HIGH TECHNOLOGY LETTERS | Vol. 31 No. 1 | Mar. 2025 | pp. 63-72

doi:10.3772/j.issn.1006-6748.2025.01.007

Multi-strategy improved honey badger algorithm based on periodic mutation and t-distribution perturbation①

WU Jin(吴　进)②, SU Zhengdong, TIAN Jinhang, WEN Fei, CHEN Wenfeng

(School of Electronic Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710121, P. R. China)

Abstract

The honey badger algorithm (HBA), as a new swarm intelligence (SI) optimization algorithm, has shown certain effectiveness in its applications. Aiming at the problems of unsatisfactory initial population distribution of HBA, poor ability to avoid local optimum, and slow convergence speed, this paper proposes a multi-strategy improved HBA based on periodical mutation and t-distribution perturbation, called MHBA. Firstly, a good point set population initialization is introduced to get a uniform initial population. Secondly, periodic mutation and t-distribution perturbation are successively used to improve the algorithm's ability to avoid local optimum. Finally, the density factor is improved for balancing exploration and exploitation. By comparing MHBA with HBA and 7 other SIs on 6 benchmark functions, it is evident that the performance of MHBA is far superior to HBA. In addition, by applying MHBA to robot path planning, MHBA can identify the shortest path more quickly and consistently compared with competitors.

Key words: periodic mutation, t-distribution, linear decreasing factor, robot path planning

0 Introduction

Recently, many fields such as image processing[1], transportation[2], civil engineering[3], and others are facing increasingly complex optimization problems. The meta-heuristic algorithm (MA) can be flexibly adapted to different scenarios due to the advantages of its high exploration and exploitation capabilities. Moreover, MA can better explore the solution space and quickly find globally optimal solution[4].

Swarm intelligence (SI) optimization algorithms, as a branch of MA, have been favored by more and more researchers. SI algorithm is mainly inspired by nature, especially biological systems, and simulates the cooperative behavior of insects, animals, and others through mathematical modeling[5]. Like MA, SI algorithm has two common steps: exploration and exploitation. A successful algorithm should be able to strike the right balance between exploration and exploitation to address the problems of local optimality and premature convergence[6].

Some classic SIs including particle swarm optimization (PSO)[7] simulates the foraging behavior of a flock of birds. The ant colony optimization (ACO)[8] simulates the foraging behavior of ants. The cuckoo search algorithm (CSA)[9] is inspired by the life habits of cuckoos. These classical SIs often have the disadvantages of slow convergence speed when facing complex problems, more difficult parameter selection, and difficulty in dealing with high-dimensional problems. To address these disadvantages, some famous SIs have been proposed in the last decade. The grey wolf optimization (GWO)[10] is inspired by the hierarchical division of gray wolves in hunting for prey. The moth-flame optimization (MFO)[11] is inspired by moths flying around flames. The Harris hawk optimization (HHO)[12] is inspired by the various hunting strategies of Harris hawks.

Some recently proposed SIs include golden jackal optimizer (GJO)[13], crested porcupine optimizer (CPO)[14] and black-winged kite algorithm (BKA)[15]. It is worth mentioning that the recently proposed SI often has the drawbacks of poor local search capability and difficulty in achieving a proper balance between exploration and exploitation. Therefore, this paper carefully studies and improves the recently proposed honey badger algorithm (HBA)[16].

HBA simulates the digging and honeying behaviors of honey badgers during foraging. HBA are com-

① Supported by the National Key Research and Development Program of China (No. 2022ZD0119001)

② To whom correspondence should be addressed. E-mail: wujin1026@126.com.

Received on Apr. 20, 2024

petitive and effective compared with traditional SIs, but it still has some shortcomings. For example, the initial population distribution is not ideal, the ability to avoid local optimum is poor and the convergence speed is slow.

To improve the optimization ability of HBA, this paper proposes a multi-strategy improved honey badger algorithm (MHBA) based on period mutation and t-distribution perturbation. Firstly, to obtain an ideal initial population, a population initialization based on the good point set is introduced. Secondly, to enhance the algorithm's ability to escape local optima, a periodic mutation mechanism is integrated. Thirdly, to balance exploration and exploitation, the density factor of HBA is improved. Finally, to further accelerate the convergence speed of the algorithm, a t-distribution perturbation is added to the global optimal individual.

1 HBA

HBA is inspired by the digging and honeying behaviors of the honey badger during foraging.

1.1 Smell intensity

In HBA, each honey badger will rely on the smell of its prey for digging. Define the smell intensity I_i^t to indicate the proximity of the prey to the honey badger, as shown in Eq. (1).

$$I_i^t = r_1 \frac{S_i^t}{4\pi (\boldsymbol{d}_i^t)^2} \tag{1}$$

$$S_i^t = (\boldsymbol{x}_i^t - \boldsymbol{x}_{i+1}^t)^2 ; \boldsymbol{d}_i^t = \boldsymbol{x}_{\mathrm{prey}}^t - \boldsymbol{x}_i^t \tag{2}$$

In Eq. (1), t denotes the tth iteration, S_i^t denotes the source strength at the position of the ith honey badger, $\boldsymbol{d}_i^t$ denotes the distance vector between the prey and the ith honey badger, and r_1 is a random number in the interval [0,1]. In Eq. (2), $\boldsymbol{x}_i^t$ denotes the position of the ith honey badger and $\boldsymbol{x}_{\mathrm{prey}}^t$ denotes the position of the prey (i.e., the location of the elite solution).

1.2 Density factor

In HBA, the density factor $\beta(t)$ is used to control the transition of the algorithm from exploration to exploitation as shown in Eq. (3).

$$\beta(t) = C\mathrm{e}^{-\frac{t}{T}} \tag{3}$$

where, T represents the maximum number of iterations; C is a constant $C \geqslant 1$, and $C = 2$ is taken in HBA.

1.3 Update individual positions

The position update of HBA is divided into two main phases: the digging phase ($r < 0.5$) and the honey phase ($r \geqslant 0.5$), r is a random number in the interval [0,1].

1.3.1 Digging phase

In digging phase, HBA uses a cardioid-like motion for position update as shown in Eq. (4).

