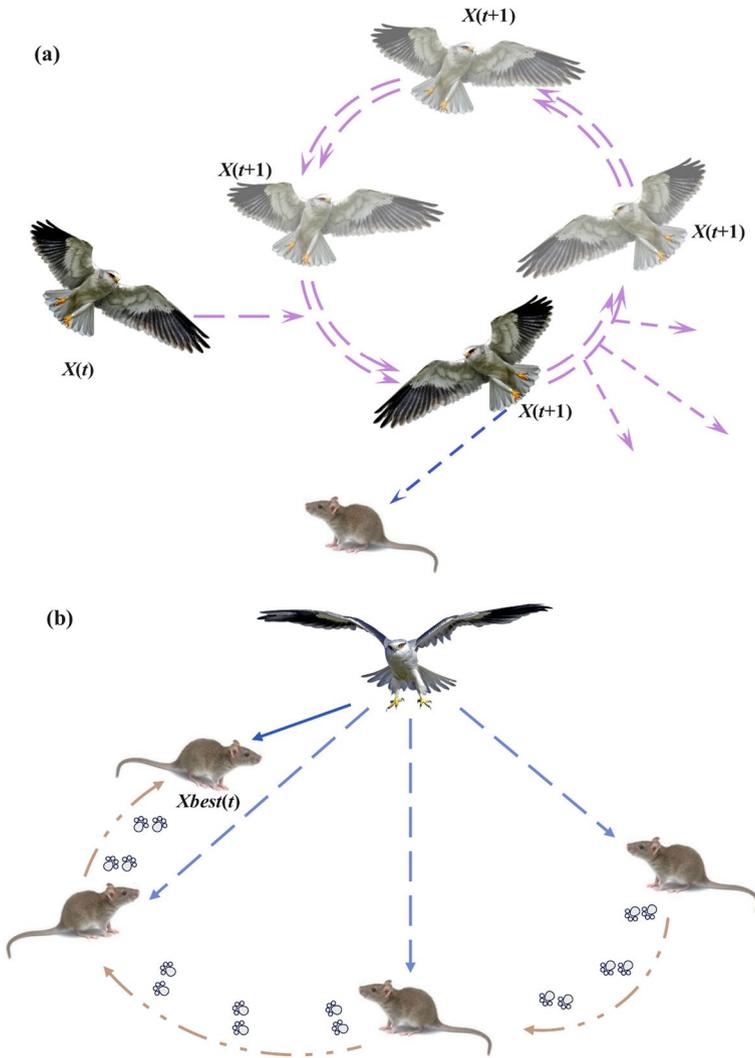


Fig. 4 Two attack strategies of Black-winged kites are **a** hovering in the air, waiting for attack, and **b** hovering in the air, searching for prey

black-winged kite rushing towards its prey at an extremely fast speed. Figure 4a shows the attack state of the black-winged kite as it hovers in the air, while Fig. 4b shows the state of the black-winged kite as it hovers in the air. The following is a mathematical model for the attack behavior of black-winged kites:

$$y_{t+1}^{ij} = \begin{cases} y_t^{ij} + n(1 + \sin(r)) \times y_t^{ij} & p < r \\ y_t^{ij} + n \times (2r - 1) \times y_t^{ij} & \text{else} \end{cases} \quad (5)$$

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

The following is a definition of the characteristics of Eqs. (5) and (6):

- $y_{i,j,t}$ and $y_{i,j,t+1}$ represent the position of the i^{th} Black-winged kites in the j^{th} dimension in the t and $(t+1)^{\text{th}}$ iteration steps, respectively.
- r is a random number that ranges from 0 to 1, and p is a constant value of 0.9.
- T is the total number of iterations, and t is the number of iterations that have been completed so far.

2.2.3 Migration behavior

Bird migration is a complex behavior influenced by environmental factors such as climate and food supply (Flack, et al. 2022). Bird migration is to adapt to seasonal changes, and many birds migrate south in winter from the north to obtain better living conditions and resources (Lees and Gilroy 2021). Migration is usually led by leaders, and their navigation skills are crucial to the success of the team. We propose a hypothesis based on bird migration: if the fitness value of the current population is less than that of the random population, the leader will give up leadership and join the migratory population, indicating that it is not suitable to lead the population forward (Cheng, et al. 2022). On the contrary, if the fitness value of the current population is greater than that of the random population, it will guide the population until it reaches its destination. This strategy can dynamically select excellent leaders to ensure a successful migration. Figure 5 shows the changes in the leading bird in the migration process of black-winged kites. The following is a mathematical model for the migration behavior of black-winged kites:

$$y_{t+1}^{i,j} = \begin{cases} y_t^{i,j} + C(0, 1) \times (y_t^{i,j} - L_t^j) & F_i < F_{ri} \\ y_t^{i,j} + C(0, 1) \times (L_t^j - m \times y_t^{i,j}) & \text{else} \end{cases} \tag{7}$$

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

The attributes of Eqs. (7) and (8) are defined as follows:

- $L_{j,t}$ represents the leading scorer of the Black-winged kites in the j^{th} dimension of the t^{th} iteration so far.
- $y_{i,j,t}$ and $y_{i,j,t+1}$ represent the position of the i^{th} Black-winged kites in the j^{th} dimension in the t and $(t+1)^{\text{th}}$ iteration steps, respectively.
- F_i represents the current position in the j^{th} dimension obtained by any Black-winged kite in the t iteration.
- F_{ri} represents the fitness value of the random position in the j^{th} dimension obtained from any Black-winged kites in the t iteration.
- $C(0,1)$ represents the Cauchy mutation (Jiang, et al. 2023). The definition is as follows:

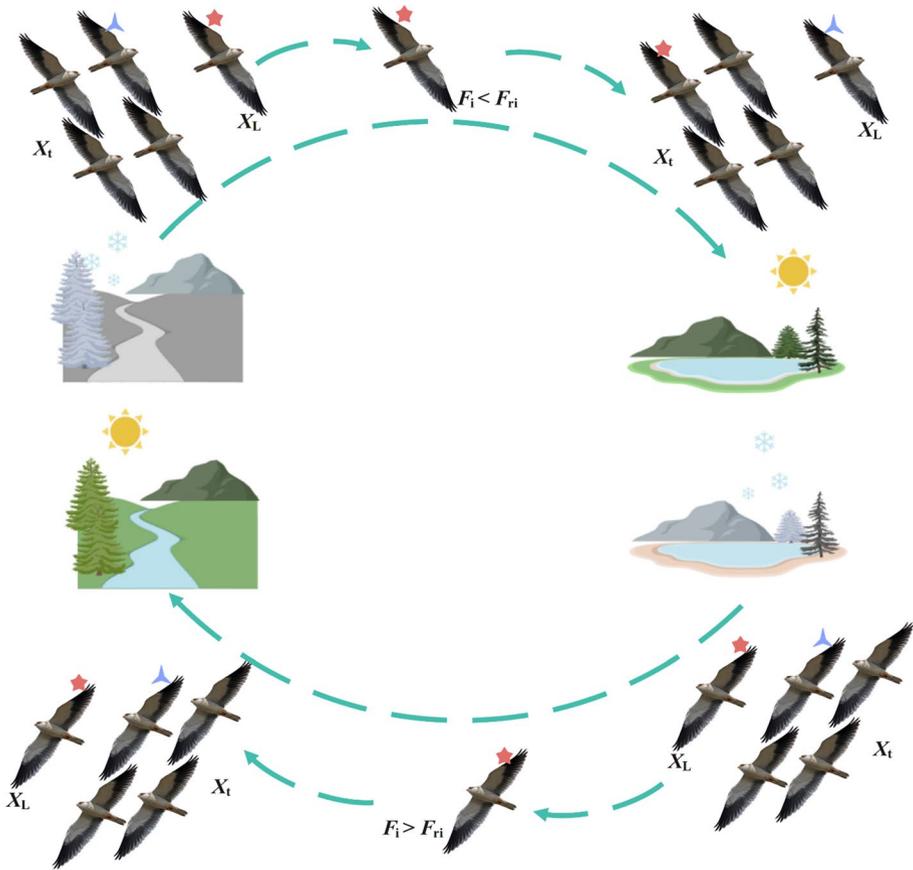


Fig. 5 The strategic changes of Black-winged kites during migration Pseudocode of BKA

A one-dimensional Cauchy distribution is a continuous probability distribution with two parameters. The following equation illustrates the probability density function of the one-dimensional Cauchy distribution:

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

When $\delta = 1, \mu = 0$, its probability density function will become the standard form. The following is the precise formula:

$$f(x, \delta, \mu) = \frac{1}{\pi} \frac{1}{x^2 + 1}, \quad -\infty < x < \infty \tag{10}$$

2.3 The balance and diversity analyses

Maintaining a good balance between global and local search is an important factor in optimizing algorithms to find the optimal solution, which involves exploring and developing

the search space. In this process, it is necessary to balance the proportion of global search and local search to ensure that the algorithm does not prematurely mature and can find the best solution. To better balance these issues, this algorithm uses parameter p to control different attack behaviors. At the same time, the variable n set in this article will decrease nonlinearly with the increase of iteration times, which can control the algorithm to shift from a global search algorithm to a local search, enabling it to find the optimal solution faster and avoid falling into local optimal solution, to better solve practical problems.

Diversity is very important in intelligent optimization algorithms, as it helps to avoid the population falling into local optima and provides a wide search range, increasing the chances of the algorithm discovering global optima. Like most intelligent optimization algorithms, the individuals in the initial population of this article are randomly generated within a given range, which results in certain differences in the positions and eigenvalues of each individual, thus giving the individuals in the population a certain degree of diversity and better exploration of the solution space. Meanwhile, during the iteration process of the algorithm, the application of the Cauchy strategy and the reasonable setting of parameters improve the diversity of the algorithm, improve its global search ability, and avoid falling into local optima.

2.4 Computational complexity

We can assess the time and space resources needed for algorithms to handle large-scale problems using computational complexity, a crucial indicator of algorithm efficiency. To better understand the effectiveness and viability of the proposed BKA algorithm, we will conduct a thorough analysis of the time and spatial complexity of the algorithm in this section.

2.4.1 Time complexity

The BKA algorithm initializes a set of potential solutions during initialization, which will be used for further search and optimization. The initialization method selected, as well as the size of the problem, typically determine how time-consuming the initialization process is. The number of candidate solutions or the size of the problem, denoted by M , determines the computational complexity of the initialization procedure in this article, which is $O(M)$. This process involves generating initial solutions, determining parameter settings, and initializing other necessary operations. This initialization process needs to be executed once before starting the algorithm. Second, one of the crucial components of the BKA algorithm, which is used to assess the effectiveness and quality of potential solutions, is fitness evaluation. The issues considered and the particular evaluation method determine how complicated the fitness assessment process is. For specific problems, fitness assessment involves complex computational or simulation techniques with a time complexity of $O(T \times M) + O(T \times M \times D)$, where T is the maximum number of iterations and D is the specific problem's dimension. Finally, updating the Black-winged kite is a critical step in the BKA algorithm, which generates new candidate solutions based on the current key and neighborhood search. The neighborhood search difficulty and the update strategy employed determine the difficulty of updating Black-winged kites. Therefore, the runtime complexity of the BKA is $O(M \times (T + T \times D + 1))$.

2.4.2 Space complexity

The spatial complexity of the BKA algorithm refers to the additional storage space required during algorithm operation. Let's analyze the spatial complexity of the BKA algorithm. The spatial complexity of the BKA algorithm is relatively low. The primary space consumption comes from storing candidate solutions and related intermediate results and temporary variables. Specifically, BKA algorithms typically only need to store the current best solution, candidate solutions, and some data structures related to the search and optimization process. In the most straightforward implementation, the spatial complexity of the BKA algorithm is approximately $O(M)$, where M represents the number of candidate solutions or the size of the problem. This is because the algorithm needs to allocate storage space for each candidate solution and update and compare it during iteration. In addition, additional storage space is needed to store other auxiliary variables and intermediate results. It should be noted that the BKA algorithm's spatial complexity can change depending on the particulars of the problem and its implementation. The spatial complexity may increase if more complex data structures or intermediate result storage are used in the algorithm.

3 Experimental results and discussion

This section conducts simulation studies and assesses the effectiveness of BKA in optimization. The experiments are conducted on MATLAB R2022b with a 3.20 GHz 64 bit Core i9 processor and 16 GB of main memory.

3.1 The benchmark set and compared algorithms

The ability of BKA to handle a variety of objective functions is tested in this article using 59 standard benchmark functions, including 18 benchmark functions, the CEC-2017 test set (Wu et al. 2016), and the CEC-2022 test set (Yazdani et al. 2021). The test results are compared with those of well-known algorithms like MVO, SCA, GWO, MPA, RIME, ALO, WOA, STOA, DO, GJO, PSO, AVOA, SHO, SCSO, SSA, AO, COA, etc. to assess the quality of the best solution offered by BKA. These algorithms' control parameters are all set to the values the algorithm proposer suggested. Three evaluation functions are also mentioned to analyze the algorithm's performance thoroughly: average (Avg), standard deviation (Std), and ranking.

(1) The definition of standard deviation is as follows:

$$Std = \sqrt{\frac{1}{m} \sum_{i=1}^m (Fi - Avg)^2} \quad (11)$$

(2) Ranking: ranking depends on the average fitness value of the algorithm. The algorithm is ranked higher when the average value is lower.

3.2 Sensitivity analysis

In this section, experiments and analysis are conducted based on the algorithm's internal parameters. The key internal parameters of the BKA algorithm are discussed and analyzed to determine the optimality and rationality of the key parameters of the proposed algorithm. In this section, when the attack mechanism takes effect, we will change the parameter p in Sect. 2.2.2. This parameter is used to control the switching between two attack behaviors and is an important parameter that affects the overall accuracy and stability of the algorithm. Set the parameter p to 0.3, 0.5, and 0.7 for experiments and compare it with the original parameter $p = 0.9$ to show the impact of parameter changes on BKA performance in the mechanism. The comparative experiment was conducted within a unified evaluation framework, with the same number of 30 populations and 30 independent runs. The experimental results are shown in Tables 1 and 2.

Table 1 The influence of parameter p on test results (CEC-2017)

Function	$p = 0.9$	$p = 0.7$	$p = 0.5$	$p = 0.3$
F1	3.27E+04	1.71E+08	5.07E+08	9.44E+08
F3	3.00E+02	3.80E+03	5.26E+03	4.76E+03
F4	4.10E+02	4.18E+02	4.52E+02	4.84E+02
F5	5.27E+02	5.39E+02	5.28E+02	5.40E+02
F6	6.17E+02	6.20E+02	6.23E+02	6.26E+02
F7	7.99E+02	7.77E+02	7.69E+02	7.88E+02
F8	8.28E+02	8.38E+02	8.45E+02	8.37E+02
F9	1.14E+03	1.22E+03	1.23E+03	1.28E+03
F10	1.80E+03	1.87E+03	1.89E+03	1.91E+03
F11	1.14E+03	1.15E+03	1.20E+03	1.18E+03
F12	2.52E+04	6.82E+05	1.14E+06	5.76E+05
F13	2.23E+03	3.91E+03	2.42E+03	4.74E+03
F14	1.48E+03	1.49E+03	1.49E+03	1.50E+03
F15	1.55E+03	1.76E+03	1.71E+03	2.21E+03
F16	1.74E+03	1.80E+03	1.79E+03	1.80E+03
F17	1.75E+03	1.76E+03	1.80E+03	1.78E+03
F18	2.29E+03	2.69E+03	9.15E+03	4.69E+03
F19	1.96E+03	1.94E+03	2.40E+03	2.00E+03
F20	2.07E+03	2.10E+03	2.08E+03	2.14E+03
F21	2.28E+03	2.28E+03	2.30E+03	2.26E+03
F22	2.35E+03	2.34E+03	2.33E+03	2.53E+03
F23	2.63E+03	2.63E+03	2.66E+03	2.65E+03
F24	2.78E+03	2.75E+03	2.78E+03	2.75E+03
F25	2.91E+03	2.93E+03	2.95E+03	2.99E+03
F26	2.99E+03	3.09E+03	3.34E+03	3.24E+03
F27	3.10E+03	3.12E+03	3.12E+03	3.13E+03
F28	3.29E+03	3.27E+03	3.33E+03	3.33E+03
F29	3.22E+03	3.26E+03	3.25E+03	3.28E+03
F30	1.12E+06	2.52E+05	1.11E+06	2.10E+06
w/t/l	21/1/7	4/1/24	2/0/27	1/1/27

Table 2 The influence of parameter p on test results (CEC-2022)

Function	$p = 0.9$	$p = 0.7$	$p = 0.5$	$p = 0.3$
F1	3.02E+02	2.57E+03	2.12E+03	2.76E+03
F2	4.03E+02	4.88E+02	4.87E+02	4.42E+02
F3	6.30E+02	6.37E+02	6.30E+02	6.29E+02
F4	8.20E+02	8.20E+02	8.22E+02	8.24E+02
F5	1.12E+03	1.15E+03	1.16E+03	1.13E+03
F6	1.94E+03	3.40E+03	2.19E+03	2.76E+03
F7	2.04E+03	2.05E+03	2.06E+03	2.05E+03
F8	2.22E+03	2.23E+03	2.23E+03	2.24E+03
F9	2.53E+03	2.58E+03	2.57E+03	2.60E+03
F10	2.67E+03	2.57E+03	2.56E+03	2.55E+03
F11	2.71E+03	2.92E+03	2.80E+03	2.70E+03
F12	2.87E+03	2.89E+03	2.87E+03	2.87E+03
w/t/l	7/3/2	0/1/11	0/2/10	2/1/9

From Table 1, we can see that for the CEC-2017 test set, BKA achieved the best results among 21 functions at parameter $p=0.9$, achieved the same optimal results as $p=0.7$ on F23, and did not achieve the best results on only 7 functions. From Table 2, we can see that for the CEC-2022 test set, BKA achieved the best results among 7 functions at parameter $p=0.9$, achieved the same optimal results as $p=0.5$ on the F3 function, achieved the same optimal results as $p=0.7$ on the F4 function, and achieved the same optimal results as $p=0.5$ and $p=0.3$ on the F12 function. Only two functions did not achieve the optimal results. Through a comprehensive analysis of Tables 1 and 2, we believe that the BKA algorithm can achieve better results in processing optimization when the parameter $p=0.9$.

