Improved Bat Algorithm Based on Fast Diving Strategy

Yanxiang Geng[1,2], Liyi Zhang[1,2], Yong Zhang[2], Zhixing Li[2], Jiahui Li[2]*

[1] School of Electric Information Engineering, Tianjin University, China
[2] School of Information Engineering, Tianjin University of Commerce, China
gengyanxiang@tjcu.edu.cn, zhangliyi@tjcu.edu.cn, zhangyong@tjcu.edu.cn, 919317812@qq.com, 2303697714@qq.com

Abstract

Bat algorithm has good global search ability, but it has some problems, such as slow convergence speed in local search stage, low convergence accuracy, easy to fall into local optimization and can not escape. In view of the above defects, inspired by Harris Hawks's strategy of catching rabbits, this paper introduces the surrounding mechanism of prey, which can quickly approach the food and judge its quality, so as to achieve the purpose of rapid convergence and improve the convergence accuracy. The experiment shows that the improved algorithm of the fast diving strategy is tested by using the test function, and compared with the basic bat algorithm, backtracking bat algorithm and HABC. The improved bat algorithm of the fast diving strategy has better optimization accuracy, faster convergence speed, simple algorithm and higher success rate.

Keywords: Bat algorithm, Fast diving strategy, Encirclement mechanism, Escape, CEC06-2019

1 Introduction

With the development of science and technology, the improvement of computer computing ability has brought new opportunities to the development of intelligent bionic algorithms. Many complex multi-mode functions and other optimization problems become easier. In recent years, intelligent bionic algorithm has achieved very good results in solving the problem of optimization strategy. Many scholars are also constantly studying intelligent bionic algorithms, such as Ant Colony Algorithm [1], Fish Swarm Algorithm [2], Bee Colony Algorithm [3], Particle Swarm Optimization Algorithm [4], Wolf Swarm Algorithm [5], Cuckoo Algorithm [6], Harris Hawks Algorithm [7], etc. It also includes the bat algorithm proposed by Professor Yang in 2010 based on the ultrasonic obstacle avoidance and predator-prey characteristics of bats [8]. Over the years, many scholars have also conducted a series of research on bat algorithm, which has been applied to multi-objective optimization problems [9], industrial inspection problems [10], constraint optimization problems [11], path optimization problems [12] and image processing [13], and achieved good results.

Bat algorithm is simple and has few parameters, but it is easy to fall into local optimization. At the same time, the convergence speed and accuracy need to be improved. Therefore, since the bat algorithm was proposed, there have been many improved methods in academic circles, which have achieved good results and solved many practical problems.

Similarly, the hunting behavior of Harris hawks is similar to that of bats, and both need to find prey and keep close to the prey. However, the hunting strategy of Harris hawks for prey is much more advanced than that of bats, and the consideration of prey energy is added. Therefore, inspired by Harris Hawks Algorithm, the update mode of ultrasonic frequency and loudness in bat algorithm is adjusted, and the fast diving mechanism is introduced. When the prey energy is large, it means that the current solution value is large and the search range is enlarged; When the prey energy is small, it means that the current solution value is small and the search range is narrow. This improvement can improve the convergence effect of local search and the overall convergence speed and accuracy of the Bat algorithm. Compared with other algorithms, it shows good performance.

This paper is divided into the following parts: the first part is the background introduction and article arrangement; The second part is literature review; In the third part, a bat algorithm with fast diving strategy (FDBAT) is proposed; The fourth part is a comparative experiment; The fifth part is the summary and outlook.

2 Literature Review of Bat Algorithm

In recent years, bat algorithm has attracted the attention of many scholars. It has been successfully applied to various fields by improving many schemes. Lijue Liu [14] et al. proposed a discrete bat algorithm, the Floyd–Warshall algorithm is first used to transform an incomplete connected graph into a complete graph whose vertex set consists of a start point and necessary points (use Floyd – warhall algorithm to construct complete graph). At the same time, the neighborhood operator is used to improve the premature phenomenon of bat algorithm, and it has achieved good application in the shortest path planning problem. Xiaofeng Yue [15] et al. proposed a hybrid bat algorithm combining genetic crossover operation and intelligent inertia weight, which enhanced the local search ability of the algorithm and achieved good results in multi-level threshold image segmentation. Minrong Chen [16] et al. proposed an improved bat algorithm with extremum optimization to improve the local search ability. In addition, the Boltzmann selection and monitoring mechanism can well balance the

*Corresponding Author: Yong Zhang; E-mail: zhangyong@tjcu.edu.cn
DOI: 10.53106/160792642023112406008

exploration ability and development ability. Yassine Saji [17] et al. added a neutral crossover operator to the bat algorithm, which effectively improved the diversity and convergence of the algorithm, and used l'evy flight to avoid falling into local optimization. Helong Yu [18] et al. used a threshold to control the step of chaotic mapping, and used the speed inertia weight to synchronize the speed of the agent, which was successfully applied in the field of I-beam design, welding beam design and pressure vessel design. It can be seen from the above documents that Bat algorithm has simple structure, few parameters and low requirements for operating environment. However, it is easy to fall into local optimum, and the convergence speed and accuracy need to be improved. In recent years, there is still much room for improvement in Bat algorithm.

3 Overview of Bat Algorithm

3.1 Basic Bat Algorithm

With the rise of intelligent algorithms, people have found that many biological habits can be transformed into mathematical models, providing good methods and means to solve practical problems. In 2010, Professor Yang proposed a meta heuristic optimization algorithm based on the ultrasonic obstacle avoidance function of bats.

Bats fly in the air at a speed V_i and a position of X_i. During the flight, ultrasonic waves of frequency f_i, wavelength λ and loudness A_o will be sent out to search. Bats judge the distance between the target and themselves based on the returned data. According to the different distance, the pulse wavelength and frequency of the emitted ultrasonic wave are adjusted, and the frequency of the transmitting pulse is gradually adjusted in the process of approaching the target $\gamma \in [0,1]$.

The mathematical model is as follows:

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta. \tag{1}$$

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X_*) f_i. \tag{2}$$

$$X_i^t = X_i^{t-1} + V_i^t. \tag{3}$$

In the formula, the frequency f_i range is $[f_{\min}, f_{\max}]$, β is a random vectort between [0,1], X_i^t and V_i^t represents the position and velocity of the ith bat at time t, and X_* represents the current bat's most position, namely the current optimal solution.

In the process of local search, the update formula of position is:

$$X_{new}(i) = X_{old} + \varepsilon A^t. \tag{4}$$

Where, ε is a random number between [0,1], A^t representing the average loudness of all bats at that moment.

When the bat detects a target, the loudness of the firing puls A_i^t decreases gradually, and then the firing rater r_i^t is increased.

$$A_i^{t+1} = \alpha A_i^t. \tag{5}$$

$$r_i^{t+1} = r_i^0 (1 - e^{-\gamma t}). \tag{6}$$

Where, $\alpha \in [0,1]$, $\gamma > 0$.

3.2 Bat Algorithm Based on Progressive Fast Diving Strategy

Harris Hawkes' hunting strategy can analyze the actual effect of the prey and use different optimization functions to improve the convergence speed of the algorithm, which is mainly reflected in the gradual rapid dive stage. The specific implementation is as follows:

Firstly, the parameter escape energy E is introduced:

$$E = 2E_0 (1 - \frac{t}{T}). \tag{7}$$

Where, E_0 is the initial energy, and its value changes randomly within (-1,1), t is the current number of cycles, and T is the total number of cycles. When $|E| \geq 1$ the food is not easy to be found, bats adopt traditional domain search; When $|E| < 1$ is used, food is easy to be found, the algorithm introduces the progressive fast diving strategy of Harris Hawks Algorithm to conduct a presearch before the traditional domain search, so as to speed up the convergence speed and improve the accuracy of the algorithm. Harris Hawk optimization algorithm determines which strategy to adopt through escape energy E and escape probability r.