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_{\mathrm{prey}}^t + F \cdot \lambda \cdot I \cdot \boldsymbol{x}_{\mathrm{prey}}^t + F \cdot r_2 \cdot \beta(t) \cdot \boldsymbol{d}_i^t \cdot | \cos(2\pi r_3) \cdot [1 - \cos(2\pi r_4)] | \tag{4}$$

$$F = \begin{cases} 1 & r_5 \leqslant 0.5 \\ -1 & r_5 > 0.5 \end{cases} \tag{5}$$

where, $\boldsymbol{x}_i^{t+1}$ denotes the position of the ith honey badger after the $t+1$ iteration update; λ represents the ability of the honey badger to capture food and $\lambda \geqslant 1$, $\lambda = 6$ is taken in HBA; F is obtained through Eq. (5) and is used to change the search direction; r_2, r_3, r_4, and r_5 are all random numbers in the interval [0,1], and independent of each other.

1.3.2 Honey phase

During the honey phase, the honey badger searching for the hive is shown in Eq. (6).

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{x}_{\mathrm{prey}}^t + F \cdot r_6 \cdot \beta(t) \cdot \boldsymbol{d}_i^t \tag{6}$$

where, r_6 is a random number in the interval [0,1], and the other variables are the same as in Eq. (4).

1.4 Greedy choice

The HBA uses Eq. (7) for greedy choice after the digging phase or the honey phase to complete the algorithm iteration.

$$\boldsymbol{x}_i^{t+1} = \begin{cases} \boldsymbol{x}_i^{t+1} & f(\boldsymbol{x}_i^{t+1}) \leqslant f(\boldsymbol{x}_i^t) \\ \boldsymbol{x}_i^t & f(\boldsymbol{x}_i^{t+1}) > f(\boldsymbol{x}_i^t) \end{cases} \tag{7}$$

where, $f(\cdot)$ denotes the fitness value of the objective function.

2 MHBA

To address the shortcomings of HBA, a series of improvement strategies is proposed to enhance the performance and accuracy of HBA.

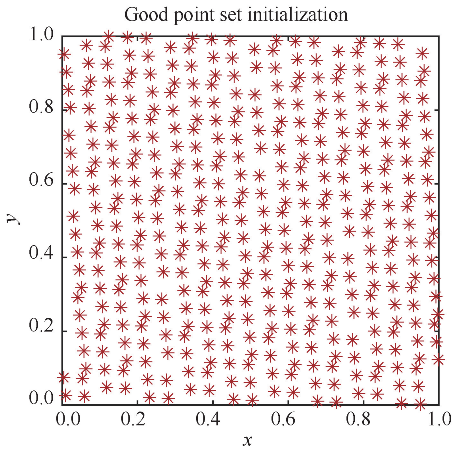
2.1 Good point set population initialization

Professor Hua Luogeng and other mathematicians proposed the good point set, which is defined as follows[17]: suppose that there exists a set $Q_n(w) = \{J_{1,n} \cdot w, J_{2,n} \cdot w, \cdots, J_{s,n} \cdot w; 1 \leqslant w \leqslant n\}$ in a Euclidean space H_s of dimensions which has a deviation $\varphi(n) = C(J,\mu) n^{-1+\mu}$, then the set $Q_n(w)$ is said to be a good point set and J is a good point. The value of the good point J is given by $J^D = 2\cos(2\pi D/p)$, where $1 \leqslant D \leqslant n$, p is the smallest prime number that satisfies the expression $p \geqslant 2s+3$. The good point set population initialization is mapped through Eq. (8).

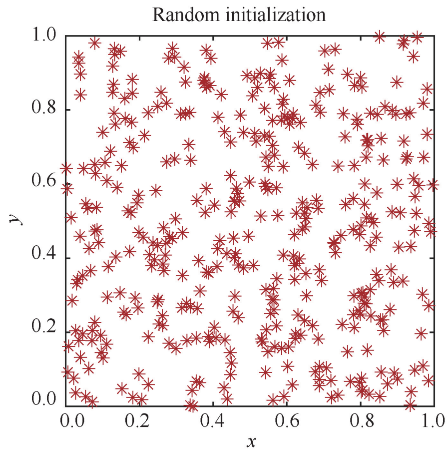
$$x_{i,D} = J^D \cdot (U_D - L_D) + L_D \tag{8}$$

where, $x_{i,D}$ denotes the Dth dimension of the ith individual, and U_D and L_D denote the upper and lower bounds of the Dth dimension.

Fig. 1 gives the distribution of 400 points generated by good point set population initialization and random initialization in two-dimensional (2D) space. From Fig. 1, it can be seen that the points generated by the good point set are more uniformly distributed than the randomly generated points, and achieve a uniform distribution globally. Thus, the search space can be explored more comprehensively by initializing the population based on the good point set.



(a) Good point set



(b) Random

Fig. 1 Distribution of 400 points

2.2 Periodic mutation mechanism

The introduction of a periodic mutation mechanism allows MHBA to make large-scale periodic jumps in the solution space, enhancing the algorithm's ability to escape local optima, as shown in Eq. (9).

$$\begin{aligned} \boldsymbol{x}_i^{t+1} &= \boldsymbol{x}_i^t \cdot [1 + b_0(0.5 - n_1) \cdot \delta] \\ &+ round(0.5 \cdot (0.05 + r_7)) \cdot n_2 \boldsymbol{U} \cdot \delta, \\ \delta &= \begin{cases} 1 & t = mT_0 \\ 0 & t \neq mT_0 \end{cases}, \quad m = 1,2,3,\cdots \end{aligned} \tag{9}$$

$$b_0 = \begin{cases} 1 & q_1 > q_2 \\ -1 & q_1 \leqslant q_2 \end{cases} \tag{10}$$

In Eq. (9), b_0 is either -1 or 1, which is determined by Eq. (10); T_0 is the mutation period, which is smaller than the maximum number of iterations, and $T_0 = 5$; m denotes that t is m times T_0, indicating the execution of the mth periodic mutation; the parameter δ is used to control whether to perform periodic mutation and perturbation; n_1 and n_2 are random numbers obeying the standard normal distribution; r_7 is a random number in the interval $[0,1]$; $round(\cdot)$ is rounding; and $\boldsymbol{U}$ is a D-dimensional column vector with elements taking 1. In Eq. (10), q_1 and q_2 are random numbers in the interval $[0,1]$.