3.3 The results of the algorithm on different test sets

This section used several test sets to gauge how well the recently created meta-heuristic algorithm BKA handled global optimization issues.

3.3.1 Evaluation of 18 functions and qualitative analysis

This test set includes both unimodal and multimodal functions to thoroughly assess the performance of the BKA algorithm (Xie and Huang 2021). The unimodal function (F1–F9) in this test set is a function with a globally optimal solution used to verify the efficacy of the optimization algorithm. Multimodal functions (F10–F18) have many local extremum values used to assess the algorithm's exploratory power. Tables 3 and 4 provide detailed information on 18 test functions. The results of all algorithms were obtained using 30 search agents with 500 iterations and 10 independent runs.

Table 5 shows the results of BKA and the comparison algorithm on 18 test functions. The value of Avg determines the ranking in Table 5; the lower the value, the higher the ranking. In Figs. 6 and 7, where the vertical axis denotes the fitness value and the horizontal axis denotes the number of iterations, the convergence curves of BKA and other optimization algorithms at dimension 10 are contrasted. In unimodal functions, BKA exhibits an advantage over other F1, F3, and F4 algorithms, even surpassing other algorithms by

Table 3 Unimodal test functions

Name	Function	D	Range	min
Sphere	$F_1(x) = \sum_{i=1}^D x_i^2$	30	[- 100, 100]	0
Schwefel 2.22	$F_{11}(x) = \sum_{i=1}^{dim} [y_i^2 - 10 \cos(2\pi y_i) + 10],$ $y_i = \{x_i, x_i < 0.5\}$	30	[- 10, 10]	0
Schwefel 1.2	$F_3(x) = \sum_{i=1}^D \left(\sum_{j=1}^i x_j \right)^2$	30	[- 100, 100]	0
Schwefel 2.21	$F_4(x) = \max_i \{ x_i , 1 \leq x_i \leq D \}$	30	[- 10, 10]	0
Quartic	$F_5(x) = \sum_{i=1}^{Dim} Dim \times x_i^2 + rand(0, 1)$	30	[- 1.28, 1.28]	0
Sum power	$F_6 = \sum_{i=1}^{Dim} x_i ^{(i+1)}$	30	[- 1, 1]	0
Sum squares	$F_7(x) = \sum_{i=1}^{Dim} Dim \times x_i^2$	30	[- 10, 10]	0
Zakharov	$F_8(x) = \sum_{i=1}^{dim} x_i^2 + \left(\sum_{i=1}^{dim} 0.5ix_i \right)^2 + \left(\sum_{i=1}^{dim} 0.5ix_i \right)^4$	30	[- 5, 10]	0
Noise	$F_9(x) = \sum_{i=1}^D ix_i^4$	30	[- 1.28, 1.28]	0

tens of orders of magnitude. However, in F2, F5, and F9, the advantage of BKA is not as apparent as before. In F6, the RIME algorithm has a weak advantage over BKA; in F7 and F8, the WOA algorithm is slightly better than BKA. Although BKA did not achieve the optimal value on all unimodal functions, compared to the optimal algorithm, the difference between BKA and the optimal algorithm is minimal in those functions where BKA did not achieve the optimal solution. It should be emphasized that although BKA has significant advantages in some unimodal functions, its performance is not entirely dominant compared to other algorithms. This means that in specific problem domains and function types, other algorithms still have competitiveness and similar performance. The BKA algorithm achieved the theoretical optimal value of 0 on F10, F11, F13, F15, and F17 in multimodal functions. On F12 and F18, the BKA algorithm achieved results similar to those of other algorithms. While other algorithms are stuck in local optima for F14 and F16, the BKA algorithm still achieves excellent results. These findings show that the BKA algorithm performs well regarding global search and optimization when dealing with multimodal functions. In most multimodal functions, the BKA algorithm can accurately find the theoretical optimal value, demonstrating its powerful effect in global optimization. The BKA algorithm's results in functions F12 and F18 are comparable to those of other algorithms, but they still exhibit the BKA algorithm's effectiveness and robustness in handling complex problems. In contrast to other algorithms, the BKA algorithm can avoid hitting local optima and produce results close to the ideal outcome.

Table 4 Multimodal test functions

Name	Function	D	Range	min
Rastrigin	$F_{10}(x) = \sum_{i=1}^{\dim} [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[- 5.12, 5.12]	0
NCRastrigin	$F_{11}(x) = \sum_{i=1}^{\dim} [y_i^2 - 10 \cos(2\pi y_i) + 10],$ $y_i = \begin{cases} x_i, & x_i < 0.5 \\ \text{round}(2x_i)/2, & x_i > 0.5 \end{cases}$	30	[- 5.12, 5.12]	0
Ackley	$f_{12}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{\dim} \sum_{i=1}^{\dim} x_i^2}\right) + \exp\left(\frac{1}{\dim} \sum_{i=1}^{\dim} \cos(2\pi x_i)\right) + 20 - \exp(1)$	30	[- 50, 50]	0
Griewank	$F_{13}(x) = \frac{1}{4000} \sum_{i=1}^{\dim} x_i^2 - \prod_{i=1}^{\dim} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[- 600, 600]	0
Alpine	$f_{14}(x) = \sum_{i=1}^{\dim} x_i \times \sin(x_i) + 0.1x_i $	30	[- 10, 10]	0
Weierstrass	$f_{15}(x) = \sum_{i=1}^{\dim} \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_{i+0.5}))] \right) - \dim \cdot \sum_{k=0}^{k_{\max}} [a^k \cos \geq (2\pi b^k \cdot 0.5)], a = 0.5, b = 3, k_{\max} = 20$	30	[- 1, 1]	0
Solomon	$f_{16}(x) = 1 - \cos\left(2\pi \sqrt{\frac{\sum_{i=1}^{\dim} x_i^2}{\dim}}\right) + 0.1 \sqrt{\sum_{i=1}^{\dim} x_i^2}$	30	[- 100, 100]	0
Bohachevsky	$f_{17}(x) = \sum_{i=1}^{D-1} \left[x_i^2 + 2x_{i+1}^2 - 0.3 \cos(3\pi x_i) \right] - 0.4 \cos(4\pi x_{i+1}) + 0.7$	30	[- 10, 10]	0
Generalized schaffer	$f_{18} = 0.5 + \pi \left(\left(\sin\left(\sum_{i=1}^D x_i^2\right) \right)^2 - 0.5 \right) \times \left(1 + 0.001 \left(\sum_{i=1}^D x_i^2 \right) \right)^{-2}$	30	[- 100, 100]	0

Figure 8 shows the search surface graph of the benchmark function, the historical search process of BKA, the average convergence curve of fitness values, and the average convergence curve. The first column displays the search space of each algorithm, and observing the search surface of the search space can provide a more precise and intuitive understanding of the characteristics of the function. The graph shows that F1, F2, and F8 have only one extreme value, while F12, F13, and F17 have multiple extreme values. The second column depicts the historical search process of BKA on a global scale, where the red dots represent the positions of the optimal individuals in each generation of BKA, and the blue dots represent the positions of ordinary individuals. Observing the images of the historical search process makes it possible to gain a more intuitive understanding of the distribution of BKA and the changes in individual positions during the iteration process. The

Table 5 Simulation results of BKA and comparative algorithm on F1-F18

function	Index	BKA	MVO	SCA	GWO	MPA	RIME	ALO	WOA	STOA	DO
F1	Avg	9.68E-81	1.52E-02	5.25E-10	4.94E-56	2.73E-30	1.68E-02	7.19E-09	1.56E-74	1.40E-16	1.04E-11
	Std	3.06E-80	5.71E-03	1.66E-09	1.53E-55	4.88E-30	1.90E-02	2.43E-09	4.16E-74	2.31E-16	1.09E-11
	Rank	1	9	7	3	4	10	8	2	5	6
F2	Avg	2.47E-49	4.95E-02	1.93E-09	3.36E-33	4.39E-17	3.10E-02	5.01E-01	4.58E-51	1.35E-09	9.53E-07
	Std	5.52E-49	3.22E-02	2.51E-09	3.23E-33	4.32E-17	9.48E-03	7.06E-01	9.00E-51	2.38E-09	7.13E-07
	Rank	2	9	6	3	4	8	10	1	5	7
F3	Avg	4.20E-92	1.54E-03	1.29E-05	1.12E-26	1.54E-16	4.21E-03	2.44E-04	3.91E+00	1.56E-12	8.79E-08
	Std	1.33E-91	1.30E-03	2.34E-05	2.98E-26	2.47E-16	2.91E-03	5.83E-04	5.58E+00	3.12E-12	9.93E-08
	Rank	1	8	6	2	3	9	7	10	4	5
F4	Avg	1.81E-38	9.71E-03	1.67E-04	2.08E-19	1.27E-13	1.55E-02	1.74E-04	3.71E-01	1.41E-07	9.24E-06
	Std	5.71E-38	3.48E-03	2.42E-04	2.71E-19	9.35E-14	5.31E-03	1.19E-04	6.62E-01	1.20E-07	9.14E-06
	Rank	1	8	6	2	3	9	7	10	4	5
F5	Avg	3.04E-04	2.56E-03	1.54E-03	4.89E-04	7.71E-04	3.43E-03	3.64E-02	2.16E-03	2.92E-03	2.53E-03
	Std	1.83E-04	1.79E-03	8.77E-04	2.77E-04	8.52E-04	1.07E-03	2.26E-02	2.52E-03	2.50E-03	1.57E-03
	Rank	1	7	4	2	3	9	10	5	8	6
F6	Avg	4.69E-118	3.60E-08	1.12E-29	5.30E-121	2.14E-62	4.97E-11	2.85E-07	1.42E-109	2.21E-39	8.00E-15
	Std	1.48E-117	3.23E-08	3.35E-29	1.67E-120	3.96E-62	9.49E-11	1.38E-07	4.50E-109	6.99E-39	1.03E-14
	Rank	2	9	6	1	4	8	10	3	5	7
F7	Avg	5.40E-79	1.44E-03	4.22E-14	1.53E-58	1.18E-31	1.27E-03	2.04E-07	5.27E-82	1.59E-18	1.58E-12
	Std	1.70E-78	1.45E-03	8.55E-14	4.49E-58	1.58E-31	9.73E-04	2.30E-07	1.04E-81	2.61E-18	1.28E-12
	Rank	2	10	6	3	4	9	8	1	5	7
F8	Avg	1.78E-70	5.92E-04	6.09E-15	7.25E-57	3.30E-31	1.45E-03	1.82E-10	3.74E-80	2.50E-18	1.98E-12
	Std	5.61E-70	2.25E-04	1.25E-14	2.09E-56	4.05E-31	5.36E-04	1.01E-10	1.18E-79	3.91E-18	3.31E-12
	Rank	2	9	6	3	4	10	8	1	5	7
F9	Avg	3.89E-04	8.34E-02	2.22E-03	5.99E-04	4.62E-04	5.66E-02	2.06E-02	2.33E-03	2.34E-03	8.16E-03
	Std	2.54E-04	4.01E-02	2.43E-03	3.25E-04	1.66E-04	3.68E-02	1.13E-02	2.93E-03	2.19E-03	4.82E-03
	Rank	1	10	4	3	2	9	8	5	6	7

Table 5 (continued)

function	Index	BKA	MVO	SCA	GWO	MPA	RIME	ALO	WOA	STOA	DO
F10	Avg	0.00E+00	1.47E+01	5.31E-03	1.42E-15	0.00E+00	3.89E+00	2.33E+01	2.84E-15	2.36E+00	3.38E+00
	Std	0.00E+00	7.96E+00	1.14E-02	4.49E-15	0.00E+00	1.78E+00	8.66E+00	8.99E-15	4.22E+00	2.35E+00
	Rank	1	8	4	2	1	7	9	3	5	6
F11	Avg	0.00E+00	1.36E+01	3.97E+00	1.73E+00	2.01E-01	2.30E+00	1.93E+01	1.78E-16	3.10E+00	1.40E+00
	Std	0.00E+00	3.78E+00	5.61E+00	1.99E+00	6.32E-01	9.47E-01	6.52E+00	5.62E-16	2.33E+00	1.26E+00
	Rank	1	9	8	5	3	6	10	2	7	4
F12	Avg	4.44E-16	6.16E+00	2.00E+01	7.19E-15	5.16E-14	2.09E+00	2.31E-01	4.71E-15	2.00E+01	9.99E+00
	Std	0.00E+00	9.54E+00	1.19E-03	1.12E-15	1.11E-13	6.28E+00	4.87E-01	1.50E-15	1.89E-04	1.05E+01
	Rank	1	7	10	3	4	6	5	2	9	8
F13	Avg	0.00E+00	3.37E-01	1.11E-01	1.80E-02	0.00E+00	1.70E-01	2.47E-01	3.67E-02	7.23E-02	8.59E-02
	Std	0.00E+00	1.09E-01	2.08E-01	1.34E-02	0.00E+00	7.36E-02	9.20E-02	8.15E-02	1.03E-01	6.98E-02
	Rank	1	9	6	2	1	7	8	3	4	5
F14	Avg	2.89E-50	2.05E-01	2.23E-03	1.17E-04	4.46E-16	1.56E-02	4.80E-01	4.08E-01	4.42E-02	4.52E-02
	Std	4.25E-50	1.40E-01	6.86E-03	2.11E-04	9.52E-16	1.64E-02	6.62E-01	1.29E+00	1.40E-01	1.02E-01
	Rank	1	8	4	3	2	5	10	9	6	7
F15	Avg	0.00E+00	1.14E+00	0.00E+00	1.67E+00	0.00E+00	1.83E-01	0.00E+00	0.00E+00	0.00E+00	2.58E-01
	Std	0.00E+00	4.22E-01	0.00E+00	1.60E+00	0.00E+00	1.52E-01	0.00E+00	0.00E+00	0.00E+00	2.14E-01
	Rank	1	3	1	4	1	2	1	1	1	2
F16	Avg	1.21E-45	1.29E-01	9.95E-02	9.95E-02	9.95E-02	1.29E-01	6.96E-01	1.29E-01	9.95E-02	1.29E-01
	Std	3.82E-45	9.44E-02	2.04E-06	4.94E-10	5.19E-17	9.44E-02	2.57E-01	9.44E-02	4.43E-08	9.44E-02
	Rank	1	3	2	2	2	3	4	3	2	3
F17	Avg	0.00E+00	1.79E-01	3.86E-14	0.00E+00	0.00E+00	4.59E-04	1.19E+00	0.00E+00	0.00E+00	4.65E-13
	Std	0.00E+00	4.25E-01	1.17E-13	0.00E+00	0.00E+00	2.54E-04	1.07E+00	0.00E+00	0.00E+00	7.95E-13
	Rank	1	5	2	1	1	4	1548nubd246789	1	1	3