(1) When $|E| \geq 0.5$, $r \geq 0.5$ is in the soft encirclement stage, the accurate position of food is not easy to be found by bats. At this time, bats will use two strategies to soft encircle more intelligently like Harris Hawks.

Strategy 1:

$$Y(i) = X_* - E(JX_* - X_{new}(i)). \tag{8}$$

Where, $J = 2(1-r)$ is a random number of (-2, 0).

When strategy 1 is invalid, execute strategy 2.

Strategy 2:

$$Z(i) = Y(i) + S \times LF(x). \tag{9}$$

Where, S is a random number and LF is an Levy flight function.

$$LF(x) = 0.01 \times \frac{\mu + \sigma}{|v|^{\frac{1}{\omega}}}. \tag{10}$$

$$\sigma = \left(\frac{\Gamma(1+\omega) \times \sin\left(\frac{\pi\omega}{2}\right)}{\Gamma\left(\frac{1+\omega}{2}\right) \times \omega \times 2\left(\frac{\omega-1}{2}\right)} \right)^{\frac{1}{\omega}}. \tag{11}$$

Where, ω = 1.5, μ, v are random numbers evenly distributed between 0 and 1, and Γ is gamma functions. Therefore, the location update strategy of bats at this stage is implemented according to the following equation:

$$X(i)=\begin{cases} Y(i) & if\ F(Y(i))<F(X(i)) \\ Z(i) & else\ if\ F(Z(i))<F(X(i)) \\ X(i) & otherwise \end{cases}. \quad \mathbf{(12)}$$

Where, $F(*)$ is fitness function.

(2) When $|E|$ < 0.5, r < 0.5 is in the hard encirclement stage, the food does not have enough energy to cover itself, but it may not be found by the bat. Therefore, the bat will narrow the distance from the food, form a hard encirclement, and adopt two strategies for hunting.

Strategy 3:

$$Y(i)=X_*-E(JX_*-X_m(i)). \quad \mathbf{(13)}$$

Where, $X_m(i)$ is the average position of the current population.

When strategy 3 is invalid, strategy 4 is used.

Strategy 4:

The bat's position update strategy is implemented in this stage according to F9 and F12.

3.3 Based on the Strategy of Incremental Rapid Diving Bat Algorithm Process

Step1: Initialize the bat population, randomly distribute the initial positions of bats, set the maximum number of iterations, and set the initial bat speed, loudness, pulse frequency and emissivity.

Step 2: Randomly assign the initial position of bats, start the search, record the search results of each bat, find the current optimal solution, and store the current optimal solution.

Step 3: Start the iteration, update each bat, judge whether it is beyond the limit, replace with the critical value if it is yes, and proceed to the next step if not, update the energy E of the prey.

Step 4: If $|E|$ < 1, further judge the values of E and r, otherwise return to Step2.

Step 5: If $|E| \geq 0.5$, $r \geq 0.5$, execute strategy 1. If the conditions are not met, execute strategy 3.

Step 6: Judge whether the current solution is better than the current optimal solution. If yes, update the bat position and the current optimal solution. Otherwise, update the bat position and the current optimal solution after executing strategy 2.

Step 7: Add 1 to the number of iterations and return to Step3 until the maximum number of iterations is reached.

Step 8: End the algorithm and outputs the optimal result.

The algorithm flowchart is shown in Figure 1.

4 Simulation Comparison and Analysis

In order to prove the bat algorithm with fast diving strategy, 23standard test functions [19] were selected for Matlab simulation. Matlab2016 was used to carry out simulation test under Windows7 system. The test functions are shown in Table 1 to Table 3. Table 1 is the single-peak test function (F1-F7) in the benchmark function, Table 2 is the multi-dimensional multi-peak test function (F8-F13), and Table 3 is the fixed-dimension multi-peak test function (F14-F23). 30 is the dimension of F1-F13, the dimension of F14, F16, F17, and F18 is 2, the dimension of F15, F21, F22, and F23 is 4, the dimension of F19 is 3, and the dimension of F20 is 6.The data are compared with the original bat algorithm and the improved RNABAT and COBAT, and also with the simulation results of the new intelligent optimization algorithm proposed in recent years, such as AOA [20], HHO [21], GWO [22], BOA [23], SOA [24], and WOA [25].

4.1 Comparison with Improved Bat Algorithm

FDBAT, RNABAT [26] and COBAT [27] are all improved methods aiming at different shortcomings of bat algorithm. In the simulation process, the same number of iterations (t_{max}=500) and population (N=100) are selected. To avoid contingency, the simulation results in Table 4 are repeated 30 times and recorded after averaging.

The convergence performance of FDBAT is better than that of BAT, RNABAT and COBAT. In the test results of unimodal and multidimensional multimodal tests, the optimization effect is more obvious, and the optimization effect in the fixed-dimensional multimodal test function is also better than other improved bat algorithms. Especially in the optimization process of F_1, F_2, F_3, F_4, F_8, F_9, F_{10}, F_{11}, F_{12}, F_{16}, F_{17}, F_{18}, FDBAT can quickly and accurately find the optimal solution of the function. Compared with other improved algorithms, the accuracy of optimization has been greatly improved. In many simulation tests, the improved bat algorithm can find the optimal solution of the test function, but the success rate of FDBAT is the highest. These algorithms have little difference in convergence speed and can quickly form convergence effect, but there are some differences in convergence accuracy.

4.2 Comparison with Other Intelligent Optimization Algorithms

The population was 100, the max iterations were 500, and 30 experiments were independently repeated to record data. Due to limited space, only the first four convergence curve sand test data of the first two types of test function are given.

Figure 2 shows that the convergence speed of FDBAT algorithm is faster than other algorithms. Especially for F_2 and F_3, the convergence speed of FDBAT algorithm in the later stage is significantly accelerated. However, in F_4 and F_9 functions, the convergence speed of FDBAT algorithm is slightly lower.

