

IMPROVED BAT ALGORITHM WITH NOVEL SEARCH MECHANISM AND ONE-DIMENSIONAL PERTURBATION LOCAL SEARCH STRATEGY

HAIBO ZHU, YUKUN WANG* AND YONG ZHANG

School of Electronic and Information Engineering
University of Science and Technology Liaoning
No. 185, Qianshan Road, Lishan Dist., Anshan 114051, P. R. China
{kdzhuhaibo; zy9091}@163.com; *Corresponding author: wyk410@163.com

Received December 2017; revised April 2018

ABSTRACT. *Bat algorithm often suffers from premature convergence, poor convergence rate and low accuracy in solving high-dimensional optimization problems because of quickly losing diversity. To enhance population diversity and search accuracy and speed up convergence rate of bat algorithm, this paper proposed a novel search mechanism and one-dimensional perturbation strategy to balance the ability of exploration and exploitation during the search process. Novel pulse emission rate, loudness, velocity and location updating functions are designed to avoid premature convergence. A novel population moving mechanism is introduced into bat algorithm to help the trapped bats escape from local optimum and novel one-dimensional perturbation local search strategy is designed to increase efficiency and accuracy of local search. The proposed novel improved bat algorithm has been evaluated on a set of typical nonlinear benchmark functions and compared with recently improved bat algorithms and other evolutionary algorithms. Experimental results and statistic analysis confirm promising performance of the improved bat algorithm in solving high-dimensional nonlinear functions.*

Keywords: Bat algorithm, Exploration and exploitation, Novel population moving mechanism, One-dimensional perturbation strategy

1. **Introduction.** Bat algorithm (BA) is an attractive intelligence algorithm by simulating echolocation behavior. Bats emit a loud and short pulse of sound when it hits into an object. And after a fraction of time, the echo returns back to their ears. Thus, bats can compute how far they are from an object. In addition, this orientation mechanism makes bats be able to distinguish the difference between an obstacle and a prey [1]. Based on the behaviour of bat hunting to search the optimal solutions, Yang developed bat algorithm in 2010 [2]. When flying and hunting, bats emit some short, ultrasonic pulses to the environment and list to their echoes. Studies show that the information from the echoes will enable bats to build a precise image of their surroundings and determine precisely the distance, shapes and prey's location. Compared with other optimization algorithms, bat algorithm has the characteristics of fewer parameters, simplicity and easy to implement, and has attracted scientific researchers' attention to solve benchmark tests or real-world engineering problems. Now it has been widely applied in engineering and scientific research to solve optimization problems and it has a very high potential research value [3-5]. Studies have proved that bat algorithm is better than other swarm intelligence algorithms in solving low-dimensional optimization problems, such as particle swarm optimization (PSO), ant colony optimization algorithm and genetic algorithm (GA).

However, in high-dimensional optimization problems, super bats in the population algorithm attract other individuals to gather rapidly around them, resulting in a sharp decline in population diversity. So bat algorithm suffers from premature convergence due to quickly losing diversity with number of iterations [6-8]. To improve the performance of bat algorithm, some researchers have proposed some improved methods and obtained some achievement. Bat algorithm can get trapped in a local minimum in some multi-dimensional functions. Thus, the phenomenon of slow convergence rate and low accuracy still exists. The strategy, proposed by Ramli et al. in 2018, aims to modify the exploitation of bat algorithm in optimizing the solution by modifying dimensional size and providing inertia weight [9]. In [10], a novel hybrid self-adaptive bat algorithm using different DE strategies is proposed for improving the current best solution directing the swarm of a solution towards the better regions within a search space. In [11], a hybrid bat algorithm with natural-inspired algorithms is proposed. It combines the reproduction step from weed algorithm and genetic algorithm. The reproduction step is applied to cloning each bat population by fitness values and the genetic algorithm is applied to expanding the population. In [12], a new modification approach on bat algorithm (EBA) is proposed. Local and global search characteristics of bat algorithm have been enhanced through three different methods. The results obtained by some test sets have proven EBA superior to the standard one. In [13], Gaussian distribution is used to improve the bat algorithm. To solve higher dimensional problems and decrease the step length size, the paper uses Gaussian distribution to get a shorter step lengths during the search. In [14], a simulated annealing Gaussian bat algorithm (SAGBA) is proposed. The algorithm not only inherits the simplicity and efficiency of the standard BA with a capability of searching for global optimization, but also speeds up the global convergence rate. In [15], a hybrid bat algorithm with path relinking for the capacitated vehicle routing problem is proposed. In [16], a new cloud model bat algorithm (CBA) is proposed. The algorithm focuses on remodeling echolocation model based on living and preying characteristics of bats, utilizing the transformation theory of cloud model to depict the qualitative concept. [17] presents the cost-based formulation to determine the optimal size of the battery energy storage in the operation management of micro-grid. At the same time, aimed at complicated optimization problem of some restrictions, a robust and strong new evolutionary technique named improved bat algorithm is proposed to use for developing corrective strategies and to perform least cost dispatches. The performance of the approach is evaluated by one grid-connected low voltage micro-grid where the optimal size of battery energy storage is determined, which is to solve short term power supply, power quality improvement, facilitating integration of battery energy storage, ancillary service and arbitrage. In [18] considering an important optimization problem of multilevel image thresholding, an improved version of bat algorithm adopted to search for multilevel thresholds using Kapur and Otsu criteria is proposed, which improved standard bat algorithm, where our modifications add some elements from the differential evolution and from the artificial bee colony algorithm. New proposed improved bat algorithm proved to be better, improving quality of results in all cases and significantly improving convergence speed. And the proposed algorithm is an excellent choice for the multilevel image thresholding problem.

