Research on Bat Optimization Algorithms Incorporating Multiple Strategies

Zhijun Teng, Yaoyao Cui, Liangcen Gu, Yushan Fu, Yizhou Chen, Hongbiao Li

Abstract—**Bat algorithm is a popular solution in numerous engineering contexts, owing to its straightforward implementation and the minimal number of parameters required. However, it is prone to local optima, and exhibits limited population diversity. This paper proposes a multi-strategy Bat optimization algorithm to address the aforementioned problems. Firstly, a point set method that can be said to be both good and effective is adopted for the purpose of initializing the population and improving the diversity of the bat population. Secondly, a nonlinear inertia weight is proposed to update the position of the bat population and to adaptively adjust its position based on the characteristics of the bat population. The purpose of this adjustment is to improve the optimization performance of the bat algorithm. Subsequent to this, the crisscross optimization that is both horizontal and vertical is introduced in order to circumvent the possibility of the algorithm becoming trapped in a local optimum. In conclusion, a total of 12 benchmark test functions have been selected for the purposes of experimentation. A comparative analysis of seven distinct algorithms is then undertaken, with the objective of substantiating the feasibility and efficacy of the novel algorithm proposed in this study.**

***Index Terms*—bat algorithm, good point set, crisscross optimization, numerical analysis**

I. INTRODUCTION

In recent years, a significant focus of research has emerged on intelligent algorithms inspired by social behavior metaphors and natural phenomena. The efficacy of intelligent algorithms in the resolution of complex problems is well-documented. Notable examples include the particle swarm algorithm [1-2], bat algorithm [3], differential evolution algorithm [4-5], artificial bee colony algorithm [6], gravitational search algorithm [7], whale optimization algorithm [8-9], genetic algorithm [10] and so on.

The concept of the bat algorithm was initially introduced by the British scholar Yang [11] in 2010. This algorithm has gained significant popularity in various engineering applications due to its straightforward implementation and minimal number of parameters. In the seminal paper by Zhang Wei et al. [12], the bat algorithm was introduced as a means of optimizing variational modal decomposition (VMD). This pioneering work led to significant advancements in the field of composite fault feature extraction for rolling bearings, significantly enhancing the performance of the process. Zhang Kexue et al. [13] pioneered a novel approach by fusing a support vector neural network with a bat algorithm. This innovative fusion enabled the construction of an intelligent comprehensive evaluation model, which exhibited superior efficacy in addressing the critical issue of coal seam impact. Xie et al. [14] introduced an adaptive bat algorithm for indoor RFID localization, utilizing a tent map and an adaptive weighting factor to formulate a position evaluation function. This approach was shown to reduce the time taken for localization whilst concomitantly enhancing the accuracy of the results. Li et al. [15] combined the improved bat algorithm with complementary integrated empirical modal decomposition to optimize the microgrid load forecasting model. The bat algorithm introduces a reverse learning mechanism to increase the diversity of the population, thereby avoiding the risk of falling into local optimums and improving the convergence performance with the help of dynamic adaptive inertia weights and Lagrange interpolation. Consequently, the performance of the microgrid load forecasting model has been enhanced.

In light of the challenges encountered by the bat algorithm, a number of domestic and international scholars have put forward various solutions. Zhao et al. [16] introduced the bat algorithm into the particle swarm algorithm to construct a collaborative optimization algorithm. They applied Gaussian perturbation to the bat population and introduced Gaussian weights in the process of generating locally optimal solutions for the bat population. The aim was to improve the performance of the bat algorithm as a whole. The bat algorithm was built upon in this study, incorporating elements of the traditional bat algorithm. Li [17] proposed a chaotic mechanism for optimizing the initialization of the bat population, which has been shown to retain superior performance as a new solution. In addition, they proposed a learning factor with inertia weights to enable the bat

Manuscript received July 9, 2024; revised March 20, 2025.
This work was supported in part by National Outstanding Youth Science Fund Project of National Natural Science Foundation of China under Grant 61901102.

Zhijun Teng is a professor of School of Electrical Engineering, Northeast Electric Power University, Jilin, 132012, China (e-mail: tengzhijun@163.com).
Yaoyao Cui is an undergraduate student of School of Electrical Engineering, Northeast Electric Power University, Jilin, 132012, China (e-mail: 1729695796@qq.com).
Liangcen Gu is a postgraduate student of School of Electrical Engineering, Northeast Electric Power University, Jilin, 132012, China (e-mail: gueranji@gmail.com).
Yushan Fu is an undergraduate student of School of Electrical Engineering, Northeast Electric Power University, Jilin, 132012, China (e-mail: 2487946254@qq.com).
Yizhou Chen is an undergraduate student of Neuroscience Institute, Carnegie Mellon University, PA, 15213, USA (e-mail: yizhoucc@gmail.com).
Hongbiao Li is a professor of School of Computing, Northeast Electric Power University, Jilin, 132012, China (corresponding author e-mail: cuiyaoyao260728@163.com).

population to adaptively adjust its speed in order to avoid falling into a local optimum. In their seminal work, Du et al. [18] pioneered the integration of Levy flight and Gaussian variation strategy with the bat algorithm. The employment of Levy flight facilitates the adjustment of bats in suboptimal positions, while the utilization of Gaussian variation strategy introduces a Gaussian factor at the position. This innovative approach serves to circumvent the pitfalls of local optima and enhance the convergence accuracy. YLIDIZDAN, GULNUR et al. [19] sought to enhance the efficacy of the bat algorithm through the judicious application of weight factor and frequency optimization strategies, drawing upon the well-established principles of the traditional bat algorithm. It is evident that the aforementioned enhancements have been effective in averting the population from entering a local optimum. Nevertheless, the collective outcomes remain unsatisfactory. IIn summary, a bat optimization algorithm (Good point-set Crisscross Bat Algorithm, abbreviated as GCBA) is proposed. This algorithm incorporates good point set and crisscross optimization. The introduction of good point set is intended to enhance the diversity of the population. Crisscross optimization is employed to circumvent the bat population from attaining a local optimum. It has been hypothesized that the proposed scheme has the potential to enhance the efficacy of the Bat algorithm. Moreover, it has been demonstrated that the update formula for the population position is optimized by means of nonlinear weight factors. Consequently, this leads to an improvement in both the convergence speed and the performance of the GCBA algorithm in terms of seeking optimization.

II. Standard Bat Algorithm

The Bat Algorithm is a swarm intelligence optimization algorithm that has been designed to take advantage of the biological behavior of bats while foraging. Bats utilize the properties of echolocation, which are based on the emission of ultrasound pulses and the subsequent reception of bounced sound waves, to navigate and capture prey during nocturnal hours. In the process of detecting prey and navigating obstacles, bats typically reduce their acoustic output following the identification of prey, concurrently increasing their pulse emission rate. The idealized process of a bat searching for prey is as follows.

