

Multi-Strategy Improved Secretary Bird Optimization Algorithm

Fengkai Wang, Bo Wang*

College of Science, Shenyang University of Technology, Shenyang, China

Email: *5692188@qq.com, 2748218791@qq.com

How to cite this paper: Wang, F.K. and Wang, B. (2025) Multi-Strategy Improved Secretary Bird Optimization Algorithm. *Journal of Computer and Communications*, 13, 90-107.

<https://doi.org/10.4236/jcc.2025.131007>

Received: December 27, 2024

Accepted: January 27, 2025

Published: January 30, 2025

Copyright © 2025 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper addresses the shortcomings of the Sparrow and Eagle Optimization Algorithm (SBOA) in terms of convergence accuracy, convergence speed, and susceptibility to local optima. To this end, an improved Sparrow and Eagle Optimization Algorithm (HS-SBOA) is proposed. Initially, the algorithm employs Iterative Mapping to generate an initial sparrow and eagle population, enhancing the diversity of the population during the global search phase. Subsequently, an adaptive weighting strategy is introduced during the exploration phase of the algorithm to achieve a balance between exploration and exploitation. Finally, to avoid the algorithm falling into local optima, a Cauchy mutation operation is applied to the current best individual. To validate the performance of the HS-SBOA algorithm, it was applied to the CEC2021 benchmark function set and three practical engineering problems, and compared with other optimization algorithms such as the Grey Wolf Optimization (GWO), Particle Swarm Optimization (PSO), and Whale Optimization Algorithm (WOA) to test the effectiveness of the improved algorithm. The simulation experimental results show that the HS-SBOA algorithm demonstrates significant advantages in terms of convergence speed and accuracy, thereby validating the effectiveness of its improved strategies.

Keywords

Secretary Bird Optimization Algorithm, Iterative Mapping, Adaptive Weight Strategy, Cauchy Variation, Convergence Speed

1. Introduction

As society and technology continue to evolve, optimization problems across various domains are becoming increasingly complex and challenging. The nature of these problems spans a wide range of areas, including manufacturing, resource

allocation, path planning, financial portfolio optimization, and more. Confronted with real-world constraints such as scarce resources, cost control, and efficiency requirements, finding the best solutions has become a top priority. Traditional mathematical optimization methods, while performing well in certain situations, often exhibit limitations when dealing with complex, high-dimensional, nonlinear, and multimodal problems. Against this backdrop, metaheuristic algorithms have emerged. These belong to a class of intelligent search algorithms inspired by natural phenomena and mechanisms, aimed at finding solutions to optimization problems through stochastic methods. Unlike traditional mathematical optimization methods, metaheuristic algorithms are more suited to complex, multimodal, high-dimensional, and nonlinear optimization problems.

Metaheuristic algorithms, which are categorized based on their fundamental principles, encompass four main categories: Evolutionary Algorithms (EA), Physics-based Heuristics (PHA), Human Behavior-based Algorithms (HBBA), and Swarm Intelligence (SI) algorithms. Evolutionary Algorithms, inspired by natural evolution, include prominent methods such as Genetic Algorithms (GA) [1] and Differential Evolution (DE) [2], both rooted in Darwin's theory of evolution. Physics-based Heuristics, on the other hand, are often motivated by physical phenomena, with Simulated Annealing (SA) [3] being a clear example that mimics the annealing process of solids to find optimal solutions. Human Behavior-based Algorithms draw inspiration from the intricate dynamics within human societies, with Social Group Optimization (SGO) [4] serving as a typical example. Swarm Intelligence algorithms take their cue from the collective behaviors observed in natural biological populations, and they include a variety of methods such as Particle Swarm Optimization (PSO) [5], Ant Colony Optimization (ACO) [6], Artificial Bee Colony Algorithm (ABC) [7], Whale Optimization Algorithm (WOA) [8], Grey Wolf Optimizer (GWO) [9], and Harris Hawks Optimization (HHO) [10]. More recent additions to the swarm intelligence algorithms include the Greater Cane Rat Algorithm (GCRA) [11], Pied Kingfisher Optimizer (PKO) [12], Honey Badger Algorithm (HBA) [13], and Secretary Bird Optimization Algorithm (SBOA) [14]. These metaheuristic approaches provide a diverse toolkit for addressing optimization problems by emulating different facets of nature and human behavior, offering innovative and effective strategies for navigating complex problem spaces. Among various optimization algorithms, the Secretary Bird Optimization Algorithm (SBOA) stands out as an emerging intelligent optimization technique, demonstrating superior performance in balancing exploration and exploitation, enhancing convergence velocity, and optimizing accuracy.

Despite the Secretary Bird Optimization Algorithm (SBOA) demonstrating commendable performance in various optimization tasks, there remains room for improvement in convergence precision, convergence speed, and susceptibility to local optima as the complexity of optimization problems increases. Consequently, this paper aims to address these shortcomings of SBOA by proposing a multi-strategy improved Sparrow and Eagle Optimization Algorithm (HS-SBOA) to

enhance its performance in complex optimization problems.

2. Secretary Bird Optimization Algorithm (SBOA)

The Secretary Bird Optimization Algorithm (SBOA) emulates the secretary bird, drawing inspiration from its survival behaviors in the natural environment. The exploration phase of this algorithm simulates the secretary bird’s behavior of preying on snakes, while the exploitation phase mimics their process of escaping from predators.

The SBOA is a population-based metaheuristic method where each secretary bird is considered a member of the algorithm’s population. The position of each bird in the search space determines the values of the decision variables. Therefore, in the SBOA, the position of a secretary bird represents a candidate solution to the problem at hand. In the initial implementation of the SBOA, Equation (1) is used to randomly initialize the positions of the secretary birds in the search space.

$$X_{i,j} = lb_j + r(ub_j - lb_j), i = 1, 2, \dots, N, j = 1, 2, \dots, Dim \quad (1)$$

where X_i denotes the position of the i -th secretary bird, ub_j and lb_j are the lower and upper bounds, respectively, and r represents a random number between 0 and 1.

