Postprint of the Beetle Antennae Search Algorithm Based on Quadratic Interpolation

Authors: Liao Liefa, Ouyang Zongying

Date: 2020-09-28T00:00:00+00:00

Abstract

This study addresses the problems of low search precision in high-dimensional spaces and proneness to local optima in the Beetle Antennae Search (BAS) algorithm, proposing a novel beetle antennae optimization algorithm—the Quadratic Interpolation-based Beetle Antennae Search algorithm, designated as QIBAS. After the beetle performs its movement, the algorithm utilizes the positions of its left and right antennae as interpolation points, generating a new solution through quadratic interpolation. It then compares the fitness value of this interpolated solution with those of the current best and global best solutions to update the global optimum. Numerical simulation tests were performed on multiple unimodal and multimodal functions with dimensions set to 100, 500, 1000, 5000, and 10000. The simulation results indicate that the introduction of quadratic interpolation effectively enhances the ability of the BAS algorithm to escape local optima. QIBAS demonstrates substantial improvements in solution precision and notable enhancements in convergence speed when solving for optimal values, thereby validating the effectiveness of the improved algorithm.

Full Text

Preamble

Vol. 38 No. 3
Application Research of Computers
Accepted Paper

Beetle Antennae Search Based on Quadratic Interpolation

Liao Liefa†, Ouyang Zongying
(School of Information Engineering, Jiangxi University of Science & Technology, Ganzhou, Jiangxi 341000, China)

Abstract: This paper addresses the low search precision and susceptibility to local optima of the beetle antennae search (BAS) algorithm in high-dimensional spaces. We propose a novel beetle antennae optimization algorithm—Quadratic Interpolation-based Beetle Antennae Search (QIBAS). After the beetle moves, the algorithm uses the positions of its left and right antennae as interpolation points to generate a new solution via quadratic interpolation. The fitness of this interpolated solution is then compared with the current best and global best solutions to update the global optimum. Numerical simulations were conducted on multiple unimodal and multimodal functions with dimensions of 100, 500, 1000, 5000, and 10000. The results demonstrate that introducing quadratic interpolation effectively enhances the BAS algorithm' s ability to escape local optima. QIBAS achieves substantial improvements in solution accuracy and noticeable gains in convergence speed, thereby validating the effectiveness of the improved algorithm.

Keywords: beetle antennae search; quadratic interpolation; high-dimensional space; global optimum; convergence speed

0 Introduction

Swarm intelligence optimization algorithms emerge from humanity's exploration of nature, simulating the behavior of various biological organisms. These intelligent algorithms have been widely applied in image processing, industrial control, production scheduling, pattern recognition, and numerous other fields. Common examples include ant colony optimization [1], particle swarm optimization [2], genetic algorithms [3], artificial bee colony algorithms [4], and whale optimization algorithms [5]. However, these algorithms exhibit certain deficiencies when solving optimization problems. For instance, ant colony algorithms require complex parameter settings and involve substantial computational overhead; particle swarm algorithms are prone to premature convergence, particularly for complex multimodal problems; and genetic algorithms necessitate encoding and decoding of problems, resulting in high computational complexity. Scholars worldwide have proposed various improvements to address these issues [6–10].

The Beetle Antennae Search (BAS) algorithm [12], proposed by Jiang and Li in 2017, is a novel intelligent optimization algorithm that simulates the foraging behavior of beetles to solve complex optimization problems. BAS has been successfully applied in diverse domains. For example, reference [13] applied BAS to multi-objective energy management in microgrids, offering a new solution for energy optimization and dispatch. Reference [14] utilized BAS to optimize neural network weights and thresholds for predicting the unconfined compressive strength of jet-grouted concrete. Nevertheless, BAS suffers from susceptibility to local optima, insufficient stability, and limitation to single-objective optimization problems [15]. Consequently, researchers have developed various improved versions. Reference [16] introduced a swarm into BAS, proposing the Beetle Swarm Antennae Search (BSAS) algorithm to overcome the random seed-induced variability in optimization results and low precision. Reference [17] incorporated simulated annealing' s Monte Carlo criterion into BAS to accelerate escape from local optima and applied the improved algorithm to distributed generation siting and sizing. Reference [18] combined BAS with particle swarm optimization to enhance convergence speed and search accuracy. Reference [19] integrated BAS with gradient descent, selecting either the gradient descent direction or the optimal direction from the beetle' s "left-right antennae" for iteration.