Unlike ordinary periodic mutations, MHBA introduces a perturbation[18]. The perturbation not only simulates the deviation in the movement process of honey badgers, but also further prevents the algorithm from falling into local optima from the perspective of algorithm performance. The last term in Eq. (9) is the introduced perturbation, where, n_2 denotes the magnitude of the perturbation and the vector $\boldsymbol{U}$ is used to add an identical perturbation to each dimension of the ith individual. Thus, the updated iteration equation for MHBA is shown in Eq. (11).

$$\boldsymbol{x}_i^{t+1} = \begin{cases} \begin{aligned} &\boldsymbol{x}_i^t \times [1 + b_0 \times (0.5 - n_1)] + \\ &round(0.5 \cdot (0.05 + r_7)) \cdot n_2 U \end{aligned} & t = mT_0 \\ \begin{cases} \begin{aligned} &\boldsymbol{x}_{\text{prey}}^t + F \cdot \lambda \cdot \boldsymbol{I} \cdot \boldsymbol{x}_{\text{prey}}^t + F \\ &\cdot r_2 \cdot \beta(t) \cdot d_i^t \cdot | \cos(2\pi r_3) \cdot \\ &[1 - \cos(2\pi r_4)] | \end{aligned} & r < 0.5 \\ \boldsymbol{x}_{\text{prey}}^t + F \cdot r_6 \cdot \beta(t) \cdot \boldsymbol{d}_i^t & r \geqslant 0.5 \end{cases} & t \neq mT_0 \end{cases} \tag{11}$$

2.3 Linearly decreasing density factor

The HBA's density factor $\beta(t)$ hopes to enable the transition from exploration to exploitation, but it does not take into account the various stochastic scenarios in which the algorithm iterates. Meanwhile, the value of $\beta(t)$ in HBA remains large in the late iteration, which is very harmful to the algorithm for exploitation behavior. In addition, the division between exploration and exploitation in the HBA is not clear, resulting in an imbalance between exploration and exploitation. Therefore, MHBA introduces a linear decreasing factor $\alpha(t)$[10] for replacing $\beta(t)$ as in Eq. (12).

$$\alpha(t) = C \cdot \alpha_0 \left(1 - \frac{t}{T}\right) \tag{12}$$

$$\alpha_0 = 2r_8 - 1 \tag{13}$$

where, r_8 is a random number in the interval $[0,1]$; C is taken as 2, which is the same as Eq. (3); α_0 is a random number between -1 and 1. The range of $\alpha(t)$ is $(-2,2)$, thus, $|\alpha(t)|$ decreases from 2 to 0 with the number of iterations.

After 1 000 iterations, as shown in Fig. 2, the value of the horizontal dashed line is 1; the vertical dashed line represents the iteration number 500. In the first 500 iterations, larger values of both $|\alpha(t)|$ and $\beta(t)$ are beneficial for the algorithm to perform exploration operations. However, the value of $\beta(t)$ remains large late in the iteration, which is not beneficial for the algorithm to perform the localized search, and $|\alpha(t)|$ can address this deficiency. Meanwhile, $\alpha(t)$ is a number that can be positive or negative during the iteration process, which can simulate the deviation of the honey badger during its actual movement.

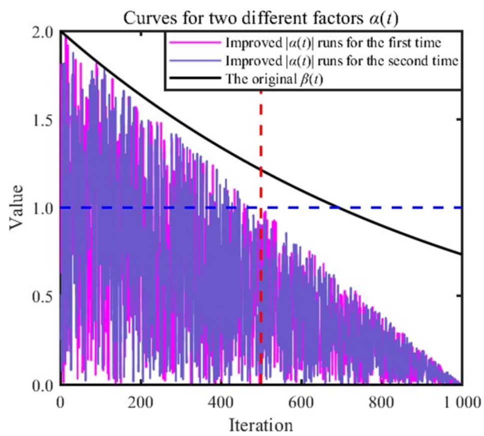


Fig. 2 The curves of the $|\alpha(t)|$ for two runs and $\beta(t)$

In addition, $\alpha(t)$ converges from 2 (or -2) to 0 during the iteration process. Therefore, Eq. (6) will converge to the global optimum as the iteration progresses, as shown in Eq. (14).

$$\lim_{t \to T} (\boldsymbol{x}_{\text{prey}}^t + F \cdot r_6 \cdot \alpha(t) \cdot \boldsymbol{d}_i^t) = \boldsymbol{x}_{\text{prey}}^t \tag{14}$$

2.4 t-distribution perturbation

The shape of the t-distribution curve is related to the degree of freedom. The greater the degree of freedom, the closer the distribution is to a normal distribution[19]. In this paper, the elite individual is perturbed by the t-distribution to realize the variation of the population, because the position update of HBA is very dependent on the elite individual, as shown in Eq. (15). In MHBA, after executing Eq. (15) Eq. (7) is required for greedy choice.

$$\boldsymbol{x}_{\text{prey}}^t = \boldsymbol{x}_{\text{prey}}^t + \boldsymbol{x}_{\text{prey}}^t \cdot trnd(t) \tag{15}$$

In Eq. (15), $trnd(t)$ is the coefficient of variance obeying a t-distribution with degrees of freedom of the current iteration number t. Fig. 3 illustrates the t-distribution function corresponding to the different degrees of freedom. Defining mutation factors obeying the t-distribution located in $[-1, 1]$, the algorithm focuses more on local search, while in other cases the algorithm focuses more on global search. P is the probability that the mutation factor lies in $[-1, 1]$. As the degrees of freedom increase from 0.2 to 10.0, the region where P is located becomes larger and larger. From Eq. (15), the distribution of mutation factors in MHBA is related to the number of iterations. Thus, the t-distribution of this paper allows MHBA to search globally at the beginning of the iteration and locally at the end of the iteration.