In the D-dimensional search space, the position and velocity of bat in the T_{th} iteration are updated according to the following formulas:

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

$$v_i^T = v_i^{T-1} + \left(X_i^{T-1} - X^*\right)f_i, \tag{2}$$

$$X_i^T = X_i^{T-1} + v_i^T, \tag{3}$$

In this context, the symbol β denotes a random number that is distributed uniformly within the interval [0, 1]. f_i is the pulse frequency at X_i . $f_i \in \left[f_{\min}, f_{\max}\right]$; X^* indices the current optimal solution. Nevertheless, it must be noted that the efficacy of the algorithm is not yet optimal. A persistent tension persists between the capacity to explore algorithms and the capacity to develop them. Furthermore, it has been observed that the accuracy of the algorithm is reduced at the subsequent stages of the iteration. Consequently, there is a significant need to enhance the performance and efficiency of the algorithms and expand their applications through various improvement methods.

III. Improved Bat Algorithm

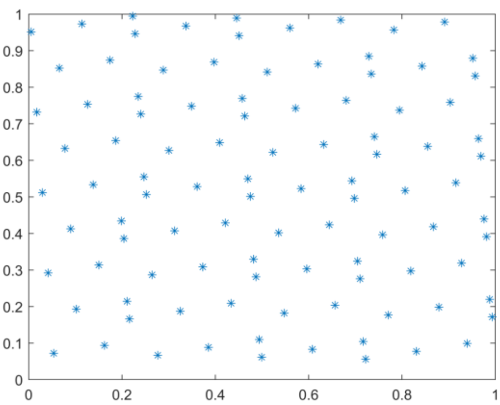
A. Good Point Set

The distribution of the initial population in the search space exerts a direct influence on two key aspects: the efficiency of the intelligent optimization algorithm and the accuracy of the search. However, the execution steps of the bat algorithm demonstrate that the initial population uniformity and traversal of the randomized distribution are poor, and it is easy to be trapped in the local optimum. The GCBA algorithm introduces good point set [20] in order to generate the initial bat population, thereby improving both the uniform distribution and the diversity of the population.

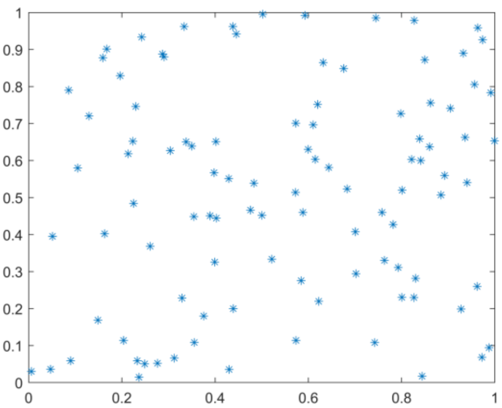
The good point set of N points taken in D-dimensional space is denoted as follows.

$$P_N(k) = \left\{b^1 * k, \cdots, b^i * k, \cdots, b^D * k\right\}, \tag{4}$$

where k=1, …, $N, b^i = 2\cos\left(2\pi i / p\right), 1 \le i \le D$; p is the smallest prime number satisfying $(p\text{-}3)/2 \ge D$.



(a) Good point set



(b) Randomization

Fig. 1 The initialization of the population

As illustrated in Fig 1, the initial population distributions were generated within the interval [0,100] using randomization and good point set, respectively. A comparison of the node distributions reveals that the uniformity, diversity and traversal of the population distribution initialized by good point set are significantly superior to those of a random distribution. Random population distributions are characterized by uneven densities, gaps in areas that are not searched, or overly dense distributions in individual areas.

B. Position Update Based on Inertia Weights

In this paper, a novel formulation for the bat position update is proposed. This is based on nonlinear inertial weights, with the aim of adjusting the global and local search capabilities. It is argued that this provides a significant improvement in the performance of the standard bat algorithm. The improved bat position update formula is as follows.

When *rand* > *r*, the position is updated according to Eq. (5).

$$\begin{cases} X_i^T = \omega * X_i^{T-1} + A_i^T \\ \omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) * \dfrac{3}{2 + e^{10-0.04T}} \\ A_i^T = e^{1-10*\frac{T_{\max}+T}{T_{\max}-T}} * A_i^{T-1} \\ r = r_0 * (1 - e^{-0.5T}) \end{cases}, \tag{5}$$

Otherwise, the position is updated according to Eq. (6).

$$\begin{cases} X_i^T = \omega * X_i^{T-1} + (1-\omega) * v_i^T \\ \omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) * \dfrac{3}{2 + e^{10-0.04T}}, \\ r = r_0 * (1 - e^{-0.5T}) \end{cases} \tag{6}$$

where ω indices the inertia weighting factor; $\omega_{\max}$ is the maximum value of inertia weights; $\omega_{\min}$ is the minimum value of inertia weights; A_i^T denotes the loudness of the sound wave emitted by the bat at the current moment; $T_{\max}$ indices the maximum number of iterations; *r* is the pulse rate of the bat;r_0 denotes the initial pulse rate of the bat.

C. Crisscross Optimization

Crisscross optimization [21] is a population-based stochastic search algorithm with two types of crossover: horizontal and vertical. This algorithm has been shown to be more effective in overcoming the issue of bat algorithms falling into local optimum.

1) Horizontal Crossover

As with the crossover operation of genetic algorithms, horizontal crossover is defined as an arithmetic crossover between the same dimensions of two different bat individuals in a bat population. The horizontal crossover operation has been demonstrated to enhance the global search capability of the algorithm by allowing bat individuals to be perturbed between different dimensions. Assuming lateral crossing between the *D* th dimension of bat individuals X_i^T and X_j^T, the update formula is as follows.

$$NX_{i,d}^T = r_1 * X_{i,d}^T + (1-r_1) * X_{j,d}^T + c_1 * (X_{i,d}^T - X_{j,d}^T), \tag{7}$$

$$NX_{j,d}^T = r_1 * X_{j,d}^T + (1-r_1) * X_{i,d}^T + c_1 * (X_{j,d}^T - X_{i,d}^T), \tag{8}$$

Where r_1 is a random number between [0,1] and c_1 is a random number between [-1,1]; $X_{i,d}^T$, $X_{j,d}^T$ denote d_{th} dimension bat individuals of individuals X_i^T and X_j^T , respectively, in the bat population; $NX_{i,d}^T$ and $NX_{j,d}^T$ are d_{th} dimensional bat individuals produced by $X_{i,d}^T$ and $X_{j,d}^T$ through horizontal crossover.