Table 5 (continued)

function	Index	BKA	MVO	SCA	GWO	MPA	RIME	ALO	WOA	STOA	DO
F18	Avg	3.98E-01	3.98E-01	4.00E-01	3.98E-01						
	Std	0.00E+00	9.55E-07	1.89E-03	2.95E-05	5.17E-15	9.99E-07	6.58E-14	8.18E-06	3.43E-04	5.09E-11
	Rank	1	1	2	1	1	1	1	1	1	1
	Average Ranking	1.22	7.33	5.00	2.50	2.61	6.78	7.29	3.50	4.61	5.33
	Final Ranking	1	10	6	2	3	8	9	4	5	7

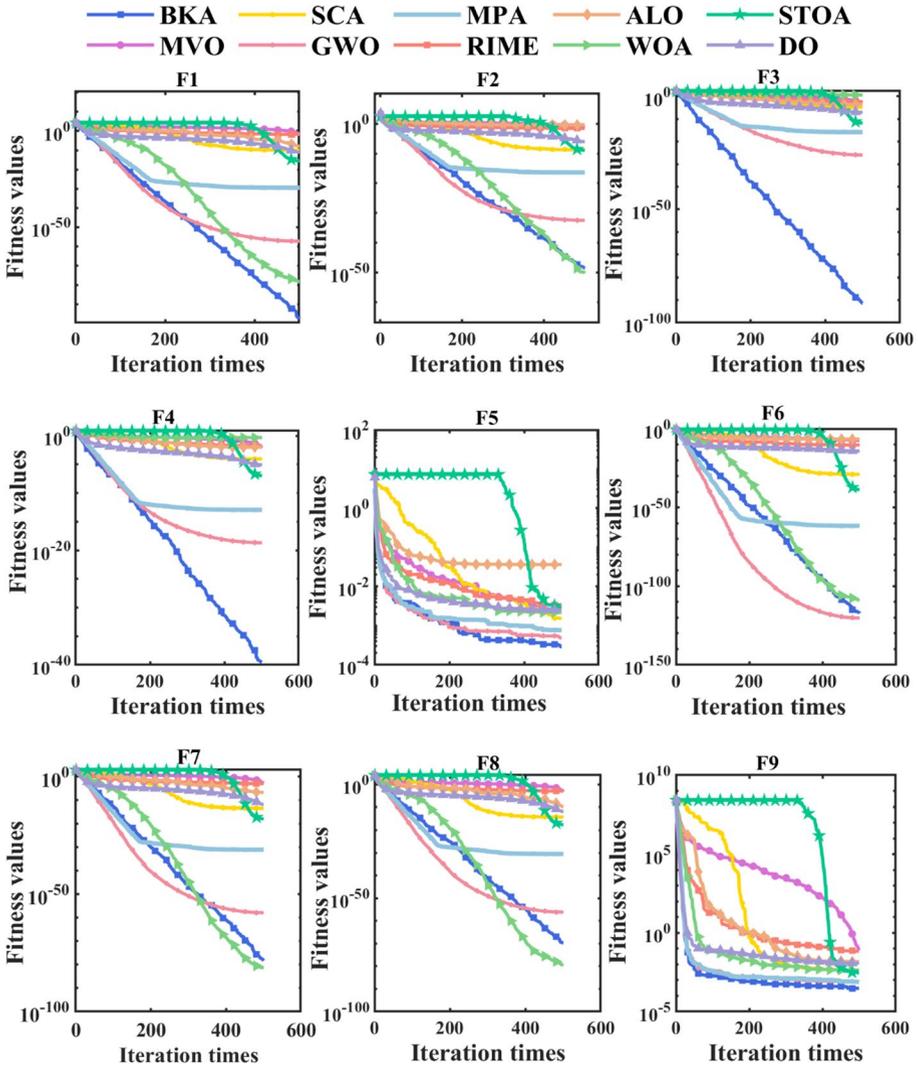


Fig. 6 Convergence analysis of the proposed BKA and competitor algorithms in unimodal functions in dimension 10

intermediate fitness image of BKA represents the average target optimal values of all dimensions during each iteration process in the third column, which also shows the average trend of the population’s evolution. The average fitness value of the BKA algorithm exhibits strong oscillations in the early iterations, which gradually weaken and tend to flatten out, as seen in the images. This reveals that the BKA algorithm has been fully explored in its early stages and extensively searched and optimized globally.

Meanwhile, in the later stages of the iteration, we can also observe significant short-term oscillations. This reflects the BKA algorithm’s continuous attempts to jump out of the local optimal value in the later stage to find higher accuracy and better solutions. This

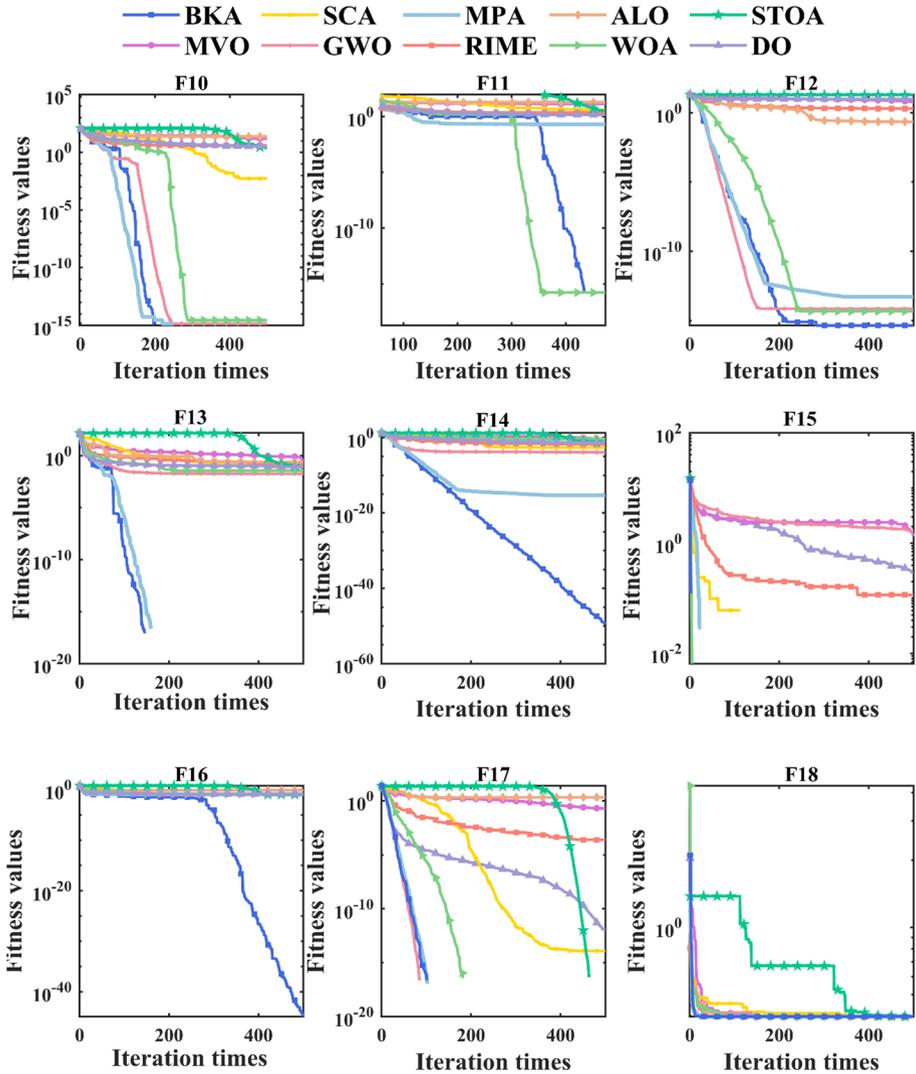


Fig. 7 Convergence analysis of the proposed BKA and competitor algorithms in multimodal functions in dimension 10

short-term oscillation indicates that the BKA algorithm has a certain degree of convergence and continuously strives to improve the quality of the key in the later stages. Overall, the BKA algorithm exhibits a strategy of exploration before development during the optimization process. The algorithm uses large oscillation amplitudes in the early stages to identify potential optimization directions. In the later stage, the BKA algorithm focuses more on fine-tuning and optimization, constantly trying to jump out of the local optimal solution to converge to higher accuracy and better results. The fourth column displays an image of the average convergence curve, which shows the optimal solution obtained by the BKA algorithm throughout the entire iteration process. The multimodal function

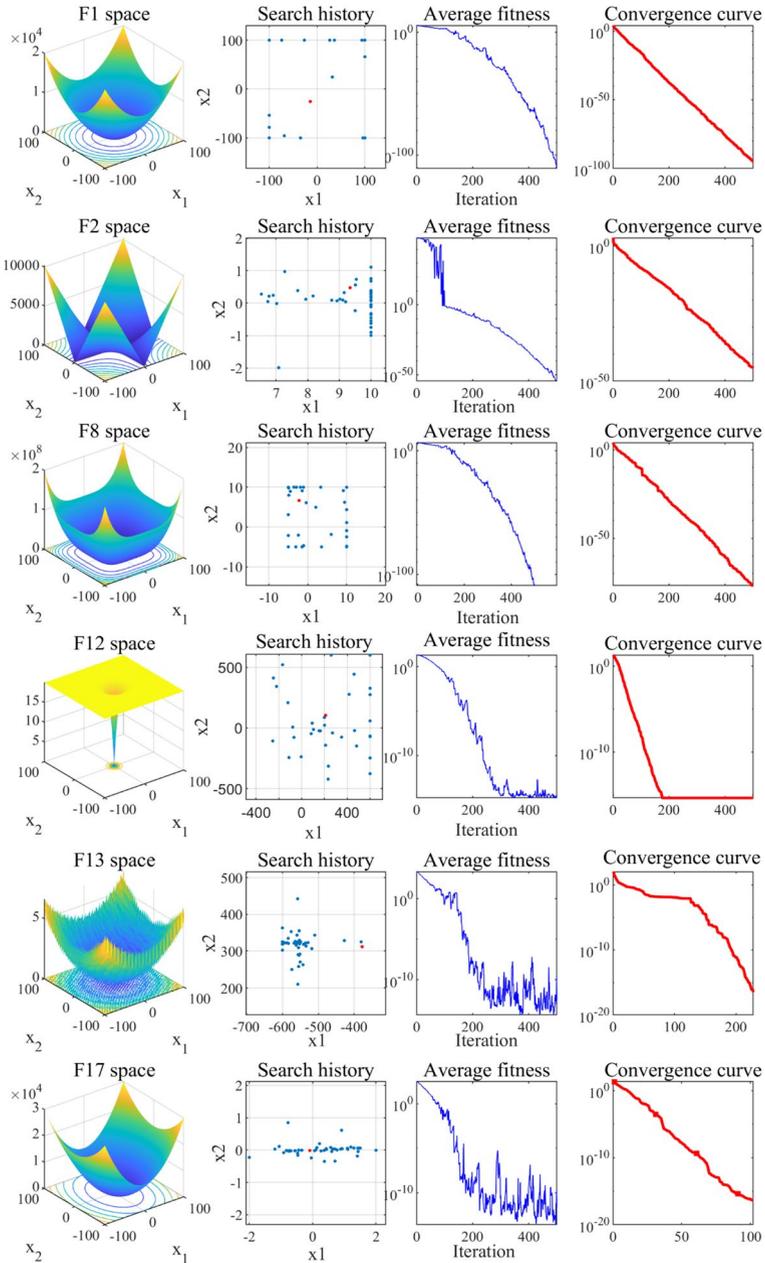


Fig. 8 Search space, search history, average fitness, and convergence curve of BKA algorithm

curve decreases gradually during convergence, while the unimodal function curve rapidly decreases as the number of iterations rises. The ability of the BKA algorithm to quickly exit the local extremum and gradually inch closer to the global optimal value during the optimization process is reflected in this trend.

3.3.2 Evaluation of the CEC-2017 suite test

The CEC-2017 suite is chosen as the testing project in this experiment to gauge BKA's effectiveness in resolving optimization issues. The CEC-2017 set contains four different kinds of benchmark functions. It should be noted that the instability of the F2 function may lead to unpredictable optimization results, resulting in uncertain and inconsistent results when evaluating algorithm performance. The decision-maker decides to eliminate the F2 function from the CEC-2017 test suite to guarantee the test set's validity and consistency. The search domain for all functions in this test suite is $[-100, 100]$, and each test function has ten dimensions. The simulation results of all algorithms are obtained using 30 search agents with 1000 iterations and 10 independent runs.

In calculating the CEC-2017 test set, the outcomes of our algorithm and the comparison algorithm are shown in Table 6, with the best effect denoted in bold. From the data in Table 6, it can be concluded that in the 29 test functions of CEC-2017, the BKA algorithm achieved 21 optimal results, accounting for 72.4%, surpassing the other eight algorithms. A typical statistical chart in the field of statistics is the box plot. Its resemblance to a box's shape led to its name. The box chart can calculate the degree of dispersion of univariate data and clearly and intuitively display the degree of dispersion and distribution interval while highlighting abnormal data values. The box's upper and lower boundaries correspond to the upper and lower quartiles of the data, respectively, and the box's median represents the middle point of the data. The shorter the length of the box, the more concentrated the data. The longer the box length, the more scattered the data, and the worse the stability. Figure 9 shows the box plots of the BKA algorithm and its comparison algorithm on F3, F8, F9, F10, F14, F15, F20, and F26. By observing the chart, we can draw some conclusions. Firstly, the box plot shows that BKA, GJO, PSO, AVOA, and SHO algorithms have almost no outliers, indicating their high stability. This means that on these benchmark functions, the performance of these algorithms is relatively consistent, without significant performance fluctuations or anomalies. Secondly, by observing the box length, we can see that the box length of the BKA algorithm is shorter and at a lower position. This means that the BKA algorithm has a slight difference in the solution set on these benchmark functions, which means its solution accuracy is relatively high. Firstly, the box plot shows that BKA, GJO, PSO, AVOA, and SHO algorithms have almost no outliers, indicating their high stability. This means that on these benchmark functions, the performance of these algorithms is relatively consistent, without significant performance fluctuations or anomalies. Secondly, by observing the box length, we can see that the box length of the BKA algorithm is shorter and at a lower position. As a result, the BKA algorithm's solution set difference for these benchmark functions is minimal, demonstrating a high level of solution accuracy.

The heat map is a graphical representation based on color coding, which represents the size of data through the strength, depth, and different colors of colors, allowing readers to have a more intuitive understanding of the correlations and trends between data. In Fig. 10, the darker the color, the greater the error of the algorithm. The figure indicates that all algorithms perform poorly for functions F1, F2, F12, F13, F15, F18, F19, and F30, indicating that these functions are relatively difficult. In addition, the SSA algorithm has significant errors in most functions, proving that its performance is weak and cannot effectively solve these problems. Figure 11 shows the total running time of each algorithm on the CEC-2017 test set. Observing the graph, it can be seen that the running time of the BKA algorithm is at a relatively high level, with a difference of no more than 20 s compared to the PSO with the shortest running time. However, it is encouraging to note that in this test set, the performance of the BKA