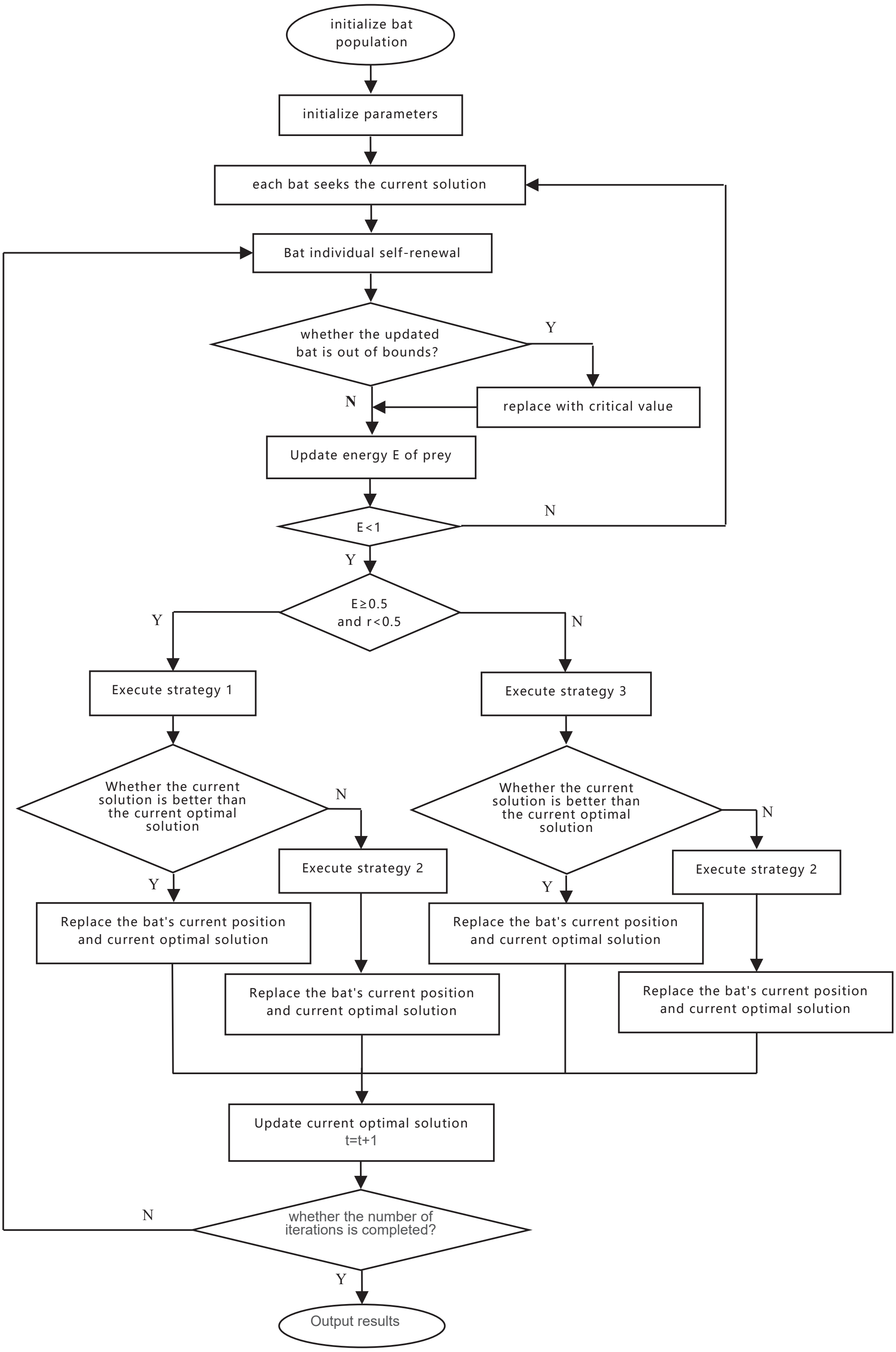


Figure 1. Algorithm flowchart

Table 1. Unimodal benchmark functions

Functions	Function name	Range	F_{min}
$F_1(x)=\sum_{i=1}^{\dim} x_i^2$	Sphere	[-100.100]	0
$F_2(x)=\sum_{i=1}^{\dim}\lvert x_i\rvert+\prod_{i=1}^{\dim}\lvert x_i\rvert$	Schwefel 222	[-10,10]	0
$F_3(x)=\sum_{i=1}^{\dim}\left(\sum_{j=1}^{i} x_j\right)^2$	Schwefel 1.2	[-100,100]	0
$F_4(x)=\max_i\left\{\lvert x_i\rvert, 1\le i\le \dim\right\}$	Schwefel 2.21	[-100,100]	0
$F_5(x)=\sum_{i=1}^{\dim-1}\left[100\left(x_{i+1}-x_i^2\right)+\left(x_i-1\right)^2\right]$	Rosenbrock	[-30,30]	0
$F_6(x)=\sum_{i=1}^{\dim}\left(\lvert x_i+0.5\rvert\right)^2$	Step	[-100,100]	0
$F_7(x)=\sum_{i=1}^{\dim} i x_i^4+\text{random}[0,1]$	Quartic	[-1.28,1.28]	0

Table 2. Multidimensional and multimodal benchmark functions

Functions	Function name	Range	F_{min}
$F_8(x)=\sum_{i=1}^{\dim} -x_i\sin(\sqrt{\lvert xi\rvert})$	Schwefel 2.26	[-500,500]	-12569.5
$F_9(x)=\sum_{i=1}^{\dim}\left[x_i^2-10\cos\left(2\pi x_i\right)+10\right]$	Rastrigin	[-5.12,5.12]	0
$F_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{\dim}d\sum_{i=1}^{\dim}x_i^2}\right)-$ $\exp\left(\frac{1}{\dim}\sum_{i=1}^{\dim}\cos\left(2\pi x_i\right)\right)+20+e$	Ackley	[-32,32]	0
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{\dim}x_i^2-\prod_{i=1}^{\dim}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	Griewank	[-600,600]	0
$F_{12}(x)=\frac{\pi}{\dim}10\sin^2\left(\pi y_1\right)+\sum_{i=1}^{\dim-1}\left(y_i-1\right)^2\left[1+10\sin^2\left(\pi y_{i+1}\right)\right]$ $+\left(y_{\dim}-1\right)^2+\sum_{i=1}^{\dim}U\,\text{fun}\left(x_i,10,100,4\right)$	Penalized1	[-50,50]	0
$F_{13}(x)=0.1\begin{Bmatrix}\sin^2\left(3\pi x_1\right)+\sum_{i=1}^{\dim}\left(x_i-1\right)^2\left[1+\sin^2\left(3\pi x_{i+1}+1\right)\right]+\\ \left(x_{\dim}-1\right)^2\left[1+\sin^2\left(2\pi x_{dim}\right)\right]\end{Bmatrix}$ $+\sum_{i=1}^{\dim}U\,\text{fun}\left(x_i,5,100,4\right)$	Penalized2	[-50,50]	0

Table 3. Fixed-dimensional multimodal benchmark functions

Functions	Function name	Range	F_{min}
$F_{14}(x)=\left(\frac{1}{500}+\sum_{j=1}^{25}\frac{1}{j+\Sigma_{i=1}^{2}\left(x_i-a_{ij}\right)^6}\right)^{-1}$	Foxholes	[-65.5360, 65.5360]	1
$F_{15}(x)=\sum_{k=1}^{11}\left[a_k-\frac{x_1\left(b_k^2+b_kx_2\right)}{b_k^2+b_kx_3+x_4}\right]^2$	Kowalik	[-5,5]	0.00030
$F_{16}(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2{}^2+4x_2^4$	Six Hump Camel Bcak	[-5,5]	-1.0316
$F_{17}(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	Branin	[-5,0 10,15]	0.398
$F_{18}(x)=\left[1+\left(x_1+x_2+1\right)^2\left(19-14x_1+3x_1{}^2-14x_2+6x_1x_2+3x_2^2\right)\right]\times$ $[30+\left(2x_1-3x_2\right)^2\times\left(18-32x_1+12x_1{}^2+48x_2-36x_1x_2+27x_2{}^2\right)\Big]$	GoldStein Price	[-5,5]	3
$F_{19}(x)=-\sum_{i=1}^{4}cH_i\exp\left(-\sum_{j=1}^{3}aH_{ij}\left(x_j-pH_{ij}\right)^2\right)$	Hartman3	[0,1]	-3.86
$F_{20}(x)=-\sum_{i=1}^{4}cH_i\exp\left(-\sum_{j=1}^{6}aH_{ij}\left(x_j-pH_{ij}\right)^2\right)$	Hartman6	[0,1]	-10.1532
$F_{21}(x)=-\sum_{i=1}^{5}\left[\left(x-a_i\right)\left(x-a_i\right)^T+c_i\right]^{-1}$	Shekel5	[0,10]	-10.1532
$F_{22}(x)=-\sum_{i=1}^{7}\left[\left(x-a_i\right)\left(x-a_i\right)^T+c_i\right]^{-1}$	Shekel7	[0,10]	-10.4028
$F_{23}(x)=-\sum_{i=1}^{10}\left[\left(x-a_i\right)\left(x-a_i\right)^T+c_i\right]^{-1}$	Shekel10	[0,10]	-10.5363