In the SBOA, optimization begins with a population of candidate solutions, as shown in Equation (2). These candidate solutions X are randomly generated within the upper bound (Ub) and lower bound (Lb) constraints of the given problem. The best solution obtained so far is approximated as the optimal solution in each iteration.

$$X = \begin{bmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,Dim} \\ x_{2,1} & x_{2,2} & \dots & x_{2,Dim} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N,1} & x_{N,2} & \dots & x_{N,Dim} \end{bmatrix} \quad (2)$$

Two distinct natural behaviors of secretary birds are used to update the members of the SBOA. These behaviors include hunting strategies and escape strategies. The hunting strategy of secretary birds is divided into three stages: searching for prey, capturing prey, and attacking prey.

The prey search phase employs a differential evolution strategy, enhancing the algorithm’s diversity and global search capability.

$$\begin{aligned} \text{When } t < \frac{1}{3}T, x_{i,j}^{new,P1} &= x_{i,j} + (x_{random_1} - x_{random_2}) \times R_1 \\ X_i &= \begin{cases} X_i^{new,P1}, & \text{if } F_i^{new,P1} < F_i \\ X_i, & \text{else} \end{cases} \end{aligned} \quad (3)$$

where t represents the current iteration number, T denotes the maximum number of iterations, $X_i^{new,P1}$ is the new status of the i -th secretary bird in the first phase, X_{random_1} and X_{random_2} are random candidate solutions in the first phase iteration. R_1 is a random vector with dimensions $1 \times Dim$ generated from the interval

[0, 1], where Dim is the dimension of the solution space. $X_{i,j}^{new,P1}$ represents its j-th dimension value, and $F_i^{new,P1}$ represents its objective function fitness value.

The prey capture phase introduces Brownian motion (RB) to simulate the random movement of secretary birds. This method not only helps individuals avoid premature convergence to local optima but also accelerates the algorithm's convergence to the best position in the solution space.

$$\text{When } \frac{1}{3}T < t < \frac{2}{3}T, x_{i,j}^{new,P1} = x_{best} + \exp\left(\left(\frac{t}{T}\right)^4\right) \times (RB - 0.5) \times (x_{best} - x_{i,j}) \quad (4)$$

$$RB = randn(1, Dim)$$

where random (1, Dim) represents a $1 \times Dim$ vector of random numbers generated from the standard normal distribution (mean 0, standard deviation 1), and x_{best} denotes the current best value.

The prey attack phase introduces a Levy flight strategy, enhancing the optimizer's global search capability, reducing the risk of SBOA getting trapped in local solutions, and improving the algorithm's convergence accuracy.

$$\text{When } t > \frac{2}{3}T, x_{i,j}^{new,P1} = x_{best} + \left(1 - \frac{t}{T}\right)^{2 \times \frac{t}{T}} \times x_{i,j} \times RL$$

$$RL = 0.5 \times Levy(Dim)$$

$$Levy(D) = S \times \frac{u \times \sigma}{|v|^{\frac{1}{\eta}}} \quad (5)$$

$$\sigma = \left(\frac{\Gamma(1 + \eta) \times \sin\left(\frac{\pi\eta}{2}\right)}{\Gamma\left(\frac{1 + \eta}{2}\right) \times \eta \times 2\left(\frac{\eta - 1}{2}\right)} \right)^{\frac{1}{\eta}}$$

where S is a fixed constant 0.01, η is a fixed constant 1.5. u and v are random numbers in the interval [0, 1]. Γ represents the Gamma function.

When threatened by predators, secretary birds typically employ various evasion strategies to protect themselves or their food. The escape strategy is divided into two categories. The first category is that secretary birds can swiftly fly away when encountering danger, seeking safer locations. The second strategy is camouflage. Secretary birds may use the colors or structures in their environment to blend in, making it more difficult for predators to detect them.

$$x_{i,j}^{new,P2} = \begin{cases} C_1 : x_{best} + (2 \times RB - 1) \times \left(1 - \frac{t}{T}\right)^2 \times x_{i,j}, & \text{if } r \text{ and } < r_i \\ C_2 : x_{i,j} + R_2 \times (x_{random} - K \times x_{i,j}), & \text{else} \end{cases} \quad (6)$$

$$K = round(1 + rand(1, 1))$$

where $r = 0.5$, R_2 represents a vector array of dimensions $1 \times Dim$ randomly generated from the normal distribution, x_{random} is the random candidate solution of the current iteration, and K is a random choice of integer 1 or 2.

The flow of the SBOA is as follows:

Step 1. Initialize the parameters of the secretary bird population.

Step 2. Calculate the fitness value of each individual secretary bird, defined according to the specific optimization problem.

Step 3. Update the population position using Equation (3) during the prey search phase.

Step 4. Update the population position using Equation (4) during the prey capture phase.

Step 5. Update the population position using Equation (5) during the prey attack phase.

Step 6. Update the secretary bird positions according to the escape strategy in the SBOA using Equation (6).

Step 7. Check the termination condition. If the stopping criteria of the algorithm are met (such as reaching the maximum number of iterations or fitness threshold), output the best fitness value and the corresponding secretary bird position information; otherwise, return to Step 3 and continue iterating.

3. Multi-Strategy Driven Secretary Bird Optimization Algorithm (HS-SBOA)

3.1. Chaotic Sequence Initialization

The basic SBOA initializes the secretary bird population randomly in the search space, which, while straightforward, can lead to uneven distribution of secretary bird positions, thereby affecting population diversity. Insufficient population diversity limits the algorithm's global search capability and increases the risk of falling into local optima. To address this issue, this paper proposes the use of Iterative chaotic mapping to improve the population initialization process.

Iterative chaotic mapping, known for its excellent ergodicity and non-repetitiveness, is widely used in optimization algorithms to enhance initial population diversity. Chaos theory suggests that even minor changes in initial conditions can lead to significantly different long-term behaviors, making chaotic mapping an ideal choice for optimizing population initialization [15].

The mathematical expression of iterative chaotic mapping is key to implementing this improvement. Through a carefully designed iterative formula, it ensures that the initial population is uniformly distributed in the search space, ensuring that the algorithm covers a broad search area in the early stages, laying a solid foundation for subsequent local searches and fine-tunings. The mathematical expression for Iterative mapping is:

$$x_{k+1} = \sin\left(\frac{b\pi}{xk}\right) \quad (7)$$

where x_k is the state after the n -th iteration, and x_{k+1} is the chaotic mapping function. In this paper, Iterative mapping is selected with control parameters $b \in (0,1)$. In experiments, iterative chaotic mapping ensures the uniform distribution of the initial population, thereby enhancing the algorithm's global search

capability.