Table 6 Simulation results of BKA and comparative algorithm on CEC-2017 test set

Index	BKA	GJO	PSO	AVOA	SHO	SCSO	SSA	AO	COA
F1	Avg	3.16E+08	8.47E+08	3.46E+03	9.75E+08	1.15E+08	2.19E+10	7.55E+05	8.44E+09
	Std	1.95E+08	3.01E+05	4.39E+03	1.10E+09	3.06E+08	8.97E+07	5.27E+05	3.49E+09
F3	Avg	3.00E+02	3.03E+02	3.00E+02	9.35E+03	5.10E+03	3.40E+04	4.08E+03	2.16E+04
	Std	1.87E-01	5.91E-01	4.10E-06	4.39E+03	5.56E+03	3.35E+02	2.55E+03	4.28E+03
F4	Avg	4.10E+02	4.10E+02	4.18E+02	4.38E+02	4.34E+02	4.81E+03	4.33E+02	1.34E+03
	Std	1.82E+01	6.77E+00	1.48E+01	1.22E+01	1.43E+01	1.03E+02	2.75E+01	2.98E+02
F5	Avg	5.27E+02	5.33E+02	5.38E+02	5.38E+02	5.35E+02	6.23E+02	5.28E+02	5.72E+02
	Std	1.58E+01	4.25E+00	1.35E+01	1.13E+01	1.33E+01	4.11E+00	1.32E+01	1.03E+01
F6	Avg	6.17E+02	6.04E+02	6.14E+02	6.11E+02	6.19E+02	6.48E+02	6.13E+02	6.33E+02
	Std	5.58E+00	2.13E+00	6.23E+00	7.66E+00	5.62E+00	1.12E+00	5.75E+00	4.45E+00
F7	Avg	7.99E+02	7.54E+02	7.89E+02	7.68E+02	7.60E+02	1.00E+03	7.62E+02	8.54E+02
	Std	3.77E+01	2.69E+01	3.13E+01	2.36E+01	1.95E+01	1.82E+01	8.43E+00	3.06E+01
F8	Avg	8.28E+02	8.31E+02	8.46E+02	8.57E+02	8.31E+02	9.54E+02	8.31E+02	8.76E+02
	Std	5.64E+00	1.11E+01	2.12E+01	1.62E+01	1.00E+01	1.01E+00	8.97E+00	1.19E+01
F9	Avg	1.14E+03	9.34E+02	1.25E+03	1.16E+03	1.09E+03	3.43E+03	1.00E+03	1.68E+03
	Std	1.16E+02	5.55E+01	1.62E+02	2.97E+02	1.93E+02	1.65E+02	3.67E+02	2.58E+02
F10	Avg	1.80E+03	1.62E+03	1.99E+03	1.98E+03	1.85E+03	4.56E+03	1.79E+03	2.62E+03
	Std	3.17E+02	3.12E+02	4.36E+02	2.79E+02	2.35E+02	2.91E+02	5.66E+01	1.94E+02
F11	Avg	1.14E+03	1.18E+03	1.14E+03	1.16E+03	1.16E+03	6.49E+04	1.19E+03	1.83E+03
	Std	3.06E+01	5.63E+01	1.16E+01	6.96E+01	1.01E+01	9.19E+02	6.53E+01	6.11E+02
F12	Avg	2.52E+04	5.53E+05	4.63E+05	1.20E+06	7.19E+05	3.12E+09	4.96E+06	2.22E+08
	Std	3.85E+04	7.60E+05	4.57E+05	1.53E+06	5.95E+05	2.98E+05	3.78E+07	6.30E+06
F13	Avg	2.23E+03	1.49E+04	1.67E+04	7.15E+03	9.06E+03	1.73E+09	1.90E+04	5.13E+05
	Std	9.92E+02	8.74E+03	1.14E+04	6.76E+03	4.88E+03	1.72E+07	9.31E+03	5.43E+05
F14	Avg	1.48E+03	2.95E+03	2.26E+03	2.21E+03	3.98E+03	1.69E+03	2.64E+03	1.57E+03
	Std	3.17E+01	1.87E+03	1.17E+03	1.06E+03	1.47E+03	1.89E+03	8.11E+00	7.75E+01
F15	Avg	1.55E+03	4.00E+03	4.62E+03	3.37E+03	4.12E+03	1.23E+04	6.24E+03	8.05E+03

Table 6 (continued)

Index	BKA	GJO	PSO	AVOA	SHO	SCSO	SSA	AO	COA
F16	Std	1.07E+03	2.29E+03	1.53E+03	1.17E+03	1.74E+03	1.81E+02	1.32E+03	3.63E+03
	Avg	1.81E+03	1.84E+03	1.83E+03	1.80E+03	1.85E+03	3.13E+03	1.77E+03	2.11E+03
F17	Std	1.17E+02	1.11E+02	1.46E+02	9.80E+01	1.50E+02	3.95E+01	1.27E+02	9.87E+01
	Avg	1.75E+03	1.77E+03	1.76E+03	1.75E+03	1.76E+03	2.95E+03	1.77E+03	1.82E+03
F18	Std	1.75E+01	4.05E+01	1.52E+01	1.29E+01	1.64E+01	1.24E+02	1.95E+01	3.01E+01
	Avg	2.29E+03	3.23E+04	9.90E+03	1.89E+04	2.12E+04	9.67E+09	2.80E+04	2.83E+06
F19	Std	4.63E+02	6.42E+03	7.69E+03	8.35E+03	1.58E+04	5.30E+07	1.39E+04	8.73E+06
	Avg	1.96E+03	3.44E+04	7.01E+03	8.67E+03	3.40E+04	3.96E+08	9.45E+03	1.54E+04
F20	Std	4.36E+01	8.16E+04	7.67E+03	5.03E+03	8.03E+04	1.20E+07	9.58E+03	2.68E+04
	Avg	2.07E+03	2.13E+03	2.12E+03	2.07E+03	2.12E+03	2.99E+03	2.10E+03	2.23E+03
F21	Std	2.33E+01	5.90E+01	7.40E+01	5.95E+01	4.57E+01	1.52E+02	5.81E+01	7.81E+01
	Avg	2.28E+03	2.32E+03	2.28E+03	2.32E+03	2.28E+03	2.79E+03	2.30E+03	2.37E+03
F22	Std	6.29E+01	3.88E+01	7.68E+01	4.15E+01	6.29E+01	1.49E+01	4.64E+01	3.99E+01
	Avg	2.35E+03	2.34E+03	2.55E+03	2.41E+03	2.31E+03	5.14E+03	2.31E+03	3.07E+03
F23	Std	1.45E+02	4.13E+01	5.30E+00	1.26E+02	9.35E+00	2.11E+01	3.02E+00	3.10E+02
	Avg	2.63E+03	2.63E+03	2.70E+03	2.66E+03	2.64E+03	4.01E+03	2.63E+03	2.71E+03
F24	Std	1.98E+01	1.26E+01	2.64E+03	1.82E+01	1.07E+01	3.71E+01	1.22E+01	3.19E+01
	Avg	2.78E+03	2.77E+03	2.88E+03	2.73E+03	2.75E+03	3.38E+03	2.74E+03	2.85E+03
F25	Std	2.43E+01	1.04E+01	3.34E+01	1.17E+02	6.91E+01	1.90E+00	8.63E+01	5.74E+01
	Avg	2.91E+03	2.96E+03	2.93E+03	2.94E+03	2.94E+03	4.73E+03	2.93E+03	3.52E+03
F26	Std	1.87E+01	4.36E+01	2.35E+01	2.99E+01	1.76E+01	5.09E+00	2.47E+01	2.05E+02
	Avg	2.99E+03	3.04E+03	3.63E+03	3.44E+03	3.06E+03	5.54E+03	3.01E+03	4.00E+03
F27	Std	2.12E+02	9.99E+01	6.53E+02	4.41E+02	1.23E+02	1.50E+02	1.80E+02	2.57E+02
	Avg	3.10E+03	3.11E+03	3.21E+03	3.10E+03	3.12E+03	4.82E+03	3.10E+03	3.20E+03
Std	1.07E+01	2.90E+01	5.54E+01	1.65E+01	2.93E+01	3.88E+01	3.03E+01	5.57E+00	2.81E+01

Table 6 (continued)

Index	BKA	GJO	PSO	AVOA	SHO	SCSO	SSA	AO	COA
F28	Avg	3.36E+03	3.29E+03	3.38E+03	3.47E+03	3.34E+03	3.94E+03	3.40E+03	3.77E+03
	Std	1.14E+02	1.26E+02	9.86E+01	1.71E+02	1.29E+02	3.74E+00	8.24E+01	1.13E+02
F29	Avg	3.22E+03	3.34E+03	3.29E+03	3.24E+03	3.24E+03	4.19E+03	3.24E+03	3.42E+03
	Std	5.33E+01	4.70E+01	5.01E+01	4.78E+01	4.34E+01	4.83E+01	5.19E+01	1.13E+02
F30	Avg	1.12E+06	7.62E+05	3.88E+05	1.45E+06	2.03E+06	1.94E+08	2.93E+05	5.41E+06
	Std	1.44E+06	8.59E+05	4.59E+05	3.63E+05	1.77E+06	3.48E+06	7.35E+06	6.00E+06

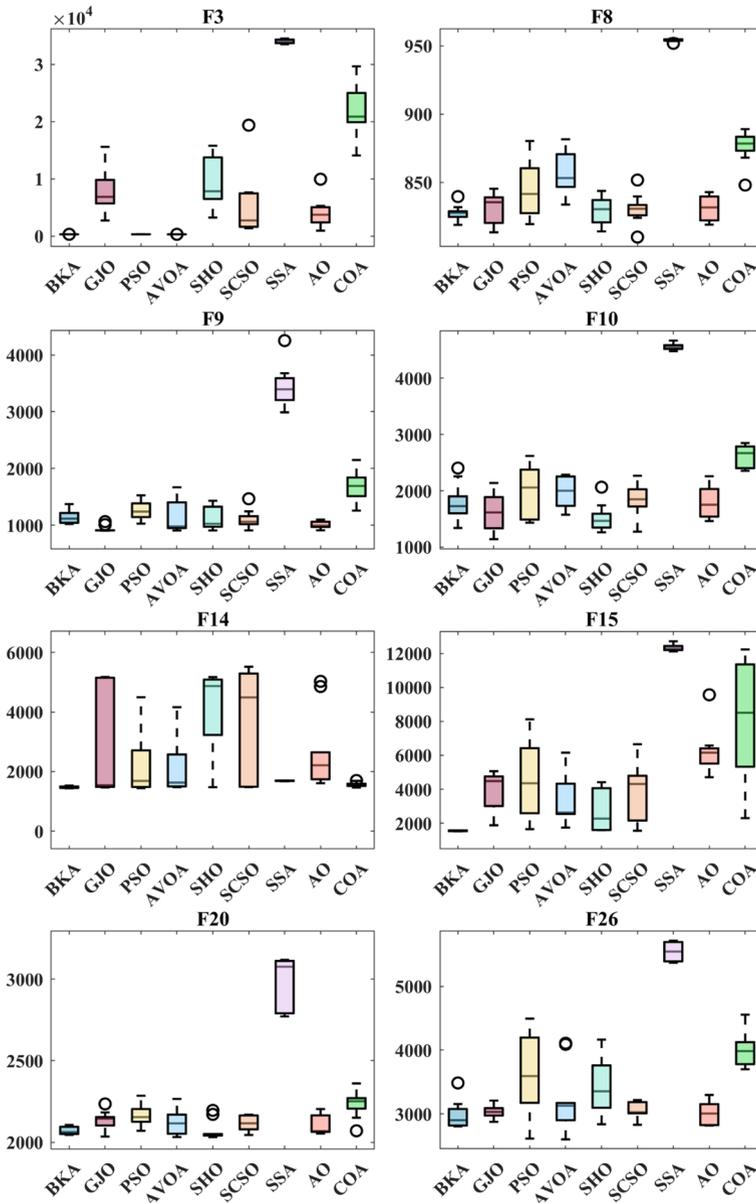


Fig. 9 Boxplot of different algorithms on partial functions of CEC-2017 in dimension 10

algorithm is significantly better than that of PSO and GJO. This indicates that although the BKA algorithm has a slightly longer runtime, it performs well.

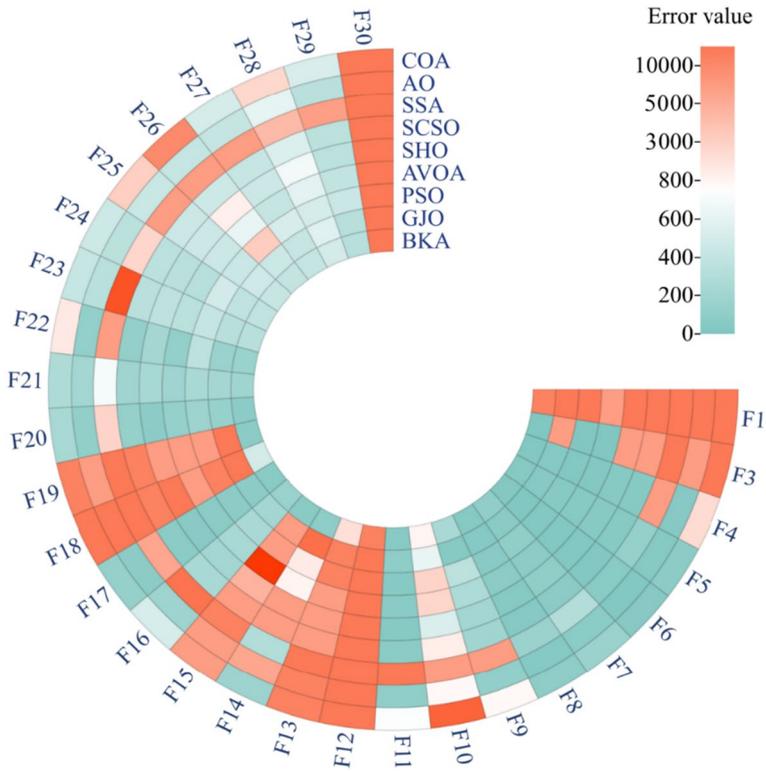


Fig. 10 The error performance of different algorithms on the CEC-2017 test set

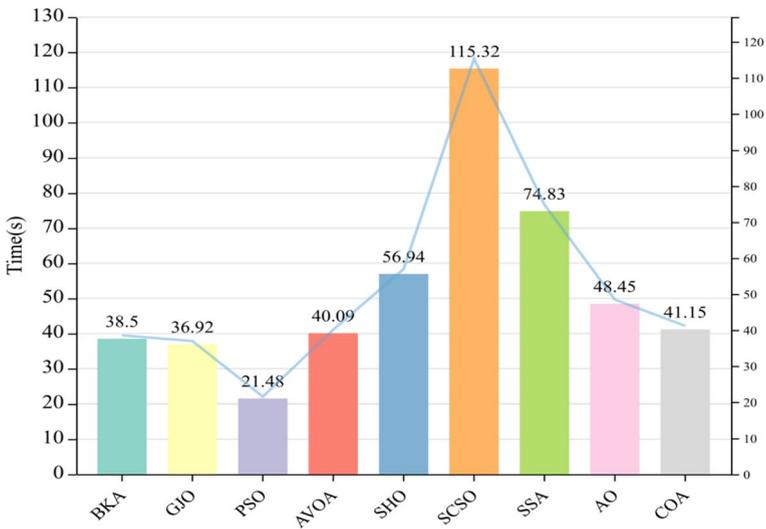


Fig. 11 The total running time of BKA and its comparison algorithm on CEC-2017

3.3.3 Evaluation of the CEC-2022 objective functions

This section further conducts experiments on the algorithm using the most recent CEC-2022 test set to highlight the uniqueness and superiority of the BKA algorithm. The CEC-2022 set includes four different kinds of benchmark functions. In the CEC-2022 test suite, the search domain for all functions is $[-100, 100]$. The CEC-2022 test set provides an updated test set and evaluation metrics aimed at comprehensively evaluating the performance of optimization algorithms. We can better understand its performance in the latest environment by comparing the BKA algorithm with the previously mentioned algorithms. The simulation results of all algorithms are obtained using 30 search agents with 1000 iterations and 10 independent runs.

Table 7 shows that the BKA algorithm outperformed the other eight algorithms by achieving 8 out of the 12 test functions with the best results, or 66.7% of the total. Figure 12 shows that the results of BKA, GJO, PSO, and AVOA perform well on F1, but all have outliers, indicating that the performance of these algorithms is relatively high but not stable enough. Other algorithms perform very well in functions F2 and F6 except for SSA. In Fig. 13, it can be seen that BKA performs stably on all functions, proving that BKA is robust. However, SSA performs poorly in various functions and cannot handle these challenging tasks. According to the results shown in Fig. 14, we can observe the error situation of different algorithms. We can see large areas of high error, especially in the color distribution of heat maps for F1 and F6 functions. This indicates that these algorithms typically perform poorly on these specific functions. This indicates that these two functions pose substantial challenges for algorithms, and optimizing these functions is a relatively complex task for most algorithms. The graph shows that, aside from the SSA and COA algorithms, the performance of other algorithms is generally reasonable. They can achieve lower error levels when processing F1 and F6 functions, demonstrating relatively good performance. Figure 15 shows the total running time of each algorithm on the CEC-2022 test set. Observing the graph, it can be seen that the running time of the BKA algorithm is at a relatively high level, with a difference of no more than 10 s compared to the PSO with the shortest running time. This indicates that although the BKA algorithm has a slightly longer runtime, it performs well.

In summary, the reasons why the BKA algorithm can achieve the best results are as follows: The BKA algorithm adopts the Cauchy distribution strategy and has a strong global search ability. Through the global search strategy, the BKA algorithm is highly likely to discover the global optimal solution. The BKA algorithm introduces a leader strategy. By selecting individuals with high fitness values as leaders, others learn and improve the solution through interaction with the leader.