2) Vertical Crossover

Vertical crossover is defined as an arithmetic crossover between two different dimensions for the same bat individual. This approach facilitates the escape of one dimension from the local optimum in a given time frame without compromising the social behavior of the other dimension. Furthermore, it prevents the system from becoming trapped in a local optimum while ensuring the diversity of the population.

Assuming a vertical crossover between the d_1 and d_2 dimensions of the bat individual X_i^T , the optimal solution is generated according to Eq. (9).

$$\begin{cases} VX_{i,d_1}^T = r_2 * X_{i,d_1}^T + (1-r_2) * X_{i,d_2}^T, & r > rand \\ VX_{i,d_1}^T = X_{i,d_1}^T, & otherwise \end{cases}, \tag{9}$$

Where $d_1, d_2 \in N(1, D)$, $r_2 \in [0,1]$; VX_{i,d_1}^T denotes the d_1 and d_2 dimensions of X_i^T produced by longitudinal crossing of the d_1 dimension bat individual.

The fitness value and loudness are utilized as constraints to assess the superiority of the local solution. Consequently, this enables the update iteration of the local optimization, which is implemented as follows: The fitness value and loudness are used as constraints to assess the superiority of the local solution.

$$\begin{cases} X_{i,d}^T = NX_{i,d}^T, & F(NX_{i,d}^T) < F(X_{i,d}^T) \,\&\, A_i^T > rand \\ X_{i,d}^T = X_{i,d}^T, & otherwise \end{cases}, \tag{10}$$

Where $F(NX_{i,d}^T)$ denotes the fitness value for individual $NX_{i,d}^T$; $F(X_{i,d}^T)$ is the fitness value for individual $X_{i,d}^T$. In the event of a decrease in the fitness value of an individual bat following a vertical crossover, when compared with the current bat's value, and a concomitant decrease in loudness, this is indicative of the bat signaling the detection of prey. It is imperative that the relevant bat individual positions be updated in a punctual manner to ensure the integrity of the original positions.

The bats are engineered to generate perturbations at a specific height within a localized area. These perturbations are generated by crossing the area horizontally. If the bats receive an acoustic response, they will continue to generate perturbations at different heights by crossing the area vertically in an upward or downward direction. This process constitutes a crisscross search path, which the bats will continue to follow until they locate the prey. The method is more efficient and accurate, but at the same time increases the time complexity of the algorithm.

D. Algorithm Steps

In summary, this paper sets out the steps to improve the bat algorithm as follows.

1) Set the relevant parameters, including the population size *M*, the maximum number of iterations $T_{\max}$, the search

dimension D, the search range ub, lb, and the pulse frequency range $f_{\min}$, $f_{\max}$.

2) Population initialization. The objective is to generate bat population individuals in the search space using the method of good point sets, and to determine the optimal bat location X^* in the current population. The fitness function calculates the fitness value of each individual in the population and then ranks them according to the magnitude of their fitness value.

3) Generate random number r_3, if $r_3> r$, update the bat position according to Eq. (5), otherwise update the bat position according to Eq. (6).

4) The vertical and horizontal crossover strategy is to be introduced, and the position is to be updated according to Eq. (7) to (9) to the extent that the algorithm is prevented from falling into a local optimum.

5) Generate a random number r_4. If $r_4< A_i^T$ and also satisfy the better fitness value of the new position, Eq. (10) is moved to the new position, otherwise the position is kept intact.

6) Determine whether the algorithm satisfies the termination condition. If satisfied, the algorithm ends and outputs the global optimal solution and the corresponding convergence curve. Otherwise, return the target value for the next search.

IV. Experimental Data and Simulation Analysis

The paper proposes a comparison and analysis of the GCBA algorithm with BA, FOSBA[22] and BBA[23]. This is achieved by means of a benchmarking function, the purpose of which is to test the performance of GCBA in relation to other bat algorithms. Moreover, the GCBA is compared with WOA, PSO and ABC in order to ascertain the performance of GCBA in relation to the other algorithms.

A. Parameters of the Algorithm

In order to ensure the objectivity of the test, the basic parameters will be kept consistent, and the specific settings are as follows. The maximum number of iterations $T_{\max}$ is 1000; Number of populations M is 20; The pulse amplitude r_0 is 0.5; The maximum value of the inertia weights $\omega_{\max}$ is 0.9. The minimum value of inertia weight $\omega_{\min}$ is 0.2; The pulse frequency increase factor $f_{\max}$ is 2; The pulse frequency attenuation factor is 0.

B. Test Results and Analysis

The paper aims to evaluate the performance of the GCBA algorithm in optimization. To this end, 12 benchmark test functions have been selected for 30 experiments. The functions selected for this study are as follows: F1-F6 are single-peak functions and F7-F12 are multi-peak functions. All tests were conducted using the software MATLAB 2017. The test is divided into two sections. On the one hand, the GCBA was subjected to longitudinal testing in comparison with other enhanced bat algorithms. Conversely, a side-by-side comparison was conducted in which the GCBA algorithm was pitted against a range of other intelligent optimization algorithms. Finally, the test results were subjected to statistical analysis.

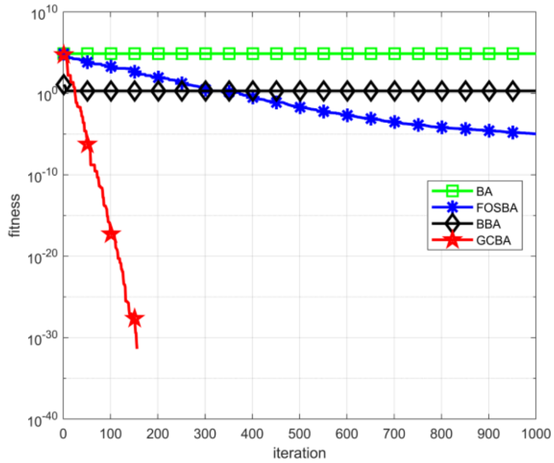
1) Comparison with Several BAs

The test data of the standard bat algorithm and several bat optimization algorithms for the 10 benchmark test functions are analyzed in Table I. The findings of this study demonstrate that three bat optimization algorithms are effective and more closely approximate the ideal value than the standard bat algorithm. As illustrated in Figure 1, a clear disparity emerges between the theoretical optimum and the observed values for BA and FOSBA. Notably, BBA exhibits a standard deviation of 0, indicating minimal relative variability. It is evident that among the six single-peak functions designated F1-F6, four single-peak functions of GCBA attain the ideal optimum with a standard deviation of 0, thereby demonstrating stability and reliability. Three of F7-F10 reach the theoretical optimum, and two of them have standard deviations of 0. It is evident that three of the F7-F10 variables reach the theoretical optimum, and two of them have standard deviations of 0. However, GCBA was found to be the closest to the ideal value when compared to FOSBA and BBA. Furthermore, it exhibits minimal standard deviation, which can attain a value of 0. The stability and aggregation of the optimization search process is optimal.