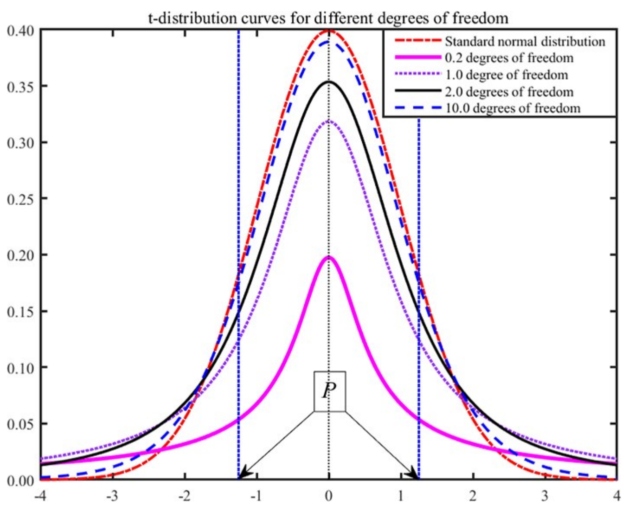


Fig. 3 The t-distribution function corresponding to different degrees of freedom

The flowchart of MHBA is shown in Fig. 4.

3 Experimental results and analysis

To test and validate the performance of MHBA, 6 benchmark functions from Ref. [10] are used to make a comparison with HBA and 7 advanced SIs. All results in the paper are from Intel(R) Core(TM) i7-8550U CPU @1.80 GHz, 1.99 GHz and Windows 10 operating system, done on Matlab R2019b software.

3.1 Test functions and SIs for comparison

The information of the benchmark function is shown in Table 1, where Range is the boundary of the function search space, D is the dimension of the function, and $f_{\min}$ is the optimal value.

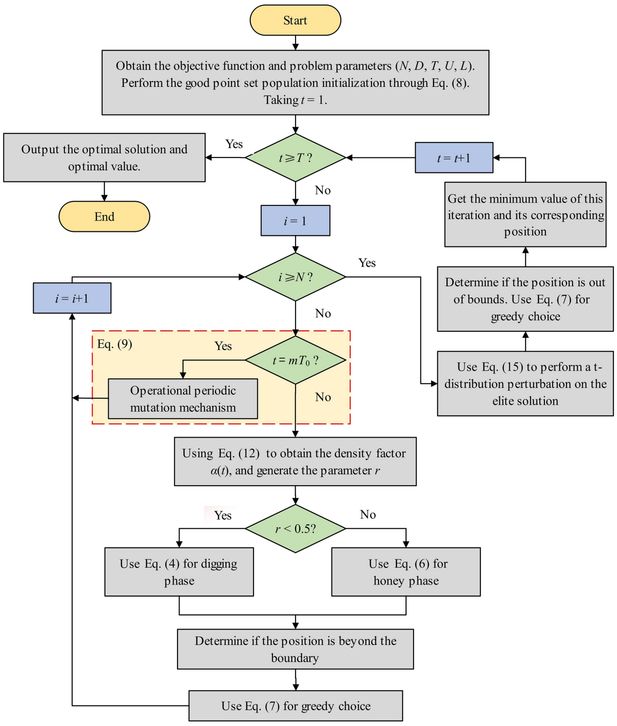


Fig. 4 Flowchart of the MHBA

Seven advanced SIs including PSO, GWO, MFO, HHO, GJO, CPO, and BKA as well as HBA are compared with the proposed MHBA. Meanwhile, to make the experimental results more fair, the population size of all algorithms is $N = 30$ and the maximum number of iterations is $T = 1\ 000$. All algorithms are run independently 30 times for each function, and the mean (*Mean*), standard deviation (*Std*), and rank (*Rank*) of these 30 results are calculated.

3.2 Qualitative analysis

In this section, 4 functions (F1, F2, F4, and F6) are selected for qualitative analysis. Since these 4 benchmark functions are sufficiently representative of unimodal and multimodal functions for various cases, the other functions are their variants. The 6 subfigures of Fig. 5 are (1) the 3D image of the benchmark function, (2) the search history of each search agent, (3) the trajectory of the first dimension of the elite individual, (4) the average fitness value of the search agent, (5) the curve of the density factor $\alpha(t)$, and (6) the convergence curves of all the algorithms.

Table 1 Algorithm parameter setting

Type	Number	Name	Range	D	f_{min}
Unimodal	F1	Schwefel's 2.22	$[-10, 10]^D$	30	0.000 0
	F2	Schwefel's 1.20	$[-100, 100]^D$	30	0.000 0
	F3	Rosenbrock	$[-30, 30]^D$	30	0.000 0
Multimodal	F4	Rastrigin	$[-5.12, 5.12]^D$	30	0.000 0
	F5	Ackley	$[-32, 32]^D$	30	0.000 0
	F6	Shekel 7	$[0, 10]^D$	4	−10.402 9