3.4 Nonparametric statistical analysis

To comprehensively evaluate the performance of BKA, we chose to use the Wilcoxon sign rank test and Friedman test to test BKA and its comparison algorithm. Wilcoxon signed-rank test is a non-parametric test method used to compare two sets of related samples. Its main function is to determine whether there is a significant difference in the median between two related samples. This method can be used to test whether the difference in the median between two sets of related samples is significant. The Friedman test is a non-parametric test used to compare multiple sets of related samples. Its main function is to determine whether there is a significant difference in the median of multiple sets of related

Table 7 Simulation results of BKA and comparative algorithm on CEC-2022 test set

	Index	BKA	GJO	PSO	AVOA	SHO	SCSO	SSA	AO	COA
F1	Avg	3.02E+02	4.27E+03	3.02E+02	3.07E+02	3.86E+03	2.25E+03	1.07E+04	1.92E+03	8.05E+03
	Std	5.22E+01	1.83E+03	5.71E-01	1.58E+01	2.16E+03	2.11E+03	9.23E+01	1.19E+03	1.98E+03
F2	Avg	4.03E+02	4.41E+02	4.03E+02	4.35E+02	4.44E+02	4.28E+02	7.61E+03	4.79E+02	1.51E+03
	Std	1.17E+02	2.39E+01	3.63E+00	3.59E+01	2.82E+01	3.43E+01	5.95E+01	1.16E+02	6.83E+02
F3	Avg	6.30E+02	6.11E+02	6.29E+02	6.21E+02	6.12E+02	6.19E+02	7.00E+02	6.13E+02	6.50E+02
	Std	9.79E+01	7.89E+00	1.52E+01	1.38E+01	5.45E+00	1.56E+01	4.78E+00	6.33E+00	9.71E+00
F4	Avg	8.20E+02	8.30E+02	8.25E+02	8.30E+02	8.24E+02	8.28E+02	9.02E+02	8.22E+02	8.50E+02
	Std	1.65E+02	1.04E+01	1.30E+01	7.41E+00	4.74E+00	6.34E+00	1.38E+00	4.76E+00	7.95E+00
F5	Avg	1.12E+03	1.02E+03	1.16E+03	1.27E+03	1.15E+03	1.18E+03	2.69E+03	1.09E+03	1.37E+03
	Std	4.31E+02	1.06E+02	1.51E+02	1.57E+02	1.31E+02	2.15E+02	2.57E+02	1.71E+02	1.35E+02
F6	Avg	1.94E+03	8.83E+03	7.84E+03	2.97E+03	4.27E+03	5.15E+03	2.18E+08	1.53E+04	1.44E+07
	Std	6.64E+01	2.26E+03	2.79E+03	1.18E+03	1.51E+03	2.20E+03	1.77E+07	1.09E+04	1.52E+07
F7	Avg	2.04E+03	2.04E+03	2.06E+03	2.04E+03	2.04E+03	2.06E+03	2.72E+03	2.04E+03	2.10E+03
	Std	9.28E+01	1.95E+01	2.71E+01	7.57E+00	2.36E+01	2.28E+01	2.39E+02	1.02E+01	1.45E+01
F8	Avg	2.22E+03	2.23E+03	2.23E+03	2.23E+03	2.22E+03	2.23E+03	2.99E+03	2.23E+03	2.24E+03
	Std	1.57E+02	4.10E+00	2.51E+00	2.60E+00	3.10E+00	3.30E+00	5.80E+01	7.31E+00	1.72E+01
F9	Avg	2.53E+03	2.60E+03	2.53E+03	2.56E+03	2.60E+03	2.56E+03	3.06E+03	2.57E+03	2.75E+03
	Std	1.92E+02	4.83E+01	1.80E-03	6.20E+01	1.58E+01	3.11E+01	6.07E+00	2.95E+01	4.37E+01
F10	Avg	2.67E+03	2.60E+03	2.56E+03	2.57E+03	2.55E+03	2.54E+03	5.18E+03	2.56E+03	2.72E+03
	Std	2.36E+02	5.31E+01	7.29E+01	6.74E+01	6.03E+01	6.33E+01	5.02E+01	6.56E+01	1.99E+02
F11	Avg	2.71E+03	2.89E+03	2.69E+03	2.76E+03	2.96E+03	2.98E+03	5.13E+03	2.67E+03	3.89E+03
	Std	1.75E+02	2.60E+02	1.28E+02	1.59E+02	2.52E+02	2.22E+02	7.62E+00	1.04E+02	3.92E+02
F12	Avg	2.87E+03	2.87E+03	2.94E+03	2.87E+03	2.90E+03	2.87E+03	4.73E+03	2.87E+03	2.98E+03
	Std	5.49E+00	7.26E+00	6.36E+01	2.66E+00	3.64E+01	9.90E+00	2.68E+01	1.90E+00	9.10E+01

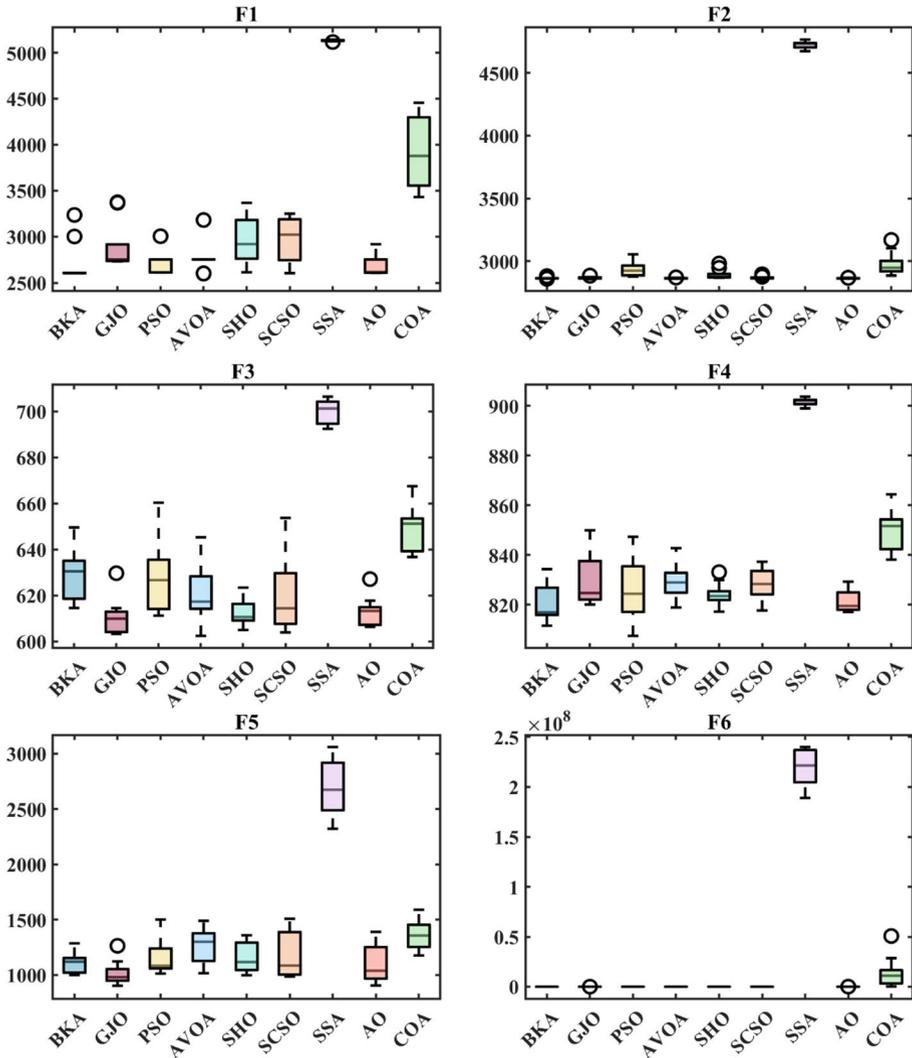


Fig. 12 Box plots of different algorithms on the CEC-2022 test set (F1–F6)

samples, which can be used to test whether there is a significant difference in the median of multiple sets of related samples.

Tables 8 and 9 list the results of Wilcoxon testing for different algorithms on different test sets, all of which are based on a 95% significance level ($\alpha=0.05$). In Tables 8 and 9, the symbol "+" indicates that the reference algorithm performs better than the comparison algorithm, the symbol "-" indicates that the reference algorithm is not as good as the comparison algorithm. The symbol "=" indicates no difference in significance between the reference and comparison algorithms. By observing the last row in the table, we can conclude that the BKA algorithm has a smaller number of '-', while there are more '+' and '='. This indicates that, in most cases, the performance of the BKA algorithm is not weaker than that of the comparison algorithm. Tables 10 and 11 list the Friedman test rankings and average rankings of

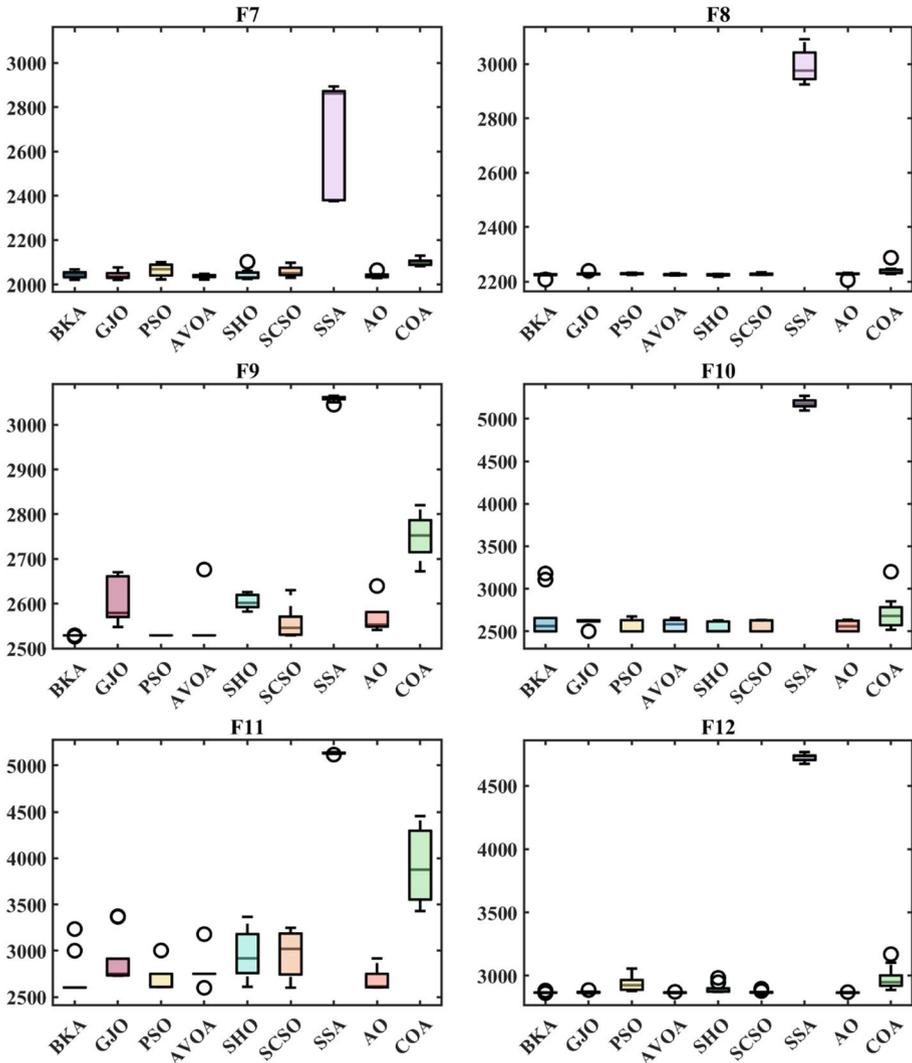


Fig. 13 Box plots of different algorithms on the CEC-2022 test set (F7–F12)

different algorithms on different test sets. By observing Tables 10 and 11, we can see that the BKA algorithm ranks first in most benchmark functions and first in average rankings. These statistical data demonstrate the BKA algorithm’s excellent performance on a single benchmark function but, more importantly, by evaluating its overall performance, its practicality in multiple optimization problems can be more reliably evaluated.

3.5 Effectiveness analysis

The overall effectiveness (OE) of the BKA algorithm and other contender algorithms are computed by Eq. (12) and reported in Table 12, where the parameter N is the total number of test

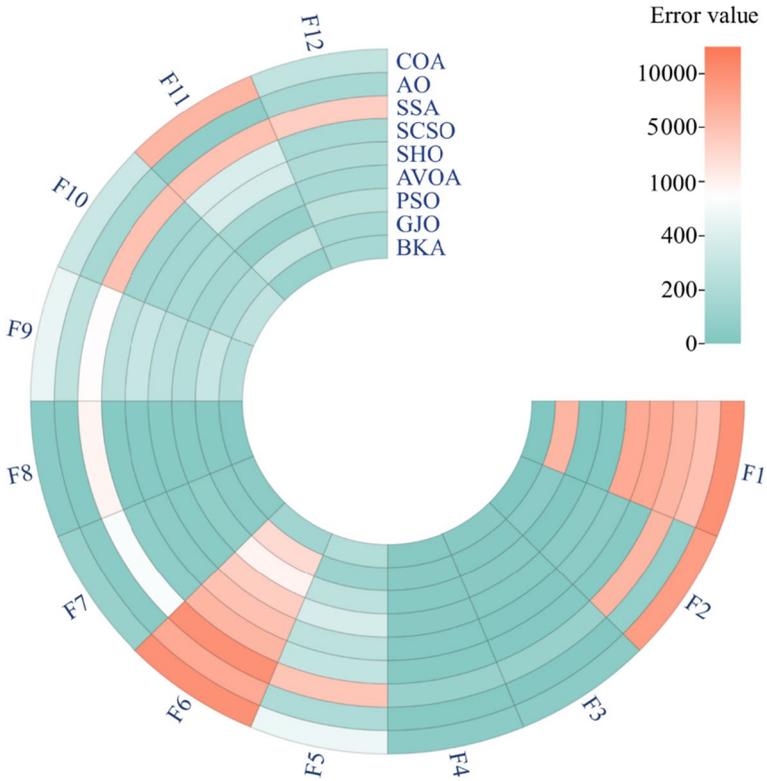


Fig. 14 The error performance of different algorithms on the CEC–2022 test set

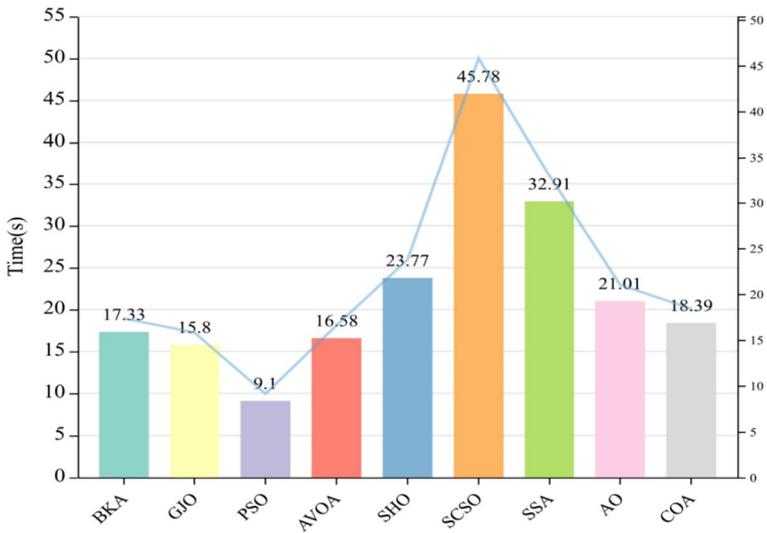


Fig. 15 The total running time of BKA and its comparison algorithm on CEC–2022

Table 8 The Wilcoxon test results of BKA and other comparative algorithms on the CEC-2017 test set ($\alpha = 0.05$)