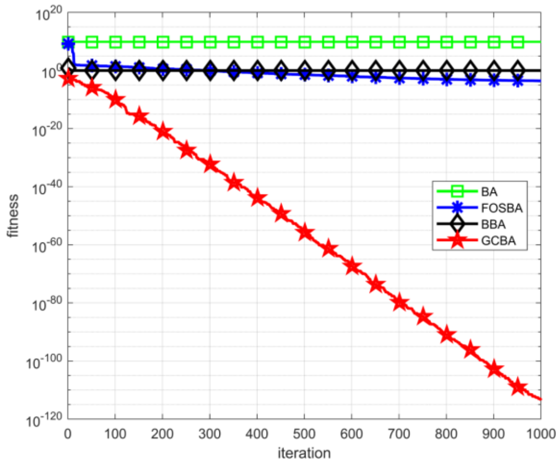
TABLE I Comparison Tests of FOSBA、BA、BBA、GCBA

Function	BA		FOSBA		BBA		GCBA	
	avg	std	avg	std	avg	std	avg	std
F1	7.23E+04	8.11E+03	8.12E-05	2.62E-04	2.80E+00	1.56E+00	**0.00E+00**	**0.00E+00**
F2	2.50E+13	3.36E+13	1.92E-04	1.76E-04	3.00E+00	1.18E+00	**0.00E+00**	**0.00E+00**
F3	8.86E+01	3.10E+00	3.58E+00	2.21E+00	1.00E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**
F4	2.78E+08	5.80E+07	2.76E+01	8.16E+00	7.45E+02	2.10E+02	2.85E+01	2.31E-01
F5	7.18E+04	8.86E+03	3.20E+00	2.56E+00	2.83E+00	1.37E+00	**0.00E+00**	**0.00E+00**
F6	1.30E+02	2.55E+01	8.74E-02	2.99E-02	3.67E+01	1.64E+01	3.91E-03	1.52E-02
F7	4.44E+02	2.71E+01	4.58E+01	1.50E+01	1.11E+00	3.93E-01	**0.00E+00**	**0.00E+00**
F8	6.35E+02	6.93E+01	1.39E-02	1.59E-02	1.24E-01	6.47E-02	**0.00E+00**	**0.00E+00**
F9	6.53E+08	1.60E+08	1.06E+00	1.20E+00	4.08E-01	2.31E-01	4.45E-02	4.80E-02
F10	3.91E-01	4.64E-01	3.14E-03	6.77E-03	1.48E-01	**0.00E+00**	4.50E-04	2.63E-04

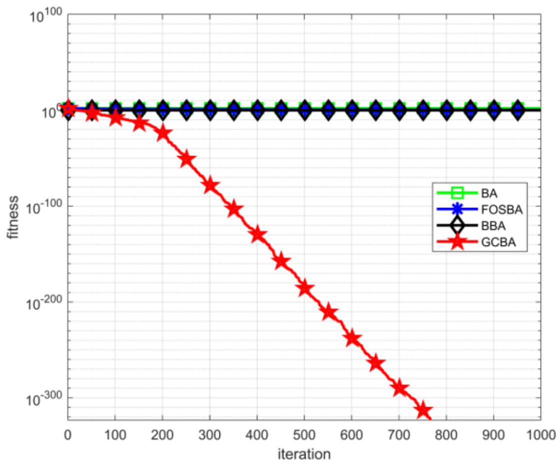
The convergence performance of the four bat algorithms on the single-peak function is represented in Fig. 2. As illustrated by the six plots, the convergence speed of this algorithm compares favorably with that of the other three algorithms. In (a) and (e), the theoretical optimum is determined in advance, consequently resulting in the premature termination of the iteration. As demonstrated in (d), the convergence performance of the GCBA algorithm exhibits superiority over that of the FOSBA algorithm. Conversely, the performance in terms of optimality discovery is comparatively deficient. As the number of convergences increases, GCBA falls into local optimality at a certain point in time, as illustrated in (f).