3.2. Adaptive Weighting Strategy

Traditional SBOA relies primarily on random walks and simple neighborhood search strategies during the exploration phase, which limits the algorithm's search capability when dealing with complex problem spaces. To overcome this limitation, this paper introduces an adaptive weighting strategy to enhance the algorithm's global search capability and population diversity. This strategy integrates dynamic search intensity and random control with time decay, enhancing the algorithm's exploration capability and exploitation efficiency in complex search spaces.

The introduction of the adaptive weighting strategy significantly improves the Secretary Bird Optimization Algorithm. Firstly, through dynamic search intensity I , the algorithm can conduct more effective searches within the solution space, reducing the likelihood of getting trapped in local optima. Secondly, by introducing a density factor α , the algorithm can smoothly transition from exploration to exploitation during the search process, enhancing the algorithm's convergence speed and solution quality. Furthermore, to further improve algorithm efficiency, the adaptive weighting strategy introduces a flag variable F that dynamically adjusts the search direction based on the current solution quality. This not only helps the algorithm escape from local optima but also maintains a certain level of diversity during the search process, preventing premature convergence. The improved prey capture and prey attack phases are as follows:

Prey Capture Phase

$$x_{new} = x_{best} + F \times \beta \times I \times x_{best} + F \times r_3 \times \alpha \times d_i \times \left[\cos(2\pi r_4) \times [1 - \cos(2\pi r_5)] \right] \quad (8)$$

where x_{best} is the historical best position, and x_{new} is the improved prey capture phase population iteration formula, r_3 , r_4 , r_5 with random numbers between 0 and 1.

$$\begin{aligned} I_i &= r_2 \frac{S}{4\pi d_i^2}, r_2 \in (0,1) \\ S &= (x_i - x_{i+1})^2 \\ d_i &= x_{best} - x_i \end{aligned} \quad (9)$$

where I_i is the search intensity of the i -th individual, S is the dynamic search intensity, and d_i is a small constant, ensuring that the search step length is inversely proportional to the distance between the individual and the best solution, enabling more effective searches within the solution space.

$$\begin{aligned} \alpha &= C \times \exp\left(\frac{-t}{t_{max}}\right), \\ C &= 2 \\ F &= \begin{cases} 1 & \text{if } r_6 \leq 0.5 \\ -1 & \text{else} \end{cases} \\ r_6 &\in (0,1) \end{aligned} \quad (10)$$

where C are constants (default values 2), and t is the current iteration number,

t_{\max} is the maximum number of iterations.

The flag variable F is used to change the search direction to help the algorithm escape from local optima.

Prey Attack Phase

$$x_{new} = x_{best} + F \times r_7 \times \alpha \times d_i, r_7 \in (0,1) \quad (11)$$

where x_{best} is the historical best position, and x_{new} is the control parameter for the improved prey attack phase population iteration formula $r_7 \in (0,1)$.

Through this adaptive weighting strategy, the algorithm can effectively narrow the search range while maintaining population diversity, accelerating convergence to the optimal solution. This strategy endows the algorithm with the ability to dynamically adjust search behavior, making it more efficient and robust when solving complex optimization problems.

3.3. Cauchy Mutation Operator

In the later stages of the SBOA algorithm's evolution, all secretary bird individuals tend to converge towards the best individual, leading to a decrease in population diversity and an increased risk of the algorithm falling into local optima. To reduce this probability, this paper proposes a Cauchy mutation operation on the position of the best secretary bird individual in the current iteration, randomly on m -dimensions. The Cauchy distribution, compared to the normal distribution, is more likely to produce random numbers far from the origin, thus better preventing the algorithm from falling into local optima [16].

Therefore, the improved mutation strategy proposed in this paper is as follows:

$$x_{new} = x_{best} + Cauchy \times ones(1, dim) \quad (12)$$

where x_{new} represents the new position after mutation, x_{best} is the position of the current best individual, and $Cauchy$ is the parameter controlling the intensity of mutation. By introducing the Cauchy mutation operator, the algorithm can enhance the probability of jumping out of local optima while maintaining exploration capabilities.

3.4. Algorithm Implementation Steps

Step 1: Initialization

Initialize the parameters of the secretary bird population. Utilize the improved iterative chaotic mapping to generate the initial positions of the secretary bird population, ensuring a uniform distribution of the population within the search space.

Step 2: Fitness Calculation

Calculate the fitness value of each secretary bird individual, defined according to the specific optimization problem.

Step 3: Update Density Factor

Update the density factor according to Equation (10) to reflect changes in the current search environment.

Step 4: Update Search Intensity

Update the search intensity I according to Equation (9), adjusting the balance between exploration and exploitation.

Step 5: Population Update

Perform differential evolution strategy to update population positions. Further adjust population positions according to Equation (8). Conduct the prey attack phase update according to Equation (11).

Step 6: Position Correction

Correct the new positions to ensure individual positions are within the preset search space boundaries. Calculate the fitness values of individuals after position updates.

Step 7: Escape Strategy

Update the positions of secretary bird individuals according to the escape strategy in the SBOA to prevent premature convergence of the population.

Step 8: Optimal Solution Update

Update the current best fitness value according to Equation (12), ensuring tracking of the global optimal solution.

Step 9: Termination Condition Check

Check if the stopping criteria of the algorithm are met (such as reaching the maximum number of iterations or fitness threshold). If satisfied, output the best fitness value and the corresponding secretary bird position information; otherwise, return to Step 2 and continue iterating.

4. Experimental Simulation and Results Analysis

To validate the performance of the HS-SBOA, experiments were conducted using the CEC2021 test function suite and three practical engineering problems, primarily to verify the optimization capabilities and convergence speed of the HS-SBOA.

The operational environment consisted of a Windows 11 operating system, an AMD Ryzen 5 4600U with Radeon Graphics 2.10 GHz CPU, and 16 GB of RAM. The programming language used was MATLAB, and the experimental platform was MATLAB R2022b.