While these improvements have achieved certain effects, none address BAS's low search precision in high-dimensional spaces. Our investigation reveals that in high-dimensional spaces, beetle individuals may even cease moving, significantly degrading the algorithm's optimization accuracy. To tackle this issue, this paper introduces quadratic interpolation into BAS. After each beetle movement, the positions of its left and right antennae are used to generate a new solution via quadratic interpolation. This interpolated solution is then compared with the current global best to update the global optimum. Simulation results demonstrate that the improved algorithm (QIBAS) achieves superior solution accuracy compared to BAS, PSO, and GA when solving high-dimensional problems.

---

1 Beetle Antennae Search Algorithm

The BAS algorithm differs from other optimization algorithms by employing only a single individual. It requires neither explicit function formulation nor gradient information, resulting in low computational cost, fast optimization, and easy implementation.

1.1 Basic Principles

The biological mechanism underlying BAS is that beetles searching for food cannot locate it precisely. Instead, they use a pair of antennae to sense food odor concentration. If the left antenna detects higher concentration, the beetle moves left; otherwise, it moves right, eventually locating the food. Figure 1 [Figure 1: see original paper] illustrates this foraging process.

The intelligent mechanism of BAS treats food odor concentration as the fitness function, which varies across spatial positions. The beetle' s two antennae sample odor values at two nearby points. By comparing these concentrations and selecting the better one iteratively, the algorithm converges when the difference between the two antennae' s detected concentrations meets a specified precision threshold. The central point found represents the location of highest food concentration.

The beetle' s food search process constitutes the BAS optimization procedure, which comprises the following steps:

a) Initialization of beetle orientation
The initial orientation is created randomly: $\mathbf{d}^0 = \frac{\text{rands}(K,1)}{\|\text{rands}(K,1)\|}$, where rands$(\cdot)$ is a random function and K is the spatial dimension.

b) Step size factor setting
The step size factor ξ^t determines the beetle' s regional search capability. To enhance search range, this paper adopts a large initial step size that decreases linearly: $\xi^t = \xi^0 \cdot \eta \cdot (1 - t/t_n)$, where η is the decreasing factor in $[0, 1]$, t is the current iteration number, and t_n is the total number of iterations.

c) Establishment of antennae positions
The left and right antennae coordinates are established as:
$\mathbf{x}_l^t = \mathbf{x}^t - d^t \cdot \mathbf{d}^t$
$\mathbf{x}_r^t = \mathbf{x}^t + d^t \cdot \mathbf{d}^t$
where $\mathbf{x}_l^t$ and $\mathbf{x}_r^t$ represent left and right antenna positions at iteration t, $\mathbf{x}^t$ is the beetle' s centroid coordinate, and d^t is the inter-antenna length, which should be sufficiently large to cover an appropriate search area and escape local minima initially.

d) Fitness function calculation
The fitness function $f(\cdot)$ represents odor concentration at the antennae: $f_l^t = f(\mathbf{x}_l^t)$ and $f_r^t = f(\mathbf{x}_r^t)$. The specific form depends on the application and will be detailed later.

e) Beetle position update
Comparing the fitness values of the left and right antennae, if $f_l^t < f_r^t$, the beetle moves left; otherwise, it moves right. The update formula is: $\mathbf{x}^{t+1} = \mathbf{x}^t - \xi^t \cdot \mathbf{d}^t \cdot \text{sign}(f_l^t - f_r^t)$.

1.2 Performance Analysis

The BAS optimization process relies primarily on the beetle' s antennae to discriminate food odor concentrations on both sides. While this mechanism accelerates position updates and improves search speed, it also predisposes the beetle to local optima. Although BAS can escape local optima to some extent, each optimization run yields different results due to its stochastic nature. For high-dimensional problems, the single-individual search approach limits the beetle' s ability to distinguish food odor concentrations adequately, potentially causing position stagnation and complete entrapment in local optima, thereby severely degrading BAS' s search precision in high-dimensional spaces.