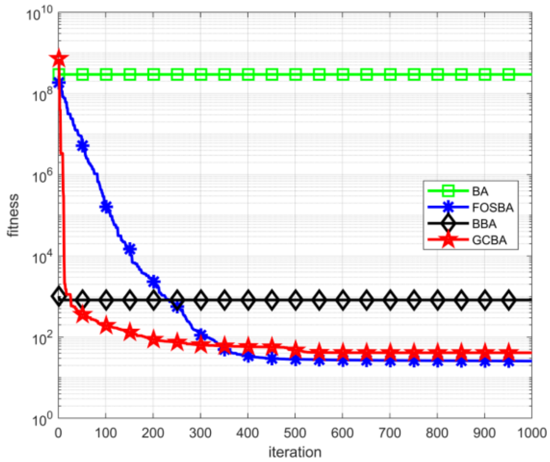
(a) 30 D-F1



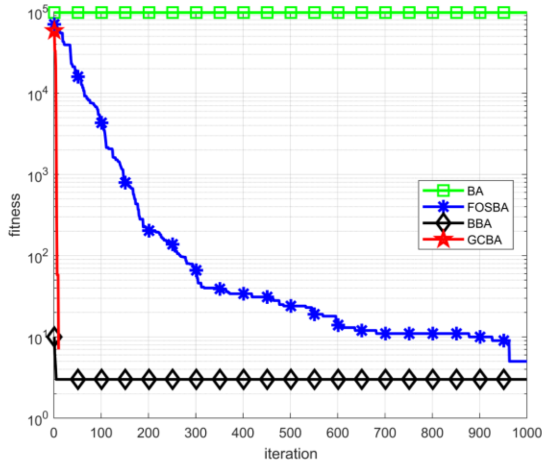
(b) 30 D-F2



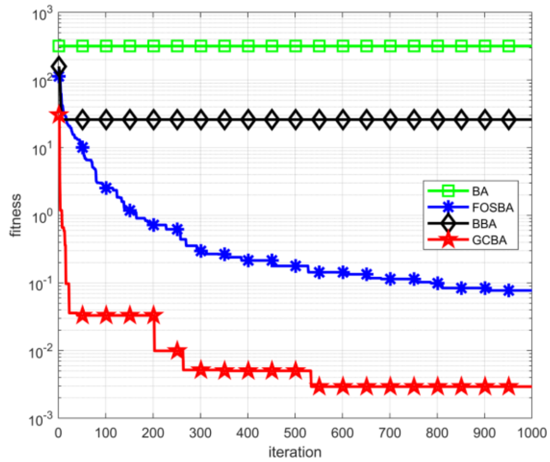
(c) 30 D-F3



(d) 30 D-F4



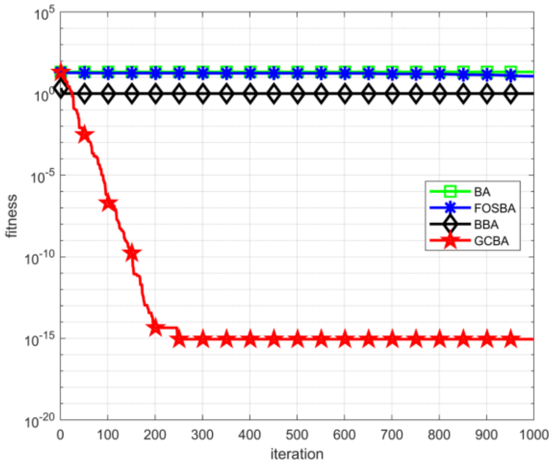
(e) 30 D-F5



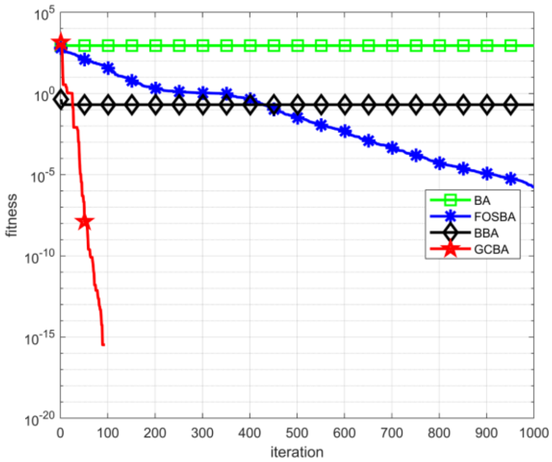
(f) 30 D-F6

Fig. 2 Convergence curves of BA, FOSBA, BBA and GCBA in the single peak function

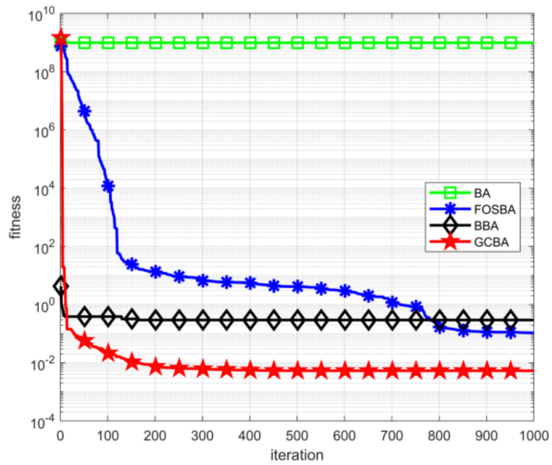
As illustrated in Fig. 3, the convergence performance of the various bat algorithms in the multi-peak function is demonstrated. It is evident that these algorithms exhibit superior performance in comparison to the randomness of the initial value, whilst exhibiting only a marginal discrepancy in their starting position. In the case of (b), the graph ceases iteration as the number of iterations increases, thereby identifying the theoretical optimal value at an earlier stage. It is evident from the provided data that, in comparison with other algorithms, the BBA exhibits minimal adaptation value fluctuation; this is particularly pronounced when the initial value is considered. However, it should be noted that the overall effect is more pronounced when there is a significant disparity between the initial value and the other algorithms.



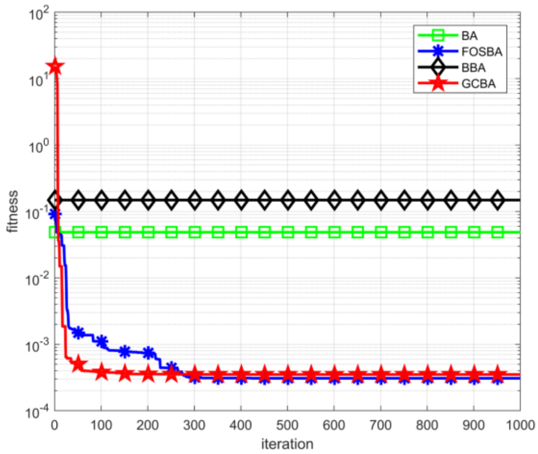
(a) 30 D-F7



(b) 30 D-F8



(c) 30 D-F9



(d) 4 D-F10

Fig. 3 Convergence curves of BA, FOSBA, BBA and GCBA in the multi-peak function

As demonstrated in Figures 2 and 3, the GCBA algorithm demonstrates a certain degree of enhancement in convergence speed and optimization accuracy in comparison to the other three algorithms. The GCBA algorithm demonstrates a substantial enhancement in the performance of both single-peak and multi-peak functions, rapidly identifying the optimal value with a reduced number of iterations. Furthermore, the GCBA algorithm demonstrates superior performance in both low and high-dimensional test functions, with the iterative curve of the test function exhibiting a rapid decrease.

It can be concluded that the GCBA algorithm demonstrates superior performance in comparison to other bat optimization algorithms with regard to stability and convergence accuracy.

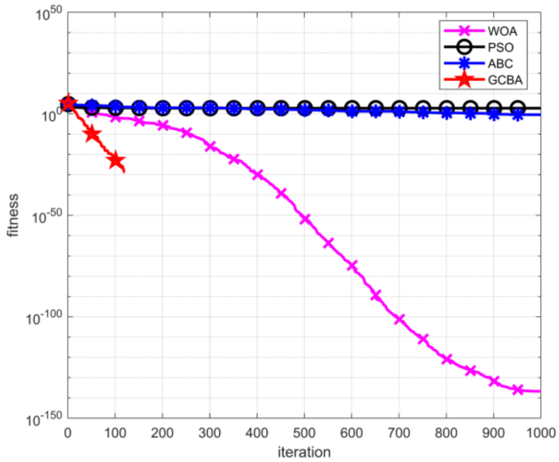
2) Comparison with Intelligent Optimization Algorithms

The test data of the four algorithms for the 12 benchmark test functions are analyzed in Table II. The table has been established that PSO and ABC are characterized by substandard data quality. However, ABC demonstrates superiority in terms of its optimization-seeking performance in the multi-peak function. Furthermore, a proportion of the data attains the theoretical value. A comparison of the two algorithms reveals that WOA is superior in identifying the optimal solution. The test data for both single-peak and multi-peak functions converged to the theoretical value. However, the overall stability of the system was found to be inadequate. The GCBA proposed in this paper utilizes a more substantial dataset to reach the theoretical values, and the data is stable.