4.1. Experiment One

The ten benchmark test functions are presented in **Table 1**. The benchmark test functions used were all 20-dimensional, with a population of 30 individuals in the algorithm, and the algorithm was independently iterated 500 times. The HS-SBOA, SBOA, GWO, HHO, PSO, and WOA algorithms were used to solve the test functions listed in **Table 1**. Thirty independent experiments were conducted using MATLAB to compare the average values and standard deviations of the two algorithms, mitigating biases due to the randomness of the algorithms and ensuring their rationality. The experimental results are shown in **Table 2**, and the convergence curves are depicted in **Figures 1-10**.

The experimental results presented in **Table 2** provide an in-depth understanding of the performance of the HS-SBOA algorithm. By comparing with four other algorithms, the HS-SBOA demonstrated significant advantages in optimization performance on the test functions, especially in terms of convergence accuracy, where the HS-SBOA algorithm achieved the top rank in these test functions. This not only proves its efficiency in solving global optimization problems but also reflects the effectiveness of the multi-scale dynamic population strategy and honey-badger-inspired mutation operations introduced in the algorithm design.

The data in **Table 2** indicates that the improved algorithm has strong search capabilities and faster convergence speed, quickly converging to the target value for most functions.

Table 1. CEC2021 test functions.

	No.	Functions	F_i^*
Unimodal Function	1	Shifted and Rotated Bent Cigar Function	100
	2	Shifted and Rotated Schwefel's Function	1100
Basic Functions	3	Shifted and Rotated Lunacek bi-Rastrigin Function	700
	4	Expanded Rosenbrock's plus Griewangk's Function	1900
	5	Hybrid Function 1 (N = 3)	1700
Hybrid Functions	6	Hybrid Function 2 (N = 4)	1600
	7	Hybrid Function 3 (N = 5)	2100
Composition Functions	8	Composition Function 1 (N = 3)	2200
	9	Composition Function 2 (N = 4)	2400
	10	Composition Function 3 (N = 5)	2500

Search range: [-100, 100]

Table 2. Experimental results for test functions.

Test Function	Performance Metric or Evaluation Criterion	SBOA	GWO	PSO	WOA	HHO	HS-SBOA
		f_1	Mean	2.817045e-158	1.968860e-37	2.333480e+03	9.443189e-81
	Std	1.542733e-157	6.629772e-37	4.301749e+03	3.018451e-80	3.732027e-91	0.000000e+00
f_3	Mean	2.651560e-01	9.791807e+00	5.317916e+02	1.023854e+02	0.000000e+00	0.000000e+00
	Std	7.724638e-01	1.629030e+01	3.037977e+02	5.607878e+02	0.000000e+00	0.000000e+00
f_4	Mean	1.831485e+00	1.056128e+02	3.461900e+01	6.573841e-33	0.000000e+00	0.000000e+00
	Std	3.902551e+00	5.917104e+01	1.102366e+01	3.600641e-32	0.000000e+00	0.000000e+00
f_5	Mean	0.000000e+00	9.167198e-01	2.654936e+00	1.361082e-01	0.000000e+00	0.000000e+00
	Std	0.000000e+00	1.823312e+00	7.446027e-01	5.757744e-01	0.000000e+00	0.000000e+00

Continued

f_6	Mean	2.232818e-01	2.770500e+00	2.275642e+04	6.001433e-23	7.645511e-87	0.000000e+00
	Std	8.224231e-01	3.708405e+00	9.465348e+04	1.986237e-22	2.435216e-86	0.000000e+00
f_7	Mean	3.303160e+00	6.440279e+00	7.751995e+01	1.030081e+02	3.608021e-05	1.770354e+00
	Std	1.149572e-03	1.691997e+01	7.529435e+01	2.565653e+02	9.705297e-05	5.995636e-04
f_8	Mean	9.207058e-02	8.846511e-01	6.722395e+03	6.770931e-02	2.016863e-06	1.081866e-06
	Std	2.074189e-06	1.869214e+00	2.888841e+04	7.844520e-02	9.113601e-06	2.534363e-06
f_9	Mean	0.000000e+00	0.000000e+00	1.090948e+02	5.473523e+01	0.000000e+00	0.000000e+00
	Std	0.000000e+00	0.000000e+00	2.664760e+02	2.997972e+02	0.000000e+00	0.000000e+00
f_{10}	Mean	2.960595e-16	2.516506e-14	3.366529e-01	6.809368e-15	9.617567e-103	2.345807e-313
	Std	1.621585e-15	8.107923e-15	1.027006e+00	6.029937e-15	3.848066e-102	0.000000e+00
f_{10}	Mean	3.027573e+01	7.409210e+01	5.828732e+01	9.767534e-02	4.818418e-04	1.040968e-03
	Std	3.838085e+01	1.453912e+01	1.540370e+01	4.285010e-02	8.616698e-04	1.227031e-03

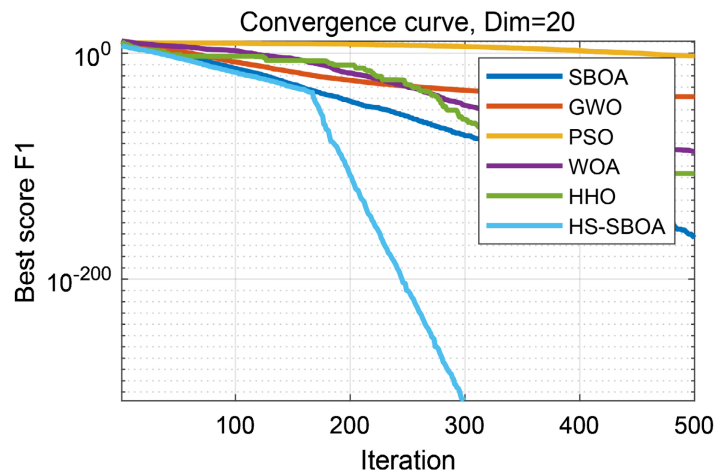


Figure 1. Convergence curve of HS-SBOA on shifted and rotated bent cigar function.