2 Quadratic Interpolation-Based Beetle Antennae Search Algorithm

2.1 Algorithm Improvement

To address BAS' s low optimization precision in high-dimensional spaces, this paper introduces the quadratic interpolation operator (Equation 6). Quadratic interpolation is a method for searching extremum points within a determined initial interval, representing a curve fitting approach. It constructs a low-degree polynomial approximating the objective function using information at several points, using the polynomial' s optimum as an approximate solution to the original function. As the interval progressively narrows, the distance between the polynomial' s optimum and the original function' s optimum decreases until meeting specified precision requirements. Figure 2 [Figure 2: see original paper] illustrates this principle.

In each beetle position update, this paper uses the left and right antenna positions as the interpolation interval to construct a low-degree polynomial approximating the objective function. The optimal solution from this polynomial is used to narrow the interval iteratively. Upon meeting precision requirements, an interval optimal solution is obtained and compared with the global best solution, with the better of the two becoming the new global optimum. This approach significantly enhances BAS' s local search capability, thereby improving its ability to escape local optima, search precision, and convergence performance in high-dimensional spaces.

Let the beetle' s centroid position in the K-dimensional space at generation t be $\mathbf{x}_b^t = (x_{b1}^t, x_{b2}^t, \dots, x_{bK}^t)^T$, with current left and right antenna coordinates $\mathbf{x}_l^t = (x_{l1}^t, x_{l2}^t, \dots, x_{lK}^t)^T$ and $\mathbf{x}_r^t = (x_{r1}^t, x_{r2}^t, \dots, x_{rK}^t)^T$, respectively, and current global best position $\mathbf{x}_{best}^t = (x_{best,1}^t, x_{best,2}^t, \dots, x_{best,K}^t)^T$. The position generated by quadratic interpolation is:

$$x_{q,k}^t = \frac{(x_{l,k}^t)^2(f(\mathbf{x}_b^t) - f(\mathbf{x}_r^t)) + (x_{b,k}^t)^2(f(\mathbf{x}_r^t) - f(\mathbf{x}_l^t)) + (x_{r,k}^t)^2(f(\mathbf{x}_l^t) - f(\mathbf{x}_b^t))}{2[x_{l,k}^t(f(\mathbf{x}_b^t) - f(\mathbf{x}_r^t)) + x_{b,k}^t(f(\mathbf{x}_r^t) - f(\mathbf{x}_l^t)) + x_{r,k}^t(f(\mathbf{x}_l^t) - f(\mathbf{x}_b^t))] + \text{eps}}$$

where $f(\cdot)$ is the fitness function and eps is a very small positive number to prevent division by zero.

The QIBAS algorithm proceeds as follows:

a) Initialization
Randomly generate the beetle' s initial orientation $\mathbf{d}^0$ using Equation (1), initial centroid position $\mathbf{x}^0 = (x_1^0, x_2^0, \dots, x_K^0)^T$, and compute the fitness value $f(\mathbf{x}^0)$ using Equation (4). Set step size factor ξ^0, decreasing factor η, initialize $t = 0$, and set $\mathbf{x}_{best} = \mathbf{x}^0$ with fitness value $f(\mathbf{x}_{best}) = f(\mathbf{x}^0)$.

b) Antennae coordinate and fitness calculation
Determine the beetle' s antennae coordinates based on its current position as

shown in Equation (3): $\mathbf{x}_l^t = \mathbf{x}^t - d^t \cdot \mathbf{d}^t$ and $\mathbf{x}_r^t = \mathbf{x}^t + d^t \cdot \mathbf{d}^t$, where $t = 1, 2, \ldots, n$ and n is the total number of iterations. Compute the antennae fitness values $f_l^t = f(\mathbf{x}_l^t)$ and $f_r^t = f(\mathbf{x}_r^t)$ using Equation (4).