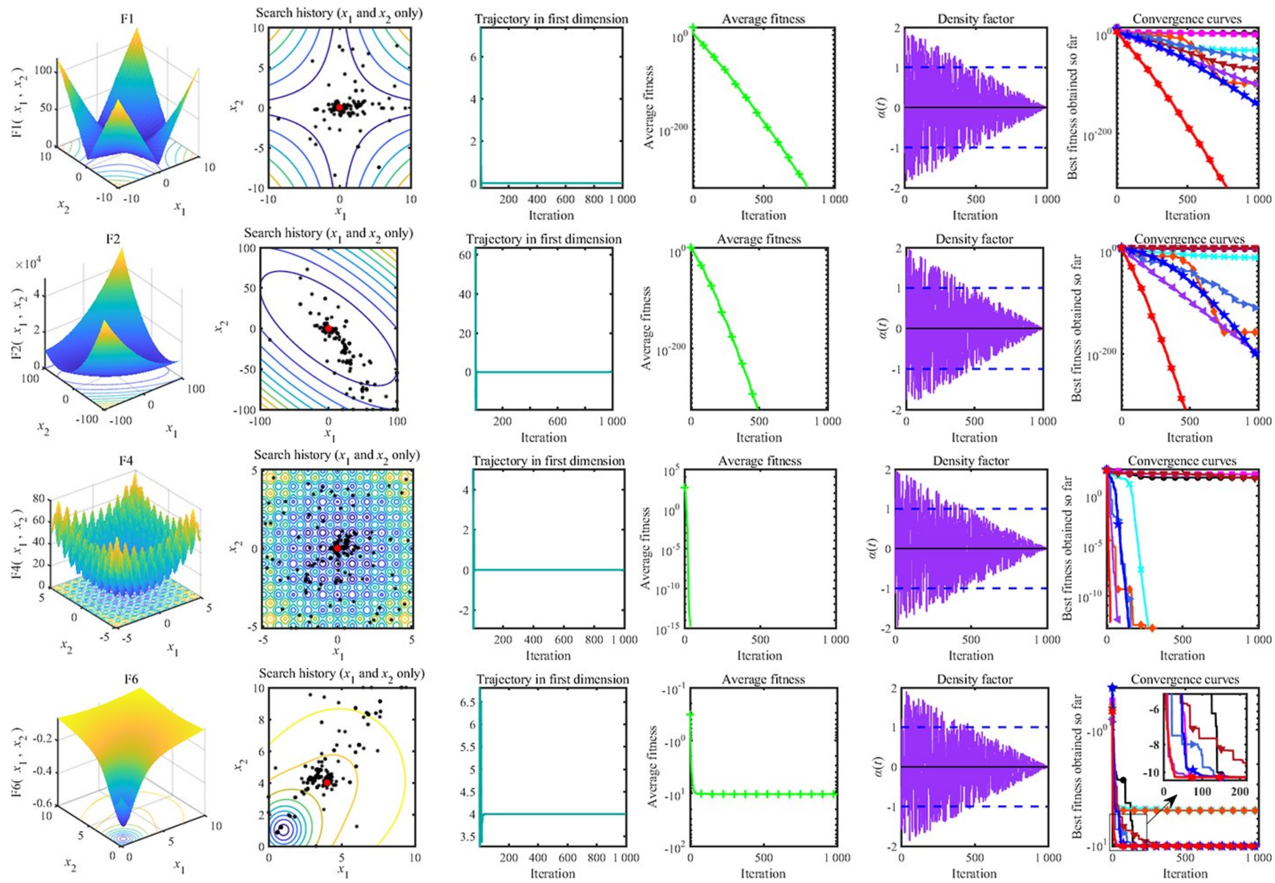
The points with asterisks in the graph of the search history represent the historical positions of the search agent, and the large dots represent the final optimal positions. From the search history in Fig. 5, it can be seen that the MHBA achieves global search with the help of the good point set population initialization.

The values of the dashed lines in the figure are 1 and −1. From the curves in Fig. 5, it can be seen that $\alpha(t)$ conforms to the theoretical analysis in subsection 2.3 during the iterative process, realizing the balance between exploration and exploitation. From the convergence curves, MHBA obtains the optimal value earlier than other optimizers, while some algorithms fail to obtain the global solution due to local optimal stagnation.

3.3 Quantitative analysis

This section provides a full study of MHBA through quantitative analysis. The results of the quantitative analysis for all benchmark functions are given in Table 2. It can be seen that MHBA outperforms all the competing algorithms on F1, F2, and F4 – F6. The results for F3 are second only to HHO, but the results obtained are far superior to HBA. MHBA's results for F1, F2, and F6 are far superior to all the algorithms and the optimal *Mean* is obtained.

Since the period mutation and t-distribution perturbation strengthen the algorithm's ability to avoid local optima, the statistical results of MHBA tend to possess smaller *Mean* and *Std* than HBA, such as F1, F2, F5, and F6 in Table 2. In conclusion, the quantitative analysis reveals that MHBA largely improves the accuracy and stability of the HBA solution.



(1) (2) (3) (4) (5) (6)

Fig. 5 Qualitative analysis results of MHBA

Table 2 Quantitative analysis of all benchmark functions

Function	Index	PSO	GWO	MFO	HHO	GJO
F1	*Mean*	0.044	1.050×10^{-34}	32.666	4.194×10^{-95}	6.037×10^{-71}
	Std	0.023	2.267×10^{-34}	24.766	2.279×10^{-94}	1.165×10^{-70}
	Rank	8	7	9	3	5
F2	*Mean*	44.478	4.173×10^{-15}	18560.200	8.220×10^{-150}	39851.000
	Std	17.035	1.531×10^{-14}	15587.600	4.499×10^{-149}	15489.200
	Rank	7	6	8	4	9
F3	*Mean*	86.914	26.725	$2.678\times10^{+06}$	0.003	92.536
	Std	29.653	0.710	$1.463\times10^{+07}$	0.003	323.775
	Rank	7	5	9	1	8
F4	*Mean*	51.882	0.158	153.626	0.000	19.177
	Std	13.210	0.864	36.962	0.000	33.863
	Rank	8	6	9	1	7
F5	*Mean*	0.102	1.64×10^{-14}	14.998	8.882×10^{-16}	8.941×10^{-15}
	Std	0.186	3.296×10^{-15}	7.154	0.000	3.724×10^{-15}
	Rank	7	6	9	1	5
F6	*Mean*	−9.445	−10.050	−8.355	−5.259	−7.277
	Std	2.213	1.343	3.010	0.941	3.040
	Rank	6	4	7	9	8

Table 2 continued

Function	Index	CPO	BKA	HBA	MHBA
F1	*Mean*	5.260×10^{-46}	2.154×10^{-76}	2.238×10^{-146}	0.000
	Std	2.875×10^{-45}	1.179×10^{-75}	9.916×10^{-146}	0.000
	Rank	6	4	2	1
F2	*Mean*	1.369×10^{-85}	2.482×10^{-156}	7.198×10^{-202}	0.000
	Std	7.500×10^{-85}	1.359×10^{-155}	0.000	0.000
	Rank	5	3	2	1
F3	*Mean*	23.469	27.086	21.830	0.017
	Std	0.492	1.188	0.509	0.612
	Rank	4	6	3	2
F4	*Mean*	0.000	0.000	0.000	0.000
	Std	0.000	0.000	0.000	0.000
	Rank	1	1	1	1
F5	*Mean*	1.125×10^{-15}	8.882×10^{-16}	0.666	8.882×10^{-16}
	Std	9.014×10^{-16}	0.000	3.645	0.000
	Rank	4	1	8	1
F6	*Mean*	−10.4029	−10.180	−9.895	−10.403
	Std	7.594×10^{-15}	1.219	1.949	8.079×10^{-16}
	Rank	2	3	5	1

4 Robot path planning based on MHBA

In this section, MHBA is applied to a 2D grid map to demonstrate its practicality.