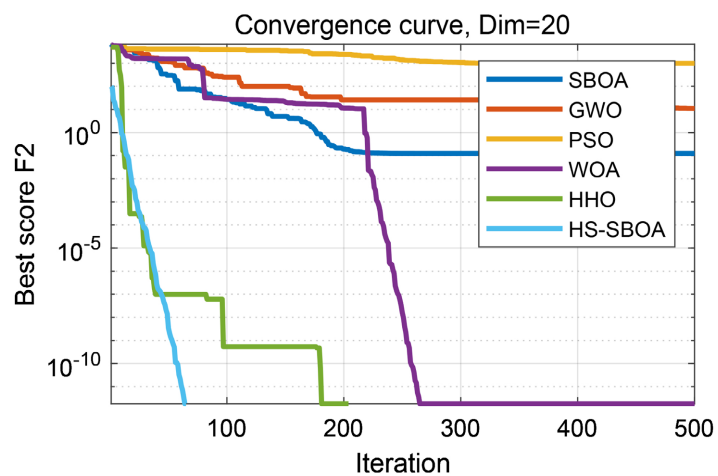


Figure 2. Convergence curve of HS-SBOA on shifted and rotated schwefel's function.

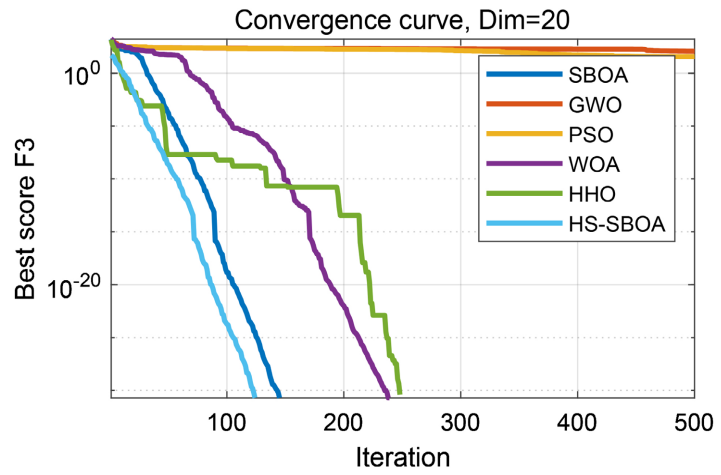


Figure 3. Convergence curve of HS-SBOA on shifted and rotated lunacek bi-raftigrin function.

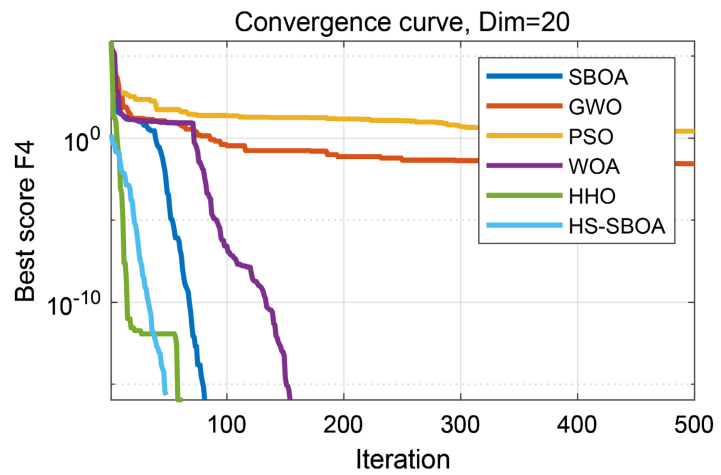


Figure 4. Convergence curve of HS-SBOA on expanded rosenbrock's plus griewangk's function.

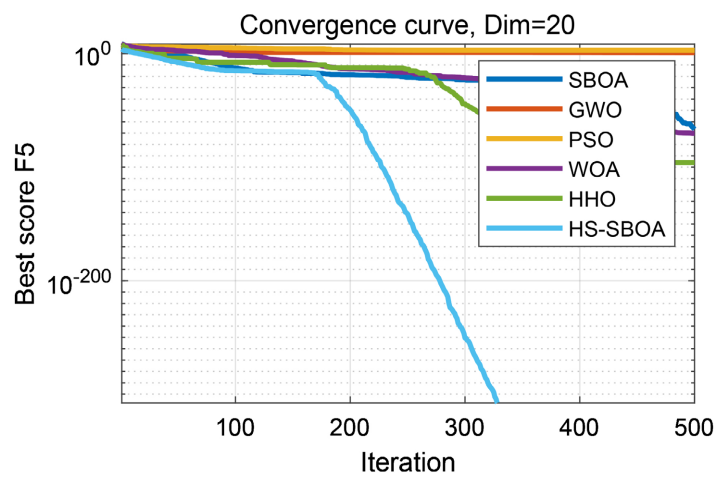


Figure 5. Convergence curve of HS-SBOA on hybrid function 1 (N = 3).

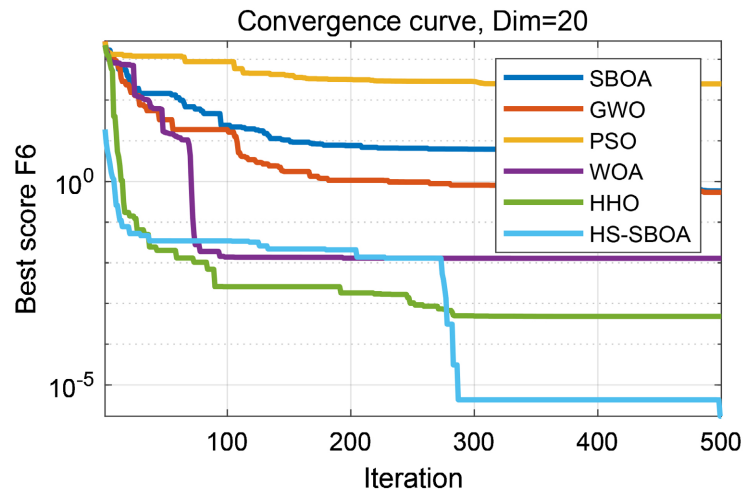


Figure 6. Convergence curve of HS-SBOA on hybrid function 2 (N = 4).