c) Quadratic interpolation and position update
Update the beetle' s position coordinate $\mathbf{x}^{t+1}$ using Equation (5) and compute the corresponding fitness value $f(\mathbf{x}^{t+1})$. Then calculate the interpolated position coordinate $\mathbf{x}_q^{t+1}$ using Equation (6) and its fitness value $f(\mathbf{x}_q^{t+1})$. If $f(\mathbf{x}_q^{t+1}) < f(\mathbf{x}^{t+1})$, update the beetle' s position to $\mathbf{x}^{t+1} = \mathbf{x}_q^{t+1}$; otherwise, maintain the current position.

d) Global best update
Compare $f(\mathbf{x}^{t+1})$ with $f(\mathbf{x}_{best})$. If $f(\mathbf{x}^{t+1}) < f(\mathbf{x}_{best})$, update $\mathbf{x}_{best} = \mathbf{x}^{t+1}$ and $f(\mathbf{x}_{best}) = f(\mathbf{x}^{t+1})$.

e) Termination
If the termination condition is met, stop; otherwise, set $t = t + 1$ and return to step b).

2.2 Time Complexity Analysis of the Improved Algorithm

The primary improvement in this paper involves generating a point distinct from the current position via quadratic interpolation after each beetle position update, then selecting the better of the two. For an optimization problem of dimension K, the computational complexity analysis is as follows: initialization complexity is $O(K)$; during iteration, computing antennae coordinates and fitness values requires $O(2K)$, quadratic interpolation generation requires $O(6K)$, and position update and fitness calculation require $O(2K)$. The per-iteration complexity is $O(10K)$, and the total complexity approximates $O(10K \cdot n)$. Table 1 compares the computational complexity of Genetic Algorithm (GA), Particle Swarm Optimization (PSO), standard BAS, and QIBAS, where N represents population size.

Table 1 Algorithm Complexity

Algorithm	Computational Complexity
GA	$O(N^2 \cdot K \cdot n)$
PSO	$O(N \cdot K \cdot n)$
BAS	$O(K \cdot n)$
QIBAS	$O(10K \cdot n)$

As shown, algorithm complexity depends on problem dimension and population size. According to Big-O notation, when optimization problem dimension K far exceeds population size N, GA and PSO have complexities of $O(N^2 \cdot K \cdot n)$ and $O(N \cdot K \cdot n)$ respectively, while BAS and QIBAS have complexity $O(K \cdot n)$.

2.3 Parameter Analysis of the Improved Algorithm

Based on the algorithm model, QIBAS requires setting parameters including: initial step size ξ^0, step decreasing factor η, maximum iterations n, and spatial dimension K. For most applications, the initial step size ξ^0 can be set as needed; this paper uses $\xi^0 = 1$. Single-factor numerical experiments analyze the impact of the remaining three parameters on algorithm performance using the test functions in Table 2 .

Table 2 Test Functions

Function	Formula
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$
Rosenbrock	$f_2(x) = \sum_{i=1}^{D-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Rastrigin	$f_3(x) = \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i) + 10]$

Empirical parameter settings are shown in Table 3 . During single-factor experiments, all parameters except one are fixed while the target parameter varies for simulation.

Table 3 Empirical Parameter Settings

Parameter	Value
Step decreasing factor η	0.1, 0.2, 0.3, 0.4, 0.9, 1
Maximum iterations n	1000, 2000, 3000, 4000
Dimension K	200

a) Impact of step decreasing factor η

With η values in the common range $[0, 1]$, experiments were conducted with $\eta = 0.1, 0.2, 0.3, 0.4, 0.9, 1$. Each test function underwent 20 independent runs. Results are shown in Table 4 .

The step decreasing factor enables different search radii during different algorithm phases. Early stages require large search ranges to accelerate convergence, while later stages need smaller ranges to pinpoint the optimum as iterations progress and solutions approach optimality. Table 4 shows that for the Sphere function, $\eta = 0.1$ yields optimal performance; for Rosenbrock, $\eta = 0.9$ is best; and for Rastrigin, $\eta = 0.1$ performs optimally.

b) Impact of maximum iterations

Maximum iterations were set to $n = 1000, 2000, 3000, 4000$, with 20 independent runs per test function. Results in Table 5 indicate that maximum iterations have minimal impact on algorithm performance. Considering all factors, this paper selects $n = 2000$.