Table 4. Comparing the optimization accuracy of different algorithms

Function	Algorithm	Optimal solution	Theworst solution	Average solution	Standard variance	Mean algebra	Success rate%
F_1	BAT	2.1023e+4	3.6691e+4	2.3652e+4	3.8861e+3	5	0
	RNABAT	0.0000e+0	3.6581e+2	1.1126e+2	1.3117e+1	15	60
	COBAT	0.0000e+0	1.3547e+4	3.8351e+3	1.7137e+2	10	80
	FDBAT	**0.0000e+0**	**0.0000e+0**	**0.0000e+0**	**0.0000e+0**	**8**	**100**
F_2	BAT	1.7809e+4	6.45520e+6	3.5841e+4	2.6984e+2	7	0
	RNABAT	3.0000e+0	3.1682e+0	3.0045e+0	0.1586e+0	15	0
	COBAT	0.0000e+0	3.5894e-52	2.8815e-44	2.3584e-28	12	70
	FDBAT	**0.0000e+0**	**8.334e-123**	**2.8415e-82**	**1.5861e-12**	**5**	**90**
F_3	BAT	4.7850e+4	8.7073e+4	5.4217e+4	3.2541e+3	70	0
	RNABAT	7.1189e+2	1.3251e+4	7.2514e+3	6.2541e+3	20	0
	COBAT	0.0000e+0	5.2369e+4	1.5267e+1	2.5864e+2	5	70
	FDBAT	**0.0000e+0**	**1.7926e+4**	1.3647e+0	1.2546e+1	10	80

F_4	BAT	6.5756e+1	7.1374e+1	6.8145e+1	3.6859e+0	9	0
	RNABAT	2.0000e+0	3.0000e+0	2.4000e+0	1.3641e+0	60	0
	COBAT	0.0000e+0	1.5321e+2	0.2513e+2	1.5198e+0	46	30
	FDBAT	**0.0000e+0**	**5.0190e+1**	**1.2501e+1**	**3.1211e+0**	**23**	**90**
F_5	BAT	7.2707e+7	8.1103e+7	7.6218e+7	5.2783e+7	6	0
	RNABAT	3.1700e+2	7.5301e+2	4.8821e+2	3.6562e+1	28	0
	COBAT	2.9012e+1	2.9835e+2	2.5893e+2	5.1238e+2	7	0
	FDBAT	**2.8739e-5**	**2.8802e-4**	**2.8783e-5**	**4.5801e-4**	**12**	**0**
F_6	BAT	2.9736e+4	3.2295e+4	3.0158e+4	1.2594e+2	50	0
	RNABAT	1.3500e+1	1.5500e+1	1.3981e+1	3.6851e+1	21	0
	COBAT	7.5000e+0	1.5632e+2	6.3252e+1	4.2140e+1	35	0
	FDBAT	**1.6385e+0**	**2.3939e+0**	**1.7852e+0**	**1.0210e+0**	**17**	**0**
F_7	RNABAT	3.6772e+1	1.1685e+2	6.2513e+1	5.2146e+1	8	0
	BAT	3.7637e+1	4.9476e+1	3.9211e+1	4.1250e+0	12	0
	COBAT	3.8668e-5	1.3322e-4	7.1251e-5	2.2145e-6	5	0
	FDBAT	**1.0112-5**	**4.4862e-5**	**2.8546e-5**	**3.5480e-3**	**16**	**0**
F_8	BAT	-3.2148e+3	-2.2377e+3	-3.0289e+3	5.1981e+4	8	0
	RNABAT	-3.2079e+3	-2.7843e+3	-3.1584e+3	4.7512e+4	15	0
	COBAT	-2.5585e+3	-1.9700e+3	-2.3581e+3	1.2896e+4	10	0
	FDBAT	**-1.2570e+4**	**-1.2502e+4**	-1.2569e+4	0.2457e+0	26	80
F_9	BAT	3.7155e+2	3.9206e+2	3.7952e+2	6.3511e+3	18	0
	RNABAT	3.0000e+0	4.0000e+0	3.5150e+0	5.2681e+2	15	0
	COBAT	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	230	100
	FDBAT	**0.0000e+0**	**0.0000e+0**	**0.0000e+0**	**0.0000e+0**	**21**	**100**
F_{10}	BAT	1.8945e+1	1.9395e+1	1.9022e+1	3.4112e+1	54	0
	RNABAT	1.0066e+0	1.2257e+0	1.0620e+0	2.6851e+1	38	0
	COBAT	8.8818e-16	8.8818e-16	8.8818e-16	0.0000e+0	236	0
	FDBAT	**0.0000e+0**	**0.0000e+0**	**0.0000e+0**	**0.0000e+0**	**13**	**100**
F_{11}	BAT	2.7792e+2	3.0688e+2	2.8879e+2	3.5821e+0	27	0
	RNABAT	0.9882e+0	0.9908e+0	0.9900e+0	8.1005e-2	11	0
	COBAT	0.0000e+0	0.0000e+0	0.0000e+0	0.0000e+0	210	100
	FDBAT	**0.0000e+0**	**0.0000e+0**	**0.0000e+0**	**0.0000e+0**	**12**	**100**
F_{12}	BAT	7.3373e+7	1.11008e+8	3.2589e+8	6.6355e+6	9	0
	RNABAT	2.3038e+0	3.0565e+0	2.6320e+0	4.5255e+2	11	0
	COBAT	0.0000e+0	1.669e+0	5.6980e-5	2.6950e-4	5	60
	FDBAT	**0.0000e+0**	**9.7154e-2**	**6.3352e-7**	**3.6533e-5**	**7**	**80**
F_{13}	BAT	6.1058e+7	1.3186e+8	7.2510e+7	1.5986e+5	14	0
	RNABAT	7.0151e-1	9.3564e-1	7.9651e-1	2.5112e-5	5	0
	COBAT	2.5894e+0	3.0000e+0	2.7312e+0	4.0281e-3	7	0
	FDBAT	**5.6244e-7**	**6.3909e-4**	5.8872e-6	5.2287e-5	9	0