4.1 Environmental modeling

The goal of the robot path planning problem is to find the shortest distance for a robot to travel from the starting point to the target point. In the map, the obstacle raster is assigned a value of 0, and the passable raster is assigned a value of 1. The dots in Fig. 6 indicate robot positions, the arrows indicate moveable paths, and the distance between neighboring grids may be 1 or $\sqrt{2}$.

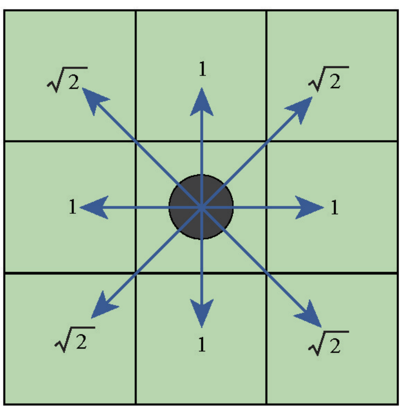


Fig. 6 Robot's feasible region

Assuming that the robot reaches the end point after M steps from the starting point, the objective function is the obstacle-free path length L of the mobile robot, as shown in Eq. (16).

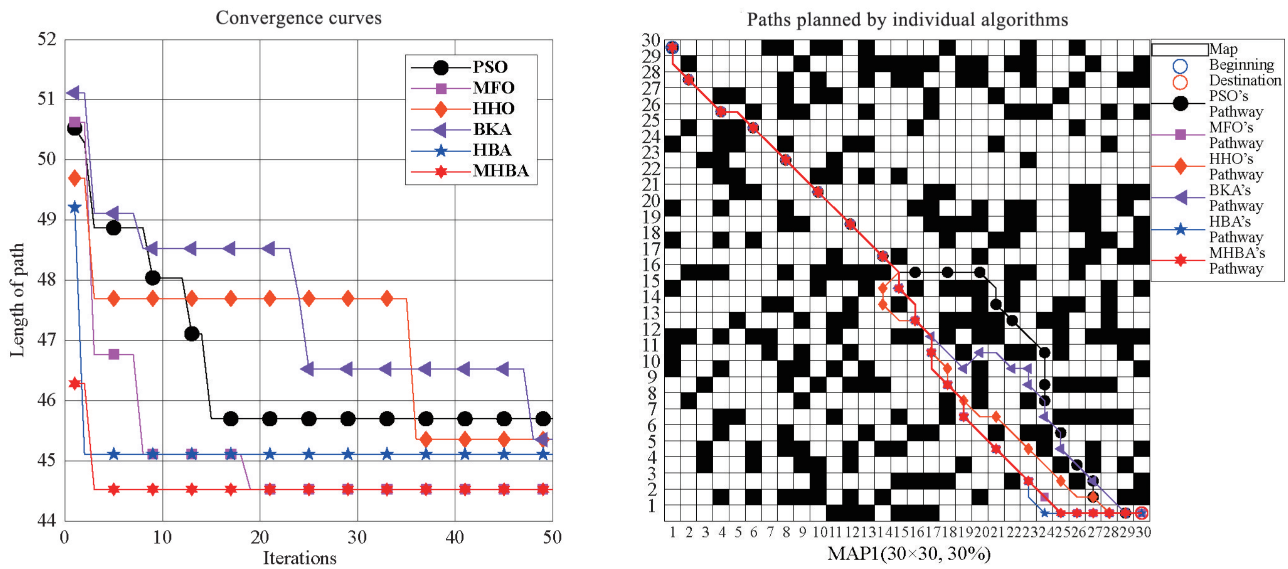
$$\min L = \sum_{m=1}^{M} d(m) \tag{16}$$

In Eq. (16), $d(m)$ is the Euclidean distance. Let the positions of the robot before and after moving be $P_m(x,y)$ and $P_{m+1}(x,y)$, then $d(m) = |P_{m+1}(x,y) - P_m(x,y)|$.

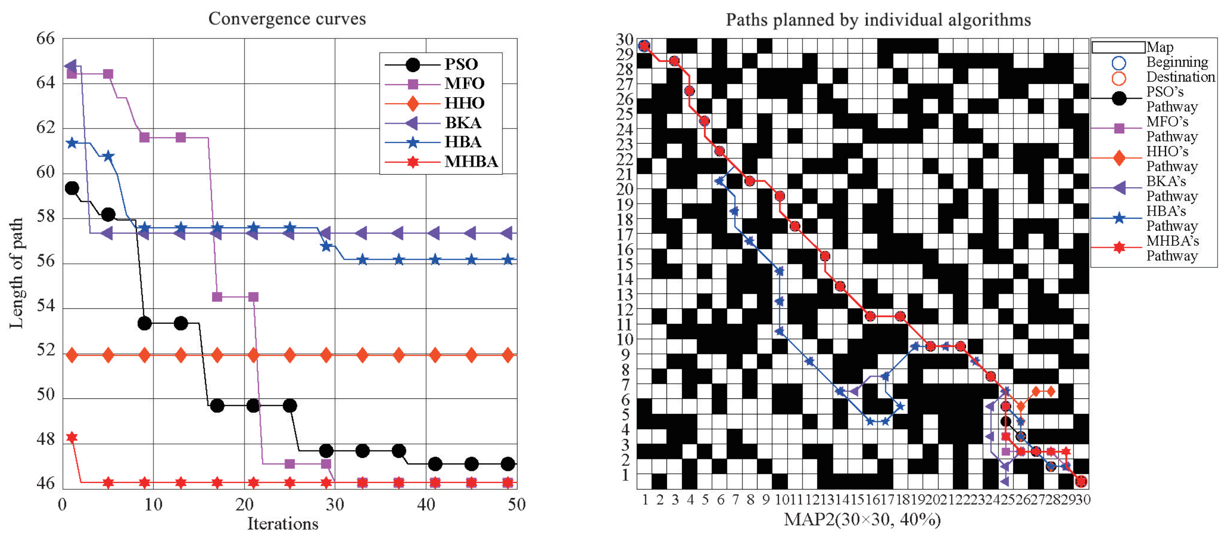
4.2 Simulation experiment

To verify the performance of MHBA in path planning, MHBA is compared with PSO, MFO, HHO, BKA, and HBA. The initial population of the 6 algorithms is 30 and the maximum number of iterations is 50. Obstacles are randomly generated for each map, starting in the upper left corner and ending in the lower right corner. In Fig. 7, the parameter settings for each map are MAP1: 30 × 30, 30%; MAP2: 30 × 30, 40%; MAP3: 40 × 40, 30%; and MAP4: 40 × 40, 45%. Percentage indicates the proportion of obstacles to the total grid.