---

3 Experiments and Analysis

3.1 Parameter Settings

All simulations were conducted on the Matlab2018b platform to test and analyze QIBAS performance on benchmark functions. For BAS and QIBAS, initial step size $\xi^0 = 1$, decreasing factor $\eta = 0.9, 0.1$, total iterations $n = 2000$, and problem dimensions $K = 100, 500, 1000$ were tested across 50 runs. GA and PSO used population sizes of 200, with PSO learning factors set to 1.

3.2 Test Functions

To validate algorithm performance, this paper selected 11 standard benchmark functions [20] including both unimodal and multimodal types, detailed in Tables 6 and 7 .

Table 6 Unimodal Functions

Function	Formula
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$
Schwefel Problem 2.22	$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$
Schwefel Problem 1.2	$f_3(x) = \sum_{i=1}^{D} (\sum_{j=1}^{i} x_j)^2$
Rosenbrock	$f_4(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$
Quartic	$f_5(x) = \sum_{i=1}^{D} i x_i^4 + \text{random}[0, 1)$
Schwefel Problem 2.21	$f_6(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le D\}$

Table 7 Multimodal Functions

Function	Formula
Schwefel Problem 2.26	$f_7(x) = -\sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$
Rastrigin	$f_8(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$
Ackley	$f_9(x) = -20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$
Griewank	$f_{10}(x) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$

3.3 Experimental Results and Analysis

QIBAS was compared against GA, PSO, and BAS, with each algorithm running 50 times. Mean values and standard deviations were computed, with results shown in Table 8 (for dimension $K = 200$). Friedman non-parametric statistical analysis was performed with significance level $\alpha = 0.05$, shown in Table 9 . Table 10 presents the computational time for 50 runs across three functions.

Table 8 Comparison of Algorithm Test Results

Function	QIBAS	BAS	PSO	GA
f_1	8.62×10^4	9.35×10^3	4.47×10^2	8.12×10^3
f_2	9.69×10^5	5.88×10^6	3.40×10^{-3}	3.499×10^2
f_3	8.65×10^{-7}	1.40×10^{-6}	7.00×10^2	2.01×10^2
f_4	7.08×10^{-7}	2.00×10^2	5.65×10^4	3.35×10^2
f_5	2.61×10^3	2.82×10^3	4.30×10^2	3.29×10^{-4}
f_6	3.15×10^{-4}	-1.69×10^{-2}	9.13×10^{-4}	-1.33×10^3
f_7	7.84×10^3	9.91×10^2	-6.25×10^2	-1.99×10^3
f_8	3.90×10^{-3}	7.91×10^{-3}	5.62×10^{-4}	2.63×10^{-4}
f_9	1.12×10^{-3}	-	-	-

Table 8 shows (with best values bolded) that QIBAS converged to the global minimum of 0 for functions f_1, f_6, and f_{11} across all 50 runs with zero standard deviation, demonstrating high stability. For functions f_2, f_3, f_4, f_5, f_7, f_8, f_9, and f_{10}, QIBAS outperformed PSO. While results for f_1 and f_5 were inferior to PSO, they remained superior to GA. Compared with standard BAS, QIBAS showed better mean and standard deviation for all functions except f_5, f_6, and f_{10}. These results confirm that QIBAS significantly improves computational accuracy and stability, validating the effectiveness of the improvement.

Table 9 Friedman Test

Algorithm	Mean Rank
QIBAS	1.45
BAS	2.73
PSO	2.82
GA	3.00

The Friedman test results show QIBAS has the smallest mean rank, indicating superior performance among the four algorithms.

Table 10 Algorithm Running Time (seconds)

Function	QIBAS	BAS	PSO	GA
f_1	0.12	0.10	15.3	18.7
f_5	0.15	0.13	16.8	19.2
f_9	0.18	0.16	17.5	20.1

Table 10 reveals that QIBAS and BAS runtimes are far shorter than PSO and GA for both unimodal and multimodal functions, with time differences exceeding 100-fold in some cases. QIBAS requires slightly more time than BAS due to the interpolation operation, while PSO and GA show varying performance. Thus, QIBAS and BAS demonstrate superior computational efficiency.