F_{14}	BAT	6.9113e+0	1.1731e+1	9.2253e+0	1.5212e+1	30	0
	RNABAT	3.1998e+0	1.2670e+1	4.2480e+0	6.9302e+0	21	0
	COBAT	1.2670e+0	1.3652e+1	1.2892e+0	2.5801e+0	17	0
	FDBAT	**9.9800e-1**	**1.0763e+1**	**1.0032e+0**	**6.2158e-1**	**11**	**0**
F_{15}	BAT	7.7682e-3	8.0793e-2	3.9325e-2	1.1975e-4	63	0
	RNABAT	1.9140e-2	8.3763e-2	3.6960e-2	4.2198e-5	151	0
	COBAT	2.4593e-2	1.4804e-1	6.8812e-2	1.0352e-1	201	0
	FDBAT	**3.6176e-4**	**1.8355e-3**	**3.2861e-4**	**2.6657e-2**	**89**	**0**
F_{16}	BAT	-1.0209e+0	-8.9778e-1	-9.5091e-1	2.4514e+1	25	0
	RNABAT	-9.9983e-1	-1.2701e-4	-1.3386e+0	6.3571e+1	9	0
	COBAT	-1.0015e+0	0.0000e+0	-9.8551e-1	7.9834e-1	32	0
	FDBAT	**-1.0316e+0**	**-1.0316e+0**	**-1.0316e+0**	**0.0000e+0**	**17**	**100**
F_{17}	BAT	3.9800e-1	3.4821e-1	3.6522e-1	6.2580e-3	5	20
	RNABAT	3.9800e-1	3.5221e-1	3.7258e-1	5.2251e-1	33	50
	COBAT	3.9800e-1	3.6214e-1	3.9632e-1	2.6651e-4	27	70
	FDBAT	**3.9800e-1**	**3.9800e-1**	**3.9800e-1**	**0.0000e+0**	**14**	**100**
F_{18}	BAT	1.0231e+1	3.4815e+1	1.3659e+1	2.5841e-1	58	0
	RNABAT	1.1735e+1	1.1663e+2	1.6527e+1	4.3229e+0	18	0
	COBAT	7.1527e+0	1.1149e+2	5.6843e+1	6.2943e-1	210	0
	FDBAT	**3.0000e+0**	**3.0006e+0**	3.0002e+0	1.5644e+0	32	90
F_{19}	BAT	-3.8434e+0	-3.7563e+0	-3.7961e+0	2.9913e-1	18	0
	RNABAT	-3.7007e+0	-2.6261e+0	-3.1305e+0	5.3620e-3	51	0
	COBAT	-3.7627e+0	-2.7113e+0	-3.5622e+0	7.2105e+0	11	0
	FDBAT	**-3.8599e+0**	**-3.8627e+0**	**-3.8625e+0**	**6.71133e-4**	**79**	**0**
F_{20}	BAT	-3.7899e+0	-3.4559e+0	-3.6352e+0	2.5855e-4	5	0
	RNABAT	-3.0606e+0	-6.6219e-1	-3.9681e+0	6.5520e+0	18	0
	COBAT	-2.2746e+0	-8.3101e-1	-1.9863e+0	4.2507e-2	42	0
	FDBAT	**-3.1897e+0**	**-3.1824e+0**	**-3.1855e+0**	**5.6250e-5**	**20**	**0**
F_{21}	BAT	-2.4178e+0	-8.2102e-1	-1.3620e+0	5.6143e+1	36	0
	RNABAT	-1.6663e+0	-6.3959e-1	-8.6902e-1	6.5931e+0	11	0
	COBAT	-3.3111e+0	-3.8791e-1	-3.4672e+0	6.1185e-1	21	0
	FDBAT	**-1.0089e+1**	**-5.0414e+0**	**-1.0035e+1**	**4.5005e-5**	**6**	**0**
F_{22}	BAT	-2.4505e+0	-6.2733e-1	-2.1003e+0	3.6086e-3	22	0
	RNABAT	-1.8046e+0	-4.6181e-1	-1.3233e+0	1.9200e+0	9	0
	COBAT	-4.4140e+0	-1.9794e+0	-3.9603e+0	1.5966e-1	11	0
	FDBAT	**-1.0259e+1**	**-5.0279e+0**	**-1.0039e+1**	**2.0284e-4**	**32**	**0**
F_{23}	BAT	-3.7032e+0	-1.0620e+0	-3.2606e+0	9.9951e-2	5	0
	RNABAT	-5.1285e+0	-5.2159e-1	-4.8560e+0	9.6522e-1	11	0
	COBAT	-1.7575e+0	-8.4214e-1	-1.3855e+0	3.6572e+1	21	0
	FDBAT	**-1.0331e+1**	**-5.1211e+0**	**-1.0116e+1**	**2.2865e-2**	**14**	0