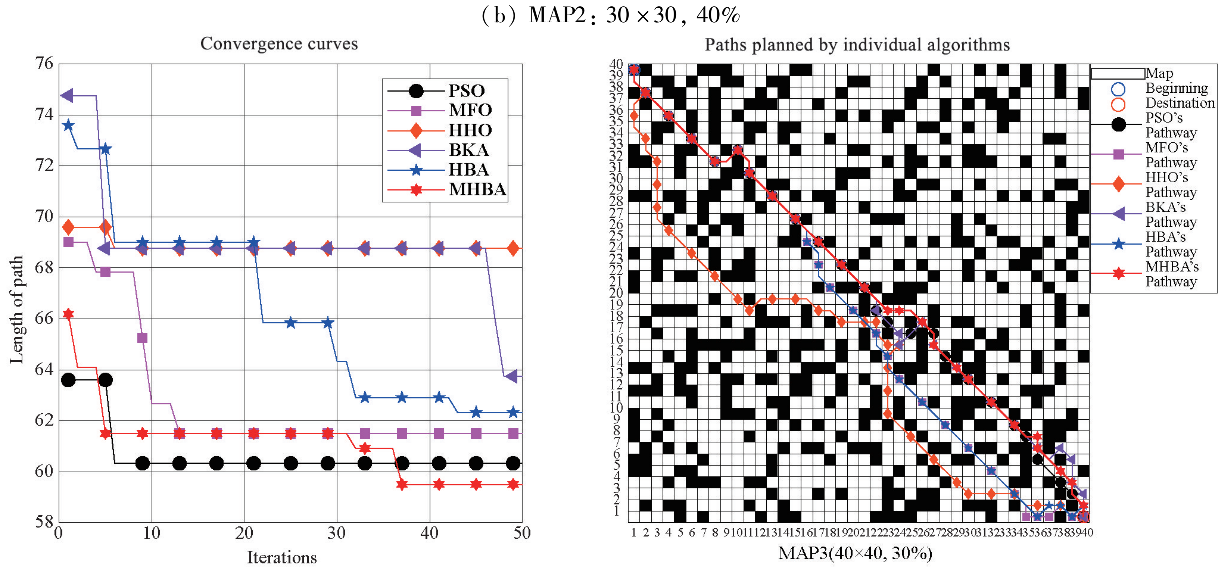
Fig. 7 shows the convergence curves of the algorithms and the optimal paths obtained by each algorithm based on the convergence curves. The metrics *Mean*, *Std*, and *Rank* given in Table 3 are calculated after 30 independent runs and Curve's data corresponds to the convergence curves and paths in Fig. 7.



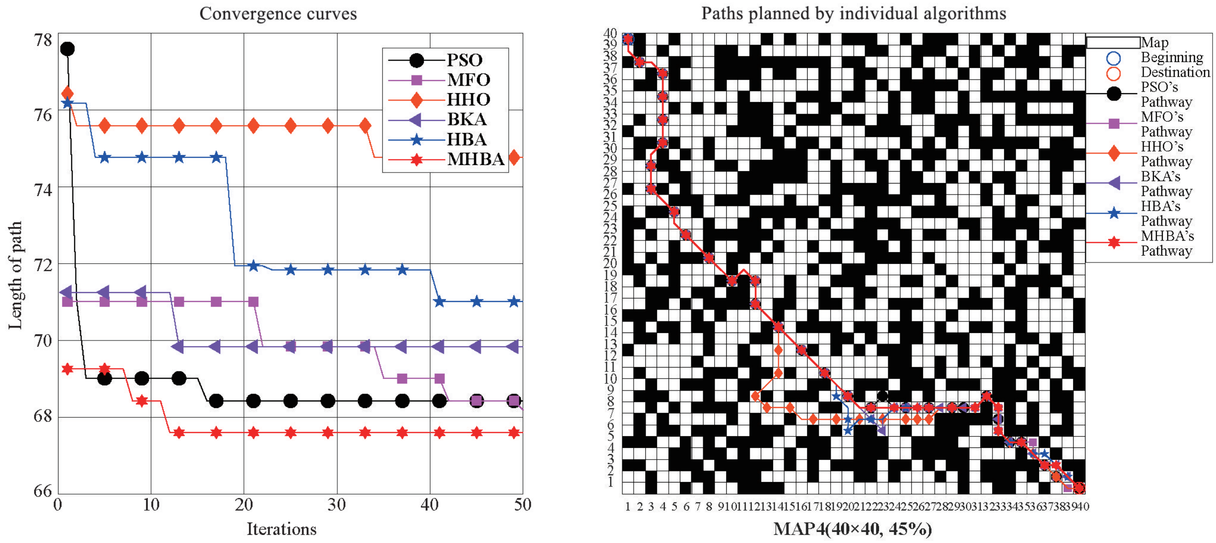
(a) MAP1: 30 × 30, 30%



(b) MAP2: 30 × 30, 40%



(c) MAP3: 40 × 40, 30%



(d) MAP4: 40 × 40, 45%

Fig. 7 Convergence curves and paths

As shown in Fig. 7, MHBA can recognize shorter paths in 4 different environments and outperforms the other 5 algorithms. Except for MAP3, MHBA enables fast convergence and outperforms all competitors. From MAP3, it is clear that MHBA has a strong ability to avoid local optima. From Table 3 it can be seen that although the *Mean* indicators are not all the smallest, the *Std* is the smallest, indicating that the results of MHBA are more stable.

MHBA tends to obtain optimal initial solutions due to the good point set population initialization, as shown by the convergence curves of MAP1, MAP2, and MAP4. Since periodic mutation and t-distribution perturbation strengthen the search capability of the algorithm, MHBA can fully utilize the search space and find optimal solutions, such as the convergence curve of MAP3. The balancing of exploration and exploitation in MHBA is achieved through the factor $\alpha(t)$, leading to optimal results in MHBA. Therefore, MHBA is more competitive than other algorithms in path planning.