To further examine quadratic interpolation' s impact, additional comparisons were made with Simulated Annealing BAS (SABAS), Variable Step BAS (VSBAS), and BSAS across 50 runs, with results in Table 11 .

Table 11 Comparison of Improved Algorithm Test Results

Function	SABAS	VSBAS	QIBAS
f_1	4.93×10^{-30}	2.22×10^{-15}	8.65×10^{-7}
f_2	9.69×10^{5}	5.88×10^{6}	7.08×10^{-7}
f_3	9.46×10^{4}	7.25×10^{4}	2.00×10^{2}
f_4	2.47×10^{2}	3.91×10^{3}	5.65×10^{4}
f_5	2.32×10^{-41}	2.68×10^{2}	3.35×10^{2}
f_6	3.40×10^{-3}	1.12×10^{-2}	2.61×10^{3}
f_7	1.57×10^{-15}	7.00×10^{2}	2.82×10^{3}
f_8	9.73×10^{2}	1.04×10^{4}	4.30×10^{2}
f_9	1.09×10^{3}	2.01×10^{2}	3.29×10^{-4}
f_{10}	7.08×10^{-7}	2.96×10^{2}	3.15×10^{-4}
f_{11}	2.82×10^{3}	4.30×10^{2}	-1.69×10^{-2}

The mean values and standard deviations indicate that SABAS and VSBAS improve upon standard BAS to varying degrees, but the improvements are not substantial. QIBAS significantly outperforms SABAS on all functions except f_1 and f_5. Compared with VSBAS, QIBAS shows slightly worse results on f_1 but superior performance on all other functions. QIBAS also outperforms BSAS on most test cases. Table 12 presents Friedman test results confirming QIBAS' s smallest mean rank among these improved algorithms.

To visually demonstrate convergence characteristics, experiments with results closest to the 50-run average were selected to plot convergence curves for four test functions at dimension $K = 200$, comparing VSBAS, SABAS, BSAS, and BAS. Figures 3 [Figure 3: see original paper]–6 [Figure 6: see original paper] show convergence performance for functions f_1, f_4, f_9, and f_{11}. Figures 7 [Figure 7: see original paper] and 8 [Figure 8: see original paper] display convergence curves at $K = 500$ and $K = 1000$.

At $K = 200$, BAS, BSAS, and SABAS show varying convergence effectiveness across different functions, while VSBAS consistently outperforms BAS but remains significantly inferior to QIBAS. The convergence plots clearly demonstrate QIBAS' s superior performance over all four comparison algorithms.

At $K = 500$ and $K = 1000$, BAS, VSBAS, and SABAS show minimal convergence or even fail to converge, while BSAS only performs well on function f_1. This confirms BAS' s limitations in high-dimensional spaces. However, QIBAS maintains good convergence properties, with solutions at $K = 1000$ approaching true optima more closely than at $K = 500$.

To further validate feasibility at even higher dimensions, experiments were conducted at $K = 5000$ and $K = 10000$, with results shown in Figures 9 [Figure 9: see original paper] and 10 [Figure 10: see original paper]. The plots demonstrate that QIBAS maintains superior optimization performance even at these extreme dimensions.

In summary, QIBAS achieves high convergence precision and rapid convergence speed as dimensions increase. Its consistent performance across different functions highlights the stability of the improvement strategy. In contrast, the other four algorithms exhibit diminished effectiveness, slower convergence, or complete failure to converge as dimensions grow, further validating QIBAS' s stability and effectiveness.

---

4 Conclusion

To address the low search precision and local optima susceptibility of the beetle antennae search algorithm in high-dimensional spaces, this paper proposes a Quadratic Interpolation-based Beetle Antennae Search algorithm. Quadratic interpolation enhances the beetle' s ability to escape local optima and improves search precision in high-dimensional spaces. Simulation experiments on 11 benchmark functions demonstrate that compared with GA, PSO, and standard BAS, the improved algorithm achieves significant enhancements in computational accuracy, stability, and convergence speed. Comparisons with SABAS, VSBAS, and BSAS further confirm through statistical analysis and computational results that the proposed improvement strategy is both effective and stable.