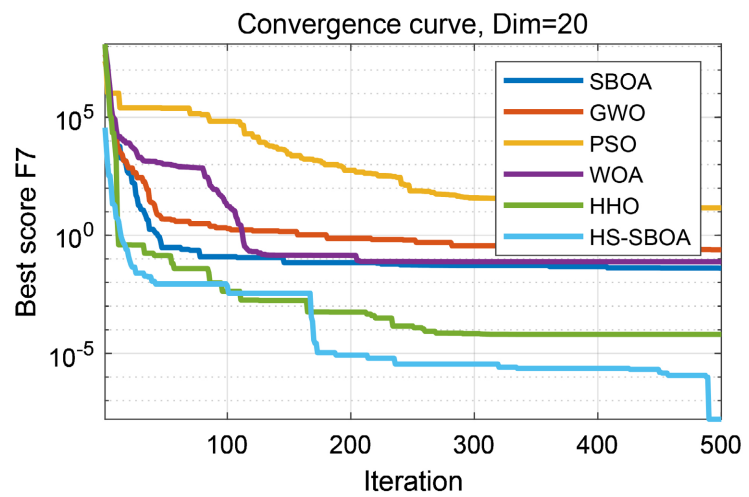


Figure 7. Convergence curve of HS-SBOA on hybrid function 3 (N = 5).

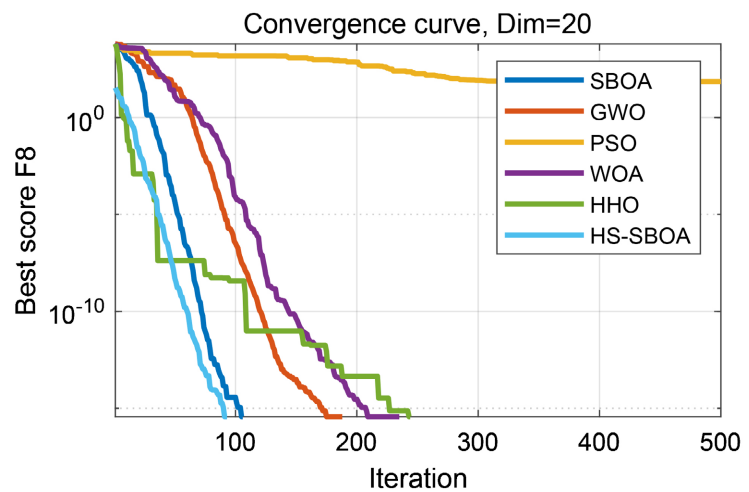


Figure 8. Convergence curve of HS-SBOA on composition function 1 (N = 3).

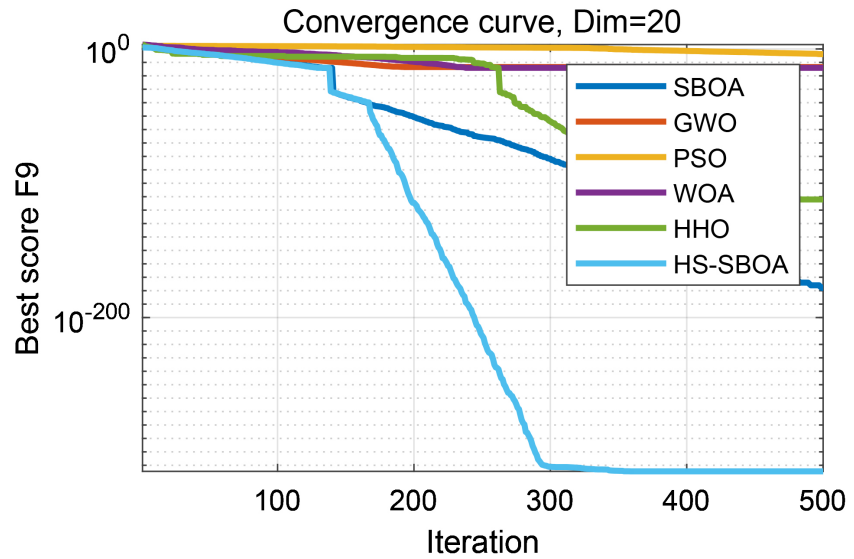


Figure 9. Convergence curve of HS-SBOA on composition function 2 (N = 4).

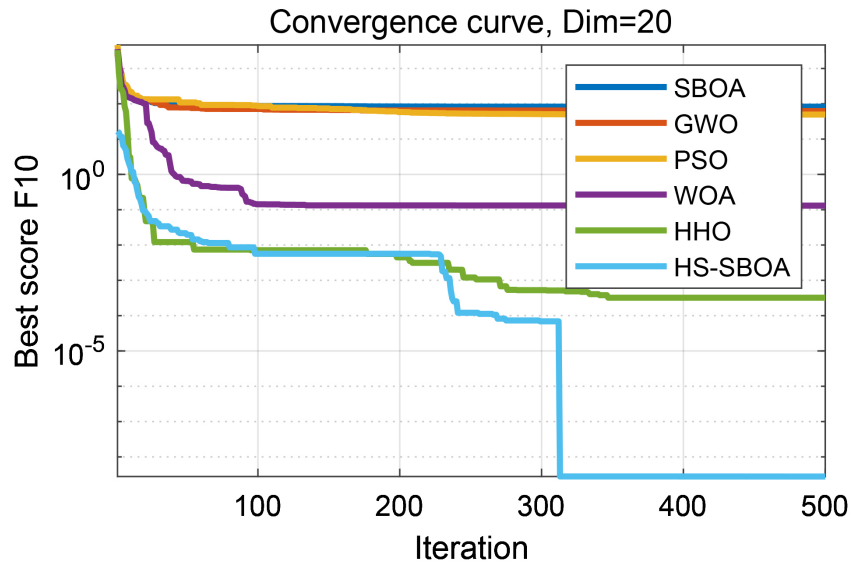


Figure 10. Convergence curve of HS-SBOA on composition function 3 (N = 5).

4.2. Experiment Two

In the data-driven modern world, constraint optimization for engineering problems has become a key task. Although the previous section has conducted optimization analysis on standard test functions, real-world engineering challenges often come with specific constraints. This section will demonstrate the application of HS-SBOA, SBOA, GWO, and BWO in practical engineering cases such as the Gear Train Design Problem (GTDP) [17], Spring Compression Design Problem (TCSD) [18], and Pressure Vessel Design Problem (PVD) [19], aiming to evaluate and prove the practicality and effectiveness of HS-SBOA in dealing with actual engineering problems. Through this practice, the performance and potential of these algorithms in real-world problems will be deeply explored.