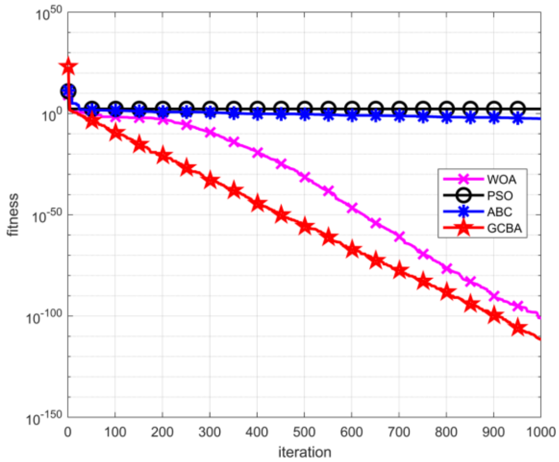
TABLE II Comparison Tests of WOA, PSO, ABC and GCBA

Function	WOA		PSO		ABC		GCBA	
	Avg	Std	Avg	Std	Avg	Std	Avg	Std
F1	3.94E-132	1.42E-131	4.73E+02	6.10E+01	2.33E-01	2.19E-01	**0.00E+00**	**0.00E+00**
F2	1.21E-95	6.49E-95	1.69E+02	1.66E+01	3.54E-03	1.58E-03	**0.00E+00**	**0.00E+00**
F3	3.30E+01	2.59E+01	9.80E+00	1.20E+00	6.35E+01	4.49E+00	**0.00E+00**	**0.00E+00**
F4	2.77E+01	5.72E-01	1.90E+06	3.94E+05	7.46E+04	7.12E+04	2.85E+01	2.31E-01

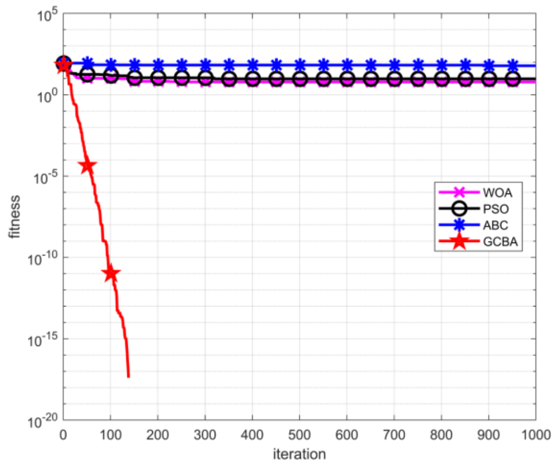
F5	**0.00E+00**	**0.00E+00**	4.98E+02	6.60E+01	1.57E+00	9.89E-01	**0.00E+00**	**0.00E+00**
F6	2.30E-03	3.31E-03	1.14E+02	2.31E+01	3.27E-01	8.99E-02	3.91E-03	1.52E-02
F7	**0.00E+00**	**0.00E+00**	4.36E+02	2.32E+01	2.39E+00	7.71E-01	**0.00E+00**	**0.00E+00**
F8	9.26E-03	3.51E-02	1.13E+00	1.43E-02	7.21E-01	1.18E-01	**0.00E+00**	**0.00E+00**
F9	4.10E-02	9.80E-02	4.92E+01	7.80E+01	4.22E+05	4.91E+05	4.45E-02	4.80E-02
F10	4.55E-01	1.93E-01	9.64E+01	3.68E+01	2.15E+05	1.89E+05	1.98E+00	8.32E-01
F11	7.74E-04	4.18E-04	1.69E-02	1.88E-02	7.91E-04	6.46E-05	4.50E-04	2.63E-04
F12	3.00E+00	1.90E-04	1.17E+01	2.59E+01	3.00E+00	1.64E-04	3.00E+00	5.83E-06