Function	BKA vs GJO	BKA vs PSO	BKA vs AVOA	BKA vs SHO	BKA vs SCSCO	BKA vs SSA	BKA vs AO	BKA vs COA
F1	1.83E-04(+)	1.83E-04(+)	1.83E-04(-)	1.83E-04(+)	5.39E-02(=)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)
F3	1.83E-04(+)	1.83E-04(+)	1.83E-04(-)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)
F4	2.83E-03(+)	1.04E-01(=)	1.76E-01(=)	2.20E-03(+)	2.20E-03(+)	1.83E-04(+)	3.61E-03(+)	1.83E-04(+)
F5	1.86E-01(=)	1.62E-01(=)	3.07E-01(=)	1.21E-01(=)	1.86E-01(=)	1.83E-04(+)	8.50E-01(=)	1.83E-04(+)
F6	7.69E-04(-)	2.73E-01(=)	6.40E-02(=)	5.39E-02(=)	9.10E-01(=)	1.83E-04(+)	1.62E-01(=)	1.83E-04(+)
F7	5.80E-03(-)	4.73E-01(=)	1.04E-01(=)	9.10E-01(=)	9.11E-03(-)	1.83E-04(+)	1.73E-02(-)	5.80E-03(+)
F8	5.21E-01(=)	3.12E-02(+)	3.30E-04(+)	6.23E-01(=)	3.07E-01(=)	1.83E-04(+)	3.85E-01(=)	1.83E-04(+)
F9	4.40E-04(-)	1.04E-01(=)	4.27E-01(=)	3.07E-01(=)	2.41E-01(=)	1.83E-04(+)	3.61E-03(-)	2.46E-04(+)
F10	2.73E-01(=)	3.85E-01(=)	2.73E-01(=)	3.76E-02(-)	5.21E-01(=)	1.83E-04(+)	8.50E-01(=)	4.40E-04(+)
F11	7.57E-02(=)	8.50E-01(=)	9.70E-01(=)	7.34E-01(=)	6.78E-01(=)	1.83E-04(+)	7.57E-02(=)	1.83E-04(+)
F12	1.31E-03(+)	5.83E-04(+)	2.83E-03(+)	1.01E-03(+)	1.31E-03(+)	1.83E-04(+)	3.30E-04(+)	1.83E-04(+)
F13	1.83E-04(+)	1.31E-03(+)	7.28E-03(+)	2.46E-04(+)	5.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)
F14	2.57E-02(+)	3.12E-02(+)	2.20E-03(+)	7.69E-04(+)	5.80E-03(+)	1.83E-04(+)	1.83E-04(+)	7.28E-03(+)
F15	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)	5.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)
F16	3.07E-01(=)	1.62E-01(=)	2.73E-01(=)	2.41E-01(=)	8.90E-02(=)	1.83E-04(+)	5.71E-01(=)	1.83E-04(+)
F17	2.57E-02(+)	1.62E-01(=)	1.40E-01(=)	9.70E-01(=)	8.90E-02(=)	1.83E-04(+)	1.73E-02(+)	2.46E-04(+)
F18	1.83E-04(+)	3.30E-04(+)	2.46E-04(+)	2.46E-04(+)	1.31E-03(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)
F19	4.59E-03(+)	1.83E-04(+)	1.83E-04(+)	2.83E-03(+)	2.83E-03(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)
F20	1.73E-02(+)	5.83E-04(+)	1.86E-01(=)	1.04E-01(=)	2.11E-02(+)	1.83E-04(+)	2.41E-01(=)	5.83E-04(+)
F21	4.73E-01(=)	3.76E-02(+)	7.91E-01(=)	1.40E-01(=)	1.00E+00(=)	1.83E-04(+)	1.00E+00(=)	5.80E-03(+)
F22	3.76E-02(+)	1.86E-01(=)	7.34E-01(=)	1.13E-02(+)	4.73E-01(=)	1.83E-04(+)	2.73E-01(=)	3.30E-04(+)
F23	4.73E-01(=)	5.83E-04(+)	2.41E-01(=)	2.83E-03(+)	7.57E-02(=)	1.83E-04(+)	1.86E-01(=)	3.30E-04(+)
F24	6.23E-01(=)	1.83E-04(+)	4.27E-01(=)	1.00E+00(=)	2.41E-01(=)	1.83E-04(+)	2.12E-01(=)	1.31E-03(+)
F25	2.83E-03(+)	6.40E-02(=)	4.59E-03(+)	5.80E-03(+)	2.20E-03(+)	1.83E-04(+)	1.40E-02(+)	1.83E-04(+)
F26	2.12E-01(=)	1.40E-02(+)	2.41E-01(=)	1.13E-02(+)	1.86E-01(=)	1.83E-04(+)	6.23E-01(=)	1.83E-04(+)
F27	2.41E-01(=)	3.30E-04(+)	1.00E+00(=)	4.40E-04(+)	9.70E-01(=)	1.83E-04(+)	1.86E-01(=)	1.83E-04(+)
F28	3.12E-02(+)	4.73E-01(=)	1.72E-02(+)	1.73E-02(+)	1.86E-01(=)	1.83E-04(+)	1.71E-03(+)	1.83E-04(+)

Table 8 (continued)

F29	9.70E-01(=)	4.52E-02(+)	1.13E-02(+)	2.41E-01(=)	3.85E-01(=)	1.83E-04(+)	5.21E-01(=)	1.83E-04(+)
F30	6.23E-01(=)	7.33E-01(=)	5.20E-01(=)	6.77E-01(=)	4.27E-01(=)	1.79E-04(+)	6.77E-01(=)	1.12E-02(+)
+/-/-	14/12/3	16/13/0	10/17/2	15/13/1	10/18/1	29/0/0	12/15/2	29/0/0

Table 9 The Wilcoxon test results of BKA and other comparative algorithms on the CEC-2022 test set ($\alpha = 0.05$)

Function	BKA vs GIO	BKA vs PSO	BKA vs AVOA	BKA vs SHO	BKA vs SCSO	BKA vs SSA	BKA vs AO	BKA vs COA
F1	1.83E-04(+)	2.83E-03(+)	4.52E-02(-)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)
F2	5.83E-04(+)	5.71E-01(=)	9.08E-03(+)	4.59E-03(+)	1.73E-02(+)	1.83E-04(+)	3.30E-04(+)	1.83E-04(+)
F3	7.69E-04(-)	5.71E-01(=)	8.90E-02(=)	7.69E-04(+)	6.40E-02(=)	1.83E-04(+)	7.69E-04(-)	1.01E-03(+)
F4	4.52E-02(+)	3.07E-01(=)	2.11E-02(+)	2.12E-01(=)	2.57E-02(+)	1.83E-04(+)	1.86E-01(=)	1.83E-04(+)
F5	2.11E-02(-)	7.91E-01(=)	3.12E-02(+)	6.78E-01(=)	9.70E-01(=)	1.83E-04(+)	3.07E-01(=)	7.69E-04(+)
F6	1.83E-04(+)	1.83E-04(+)	1.40E-02(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)
F7	7.34E-01(=)	1.04E-01(=)	2.41E-01(=)	4.73E-01(=)	2.12E-01(=)	1.83E-04(+)	3.85E-01(=)	1.83E-04(+)
F8	4.52E-02(+)	1.13E-02(+)	4.27E-01(=)	9.10E-01(=)	1.86E-01(=)	1.83E-04(+)	4.52E-02(+)	5.83E-04(+)
F9	1.83E-04(+)	1.01E-03(+)	1.04E-01(=)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)	1.83E-04(+)
F10	9.10E-01(=)	1.04E-01(=)	9.10E-01(=)	5.21E-01(=)	3.07E-01(=)	1.83E-04(+)	8.50E-01(=)	1.04E-01(=)
F11	1.13E-02(+)	2.11E-02(+)	2.73E-01(=)	7.28E-03(+)	2.11E-02(+)	1.83E-04(+)	2.57E-02(+)	1.83E-04(+)
F12	6.78E-01(=)	2.46E-04(+)	7.34E-01(=)	4.40E-04(+)	2.73E-01(=)	1.83E-04(+)	4.73E-01(=)	1.83E-04(+)
+/-/-	7/3/2	6/6/0	4/7/1	5/7/0	6/6/0	12/0/0	6/5/1	11/1/0

Table 10 The Friedman test ranking of BKA and its comparison algorithm on the CEC-2017 test set

Function	BKA	GJO	PSO	AVOA	SHO	SCSO	SSA	AO	COA
F1	2.20	6.00	4.40	1.00	6.60	3.70	9.00	4.10	8.00
F3	2.00	6.30	3.00	1.00	6.20	4.60	9.00	4.90	8.00
F4	2.00	5.30	2.20	2.70	5.60	5.50	9.00	4.70	8.00
F5	2.90	3.70	4.40	4.20	4.70	4.70	9.00	3.40	8.00
F6	5.20	1.70	4.30	3.20	4.20	5.40	9.00	4.10	7.90
F7	5.30	2.40	5.00	3.20	5.60	3.40	9.00	3.30	7.80
F8	2.80	3.60	5.00	6.60	3.60	3.30	9.00	3.60	7.50
F9	5.00	1.80	5.60	4.00	4.40	4.40	9.00	3.20	7.60
F10	4.20	3.10	4.70	5.40	2.20	4.60	9.00	3.90	7.90
F11	3.60	5.20	3.20	3.50	3.40	3.90	9.00	5.20	8.00
F12	1.60	3.90	4.60	4.30	4.90	3.30	9.00	5.40	8.00
F13	1.10	5.40	5.40	3.10	4.10	4.00	9.00	5.70	7.20
F14	1.60	5.00	4.30	4.70	7.00	6.50	5.40	6.60	3.90
F15	1.10	4.90	5.00	3.60	3.10	4.70	9.00	6.70	6.90
F16	3.20	4.30	4.60	4.20	4.30	4.60	9.00	3.10	7.70
F17	2.90	5.40	4.30	3.80	2.80	4.30	9.00	4.80	7.70
F18	1.10	6.50	4.10	3.10	4.40	4.40	9.00	6.00	6.40
F19	1.30	5.00	3.90	4.50	5.30	5.70	9.00	5.40	4.90
F20	2.70	4.90	5.50	4.50	2.50	4.30	9.00	4.30	7.30
F21	3.50	4.50	5.50	4.00	4.60	4.10	9.00	3.20	6.60
F22	2.90	5.40	3.60	2.90	6.30	3.50	9.00	3.50	7.90
F23	2.40	2.60	7.30	3.80	5.50	3.70	9.00	3.20	7.50
F24	4.00	3.40	7.60	3.70	4.30	2.90	9.00	3.20	6.90
F25	1.60	4.70	3.30	4.90	4.20	4.90	9.00	4.40	8.00
F26	2.80	3.50	5.90	4.30	5.40	3.70	9.00	3.10	7.30
F27	2.70	2.70	7.00	3.20	5.70	3.50	9.00	3.90	7.30
F28	2.50	4.00	2.80	3.90	5.50	4.20	9.00	5.20	7.90
F29	3.20	2.60	5.60	5.40	4.10	3.90	9.00	3.80	7.40
F30	4.00	4.50	3.60	3.00	5.50	5.00	9.00	3.00	7.40
Average	2.81	4.22	4.68	3.78	4.69	4.30	8.88	4.31	7.34
Ranking	1	4	6	2	7	3	9	5	8

functions, and L_i is the number of test functions in which the i -th algorithm is a loser (Nadimi-Shahraki and Zamani 2022). From Table 12, it can be seen that BKA demonstrated its effectiveness with 70.7% excellent results on the CEC-2017 and CEC-2022 test sets, far surpassing other comparative algorithms.

$$OE_i(\%) = \frac{N - L_i}{N} \times 100 \quad (12)$$

Table 11 The Friedman test ranking of BKA and its comparison algorithm on the CEC-2022 test set

Function	BKA	GJO	PSO	AVOA	SHO	SCSO	SSA	AO	COA
F1	1.90	6.40	2.70	1.40	5.70	5.00	8.90	5.00	8.00
F2	2.10	5.40	1.90	4.50	4.90	4.30	9.00	4.90	8.00
F3	5.70	2.40	5.90	4.60	2.80	4.00	9.00	2.90	7.70
F4	2.70	5.40	4.10	5.10	3.70	4.70	9.00	2.60	7.70
F5	4.10	2.60	4.40	6.30	4.20	4.00	9.00	3.30	7.10
F6	1.20	6.00	5.40	2.50	3.10	3.80	9.00	6.00	8.00
F7	4.00	3.80	5.80	3.10	3.40	4.80	9.00	3.50	7.60
F8	2.90	4.20	5.60	3.50	2.80	4.10	9.00	5.40	7.50
F9	1.90	5.60	2.70	2.20	6.20	4.40	9.00	5.00	8.00
F10	4.60	5.30	3.20	4.60	3.70	3.40	9.00	4.70	6.50
F11	2.30	4.70	3.60	3.60	5.60	5.30	9.00	2.90	8.00
F12	3.00	2.90	7.10	3.20	6.00	3.30	9.00	2.90	7.60
Average	3.03	4.56	4.37	3.72	4.34	4.26	8.99	4.09	7.64
Ranking	1	7	6	2	5	4	9	3	8

3.6 Limitation analysis

Although BKA has achieved good results in dealing with optimization problems, it cannot be ignored that this algorithm still has some shortcomings, which can be summarized as follows: this algorithm has not achieved optimal results in solving specific types of optimization problems and has shown insufficient stability in multiple runs. Specifically, the insufficient stability of the algorithm may be due to the uneven distribution of initial parameters, resulting in the search strategy exhibiting variation in multiple runs. In addition, when dealing with complex problems with high-dimensional search spaces, the algorithm may experience premature convergence or repeated convergence during the iteration process, reducing the consistency of the results. Meanwhile, although slightly superior in performance, BKA's running speed is relatively low, which may become a disadvantage in application scenarios that require fast iteration. To improve these limitations, it is recommended to further adjust the initial value distribution, optimize the exploration and utilization mechanism, and consider algorithm acceleration strategies in subsequent research in order to improve the stability and efficiency of the algorithm and better adapt to various complex optimization problems.

4 BKA for solving engineering problems

This section evaluates how well BKA performed in resolving five elaborate engineering design issues: the design of a tension/compression spring, a pressure vessel, a welded beam, a speed reducer, and a three-bar truss design issue. These well-known engineering problems contain numerous equality and inequality constraints, and the ability of BKA to optimize real-world and constrained problems is evaluated from the perspective

Table 12 Effectiveness of the BKA and other competitor algorithms

	BKA (w/t/l)	GJO (w/t/l)	PSO (w/t/l)	AVOA (w/t/l)	SHO (w/t/l)	SCSO (w/t/l)	SSA (w/t/l)	AO (w/t/l)	COA (w/t/l)
CEC-2017	11/10/8	3/2/24	0/3/26	3/3/23	1/3/25	0/1/28	0/0/29	0/2/27	0/0/29
CEC-2022	2/6/4	2/2/8	0/3/9	0/2/10	0/2/10	1/1/10	0/0/12	1/2/9	0/0/12
Total	13/16/12	5/4/32	0/6/35	3/5/33	1/5/35	1/2/38	0/0/41	1/4/36	0/0/41
OE (%)	70.7%	22.0%	14.6%	19.5%	14.6%	7.3%	0%	12.2%	0%

of constraint processing. Here, the constrained issues are transformed into unconstrained problems using a straightforward method of the death penalty.

Solving constrained optimization problems is a crucial task in both optimization theory and applications. There are numerous methods for processing constraints, including operators, decoder functions, representations that preserve feasibility, repair algorithms, and penalty functions. Constrained optimization issues are typically solved using the penalty function method, a popular technique from optimization theory. The objective of the penalty function approach is to introduce a penalty function that transforms the constraint conditions into a component of the objective function, thereby changing the original constraint problem into an unconstrained one. Without considering constraints, the ideal answer to the original issue can be found by modifying the shape and parameters of the penalty function. This study resolves these practical engineering issues using the penalty function method.

4.1 Pressure vessel design

This engineering challenge aims to reduce the cost of producing cylindrical pressure vessels while meeting four constraints. This problem’s resolution can be mathematically stated as follows:

Consider variable $H = [h_1, h_2, h_3, h_4] = [T_s, T_h, R, L]$

$$\text{Minimize } f(H) = 0.6224h_1h_3h_4 + 1.7781h_2h_3^2 + 3.1661h_1^2h_4 + 19.84h_1^2h_3 \quad (13)$$

$$\text{Subject to : } l_1(H) = 0.0193h_3 - h_1 \leq 0, \quad (14)$$

$$l_2(H) = 0.00954h_3 - h_2 \leq 0, \quad (15)$$

$$l_3(H) = 1,296,000 - \pi h_3^2 h_4 - \frac{4}{3} \pi h_3^3 \leq 0, \quad (16)$$

Table 13 The best solutions to the Pressure vessel design problem using various algorithms

Algorithm	Optimal values for variables				Optimal cost
	T_s	T_h	R	L	
BKA	0.778433	0.384690	40.319619	200	5887.364927
GJO	0.778523	0.403115	40.332023	200	5908.557674
PSO	0.958559	0.510067	49.120228	105.66659	6289.337794
AVOA	1.244248	0.615032	64.468805	13.297390	7254.449800
SHO	0.779035	0.384660	40.327793	199.650290	5889.368900
SCSO	0.987235	0.488050	51.151977	89.460729	6347.596799
SSA	56.132650	55.950231	56.158215	55.928870	4.49E+06
AO	1.037249	0.514761	53.556647	74.087978	6582.536753
COA	1.134869	1.674960	56.788030	52.204932	13362.472751

$$l_4(H) = -240 + h_4 \leq 0 \tag{17}$$

Variables range $0 \leq h_j \leq 100, j = 1, 2, 10 \leq h_j \leq 200, j = 3, 4$

BKA has optimized this issue. BKA can obtain the optimal function value $f(H) = 5887.364927$ with the structure variables $H = (0.778433, 0.384690, 40.319619, 200)$. Table 13 displays the optimal values and variables that BKA and its comparison algorithm arrived at, demonstrating how well the algorithm resolved this issue. The algorithm performs better when the numerical value is lower. The results indicate that BKA has discovered a new structure that can achieve lower manufacturing costs than other structures.