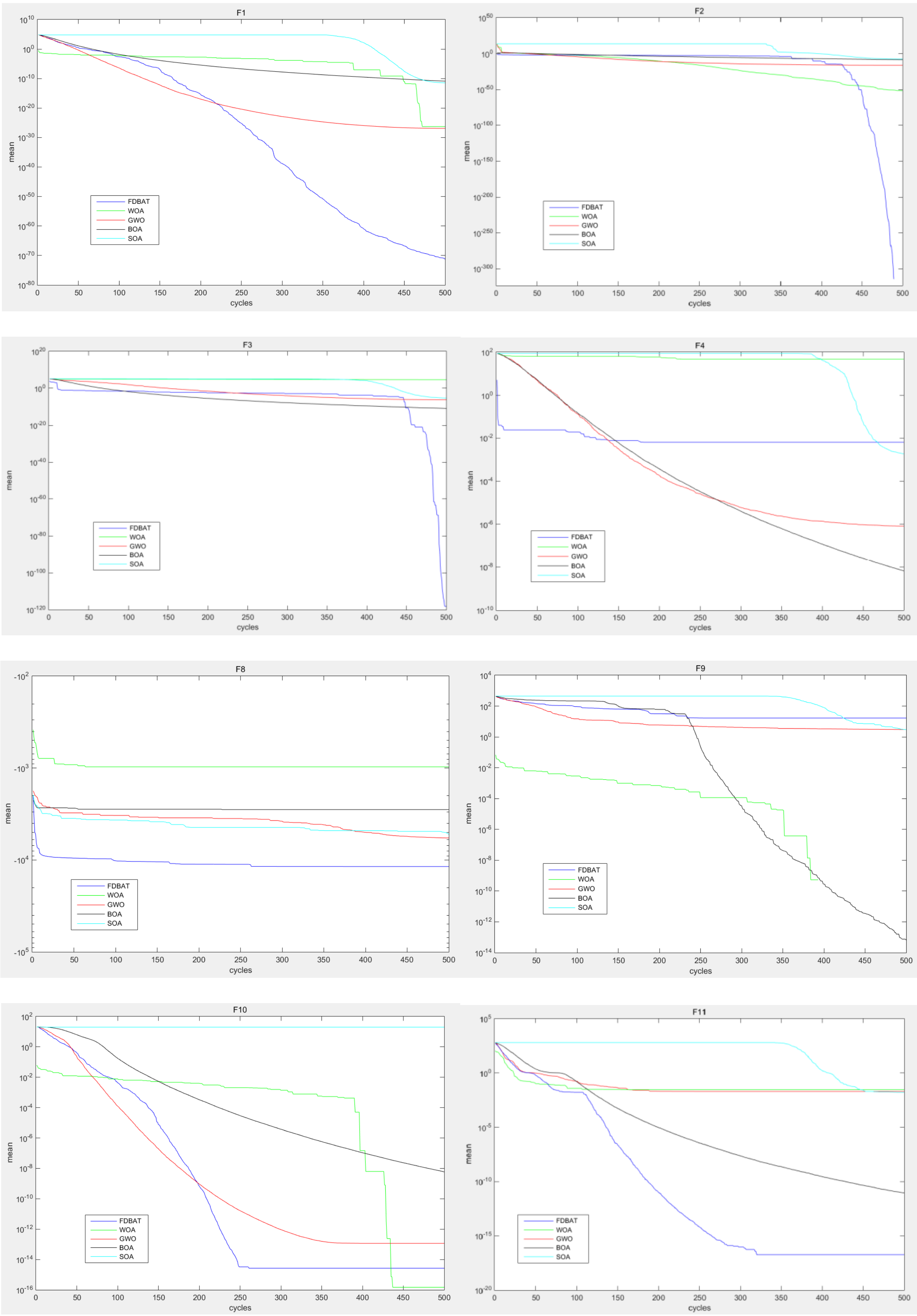


Figure 2. Convergence curves of FDBAT and WOA, GWO, BOA, SOA algorithms

Table 5. Optimization data of test function

Function	Algorithm	Average solution	Standard variance	Optimal solution	Theworst solution
F_1	FDBAT	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	WOA	4.1376e-73	9.9738e-73	7.3413e-82	3.1007e-72
	GWO	8.1469e-28	9.2628e-28	3.2289e-29	2.9190e-27
	BOA	1.2450e-11	1.2414e-12	1.0883e-11	1.4551e-11
	SOA	7.5124e-12	9.5911e-12	3.6619e-15	3.0816e-11
F_2	FDBAT	2.8415e-82	1.5861e-12	0.0000e+00	8.334e-123
	WOA	1.3191e-52	3.2698e-52	5.3042e-56	1.0546e-51
	GWO	1.4988e-16	1.1774e-16	3.7440e-17	3.7634e-16
	BOA	4.5281e-09	1.5278e-09	1.5922e-09	5.7314e-09
	SOA	2.1193e-08	2.1217e-08	2.5283e-09	6.3267e-08
F_3	FDBAT	1.3647e+0	1.2546e+1	0.0000e+0	1.7926e+4
	WOA	**2.1193e-08**	**2.1217e-08**	**2.5283e-09**	**6.3267e-08**
	GWO	8.7996e-06	1.4506e-05	1.7517e-07	4.7553e-05
	BOA	8.7996e-06	1.4506e-05	1.7517e-07	4.7553e-05
	SOA	9.9182e-05	2.5499e-04	1.6446e-07	8.1534e-04
F_4	FDBAT	1.2501e+1	3.1211e+0	**0.0000e+00**	**5.0190e+01**
	WOA	5.5716e+01	2.4163 e+01	1.8495 e+01	8.8377 e+01
	GWO	6.6159e-07	4.3484e-07	2.4970e-07	1.6898e-06
	BOA	6.0622e-09	4.0783e-10	5.5496e-09	6.7156e-09
	SOA	1.5000e-03	2.2000e-03	4.2154e-05	6.4000e-03
F_8	FDBAT	-1.2569e+04	0.2457e+00	-1.2570e+04	-1.2502e+04
	WOA	-9.4921e+03	1.4130e+03	-1.1944e+04	-7.6396e+03
	GWO	-6.2764e+03	9.3768e+02	-7.9681e+03	-4.9634e+03
	BOA	-3.0059e+03	572.1172	-3.8350e+03	-2.0165e +03
	SOA	-5.0974e+03	610.6941	-6.3931e+03	-4.4614e+03
F_9	FDBAT	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	WOA	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	GWO	3.6162e+00	3.5242e+00	5.6843e-14	8.5607e+00
	BOA	1.0232e-13	2.0195e-13	0.0000e+00	5.1159e-13
	SOA	3.5442e+00	9.5244e+00	7.9581e-13	3.0250e+01
F_{10}	FDBAT	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	4.7962e-15	2.6214e-15	8.8818e-16	4.7962e-15	7.9936e-15
	GWO	1.0534e-13	9.6499e-15	8.9706e-14	1.1458e-13
	BOA	5.6517e-09	2.0555e-10	5.4120e-09	6.0208e-09
	SOA	1.9961e+01	9.8963e-04	1.9959e+01	1.9962e+01
F_{11}	FDBAT	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	WOA	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
	GWO	1.1500e-02	1.3900e-02	0.0000e+00	3.4500e-02
	BOA	7.9893e-12	3.2356e-12	2.9752e-12	1.2178e-11
	SOA	1.1600e-02	1.8600e-02	4.6448e-12	5.6000e-02

In the comparison with GWO, BOA, SOA and WOA algorithms, the FDBAT algorithm shows better optimization ability in Table 5. In F_3 and F_6, woa function finds a better optimal solution than FDBAT function. In F_{20} and F_{22}, gwo shows better optimization ability, which is slightly different from the extreme value found by the FDBAT function, and is better with a weak advantage. Therefore, in this comparison process, the convergence accuracy and speed of the FDBAT algorithm are higher than other intelligent algorithms.

4.3 CEC06-2019 Benchmark Function

To further evaluate the effectiveness of FDBAT, the IEEE CEC06-2019 benchmark function [28] is used for testing in Table 6. Using the CEC06-2019 benchmark test function, the FDBAT algorithm is compared with GWO, BOA, HHO, SOA, WOA, AOA. During the test and comparison process, the population number N is defined as 30. The $t_{\max} = 500$, and every function is repeated 30 times to obtain the average value.

It can be seen from Table 7 that none of the seven intelligent optimization algorithms can find the optimal solution of CEC06-2019 benchmark function, and the optimization results of $F24$, $F25$, $F27$ and $F30$ functions have a large deviation. However, FDBAT algorithm showed the best optimization effect in the test process, and 70% of the test results were better than the other six intelligent algorithms.

To sum up, the data in Table 7 fully proves that FDBAT algorithm has shown excellent optimization ability in the optimization process of CEC06-2019 test function.

Table 6. IEEE CEC06-2019 benchmark function

Function	Function name	F_{min}	Dim	Range
F_{24}	Storn's Chebyshev Polynomial Fitting Problem	1	9	[-8192,8192]
F_{25}	Inverse Hilbert Matrix Problem	1	16	[-16.384,16.384]
F_{26}	Lennard–Jones Minimum Energy Cluster	1	18	[-4,4]
F_{27}	Rastrigin's Function	1	10	[-100,100]
F_{28}	Griewank's Function	1	10	[-100,100]
F_{29}	Weierstrass Function	1	10	[-100,100]
F_{30}	Modified Schwefel's Function	1	10	[-100,100]
F_{31}	Expanded Schaffer's F6 Function	1	10	[-100,100]
F_{32}	Happy Cat Function	1	10	[-100,100]
F_{33}	Ackley's Function	1	10	[-100,100]