It can be seen from above literature that there are many methods to improve bat algorithm, but they mainly focus on the improvement of searching precision and global searching ability of the algorithm. In order to overcome the shortcomings of basic bat algorithm, in this paper, a novel improved bat algorithm (IBA) with reducing step-size local searching strategy and novel population moving mechanism is proposed. The novel pulse emission rate, loudness and velocity updating methods, novel local search method

with reducing step-size local searching strategy and novel population moving mechanism are used to improve the performance of the algorithm.

The remainder of the paper is structured as follows. In Section 2, the basic bat algorithm and analyses of its shortcomings are introduced. An improved bat algorithm with reducing step-size searching strategy and novel population moving mechanism is given in Section 3. In Section 4, simulation results and analysis are given. The paper concludes by summarizing the performed work and outlines directions for the further development in Section 5.

2. Classical Bat Algorithm. Bat algorithm is inspired by the behavior of bat echolocation in hunting, and it is firstly introduced by Yang and Deb [26] in 2010 as a new heuristic algorithm. For solving the optimization problem with the variable $X = (x_1, x_2, \dots, x_d)$ and objective function $f(X)$, the steps of bat algorithm are as follows.

Step 1: Define the population number M , initial pulse loudness A_0 and pulse emission rate R_0 , optimal range $[X_{\min}, X_{\max}]$, velocity range $[V_{\min}, V_{\max}]$, pulse frequency range $[f_{\min}, f_{\max}]$, loudness reduce coefficient α , pulse emission rate increase coefficient γ and max iterations N . Set the current iteration number $n = 1$, randomly initialize the position $X_0(i)$ and speed $V_0(i)$ of each bat and calculate its fitness:

$$Fitness(i) = f(X_0(i)) \quad i = 1, 2, \dots, M \quad (1)$$

Find the global optimal solution X^* and the optimal value $Fitness_Best = f(X^*)$.

Step 2: Randomly generate new pulse frequency $f(i)$ and calculate the current velocity $V_n(i)$ and position $X_{new}(i)$ of each bat according to the following formulas:

$$f(i) = f_{\min} + (f_{\max} - f_{\min})\beta(i) \quad (2)$$

$$V_n(i) = V_{n-1}(i) + [X_n(i) - X^*]f(i) \quad (3)$$

$$X_{new}(i) = X_{n-1}(i) + V_n(i) \quad (4)$$

where $\beta(i) \in [0, 1]$ is a random vector drawn from a uniform distribution, n is the current iteration number, and X^* is the current global optimal solution which is located after comparing all solutions among all the N bats.

Step 3: Generating a random number $r \in [0, 1]$, $R_n(i)$ is the current pulse emission rate of the i th bat. For the bat i , if $(r > R_n(i))$ do the following:

$$X_{new}(i) = X^* + \varepsilon \bar{A}_n \quad (5)$$

where $\varepsilon \in [0, 1]$ is a d dimensional random vector drawn from a uniform distribution and \bar{A}_n is the average pulse loudness of the current bat population.