4.2.1. Gear Train Design Problem (GTDP)

The gear train design problem is a common optimization problem in the field of mechanical engineering, with its core objective being to reduce the manufacturing and operational costs of gear transmission systems.

In this problem, designers need to determine four key variables: the number of teeth for the input gear, intermediate gear, driver gear, and output gear. These variables directly affect the gear ratio, efficiency, strength, durability, and overall size and weight of the gear train. The optimization process requires a comprehensive consideration of gear material costs, manufacturing process complexity, system reliability, and maintenance needs. By applying mathematical modeling, computer-aided design, finite element analysis, and optimization algorithms, designers can explore different gear configurations to achieve maximum cost-effectiveness and optimal performance. The specific description is given by the following equation:

$$\begin{aligned} \min f(x) &= \left(\frac{1}{6.931} - \frac{x_2 x_3}{x_1 x_4} \right)^2 \\ \text{s.t. } & 12 \leq x_i \leq 60, i = 1, 2, 3, 4 \end{aligned} \quad (13)$$

Thirty independent experiments were conducted to validate the effectiveness of the algorithms, with the results shown in **Table 3**.

Table 3. GTDP optimal solutions and optimal values.

Algorithm	Optimal Solution				Optimal Value
	x_1	x_2	x_3	x_4	
BWO	52.521163	16.921864	18.486170	40.148348	0.0000000036
SBOA	23.492901	13.049456	12.000000	47.375796	0.0000000010
GWO	23.147780	13.221766	12.000000	46.608512	0.0000000010
HS-SBOA	44.468705	13.317390	20.770835	42.711939	0.0000000002

Through **Table 3**, the optimal solutions and optimal values for HS-SBOA, SBOA, GWO, and BWO can be obtained, indicating that the proposed HS-SBOA is feasible in solving the GTDP.

4.2.2. Spring Compression Design Problem (TCSD)

The spring compression design problem plays a fundamental and critical role in mechanical design, involving the precise calculation and design of springs to accommodate different compression loads and working environments.

It requires engineers to find the optimal spring design within a limited parameter space to meet specific functional requirements and performance standards. The main objective of the spring compression problem is to minimize weight by selecting three variables: wire diameter, mean coil diameter, and the number of active coils. The mathematical expression is given by:

$$\begin{aligned}
 \min \quad & f(x) = (x_3 + 2)x_2x_1^2 \\
 \text{s.t.} \quad & g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \\
 & g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5180x_1^2} \leq 0 \\
 & g_3(x) = 1 - \frac{140.45x_1}{x_2^3x_3} \leq 0 \\
 & g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \\
 & 0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15
 \end{aligned} \tag{14}$$

Thirty independent experiments were conducted to validate the effectiveness of the algorithms, with the results shown in **Table 4**.

Table 4. TCSD optimal solutions and optimal values.

Algorithm	Optimal Solution			Optimal Value
	x_1	x_2	x_3	
BWO	0.050000	0.312892	15.000000	0.013298
SBOA	0.054359	0.424365	8.212425	0.012806
GWO	0.053487	0.401538	9.075201	0.012722
HS-SBOA	0.052857	0.385477	9.782878	0.012690

In **Table 4**, the optimal solutions and optimal values for HS-SBOA, SBOA, GWO, and BWO can be obtained, indicating that the proposed HS-SBOA is feasible in solving the TCSD problem.

4.2.3. Pressure Vessel Design Problem (PVD)

The pressure vessel design problem is a classic structural optimization problem that requires minimizing the weight or cost of the vessel while satisfying a series of engineering constraints and safety standards.

Such problems often involve complex geometric shapes, material properties, and loading conditions, requiring a comprehensive consideration of strength, stability, and durability. In the design process, engineers must ensure that the vessel can withstand internal and external pressures while also considering manufacturing processes and economic viability. To achieve this goal, advanced numerical methods and optimization algorithms are often employed to find the optimal design solution, which not only improves the performance of the vessel but also reduces material usage, thereby reducing costs and increasing efficiency. In both academic research and industrial applications, the pressure vessel design problem is an extremely challenging field that continues to drive the development of material science, computational mechanics, and optimization technology. The specific description is given by the following equation:

$$\begin{aligned}
\min \quad & f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \\
s.t. \quad & g_1(x) = -x_1 + 0.0193x_3 \leq 0 \\
& g_2(x) = -x_3 + 0.00954x_3 \leq 0 \\
& g_3(x) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\
& g_4(x) = x_4 - 240 \leq 0 \\
& 0 \leq x_1, x_2 \leq 99, 10 \leq x_3, x_4 \leq 200
\end{aligned} \tag{15}$$

Thirty independent experiments were conducted to validate the effectiveness of the algorithms, with the results shown in **Table 5**.

Table 5. PVD optimal solutions and optimal values.

Algorithm	Optimal Solution				Optimal Value
	x_1	x_2	x_3	x_4	
BWO	1.1946	0.5565	56.6123	56.9943	7430.560614
SBOA	0.7914	0.3912	41.0028	190.7038	5908.253838
GWO	0.7853	0.3903	40.6890	195.0212	5906.288701
HS-SBOA	0.7836	0.3892	40.5975	196.1701	5900.307258

Through **Table 5**, the optimal solutions and optimal values for HS-SBOA, SBOA, BWO, and GWO can be obtained, indicating that the proposed HS-SBOA is feasible for solving the PVD problem.

5. Conclusions and Implications

This paper addresses the deficiencies of the Secretary Bird Optimization Algorithm (SBOA) in terms of convergence accuracy, convergence speed, and susceptibility to local optima by proposing a multi-strategy enhanced Secretary Bird Optimization Algorithm (HS-SBOA). By introducing iterative chaotic mapping, adaptive weighting strategies, and Cauchy mutation operators, the algorithm's global and local search capabilities are optimized, achieving a balance between the exploration and exploitation phases. To validate the performance of HS-SBOA, this study employs the CEC2021 test function suite and three engineering application problems as experimental subjects. The experimental results demonstrate that the improved HS-SBOA outperforms traditional SBOA, GWO, PSO, HHO, and WOA algorithms in terms of global convergence, convergence speed, and algorithmic robustness, thereby validating the effectiveness of the proposed enhancement strategies.