4.2 Design issue with tension/compression springs

This engineering challenge aims to reduce the coil’s weight while meeting three criteria. These limitations ensure the coil design adheres to certain engineering limitations and requirements. We can use the following mathematical expression to explain this issue:

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

$$\text{Minimize } f(H) = (h_3 + 2) \times h_2 h_1^2 \tag{18}$$

$$\text{Subject to : } l_1(H) = -\frac{h_2^3 h_3}{71,785 h_1^4} + 1 \leq 0, \tag{19}$$

$$l_2(H) = \frac{4h_2^2 - h_1 h_2}{12,566(h_1^3 h_2 - h_1^4)} + \frac{1}{5,108 h_1^2} - 1 \leq 0, \tag{20}$$

$$65454555 \tag{21}$$

$$l_4(H) = -1 + \frac{h_1 + h_2}{1.5} \leq 0. \tag{22}$$

Table 14 Tension/compression spring design problem optimal outcomes of various algorithms

Algorithm	Optimal values for variables			Optimal cost
	<i>d</i>	<i>D</i>	<i>N</i>	
BKA	0.051173	0.344426	12.047782	0.01267027
GJO	0.050468	0.3276388	13.255784	0.01268300
PSO	0.05	0.3170802	14.076339	0.01274300
AVOA	0.05	0.317425	14.02777	0.012719054
SHO	0.0508	0.334800	11.702	0.012681
SCSO	0.051592	0.354395	11.426859	0.01271702
SSA	0.077347	1.34212	1.960143	0.03179716
AO	0.061211	0.624604	4.419888	0.015023988
COA	0.05571	0.4614	7.11158	0.013048048

Variables range $0.05 \leq h_1 \leq 2, 0.25 \leq h_2 \leq 1.3, 2 \leq h_3 \leq 15$

Table 14 displays the optimal values and variables that BKA and its comparison algorithm arrived at, illustrating how well the algorithm resolved this issue. BKA can obtain the optimal function value $f(H) = 0.01267027$ with the structure variables $H = (0.051173, 0.344426, 12.047782)$. The experiments and comparative analysis results demonstrate that the BKA algorithm can produce better solutions when tackling these issues. This discovery provides engineers and decision-makers with a reliable tool and method to improve the design, planning, and decision-making processes and achieve higher-quality engineering solutions.

4.3 Welded beam design

This engineering challenge aims to minimize the welded beam’s weight while satisfying the four constraints. The welding thickness, rod connection length, rod height, and rod thickness are the four decision variables that we must optimize to describe this issue. For this engineering problem, we can define an objective function to represent the weight of the welded beam, namely:

Consider variable $H = [h_1, h_2, h_3, h_4] = [h, l, t, b]$

Minimize: $(H) = 1.10471h_2h_1^2 + (14 + h_2) \times 0.04811h_3h_4$

$$\text{Subject to : } l_1(H) = -\tau_{\max} + \tau(h) \leq 0, \tag{23}$$

$$l_2(H) = -\sigma_{\max} + \sigma(h) \leq 0, \tag{24}$$

$$l_3(H) = -h_4 + h_1 \leq 0, \tag{25}$$

$$l_4(H) = -5 + 0.10471h_1^2 + (14 + h_2) \times 0.04811h_3h_4 \leq 0, \tag{26}$$

$$l_5(H) = -h_1 + 0.125 \leq 0, \tag{27}$$

$$l_6(H) = -\delta_{\max} + \delta(h) \leq 0, \tag{28}$$

$$l_7(H) = -P_c(h) + P \leq 0, \tag{29}$$

where

$$\tau(h) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{h_2}{2R} + (\tau'')^2}, \tag{30}$$

$$\tau' = \frac{P}{\sqrt{2}h_1h_2}, \tau'' = \frac{MR}{J}, \tag{31}$$

$$M = P\left(L + \frac{h_2}{2}\right), R = \sqrt{\frac{h_2^2}{4} + \left(\frac{h_1 + h_3}{2}\right)^2}, \delta(h) = \frac{4PL^3}{Eh_3^3h_4}, \tag{32}$$

$$J = 2 \left[\sqrt{2}h_1h_2 \left\{ \frac{h_2^2}{12} + \left(\frac{h_1 + h_3}{2} \right)^2 \right\} \right], \sigma(h) = \frac{6PL}{h_4h_3^2}, \tag{33}$$

$$P_c(h) = \frac{4.013E \sqrt{\frac{h_4^2 h_3^2}{36}}}{L^2} \left(1 - \frac{h_3}{2L} \sqrt{\frac{E}{4G}} \right) \tag{34}$$

Variables range $P = 6,000lb, L = 14in, E = 30e6psi, G = 12e6psi,$

$\tau_{max} = 13,000psi, \sigma_{max} = 30,000psi, \delta_{max} = 0.25in, 0.1 \leq h_1 \leq 2, 0.1 \leq h_2 \leq 10, 0.1 \leq h_3 \leq 10, 0.1 \leq h_4 \leq 2.$

BKA can obtain the optimal function value $f(H) = 1.724853$ with the structure variables $H = (0.205730, 3.470488, 9.036622, 0.205730)$. The results in Table 15 indicate that BKA can bring better solutions to solving such problems. After analysis and comparison, the BKA algorithm can obtain better solutions under given constraints through flexible heuristic search methods and optimization mechanisms. It can adapt to different problem characteristics and solving requirements and has a high success rate and accuracy. This discovery gives engineers and decision-makers a reliable tool and method to improve the design and decision-making process and achieve higher-quality engineering solutions.

4.4 Speed reducer design problem

This issue aims to reduce the reducer device’s weight while meeting 11 constraints. To describe this problem, we can use the following mathematical expression:

Consider variable $H = [h_1, h_2, h_3, h_4, h_5, h_6, h_7] = [b, m, p, l_1, l_2, d_1, d_2]$

$$\text{Minimize } f(H) = 0.7854h_1h_2^2(3.3333h_3^2 + 14.9334z_3 - 43.0934) - 1.508h_1(h_6^2 + h_7^2) + 7.4777(h_6^3 + h_7^3) + 0.7854(h_4h_6^2 + h_5h_7^2) \tag{35}$$

$$l_1(H) = \frac{27}{(h_1h_2^2h_3)} - 1 \leq 0 \tag{36}$$

Table 15 The Welded Beam Design Problem’s best outcomes from the various algorithms

Algorithm	Optimal values for variables				Optimal cost
	<i>h</i>	<i>l</i>	<i>t</i>	<i>b</i>	
BKA	0.205730	3.470488	9.036622	0.205730	1.724853
GJO	0.205803	3.468938	9.036642	0.205837	1.725582
PSO	0.209488	3.45523	8.927898	0.215478	1.783039
AVOA	0.20592	3.468021	9.032414	0.205921	1.725545
SHO	0.20585	3.46946	9.03276	0.20591	1.7259
SCSO	0.205709	3.471169	9.036666	0.205731	1.724928
SSA	2.330378	2.682921	2.616964	2.453113	5.20E+14
AO	0.200517	3.654022	9.057898	0.206271	1.749188
COA	0.25499	2.998089	7.969305	0.330805	2.371241

$$l_2(H) = \frac{397.5}{(h_1 h_2^2 h_3^2)} - 1 \leq 0 \tag{37}$$

$$l_3(H) = \frac{1.93h_4^3}{(h_1 h_3 h_6^4)} - 1 \leq 0 \tag{38}$$

$$l_4(H) = \frac{1}{(110h_6^3)} \times \sqrt{16.9 \times 10^6 + \left(\frac{745h_4}{h_2 h_3}\right)^2} - 1 \leq 0 \tag{39}$$

$$l_5(H) = \frac{1.93h_5^3}{(h_2 h_3 h_7^4)} - 1 \leq 0 \tag{40}$$

$$l_6(H) = \frac{1}{(85h_7^3)} \times \sqrt{157.5 \times 10^6 + \left(\frac{745h_5}{h_2 h_3}\right)^2} - 1 \leq 0 \tag{41}$$

$$l_7(H) = \frac{h_2 h_3}{40} - 1 \leq 0 \tag{42}$$

$$l_8(H) = 5 \times \frac{h_2}{h_1} - 1 \leq 0 \tag{43}$$

$$l_9(H) = \frac{h_1}{12h_2} - 1 \leq 0 \tag{44}$$

$$l_{10}(H) = \frac{1.5h_6 + 1.9}{h_4} - 1 \leq 0 \tag{45}$$

Table 16 The best solutions to the Speed reducer design problem using various algorithms

Algorithm	Optimal values for variables							Optimal cost
	<i>b</i>	<i>m</i>	<i>p</i>	<i>l₁</i>	<i>l₂</i>	<i>d1</i>	<i>d2</i>	
BKA	3.5	0.7	17	7.3	7.71532	3.350215	5.286654	2994.47107
GJO	3.5	0.7	17	7.32	7.72122	3.35025	5.28665	2994.80495
PSO	3.5	0.7	17	8.3	7.8	3.352412	5.286715	3005.763
AVOA	3.5	0.7	17	7.3	7.71532	3.350215	5.286654	2994.47109
SHO	3.5	0.7	17	7.3	7.7163	3.3502	5.2867	2994.504
SCSO	3.5	0.7	17	7.32	8.029658	3.350294	5.286794	3001.69686
SSA	7.62	8.33	8.27	8.04	8.038538	7.823429	7.862038	2.11E+16
AO	3.58	0.7	17	7.41	7.843836	3.363452	5.319559	3052.2253
COA	3.5	0.7	25.72	7.33	7.974521	3.569454	5.339685	4940.841

$$l_{11}(H) = \frac{1.1h_7 + 1.9}{h_5} - 1 \leq 0 \tag{46}$$

Variable range $2.6 \leq h_1 \leq 3.6, 0.7 \leq h_2 \leq 0.8, 17 \leq h_3 \leq 28, 7.3 \leq h_4 \leq 8.3$

$$7.3 \leq h_5 \leq 8.3, 2.9 \leq h_2 \leq 3.9, 5 \leq h_3 \leq 5.5$$

The optimal values and corresponding optimal variables that the BKA algorithm and its comparison algorithm arrived at are listed in Table 16. These values offer a simple way to compare how well various algorithms perform when solving problems. BKA can obtain the optimal function value $f(H) = 2994.47107$ with the structure variables $H = (3.5, 0.7, 17, 7.3, 7.71532, 3.350215, 5.286654)$. We can see from comparing the BKA algorithm’s results to those of other algorithms that it solves problems more efficiently and produces better optimal values. This suggests that the BKA algorithm does a better job locating the optimal solution and may be closer to the problem’s overall optimal solution. These optimal variables serve as crucial guides and references for a deeper comprehension of the problem’s solution space and the viability of obtaining optimization results.

4.5 Three-bar truss design problem

This problem aims to reduce the member structure’s weight while maintaining a constant total load. To achieve this goal, we need to consider three constraint conditions: the stress, buckling, and deflection constraints of each steel bar. Firstly, the stress constraint of each steel bar is to ensure that under the design working load, the stress borne by the steel bars in the member will not exceed the limit of their bearing capacity. This is to ensure the safety and reliability of the structure. The limitation of steel bar stress is determined by calculating the strength of the material and the force borne by the steel bar. Secondly, buckling constraint ensures that the member will not experience buckling under stress. Buckling refers to the instability phenomenon of a member under pressure, which may lead to structural failure. To avoid buckling, we need to limit the members’ length, cross-sectional shape, and material selection to ensure that the structure can withstand the design load. Finally, deflection constraints ensure the member has sufficient stiffness and stability under stress. Deflection refers to the bending and deformation of a member under external forces. To control deflection, we need to limit the rod’s geometric shape, the material’s stiffness, and the design conditions’ requirements. By simultaneously satisfying these three constraints, engineers can achieve maximum weight reduction in the member structure while maintaining the total load unchanged. This optimization design can reasonably utilize materials and reduce engineering costs while ensuring structural safety and performance. The following is the mathematical expression:

variable Consider $H = [h_1, h_2] = [x_1, x_2]$

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

$$\text{Subject to : } l_1(H) = \frac{\sqrt{x_1x_1} + x_2}{\sqrt{2x_1^2} + 2x_1x_2} P - \sigma \leq 0v \tag{47}$$

$$l_2(H) = \frac{x_2}{\sqrt{2x_1^2 + 2x_1x_2}}P - \sigma \leq 0 \tag{48}$$

$$l_3(H) = \frac{1}{\sqrt{2x_2 + x_1}}P - \sigma \leq 0 \tag{49}$$

$$l = 100cm, P = 2KN/cm_2, \sigma = 2KN/cm_2$$

Variables range ($0 \leq x_i \leq 1, i = 1, 2$)

Table 17, which compares the BKA algorithm to other algorithms, shows the optimal values and corresponding optimal variables. This table offers comparative analysis information that will allow us to assess how well various algorithms perform when solving problems. BKA can obtain the optimal function value $f(H) = 263.895843$ with the structure variables $H = (0.788675, 0.408248)$. By analyzing the data, it can be deduced that the BKA algorithm offers a superior solution to these engineering problems.

4.6 Analysis of the results of engineering design problems

By observing the results of the five different types of constrained engineering design problems mentioned above, BKA achieved the optimal results. Below is an analysis of the reasons why BKA achieved optimal results in constraint design problems:

1. Advantages of swarm intelligence: The BKA algorithm is based on swarm intelligence, which enables interaction and information exchange between individuals in a group. The swarm intelligence algorithm can search for the optimal solution through individual cooperation and collaboration and has strong robustness and global search ability. Therefore, individuals in the BKA algorithm can better explore the solution space and find optimal results through the collaborative effect of swarm intelligence.
2. Parameter optimization and adjustment: The BKA algorithm includes some parameters, such as Cauchy distribution’s control parameters and individual leaders’ selection strat-

Table 17 Optimal results of the different algorithms on the Three-bar truss design problem

Algorithm	Optimal values for variables		Optimal weight
	x_1	x_2	
BKA	0.788675	0.408248	263.895843
GJO	0.788657	0.408299	263.895844
PSO	0.788919	0.404741	263.896200
AVOA	0.788983	0.407378	263.895913
SHO	0.788898	0.40762	263.895881
SCSO	0.788334	0.409214	263.895959
SSA	0.707614	0.704996	270.642993
AO	0.788981	0.407368	263.895929
COA	0.788496	0.408285	263.895844

egy. The BKA algorithm can better adapt to different engineering examples by optimizing and adjusting reasonable parameters. Reasonably setting parameters can improve the performance and effectiveness of the algorithm in specific problems, thus achieving optimal results.

3. The BKA algorithm adopts the Cauchy distribution strategy, which gives the algorithm a strong global search ability. The Cauchy distribution has a relatively wide tail, which allows for a wider search of the solution space and avoids falling into local optima. Therefore, BKA can traverse more solution spaces in different engineering examples and has a greater probability of finding the global optimal solution.
4. The BKA algorithm introduces a leader strategy to guide the algorithm's entire optimization process. By selecting individuals with high fitness values as leaders, other individuals learn and improve solutions through interaction with the leader. Leaders usually have relatively good solutions; through their guidance, the entire group can evolve toward a more optimal solution. Therefore, BKA can accelerate convergence and achieve optimal results through leader strategy in different engineering examples.

5 Conclusion and future works

This article presents the Black Kite Algorithm (BKA), a new swarm intelligence optimization algorithm inspired by the attack and migration behaviors of Black-winged kites. The algorithm mimics the Black-winged kites' high predatory skills and integrates a migratory strategy to enhance search capabilities, striking a balance between local and global optima. The study's main contents are:

- Evaluate the performance of BKA using the CEC-2017 test set, CEC-2022 test set, and 18 complex functions, demonstrating superior results across various characteristics and complexities.
- Statistical validation using the Friedman and Wilcoxon sign rank tests, with BKA securing first place, confirming its effectiveness and scientific reliability.
- Practical application of BKA in five engineering cases involving challenging conditions and constrained search spaces, where it shows significant superiority by quickly converging to high-quality solutions and exhibiting excellent performance.

In future research, BKA can be integrated with other well-known strategies, such as adversarial learning mechanisms (Lian et al. 2023) and chaotic mapping (Liu et al. 2023), to further enhance the optimization performance of the algorithm. BKA can also be used to optimize various engineering problems in the future, such as multi-disc clutch brake design problems (Yu et al. 2020), step cone pulley problems (Nematollahi et al. 2021), etc.

Acknowledgements The authors are grateful for supporting the special project for collaborative science and technology innovation in 2021 (No: 202121206) and Henan Province University Scientific and Technological Innovation Team (No: 18IRTSTHN009).