Table 7. CEC06-2019 benchmark function test results

Function	Algorithm	Average solution	Standard variance	Optimal solution	Theworst solution
F_{24}	FDBAT	**2.7945e+03**	**2.5397e+03**	**3.16978e+01**	**6.9508e+03**
	AOA	5.8608e+09	1.6461e+10	8.2220e+05	5.2581e+10
	HHO	5.0974e+04	3.6543e+03	4.4985e+04	5.4750e+04
	WOA	4.0960e+10	3.7607e+10	1.9577e+09	9.3214e+10
	GWO	3.9246e+08	6.9371e+08	8.4667e+06	1.8693e+09
	BOA	6.9599e+04	1.5630e+04	5.5122e+04	1.0913e+05
	SOA	8.4193e+06	1.3062e+07	4.7119e+04	3.2638e+07
F_{25}	FDBAT	9.5792e+03	2.2873e+03	7.4367e+03	4.0000e+03
	AOA	1.9357e+01	4.5840e-01	1.8507e+01	1.9843e+01
	HHO	1.7363e+01	8.8000e-03	1.7349e+01	1.7378e+01
	WOA	1.7351e+01	4.1000e-03	1.7344e+01	1.7371e+01
	GWO	**1.7343e+01**	**1.6976e-04**	**1.7344e+01**	**1.7371e+01**
	BOA	1.9357e+01	4.5840e-01	1.7525e+01	1.8672e+01
	SOA	1.7418e+01	1.4280e-01	1.7344e+01	1.7700e+01
F_{26}	FDBAT	**2.3185e+00**	**1.8945e+00**	**9.9950e-01**	**5.2784e+00**
	AOA	1.2702e+01	1.0000e-03	1.2702e+01	1.2702e+01
	HHO	1.2702e+01	8.4636e-06	1.2702e+01	1.2702e+01
	WOA	1.2702e+01	1.2778e-06	1.2702e+01	1.2702e+01
	GWO	1.2702e+01	2.8289e-06	1.2702e+01	1.2702e+01
	BOA	1.2702e+01	1.0000e-03	1.2702e+01	1.2702e+01
	SOA	1.2702e+01	5.2099e-06	1.2702e+01	1.2702e+01
F_{27}	FDBAT	**7.0125e+01**	**3.8177e+00**	**6.5421e+01**	**7.8402e+01**
	AOA	1.4241e+04	5.0026e+03	4.7673e+03	2.1838e+04
	HHO	1.7952e+02	1.3990e+02	6.6652e+01	5.4331e+02
	WOA	3.6054e+02	1.0812e+02	1.3057e+02	7.8024e+02
	GWO	4.9837e+02	3.0729e+02	3.0608e+01	2.4626e+03

	BOA	1.4241e+04	5.0026e+03	9.7687e+03	2.8853e+04
	SOA	3.1668e+02	3.6252e+02	6.5873e+01	1.1244e+03
F_{28}	FDBAT	5.7877e+01	1.7610e+01	2.5027e+01	8.1528e+01
	AOA	4.7269e+00	1.3176e+00	3.1143e+00	7.3223e+00
	HHO	2.4260e+00	4.5310e-01	1.4791e+00	2.8972e+00
	WOA	1.99389e+00	2.6600e-01	1.4542e+00	2.5619e+00
	GWO	**1.50308e+00**	**2.5590e-01**	**1.0424e+00**	**1.7747e+00**
	BOA	4.7269e+00	1.3176e+00	3.3888e+00	7.0098e+00
	SOA	1.7122e+00	2.3370e-01	1.4387e+00	2.2271e+00
F_{29}	FDBAT	**4.2639e+00**	3.8401e+01	5.5038e+00	9. 9945e+01
	AOA	9.2803e+00	6.5180e-01	8.0382e+00	9.8727e+00
	HHO	9.6784e+00	1.1292e+00	7.4889e+00	1.1437e+01
	WOA	9.0908e+00	6.7740e-01	7.0853e+00	1.0600e+01
	GWO	1.0782e+01	1.0116e+00	9.5097e+00	1.1450e+01
	BOA	9.2803e+00	6.5180e-01	9.7938e+00	1.2056e+01
	SOA	1.0987e+01	5.1270e-01	1.0111e+01	1.1765e+01
F_{30}	FDBAT	**5.6737e+01**	**4.3081e+01**	**2.4407e+01**	**9.8863e+01**
	AOA	1.3331e+02	1.5796e+02	1.4124e+02	4.0290e+02
	HHO	4.0373e+02	1.2222e+02	2.0930e+02	6.0341e+02
	WOA	6.4097e+02	3.6367e+02	3.6646e+02	1.1537e+03
	GWO	5.5683e+02	3.2100e+02	8.1499e+01	1.0480e+03
	BOA	1.3331e+02	1.5796e+02	7.5303e+02	1.2470e+03
	SOA	3.2513e+02	1.6760e+02	5.3073e+01	6.0443e+02
F_{31}	FDBAT	**4.4584e+00**	**3.4930e+01**	**2.5400e+00**	**1.0915e+02**
	AOA	5.2766e+00	7.7380e-01	4.3705e+00	6.1189e+00
	HHO	5.9375e+00	2.3380e-01	5.5235e+00	6.3035e+00
	WOA	5.8726e+00	7.2600e-01	4.653e+00	6.8546e+00
	GWO	5.0023e+00	1.209e+00	4.0726e+00	6.3448e+00
	BOA	5.2766 e+00	7.7380e-01	6.0551e+00	6.8986e+00
	SOA	6.2496e+00	3.9490e-01	5.7407e+00	6.9486e+00
F_{32}	FDBAT	6.0872e+01	7.1920e-01	5.9877e+01	6.2055e+01
	AOA	8.9147e+02	4.4459e+02	2.5333e+02	1.6039e+03
	HHO	**3.2182e+00**	**5.7420e-01**	**2.6603e+00**	**4.2324e+00**
	WOA	5.0486e+00	1.0545e+00	3.3253e+00	6.4439e+00
	GWO	4.6153e+00	8.8260e-01	3.3762e+00	5.8272e+00
	BOA	8.9147e+02	4.4459e+02	1.7656e+03	2.9974e+03
	SOA	2.3472e+01	5.5074e+01	4.6762e+00	1.8019e+02
F_{33}	FDBAT	**1.7183e+01**	**3.7853e-01**	**8.8863e+00**	**1.3152e+02**
	AOA	2.0125e+01	5.6400e-02	2.0046e+01	2.0240e+01
	HHO	2.0147e+01	1.2430e-01	2.0031e+01	2.0275e+01
	WOA	2.0287e+01	1.0180e-01	2.0175e+01	2.0475e+01
	GWO	2.0484e+01	8.400e-02	2.0388e+01	2.0600e+01
	BOA	2.0125e+01	5.6400e-02	2.0387e+01	2.0616e+01
	SOA	2.0602e+01	8.6500e-02	2.0411e+01	2.0692e+01

5 Conclusion

Bat algorithm is an intelligent optimization algorithm with good search performance, but its local search ability is poor. The reason is that bat algorithm has insufficient ability to analyze and judge the current solution. In this paper, the gradual rapid dive mechanism of Harris Hawks algorithm is introduced into bat algorithm, and the parameter update method of local search in the bat algorithm is improved. The parameter escape energy is introduced to distinguish the quality of prey. The improved bat algorithm can avoid wasting a lot of time near the local optimum and quickly find the global optimum. In order to test the effect of the improved algorithm, the different types of 23test functions are selected and compared with BAT, RNABAT and COBAT. The results show that FDBAT is better than other algorithms. At the same time, it is also compared with WOA, GWO, BOA and SOA algorithms, showing the best results in terms of convergence speed and accuracy. In addition, the IEEE CEC06-2019 benchmark function is used to test FDBAT. Compared with other six algorithms, it is concluded that FDBAT has obvious advantages in convergence performance.

FDBAT needs to be further studied in future research: 1) When dealing with the fixed-dimensional multimodal benchmark functions, the convergence speed still needs to be improved. 2) High-dimensional problems are not tested in this study, and it needs to handle projects with high dimensional and analyze the algorithm's performance.

Acknowledgments

The study is subsidized by Tianjin Research Innovation Project for Postgraduate Students (No. 2021YJSS293), Tianjin Natural Science Foundation under Grant No. 20JCZDJC00150 and No. 20JCYBJC00320, Tianjin University Student Innovation and Entrepreneurship Training Program Project (202210069064), Tianjin University of Commerce College Student Innovation and Entrepreneurship Training Program Project (202310069198).

References

[1] U. Rugwiro, C. H. Gu, W. C. Ding, Task Scheduling and Resource Allocation Based on Ant-Colony Optimization and Deep Reinforcement Learning, *Journal of Internet Technology*, Vol. 20, No. 5, pp. 1463-1475, September, 2019.