Despite the excellent performance of HS-SBOA in various test problems, there are still some limitations. Firstly, when dealing with large-scale problems, the computational complexity of the algorithm is high, which may lead to longer running times. Secondly, HS-SBOA may still experience slower convergence in certain types of multimodal problems, especially when the objective function has numerous local optima that are densely distributed. Additionally, some parameters

in the algorithm (such as those in the adaptive weighting strategy and the intensity parameter of the Cauchy mutation) significantly affect performance.

Future research can further explore the application of HS-SBOA in more complex practical engineering problems and integrate other optimization strategies, such as dynamic parameter adjustment and multi-objective optimization, to enhance the algorithm's adaptability and solution capabilities. Moreover, incorporating distributed computing and parallel processing technologies is expected to improve the efficiency of HS-SBOA further and expand its application scope in large-scale optimization problems. Additionally, conducting a more in-depth sensitivity analysis of the parameters will provide clearer guidance for parameter selection, which is also an important direction for future research. Finally, considering the integration of HS-SBOA with other advanced optimization algorithms to produce new, more efficient optimization algorithms could further expand its application potential in various fields.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Holland, J.H. (1992) Genetic Algorithms. *Scientific American*, **267**, 66-72. <https://doi.org/10.1038/scientificamerican0792-66>
- [2] Price, K.V. (2013) Differential Evolution. In: Zelinka, I., Sňášel, V. and Abraham, A., Eds., *Intelligent Systems Reference Library*, Springer, 187-214. https://doi.org/10.1007/978-3-642-30504-7_8
- [3] Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. (1983) Optimization by Simulated Annealing. *Science*, **220**, 671-680. <https://doi.org/10.1126/science.220.4598.671>
- [4] Satapathy, S. and Naik, A. (2016) Social Group Optimization (SGO): A New Population Evolutionary Optimization Technique. *Complex & Intelligent Systems*, **2**, 173-203. <https://doi.org/10.1007/s40747-016-0022-8>
- [5] Kennedy, J. and Eberhart, R. (n.d.). Particle Swarm Optimization. *Proceedings of ICNN95—International Conference on Neural Networks*, Vol. 4, 1942-1948. <https://doi.org/10.1109/icnn.1995.488968>
- [6] Dorigo, M., Birattari, M. and Stutzle, T. (2006) Ant Colony Optimization. *IEEE Computational Intelligence Magazine*, **1**, 28-39. <https://doi.org/10.1109/mci.2006.329691>
- [7] Karaboga, D. (2010) Artificial Bee Colony Algorithm. *Scholarpedia*, **5**, Article No. 6915. <https://doi.org/10.4249/scholarpedia.6915>
- [8] Mirjalili, S. and Lewis, A. (2016) The Whale Optimization Algorithm. *Advances in Engineering Software*, **95**, 51-67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [9] Mirjalili, S., Mirjalili, S.M. and Lewis, A. (2014) Grey Wolf Optimizer. *Advances in Engineering Software*, **69**, 46-61. <https://doi.org/10.1016/j.advengsoft.2013.12.007>
- [10] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H. (2019) Harris Hawks Optimization: Algorithm and Applications. *Future Generation Computer Systems*, **97**, 849-872. <https://doi.org/10.1016/j.future.2019.02.028>
- [11] Agushaka, J.O., Ezugwu, A.E., Saha, A.K., Pal, J., Abualigah, L. and Mirjalili, S. (2024)

- Greater Cane Rat Algorithm (GCRA): A Nature-Inspired Metaheuristic for Optimization Problems. *Heliyon*, **10**, e31629. <https://doi.org/10.1016/j.heliyon.2024.e31629>
- [12] Bouaouda, A., Hashim, F.A., Sayouti, Y. and Hussien, A.G. (2024) Pied Kingfisher Optimizer: A New Bio-Inspired Algorithm for Solving Numerical Optimization and Industrial Engineering Problems. *Neural Computing and Applications*, **36**, 15455-15513. <https://doi.org/10.1007/s00521-024-09879-5>
- [13] Hashim, F.A., Houssein, E.H., Hussain, K., Mabrouk, M.S. and Al-Atabany, W. (2022) Honey Badger Algorithm: New Metaheuristic Algorithm for Solving Optimization Problems. *Mathematics and Computers in Simulation*, **192**, 84-110. <https://doi.org/10.1016/j.matcom.2021.08.013>
- [14] Fu, Y., Liu, D., Chen, J. and He, L. (2024) Secretary Bird Optimization Algorithm: A New Metaheuristic for Solving Global Optimization Problems. *Artificial Intelligence Review*, **57**, Article No. 123. <https://doi.org/10.1007/s10462-024-10729-y>
- [15] Wang, M.N., Wang, Q.P. and Wang, X.F. (2018) Improved Grey Wolf Optimization Algorithm Based on Iterative Mapping and Simplex Method. *Journal of Computer Applications*, **38**, 16-20+54.
- [16] Tan, F.M., Zhao, J.J. and Wang, Q. (2019) A Grey Wolf Optimization Algorithm with Improved Nonlinear Convergence. *Microelectronics & Computer*, **36**, 89-95.
- [17] Chen, P., Zhou, S., Zhang, Q. and Kasabov, N. (2022) A Meta-Inspired Termite Queen Algorithm for Global Optimization and Engineering Design Problems. *Engineering Applications of Artificial Intelligence*, **111**, Article ID: 104805. <https://doi.org/10.1016/j.engappai.2022.104805>
- [18] Si, G.L., Yang, F.Y. and Wang, W.J. (2012) Design and Experimental Study on Relief Valve with Permanent Magnetic Compression Spring. *Journal of Drainage and Irrigation Machinery Engineering*, **30**, 214-218+230.
- [19] Zhao, S., Zhang, T., Ma, S. and Chen, M. (2022) Dandelion Optimizer: A Nature-Inspired Metaheuristic Algorithm for Engineering Applications. *Engineering Applications of Artificial Intelligence*, **114**, Article ID: 105075. <https://doi.org/10.1016/j.engappai.2022.105075>