Table 3 Statistical results of route planning

	Index	PSO	MFO	HHO	BKA	HBA	MHBA
MAP1	*Mean*	45.394	45.347	46.389	46.209	45.843	44.683
	Std	0.713	0.433	1.420	1.381	1.245	0.364
	Rank	3	2	6	5	4	1
	Curve	45.699	44.527	45.355	45.355	45.113	44.527
MAP2	*Mean*	49.522	46.697	55.024	51.509	47.591	46.622
	Std	1.939	1.848	3.250	4.470	1.958	0.756
	Rank	4	2	6	5	3	1
	Curve	47.113	46.284	51.941	57.355	56.184	46.284
MAP3	*Mean*	62.058	61.527	65.247	63.604	62.679	61.260
	Std	1.262	1.878	2.010	1.153	2.353	0.958
	Rank	3	2	6	5	4	1
	Curve	60.326	61.498	68.770	63.740	62.326	59.498
MAP4	*Mean*	68.401	68.037	71.695	68.635	70.187	68.108
	Std	0.510	1.344	2.391	1.280	5.305	0.456
	Rank	3	1	6	4	5	2
	Curve	68.426	68.184	74.770	69.841	71.012	67.598

5 Conclusion

To address the shortcomings of HBA, such as the initial population distribution not being ideal, poor ability to avoid local optima, and slow convergence speed, the MHBA proposed in this paper introduces a good point set population initialization and a linear decreasing factor based on the inclusion of period mutations and t-distribution perturbations. To test the performance of MHBA, 7 SIs besides HBA are used for comparison on 6 benchmark functions including PSO, GWO, MFO, HHO, GJO, CPO, and BKA. By performing both qualitative and quantitative analysis, it is learned that MHBA can determine the global optimum of the test functions more quickly than competitors. Finally, MHBA is applied to robot path planning. By comparing PSO, MFO, HHO, BKA, and HBA in the 4 scenarios, it is known that MHBA tends to identify the optimal path faster and also has a strong ability to avoid local optima. In short, the MHBA largely solves the shortcomings of the HBA and can face more challenging problems. In the future, multi-objective and binary versions of MHBA could be investigated, allowing it to face a wider range of problems. In addition, MHBA can be extended to other domains to solve more practical problems, such as neural networks, workshop scheduling, fault detection, and so on.

References

[1] CHAO Y, XU W, LIU W H, et al. Multi-threshold image segmentation method of QFN chip based on improved grey wolf optimization[J]. Optics and Precision Engineering, 2024, 32(6):930-944. (In Chinese)

[2] ZHAO J Y, LI J G, XUE Q S. Modeling and simulation of stereo garage handler path optimization based on improved tabu search algorithm[J]. Science Technology and Engineering, 2023, 23(35):15279-15285. (In Chinese)

[3] LEE E H, LEE H M, YOO D G, et al. Application of a meta-heuristic optimization algorithm motivated by a vision correction procedure for civil engineering problems[J]. KSCE Journal of Civil Engineering, 2018, 22(7): 2623-2636.

[4] EZUGWU A E, SHUKLA A K, NATH R, et al. Metaheuristics: a comprehensive overview and classification along with bibliometric analysis[J]. Artificial Intelligence Review, 2021, 54(6): 4237-4316.

[5] KHALID O W, ISA N A M, SAKIM H A M. Emperor penguin optimizer: a comprehensive review based on state-of-the-art meta-heuristic algorithms[J]. Alexandria Engineering Journal, 2023, 63: 487-526.

[6] LYNN N, SUGANTHAN P N. Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation[J]. Swarm and Evolutionary Computation, 2015, 24: 11-24.

[7] KENNEDY J, EBERHART R. Particle swarm optimization[C]// Proceedings of the International Conference on Neural Networks. Perth, Australia: Springer, 1995: 1942-1948.

[8] CHRISTIAN B. Ant colony optimization: introduction and recent trends[J]. Physics of Life Reviews, 2005, 2(4): 353-373.

[9] YANG X S, SUASH D. Cuckoo search via levy flights[C]// 2009 World Congress on Nature and Biologically Inspired Computing. Coimbatore, India: NaBIC, 2009, 210-214.

[10] MIRJALILI S, MIRJALILI S M, ANDREW L. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.

[11] MIRJALILI S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm[J]. Knowledge-Based Systems, 2015, 89: 228-249.

[12] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris hawks optimization: algorithm and applications[J]. Future Generation Computer Systems, 2019, 97: 849-872.

[13] CHOPRA N, ANSARI M M. Golden jackal optimization: a novel nature-inspired optimizer for engineering applications[J]. Expert Systems with Applications, 2022, 198: 116924.

[14] ABDEL-BASSET M, MOHAMED R, ABOUHAWWASH M. Crested porcupine optimizer: a new nature-inspired metaheuristic[J]. Knowledge-Based Systems, 2024, 284: 111257.

[15] WANG J, WANG W C, HU X X, et al. Black-winged kite algorithm: a nature-inspired meta-heuristic for solving benchmark functions and engineering problems[J]. Artificial Intelligence Review, 2024, 57(4): 98.

[16] HASHIM F A, HOUSSEIN E H, HUSSAIN K, et al. Honey badger algorithm: new metaheuristic algorithm for solving optimization problems[J]. Mathematics and Computers in Simulation, 2022, 192: 84-110.

[17] HUA L G, WANG Y. Applications of number theory to approximate analysis[M]. Beijing: Science Press, 1978. (In Chinese)

[18] WANG L, CAO Q, ZHANG Z, et al. Artificial rabbits optimization: a new bio-inspired meta-heuristic algorithm for solving engineering optimization problems[J]. Engineering Applications of Artificial Intelligence, 2022, 114: 105082.

[19] ZHU F, LI G S, TANG H, et al. Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems[J]. Expert Systems with Applications, 2024, 236: 121219.

WU Jin, born in 1975. She received her B. S. degree from Xi'an Jiaotong University in 1998, and she also received her M. S. degree from Xi'an Jiaotong University in 2001. Her research focuses on key techniques for signal and information processing.