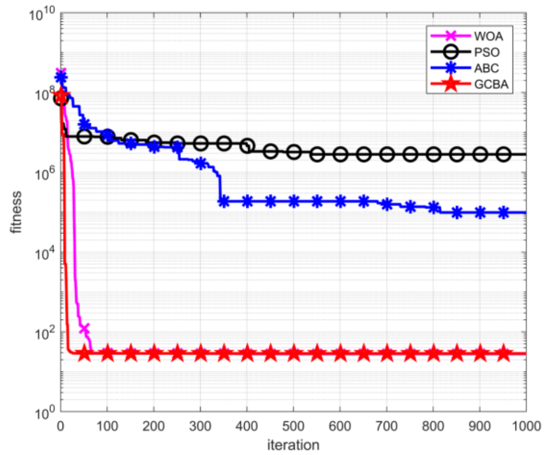
(a) 30 D-F1



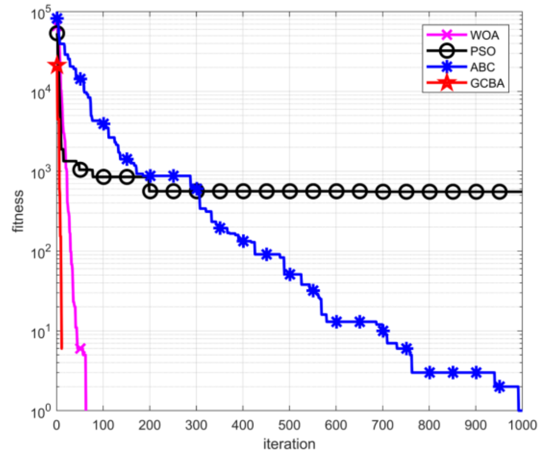
(b) 30 D-F2



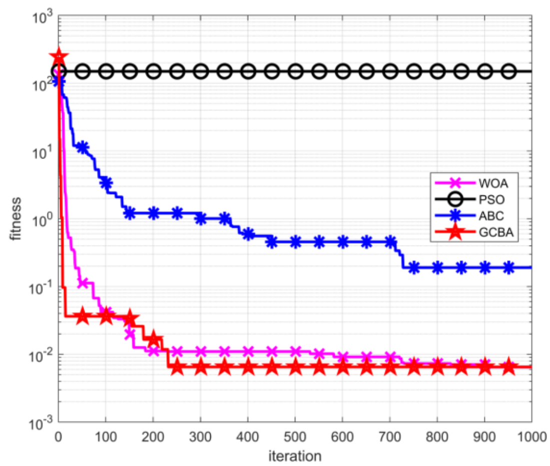
(c) 30 D-F3



(d) 30 D-F4



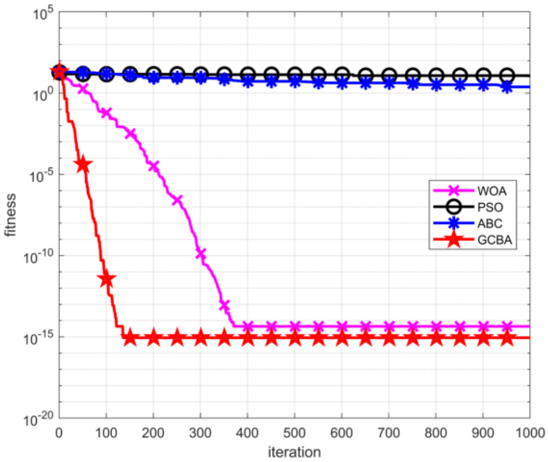
(e) 30 D-F5



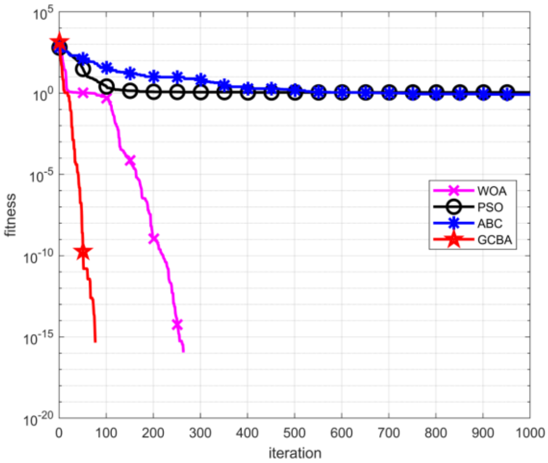
(f) 30 D-F6

Fig. 4 Convergence curves of WOA, PSO, ABC and GCBA in single peak function

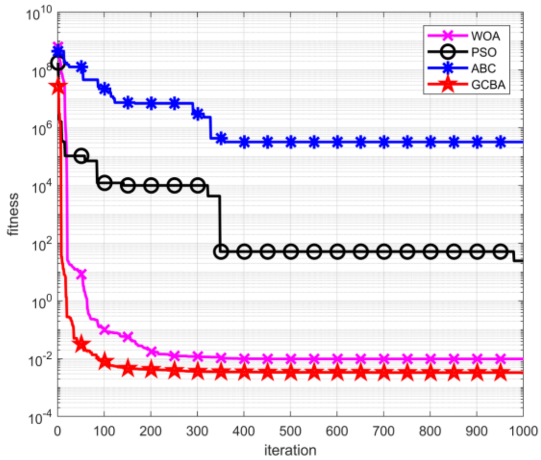
As illustrated in Fig. 4, the convergence performance of diverse intelligent algorithms on the single-peak function is demonstrated. The convergence of the algorithms is terminated by the early identification of the theoretical optimum in plots (a), (c) and (e). The GCBA experiences a brief descent into a local optimum in (f), and the performance of the GCBA algorithm is found to be inferior to that of the WOA algorithm when the number of iterations is in the range of 100-200 iterations. However, the GCBA algorithm's discrepancy with the theoretical optimum is much smaller.



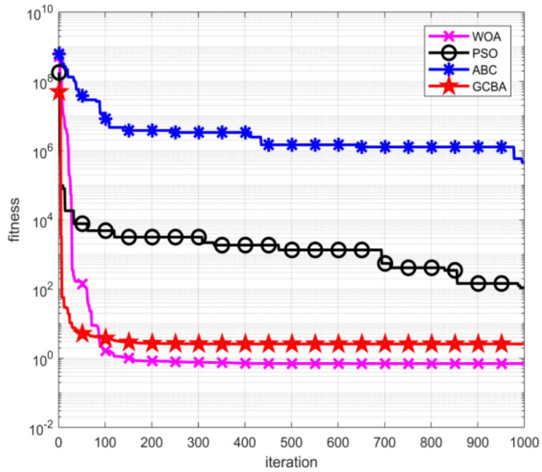
(a) 30 D-F7



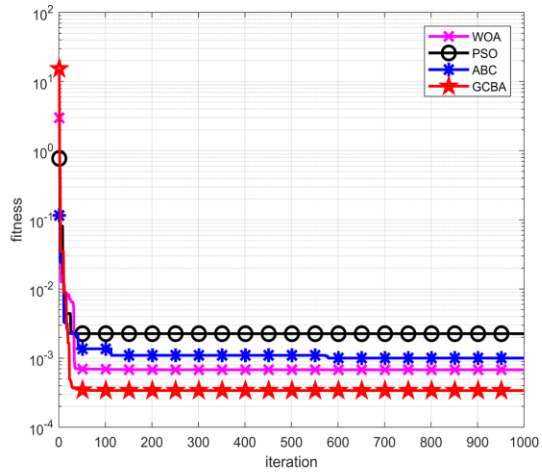
(b) 30 D-F8



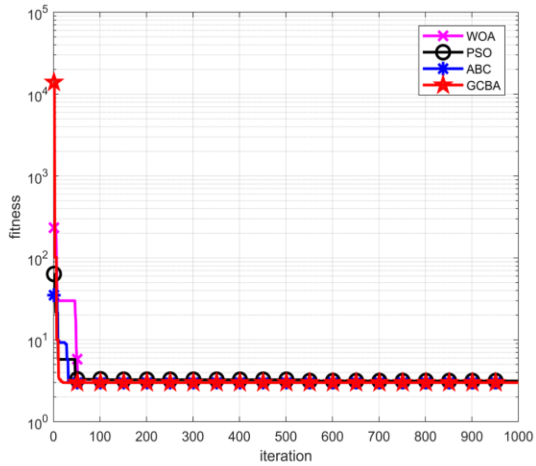
(c) 30 D-F9



(d) 30 D-F10



(e) 4 D-F11



(f) 2 D-F12

Fig. 5 Convergence curves of WOA, PSO, ABC and GCBA in the multi-peak function

As illustrated in Fig. 5, the convergence performance of multiple intelligent algorithms on multi-peak functions is demonstrated. As demonstrated by the figures, the GCBA circumvents the tendency towards local optimality and the subsequent stagnation. It is notable that both the GCBA and WOA in (b) are able to identify the theoretical optimum in advance and consequently terminate the optimization search. In scenario (d), GCBA demonstrates superior convergence speed in comparison to WOA. However, it should be noted that the efficacy of the optimization search is diminished.

As illustrated in Figs. 4 and 5, PSO and ABC demonstrate analogous convergence trends. However, it is important to note that each algorithm possesses distinct advantages and disadvantages when applied to diverse test functions. It is evident that both algorithms are more suitable for low-dimensional scenarios. A comparative analysis reveals that WOA exhibits superior convergence performance, facilitating the identification of the optimal solution and its subsequent approach with a reduced number of iterations. The GCBA algorithm has been demonstrated to exhibit the fastest descending iterative curve for the test function in comparison to the other three algorithms, and requires the least number of iterations to identify the optimal solution.

It can be concluded that the GCBA proposed in this paper exhibits superior performance in comparison to other intelligent optimization algorithms with regard to stability and convergence accuracy.