References

[1] Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents [J]. IEEE Trans on Systems Man & Cybernetics Part B Cybernetics A, 1996, 26 (1): 29.

[2] Kennedy J, Eberhart R. Particle swarm optimization [C]// Proc of IEEE International Conference on Neural Networks. Perth, Australia, USA: IEEE Press, 1995: 1942-1948.

[3] Coit D. Genetic algorithms and engineering design [J]. Engineering Economist, 1998, 43 (4): 379-381.

[4] Karaboga D. An idea based on honey bee swarm for numerical optimization: Technical Report-TR06 [R]. Kayseri: Erciyes University, 2005.

[5] Mirjalili S, Lewis A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.

[6] Deng Wu, Xu Junjie, Zhao Huimin. An improved ant colony optimization algorithm based on hybrid strategies for scheduling problem [J]. IEEE Access, 2019, 7: 20281-20292.

[7] Nouiri M, Bekrar A, Jemai A, et al. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem [J]. Journal of Intelligent Manufacturing, 2018, 29 (3): 603-615.

[8] Abdelhalim H, Ali D, Iyad R. A genetic algorithm approach for location-inventory-routing problem with perishable products [J]. Journal of Manufacturing Systems, 2017, 42.

[9] Wang Wenjie, Tian Guangdong, Chen Maoning, et al. Dual-objective program and improved artificial bee colony for the optimization of energy-conscious milling parameters subject to multiple constraints [J]. Journal of Cleaner Production, 2020, 245.

[10] Chu Dingli, Chen Hong, Wang Xuguang. Whale optimization algorithm based on adaptive weights and simulated annealing [J]. Acta Electronica Sinica, 2019, 47 (05): 992-999.

[11] Xiao Liqing, Wang Huaxiang. Optimization of ERT finite element model using improved particle swarm optimization [J]. Application Research of Computers, 2017, 34 (05): 1581-1584.

[12] Jiang Xiangyuan, Li Shuai. BAS: beetle antennae search algorithm for optimization problems [J]. arXiv preprint arXiv: 1710. 10724, 2017.

[13] Zhu Zongyao, Zhang Zhiyu, Man Weishi, et al. A new beetle antennae search algorithm for multi-objective energy management in microgrid [C]// Pro of the 13th IEEE Conference on Industrial Electronics and Applications (ICIEA). IEEE, 2018: 1599-1603.

[14] Sun Yuantian, Zhang Junfei, Li Guichen, et al. Optimized neural network using beetle antennae search for predicting the unconfined compressive strength of jet grouting coalcretes [J]. International Journal for Numerical and Analytical Methods in Geomechanics, 2019, 43 (4): 801-815.

[15] Jiang Xiangyuan, Li Shuai. Beetle antennae search without parameter tuning (BAS-WPT) for multiobjective optimization [J]. arXiv preprint arXiv: 1711. 02395, 2017.

[16] Wang Jiangyu, Chen Huanxin. BSAS: Beetle Swarm Antennae Search Algorithm for Optimization Problems [J]. arXiv preprint arXiv: 1807. 10470, 2018.

[17] Lu Guanghui, Teng Huan, Liao Hanxun, et al. Location and volume of distributed power supply based on improved beetle search algorithm [J]. Electrical Measurement and Instrumentation, 2019, 56 (17): 6-12.

[18] Chen Tingting, Zhu Yongjian, Teng Jun. Beetle swarm optimization for solving investment portfolio problems [J]. The Journal of Engineering, 2018, 2018 (16): 1600-1605.

[19] Kong Huihua, Sun Yingbo, Zhang Yanxia. The application of the full variational minimization algorithm based on the beetle whisker search in the reconstruction of computer tomography [J]. Las Optoelect Prog, 2019, 56 (21): 90-96.

[20] Fang Wei, Sun Jun, Chen Huanhuan, et al. A decentralized quantum-inspired particle swarm optimization algorithm with cellular structured population [J]. Information Sciences, 2016, 330 (C): 19-48.

Note: Figure translations are in progress. See original paper for figures.