Author contributions Jun Wang: conceptualization, methodology, data curation, writing—original draft. Wen-chuan Wang: conceptualization, methodology, writing—original draft, formal analysis, data curation. Xiao-xue Hu: writing—original draft, preparing figures. Lin Qiu: investigation, formal analysis. Hong-fei Zang: writing—original draft, formal analysis.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Consent to participate Informed consent was obtained from all participants included in the study.

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

References

- Abdel-Basset M, Mohamed R, Sallam KM, Chakraborty RK (2022) Light spectrum optimizer: a novel physics-inspired metaheuristic optimization algorithm. *Mathematics* 10:3466
- Abdel-Basset M, Mohamed R, Azeem SAA, Jameel M, Abouhawwash M (2023a) Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion. *Knowl-Based Syst* 268:110454
- Abdel-Basset M, Mohamed R, Jameel M, Abouhawwash M (2023b) Spider wasp optimizer: a novel metaheuristic optimization algorithm. *Artif Intell Rev* 10:11675–11738
- Abdel-Basset M, Mohamed R, Zidan M, Jameel M, Abouhawwash M (2023c) Mantis search algorithm: a novel bio-inspired algorithm for global optimization and engineering design problems. *Comput Methods Appl Mech Eng* 415:116200
- Abdollahzadeh B, Gharehchopogh FS, Mirjalili S (2021) African vultures optimization algorithm: a new nature-inspired metaheuristic algorithm for global optimization problems. *Comput Ind Eng* 158:107408
- Abualigah L, Youstri D, Abd Elaziz M, Ewees AA, Al-qaness MAA, Gandomi AH (2021) Aquila optimizer: a novel metaheuristic optimization algorithm. *Comput Ind Eng* 157:107250
- Al-Masri E, Soury A, Mohamed H, Yang W, Olmsted J, Kotevska O (2023) Energy-efficient cooperative resource allocation and task scheduling for internet of things environments. *Int Things* 23:100832
- Atban F, Ekinci E, Garip Z (2023) Traditional machine learning algorithms for breast cancer image classification with optimized deep features. *Biomed Signal Process Control* 81:104534
- Azizi M, Aickelin U, Khorshidi A, H, & Baghalzadeh Shishehgarkhaneh, M, (2023) Energy valley optimizer: a novel metaheuristic algorithm for global and engineering optimization. *Sci Rep* 13:226
- Banaie-Dezfouli M, Nadimi-Shahraki MH, Beheshti Z (2023) BE-GWO: binary extremum-based grey wolf optimizer for discrete optimization problems. *Appl Soft Comput* 146:110583
- Berger L, Bosetti V (2020) Characterizing ambiguity attitudes using model uncertainty. *J Econ Behav Organ* 180:621–637
- Beyer H-G, Schwefel H-P (2002) Evolution strategies—a comprehensive introduction. *Nat Comput* 1:3–52
- Bingi J, Warriar AR, Cherianath V (2023) Dielectric and plasmonic materials as random light scattering media. In: Haseeb ASMA (ed) *Encyclopedia of materials: electronics*. Academic Press, Oxford, pp 109–124
- Boukroune A, Haddad M, Li H (2023) Adaptive fuzzy control design for nonlinear systems with actuation and state constraints: an approach with no feasibility condition. *ISA Trans* 142:1–11
- Braik MS (2021) Chameleon swarm algorithm: a bio-inspired optimizer for solving engineering design problems. *Expert Syst Appl* 174:114685
- Chakraborty S, Nama S, Saha AK (2022) An improved symbiotic organisms search algorithm for higher dimensional optimization problems. *Knowl-Based Syst* 236:107779
- Chakraborty P, Nama S, Saha AK (2023) A hybrid slime mould algorithm for global optimization. *Multimed Tools Appl* 82:22441–22467
- Chen Y, Dang B, Wang C, Wang Y, Yang Y, Liu M, Bi H, Sun D, Li Y, Li J, Shen X, Sun Q (2023) Intelligent designs from nature: biomimetic applications in wood technology. *Prog Mater Sci* 139:101164

- Cheng M-Y, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput Struct* 139:98–112
- Cheng Y, Wen Z, He X, Dong Z, Zhangshang M, Li D, Wang Y, Jiang Y, Wu Y (2022) Ecological traits affect the seasonal migration patterns of breeding birds along a subtropical altitudinal gradient. *Avian Research* 13:100066
- Chopra N, Ansari MM (2022) Golden jackal optimization: a novel nature-inspired optimizer for engineering applications. *Expert Syst Appl* 198:116924
- Dehghani M, Montazeri Z, Trojovská E, Trojovský P (2022) Coati optimization algorithm: a new bio-inspired metaheuristic algorithm for solving optimization problems. *Knowl Based Syst* 259:110011
- Fan J, Zhou X (2023) Optimization of a hybrid solar/wind/storage system with bio-generator for a household by emerging metaheuristic optimization algorithm. *J Energy Storage* 73:108967
- Faramarzi A, Heidarinejad M, Mirjalili S, Gandomi AH (2020a) Marine predators algorithm: a nature-inspired metaheuristic. *Expert Syst Appl* 152:113377
- Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2020b) Equilibrium optimizer: a novel optimization algorithm. *Knowl-Based Syst* 191:105190
- Feng R, Shen C, Guo Y (2024) Digital finance and labor demand of manufacturing enterprises: theoretical mechanism and heterogeneity analysis. *Int Rev Econ Financ* 89:17–32
- Flack A, Aikens EO, Kölzsch A, Nourani E, Snell KRS, Fiedler W, Linek N, Bauer H-G, Thorup K, Partecke J, Wikelski M, Williams HJ (2022) New frontiers in bird migration research. *Curr Biol* 32:R1187–R1199
- Hansen N, Kern S (2004) Evaluating the CMA evolution strategy on multimodal test functions. In: Yao X, Burke EK, Lozano JA, Smith J, Merelo-Guervós JJ, Bullinaria JA, Rowe JE, Tiño P, Kabán A, Schwefel H-P (eds) *Parallel problem solving from nature—PPSN VIII*. Springer, Berlin, pp 282–291
- Hu G, Chen L, Wei G (2023) Enhanced golden jackal optimizer-based shape optimization of complex CSGC-Ball surfaces. *Artif Intell Rev* 56:2407–2475
- Inceyol Y, Cay T (2022) Comparison of traditional method and genetic algorithm optimization in the land reallocation stage of land consolidation. *Land Use Policy* 115:105989
- Jain M, Singh V, Rani A (2019) A novel nature-inspired algorithm for optimization: squirrel search algorithm. *Swarm Evol Comput* 44:148–175
- Jiang M-r, Feng X-f, Wang C-p, Fan X-l, Zhang H (2023) Robust color image watermarking algorithm based on synchronization correction with multi-layer perceptron and Cauchy distribution model. *Appl Soft Comput* 140:110271
- Kennedy J, Eberhart R (1995) Particle swarm optimization[C]. *Proc of the IEEE Int Conf Neural Netw Piscataway IEEE Serv Center* 12:1941–1948
- Koza JR (1992) *Genetic programming: on the programming of computers by means of natural selection*. MIT Press, Cambridge
- Kumar M, Kulkarni AJ, Satapathy SC (2018) Socio evolution & learning optimization algorithm: a socio-inspired optimization methodology. *Futur Gener Comput Syst* 81:252–272
- Kumar P, Govindaraj V, Erturk VS, Nisar KS, Inc M (2023) Fractional mathematical modeling of the stuxnet virus along with an optimal control problem. *Ain Shams Eng J* 14:102004
- Kuo HC, Lin CH (2013) Cultural evolution algorithm for global optimizations and its applications. *J Appl Res Technol* 11:510–522
- Lees AC, Gilroy JJ (2021) Bird migration: when vagrants become pioneers. *Curr Biol* 31:R1568–R1570
- Lian B, Xue W, Xie Y, Lewis FL, Davoudi A (2023) Off-policy inverse Q-learning for discrete-time antagonistic unknown systems. *Automatica* 155:111171
- Liu L, Xu X (2023) Self-attention mechanism at the token level: gradient analysis and algorithm optimization. *Knowl-Based Syst* 277:110784
- Liu R, Liu H, Zhao M (2023) Reveal the correlation between randomness and Lyapunov exponent of n-dimensional non-degenerate hyper chaotic map. *Integration* 93:102071
- Melman A, Evsutin O (2023) Comparative study of metaheuristic optimization algorithms for image steganography based on discrete Fourier transform domain. *Appl Soft Comput* 132:109847
- Mirjalili S (2015) The ant lion optimizer. *Adv Eng Softw* 83:80–98
- Mirjalili S, Lewis A (2016) The whale optimization algorithm. *Adv Eng Softw* 95:51–67
- Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. *Adv Eng Softw* 69:46–61
- Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. *Neural Comput Appl* 27:495–513
- Mirrashed M, Naderpour H (2023) Incomprehensible but intelligible-in-time logics: theory and optimization algorithm. *Knowl-Based Syst* 264:110305
- Moghdani R, Salimifard K (2018) Volleyball premier league algorithm. *Appl Soft Comput* 64:161–185

- Nadimi-Shahraki MH, Zamani H (2022) DMDE: diversity-maintained multi-trial vector differential evolution algorithm for non-decomposition large-scale global optimization. *Expert Syst Appl* 198:116895
- Nadimi-Shahraki MH, Zamani H, Asghari Varzaneh Z, Mirjalili S (2023a) A systematic review of the whale optimization algorithm: theoretical foundation, improvements, and hybridizations. *Arch Comput Methods Eng* 30:4113–4159
- Nadimi-Shahraki MH, Zamani H, Fatahi A, Mirjalili S (2023b) MFO-SFR: an enhanced moth-flame optimization algorithm using an effective stagnation finding and replacing strategy. *Mathematics* 11:862
- Nama S (2021) A modification of I-SOS: performance analysis to large scale functions. *Appl Intell* 51:7881–7902
- Nama S (2022) A novel improved SMA with quasi reflection operator: performance analysis, application to the image segmentation problem of Covid-19 chest X-ray images. *Appl Soft Comput* 118:108483
- Nama S, Saha AK (2020) A new parameter setting-based modified differential evolution for function optimization. *Int J Model Simul Sci Comput* 11:2050029
- Nama S, Saha AK (2022) A bio-inspired multi-population-based adaptive backtracking search algorithm. *Cogn Comput* 14:900–925
- Nama S, Saha AK, Sharma S (2022a) Performance up-gradation of symbiotic organisms search by backtracking search algorithm. *J Ambient Intell Humaniz Comput* 13:5505–5546
- Nama S, Sharma S, Saha AK, Gandomi AH (2022b) A quantum mutation-based backtracking search algorithm. *Artif Intell Rev* 55:3019–3073
- Nama S, Saha AK, Chakraborty S, Gandomi AH, Abualigah L (2023) Boosting particle swarm optimization by backtracking search algorithm for optimization problems. *Swarm Evol Comput* 79:101304
- Naruei I, Keynia F (2022) Wild horse optimizer: a new meta-heuristic algorithm for solving engineering optimization problems. *Eng Comput* 38:3025–3056
- Naruei I, Keynia F, Molahosseini AS (2021) Hunter–prey optimization: algorithm and applications. *Soft Comput* 26:1279–1314
- Nematollahi E, Zare S, Maleki-Moghaddam M, Ghasemi A, Ghorbani F, Banisi S (2021) DEM-based design of feed chute to improve performance of cone crushers. *Miner Eng* 168:106927
- Ramli R, Fauzi A (2018) Nesting biology of black-shouldered kite (*Elanus caeruleus*) in oil palm landscape in Carey Island, Peninsular Malaysia. *Saudi J Biol Sci* 25:513–519
- Sahoo SK, Saha AK, Nama S, Masdari M (2023) An improved moth flame optimization algorithm based on modified dynamic opposite learning strategy. *Artif Intell Rev* 56:2811–2869
- Satapathy S, Naik A (2016) Social group optimization (SGO): a new population evolutionary optimization technique. *Complex Intell Syst* 2:173–203
- Seyyedabbasi A, Kiani F (2023) Sand cat swarm optimization: a nature-inspired algorithm to solve global optimization problems. *Eng Comput* 39:2627–2651
- Sharma S, Chakraborty S, Saha AK, Nama S, Sahoo SK (2022a) mLBOA: a modified butterfly optimization algorithm with lagrange interpolation for global optimization. *J Bionic Eng* 19:1161–1176
- Sharma S, Saha AK, Roy S, Mirjalili S, Nama S (2022b) A mixed sine cosine butterfly optimization algorithm for global optimization and its application. *Clust Comput* 25:4573–4600
- Sharma, A (2015). Gene Expression Programming:-A New Adaptive Algorithm for Solving Problems, arXiv preprint cs/0102027
- Simon D (2008) Biogeography-based optimization. *IEEE Trans Evol Comput* 12:702–713
- Su H, Zhao D, Heidari AA, Liu L, Zhang X, Mafarja M, Chen H (2023) RIME: a physics-based optimization. *Neurocomputing* 532:183–214
- Wan M, Ye C, Peng D (2023) Multi-period dynamic multi-objective emergency material distribution model under uncertain demand. *Eng Appl Artif Intell* 117:105530
- Wang W-c, Xu L, Chau K-w, Xu D-m (2020) Yin-Yang firefly algorithm based on dimensionally Cauchy mutation. *Expert Syst Appl* 150:113216
- Wang W-c, Xu L, Chau K-w, Zhao Y, Xu D-m (2022) An orthogonal opposition-based-learning Yin-Yang-pair optimization algorithm for engineering optimization. *Eng Comput* 38:1149–1183
- Wang L, Gao K, Lin Z, Huang W, Suganthan PN (2023a) Problem feature based meta-heuristics with Q-learning for solving urban traffic light scheduling problems. *Appl Soft Comput* 147:110714
- Wang W-c, Xu L, Chau K-w, Liu C-j, Ma Q, Xu D-m (2023b) Cε-LDE: a lightweight variant of differential evolution algorithm with combined ϵ constrained method and Lévy flight for constrained optimization problems. *Expert Syst Appl* 211:118644
- Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. *IEEE Trans Evol Comput* 1:67–82

- Wu G, Mallipeddi R, Suganthan P (2016) 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
- Wu C-F, Lai J-H, Chen S-H, Trac LVT (2023) Key factors promoting the niche establishment of black-winged kite *Elanus caeruleus* in farmland ecosystems. *Ecol Ind* 149:110162
- Xie W, Huang P (2021) 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* 214:104663
- Xu W, Zhao H, Lv S (2023a) Robust multitask diffusion normalized M-estimate subband adaptive filter algorithm over adaptive networks. *J Franklin Inst* 360:11197–11219
- Xu Y, Du R, Pei J (2023b) The investment risk evaluation for onshore and offshore wind power based on system dynamics method. *Sustain Energy Technol Assess* 58:103328
- Yazdani, D, Branke, J, Omidvar, MN, Li, X, Li, C, Mavrovouniotis, M, Nguyen, T, & Yao, X (2021). IEEE CEC 2022 competition on dynamic optimization problems generated by generalized moving peaks benchmark, arXiv preprint arXiv:2106.06174
- Yu L, Ma B, Chen M, Li H, Liu J (2020) Investigation on the thermodynamic characteristics of the deformed separate plate in a multi-disc clutch. *Eng Fail Anal* 110:104385
- Zaman SI, Khan S, Zaman SAA, Khan SA (2023) A grey decision-making trial and evaluation laboratory model for digital warehouse management in supply chain networks. *Dec Anal J* 8:100293
- Zamani H, Nadimi-Shahraki MH, Gandomi AH (2021) QANA: quantum-based avian navigation optimizer algorithm. *Eng Appl Artif Intell* 104:104314
- Zamani H, Nadimi-Shahraki MH, Gandomi AH (2022) Starling murmuration optimizer: a novel bio-inspired algorithm for global and engineering optimization. *Comput Methods Appl Mech Eng* 392:114616
- Zervoudakis K, Tsafarakis S (2020) A mayfly optimization algorithm. *Comput Ind Eng* 145:106559
- Zhao S, Zhang T, Ma S, Chen M (2022) Dandelion optimizer: a nature-inspired metaheuristic algorithm for engineering applications. *Eng Appl Artif Intell* 114:105075
- Zhao H, Ning X, Liu X, Wang C, Liu J (2023a) What makes evolutionary multi-task optimization better: a comprehensive survey. *Appl Soft Comput* 145:110545
- Zhao S, Zhang T, Ma S, Wang M (2023b) Sea-horse optimizer: a novel nature-inspired meta-heuristic for global optimization problems. *Appl Intell* 53:11833–11860

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