V. Conclusion

The present paper proposes a bat optimization algorithm that incorporates both good point set and crisscross optimization. The algorithm introduces good point set with a view to enhancing the diversity of the population and providing beneficial conditions for the subsequent optimization search. The employment of nonlinear weight factors facilitates the adaptive adjustment of bat population positions, thereby expediting the identification of the optimal solution and maintaining it in closer proximity. Crisscross optimization is employed with the objective of ensuring that the population of the algorithm evolves sufficiently in order to avoid a local optimum and attain a global optimum in a satisfactory amount of time. A side-by-side comparison of the experiments reveals that the test data of the GCBA is more closely aligned with the ideal value, exhibiting a reduced standard deviation. The stability and aggregation of the algorithm are better, and the optimal value can be found in fewer iterations with faster convergence, which verifies the effectiveness of the algorithmic improvement strategy. The results of the longitudinal comparison experiments show that the test data of the GCBA algorithm outperforms the other three intelligent algorithms in terms of stability and aggregation. The algorithm has faster convergence speed, higher accuracy of optimization search and the least number of iterations to find the global optimal solution, which verifies the feasibility of the GCBA algorithm. The main work in the next phase is to apply the GCBA algorithm to 3D wireless sensor networks in order to improve the coverage and enhance the comprehensive performance of the network.

References

[1] HUANG Y, LIANG F, FAN C, et al. "Improved particle swarm optimization algorithm with random mutation and perception," Journal of Northwestern Polytechnical University, vol. 41, no. 2, pp 428-438, 2023,

[2] BEN I D, SAMET, M. "Design and Optimization of Dual-Band Energy-Efficient OOK UWB Transmitter Via PSO Algorithm," Journal of Circuits, Systems and Computers, pp. 1-23, 2020.

[3] PRADIP KUMAR SAHU, GUNJI BALAMURALI, BIBHUTI BHUSAN BISWAL. "Robotic manipulator trajectory optimization using an improved modified bat algorithm," International Journal of Mechatronics and Automation, vol. 7, no. 1, pp. 11-22, 2020.

[4] WANG B, REN L, WANG X, et al. "Cooperative coevolution algorithm with covariance analysis for differential evolution," Journal on Communications, vol. 44, no. 1, pp. 189-199, 2023.

[5] Xiao-Rui Zhao, Xiao-Hong Chen, and Jie-Sheng Wang. "An Improved Wild Horse Optimizer Incorporating Dual Weight Starvation Strategy and Randomized Convergence Factor," Engineering Letters, vol. 32, no. 9, pp. 1794-1813, 2024.

[6] ZHOU X, HU J, WU Y, et al. "A Multi-strategy Artificial Bee Colony Algorithm Based on Fitness Grouping," Pattern Recognition and Artificial Intelligence, vol.35, no. 8, pp. 688-700, 2022.

[7] Yadav U, Tyagi T, Nagpal S." Fuzzy Optimized Gravitational Search Algorithm for Disease Prediction," International Journal of Swarm Intelligence Research, vol. 11, no. 3, pp. 89-103, 2020.

[8] Bui D T, Abdullahi M M, Ghareh S, et al. "Fine-tuning of neural computing using whale optimization algorithm for predicting compressive strength of concrete," Engineering with Computers, vol. 37, no. 1, pp. 701-712, 2021.

[9] Rania Saadeh, Mohammad-Kheir Allubbad, and Ahmad Qazza, "Investigating the Effectiveness of the Laplace Residual Power Series Approach for Volterra Integro-Differential Equation," Engineering Letters, vol. 32, no. 8, pp. 1594-1602, 2024.

[10] DENG M, WU Z, YAO Z, et al." Unmanned Aerial Vehicle Jamming Resource Scheduling Based on Parallel Genetic Algorithm with Elite Set, "Journal of Electronics & Information Technology, vol. 44, no. 6, pp. 2158-2165, 2022.

[11] YANG X. "Bat algorithm for multi-objective optimization," International Journal of Bio-Inspired Computation, vol. 3, no. 5, pp. 267-274, 2011.

[12] ZHANG W, LI J, CHEN W. "A compound fault feature separation method of rolling bearings based on VMD optimized by the bat algorithm," Journal of Vibration and Shock, vol. 41, no. 20, pp. 133-141, 2022.

[13] ZHANG K, WU Y, HE M, et al. "Research on the Intelligent Comprehensive Evaluation of Coal Seam Impact Risk Based on the BP Neural Network Optimized by the Bat Algorithm," Modern Tunnelling Technology, vol. 60, no. 2, pp. 38-46, 2023.

[14] XIE L, LI Y, WANG Y, et al. "Indoor RFID localization algorithm based on adaptive bat algorithm," Journal on Communications, vol. 43, no. 8, pp. 90-99, 2022.

[15] LI X, TONG X, CAO J, et al. "Research and Application of Microgrid Short-term Load Power Prediction Based on CEEMD-IBA-LSSVM," Computer Measurement & Control, vol. 31, no. 3, pp. 49-55, 2023.

[16] ZHAO Z, MO H, WEN T, et al. "Hybrid bat and PSO algorithm with gaussian disturbance and cooperative optimization," Journal of Guangxi University (Natural Science Edition), vol. 43, no. 6, pp. 2235-2242, 2018.

[17] LI Z. "Adaptive Chaotic Bat Algorithm with Speed Limit," Mathematics in Practice and Theory, vol. 50, no. 20, pp. 183-190, 2020.

[18] DU Y, LIU S. "An Improved Bat Algorithm with Gauss Mutation and Lévy Flights," Microelectronics & Computer, vol. 35, no. 3, pp. 83-87+92L, 2018.

[19] YLIDIZDAN G, BAYKAN O K. "A novel modified bat algorithm hybridizing by differential evolution algorithm, "Expert Systems with Application, vol. 141, no. 3, pp. 112949.1-112949.19, 2020.

[20] CUI L, ZHANG Y. "Research on spectral peak searching of two-dimensional MUSIC based on improved chicken swarm optimization algorithm, "Journal of Electronic Measurement and Instrumentation, vol. 34, no. 3, pp. 142-148, 2020.

[21] ZHAO P, ZHANG D, ZHANG L, et al. "Bald eagle search optimization algorithm with golden sine algorithm and crisscross strategy," Journal of Computer Applications, vol. 43, no. 1, pp. 192-201, 2023.

[22] LI M, WANG Q, HUI H. "Enhanced Bat Algorithm Based on Fractional-Order Strategy and Spiral with Lévy Flight," Computer Engineering and Applications, vol. 57, no. 18, pp. 75-81, 2021.

[23] Mirjalili S, Mirjalili M S, Yang X. "Binary bat algorithm," Neural Computing and Applications, vol. 25, no. 1, pp. 3-4, 2014.