[2] G. L. Zhou, Y. M. Li, Y. C. He, X. L. Wang, M.C. Yu, Artificial fish swarm based power allocation algorithm for MIMO-OFDM relay underwater acoustic communication, *IET Communications*, Vol. 12, No. 9, pp. 1079-1085, June, 2018.

[3] G. Yavuz, B. Durmuş, D. Aydın, Artificial Bee Colony Algorithm with Distant Savants for constrained optimization, *Applied Soft Computing*, Vol. 116, Article No. 108343, February, 2022.

[4] S. C. Zhao, D. Wang, Elite-ordinary synergistic particle swarm optimization, *Information Sciences*, Vol. 609, pp. 1567-1587, September, 2022.

[5] I. A. Zamfirache, R.-E. Precup, R.-C. Roman, E. M. Petriu, Policy Iteration Reinforcement Learning-based control using a Grey Wolf Optimizer algorithm, *Information Sciences*, Vol. 585, pp. 162-175, March, 2022.

[6] X. F. Ou, M. Wu, Y. Y. Pu, B. Tu, G. Y. Zhang, Z. Xu, Cuckoo search algorithm with fuzzy logic and Gauss–Cauchy for minimizing localization error of WSN, *Applied Soft Computing*, Vol. 125, Article No. 109211, August, 2022.

[7] L.Y. Zhang, Z.C. Ren, T. Liu, J.Y. Tang, Improved Artificial Bee Colony Algorithm Based on Harris Hawks Optimization, *Journal of Internet Technology*, Vol. 23, No. 2, pp. 379-389, March, 2022.

[8] X. S. Yang, A new metaheuristic Bat-Inspired Algorithm, In: J. R. González, D. A. Pelta, C. Cruz, G. Terrazas, N. Krasnogor (Eds.), *Nature Inspired Cooperative Strategies for Optimization (NICSO)*, Springer, Berlin, Heidelberg, 2010, pp. 65-74.

[9] W. J. Ding, X. S. He, X. S. Yang, M. L. Sheng, The application of the improved bats algorithm in multi-objective optimization, *Basic Sciences Journal of Textile Universities*, Vol. 26, No. 4, pp. 537-542, December, 2013.

[10] S. Y. Sun, Q. Zhou, Application of Semi-supervised Exetreme Learning Machine Based on Bat Algorithm in Industrial Inspection, *Computer & Digital Engineering*, Vol. 50, No. 3, pp. 667-673, March, 2022.

[11] Y. L. Liu, T. B. Wu, J. N. Wang, T. Y. Zhou, Y. Cheng, Application of Modified Penalty Function Method and Bats Algorithm in Constrained Optimization, *Computer Engineering and Applications*, Vol. 51, No. 9, pp. 62-67, September, 2015.

[12] J. Zhang, L. Hong, E. Z. Dai, An improved bat algorithm for the Vehicle Routing Problem with Time Windows and Capacity Constraints, *Computer Engineering & Science*, Vol. 43, No. 8, pp. 1479-1487, August, 2021.

[13] Y. Chen, S. Chen, Research on Application of Dynamic Weighted Bat Algorithm in Image Segmentation, *Computer Engineering and Applications*, Vol. 56, No. 14, pp. 207-215, July, 2020.

[14] L. J. Liu, S. N. Luo, F. Guo, S. Y. Tan, Multi-point shortest path planning based on an Improved Discrete Bat Algorithm, *Applied Soft Computing*, Vol. 95, Article No. 106498, October, 2020.

[15] X. F. Yue, H. B. Zhang, Modified hybrid bat algorithm with genetic crossover operation and smart inertia weight for multilevel image segmentation, *Applied Soft Computing*, Vol. 90, Article No. 106157, May, 2020.

[16] M. R. Chen, Y. Y. Huang, G. Q. Zeng, K. D. Lu, L. Q. Yang, An improved bat algorithm hybridized with extremal optimization and Boltzmann selection, *Expert Systems with Applications*, Vol. 175, Article No. 114812, August, 2021.

[17] Y. Saji, M. Barkatou, A discrete bat algorithm based on Lévy flights for Euclidean traveling salesman problem, *Expert Systems with Applications*, Vol. 172, Article No. 114639, June, 2021.

[18] H. L. Yu, N. N. Zhao, P. J. Wang, H. L. Chen, C. Y. Li, Chaos-enhanced synchronized bat optimizer, *Applied Mathematical Modelling*, Vol. 77, pp. 1201-1215, January, 2020.

[19] Y. N. Zhang, R. J. Liu, X. Wang, H. L. Chen, C. Y. Li, Boosted binary Harris hawks optimizer and feature selection, *Engineering with Computers*, Vol. 37, No. 4, pp. 3741-3770, October, 2021.

[20] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, A. H. Gandomi, The Arithmetic Optimization Algorithm, *Computer methods in applied mechanics and engineering*, Vol. 376, Article No. 113609, April, 2021.

[21] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. L. Chen, Harris hawks optimization: Algorithm and applications, *Future Generation Computer Systems*, Vol. 97, pp. 849-872, August, 2019.

[22] S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey Wolf Optimizer, *Advances in Engineering Software*, Vol. 69, pp. 46-61, March, 2014.

[23] S. Arora, S. Singh, Butterfly optimization algorithm: a novel approach for global optimization, *Soft Computing*, Vol. 23, No. 3, pp. 715-734, February, 2019.

[24] G. Dhiman, V. Kumar, Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems, *Knowledge-Based Systems*, Vol. 165, pp. 169-196, February, 2019.

[25] C. Tang, W. Sun, M. Xue, X. Zhang, H. Tang, W. Wu, A hybrid whale optimization algorithm with artificial bee colony, *Soft Computing*, Vol. 26, No. 5, pp. 2075-2097, March, 2022.

[26] Y. X. Geng, L. Y. Zhang, Y. S. Sun, T. Fei, S. X. Jiang, J. J. Ma, Improved Bat Algorithm Based on RNA Genetic Algorithm, *Journal of Tianjin University (Science and Technology)*, Vol. 52, No. 3, pp. 315-320, March, 2019.

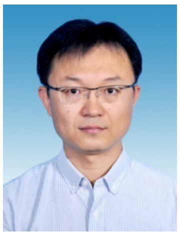
[27] Y. X. Geng, L. Y. Zhang, J. N. Guo, Y. Jiang, Adaptive Bat Algorithm Based on Historical Population Control, *Automatic Control and Computer Sciences*, Vol. 56, No. 5, pp. 438-446, October, 2022.

[28] J. Wang, M. Khishe, M. Kaveh, H. Mohammadi, *Binary Chimp Optimization Algorithm (BCHOA): a New Binary Meta-heuristic for Solving Optimization Problems*, *Cognitive Computation*, Vol. 13, No. 5, pp. 1297-1316, September, 2021.

Biographies

Yanxiang Geng is a doctoral candidate in Tianjin University, China. Her research interests include signal detection and information processing.

Liyi Zhang is a professor and PhD candidate supervisor in Tianjin University, China. His research interests include signal detection and information processing.



Yong Zhang is a Professor with Information Engineering College of Tianjin University of Commerce. His research interests include wireless sensor networks and information processing.

Zhixing Li is a under graduate majoring with Information Engineering College of Tianjin University of Commerce, His research interests include intelligent computing and information processing.

Jiahui Li is a undergraduate majoring with Information Engineering College of Tianjin University of Commerce, His research interests include intelligent computing and information processing.