Step 4: Generate a random number $\mu \in [0, 1]$, $A_n(i)$ is the current pulse loudness of the i th bat, if $\mu < A_n(i)$ and $f(X_{new}(i)) < f(X^*)$, accept the new solution:

$$X_n(i) = X_{new}(i) \quad (6)$$

Calculate and update the fitness value of $X_n(i)$, and then update pulse emission rate and pulse loudness according to the following formulas:

$$R_{n+1}(i) = R_0(i)[1 - \exp(-\gamma n)] \quad (7)$$

$$A_{n+1}(i) = \alpha A_n(i) \quad (8)$$

where n is the current iteration number, α is loudness reduce coefficient and $R_0(i)$ is the initial pulse emission rate of the bat i . If $f(X_{new}(i)) < f(X^*)$, record the optimal fitness and corresponding solution:

$$X^* = X_{new}(i) \quad (9)$$

$$Fitness_best = f(X_{new}(i)) \quad (10)$$

Step 5: If the maximum number of iterations is reached, output the optimal fitness value and the optimal solution. Otherwise, return to Step 2 to continue searching [19-22].

The shortcomings of basic bat algorithm are mainly in the following aspects.

(1) The pulse emission rate and pulse loudness update functions are easy to lead to premature convergence of the algorithm.

(2) Velocity update function is easy to take the population gather to the best individuals and fall into local optimum and bat algorithm itself does not have an effective mechanism to help the algorithm escape from local optimum.

(3) Low local search efficiency and the fixed local search step-size limit the accuracy of the algorithm.

This paper mainly focuses on the above shortcomings to improve the bat algorithm.

3. Improved Bat Algorithm.

3.1. Novel pulse emission rate & loudness updating function. The pulse emission rate updating function and loudness updating function in Formula (7) and Formula (8) determine the convergence speed of bat algorithm. In order to avoid premature convergence, this paper designed a novel pulse emission rate and loudness updating functions as follows:

$$R_{n+1}(i) = R_0(i) \frac{N - \lambda n}{N} \exp\left(\frac{N - \lambda n}{N}\right) \quad (11)$$

where N is the max iterations, n is the current iteration number, $R_0(i)$ is the initial pulse emission rate of the bat i and $\lambda \in [0.5, 0.8]$ is an adjustable coefficient.

The pulse emission rate update function of the basic bat algorithm shows a monotonous upward trend. In the initial stage of search, the conditional $R_{rnd} > R_n(i)$ is easy to establish. A large number of new solutions will quickly aggregate to the current optimal solution, and the algorithm will easily fall into local optimum. However, the novel frequency update function shows a monotonous downward trend. In the initial stage of search, the conditional $R_{rnd} > R_n(i)$ is not easy to establish. So population aggregation is slow and the probability of global search is large. As the number of iterations increases, the population gathering speeds up and the probability of local search increases, which can avoid algorithm precocity and accelerate the convergence of late search.

$$\begin{cases} A_0(i) = \alpha A_0(i) \\ A_{n+1}(i) = 1 - \xi \exp(-A_0(i)) \end{cases} \quad (12)$$

where n is the current iteration number, α is loudness reduce coefficient, $A_0(i)$ is the initial pulse loudness of the bat i and $\xi \in [0.3, 0.9]$ is an adjustable coefficient.

Figure 1 shows the pulse emission rate function curves for different values of initial pulse emission rate R_0 when $\gamma = 0.9$ in Formula (7) and $\lambda = 0.5$ in Formula (11). Figure 2 shows the pulse loudness function curves for different values of loudness reduce coefficient α when $\xi = 0.5$ in Formula (12). Here $N = 200$.

It can be seen from Figure 1 and Figure 2, for the original pulse emission rate and pulse loudness update functions, at the early stage of the search, judgment condition $r > R_n(i)$ is easy to satisfy, large number of new solutions will gather to the optimal solution quickly, and the algorithm is easy to premature. And in the late stages of the search, judgment condition $\mu < A_n(i)$ is difficult to satisfy, new solutions are difficult to be accepted, and the algorithm converges slowly and difficultly. For the novel pulse emission rate and pulse loudness update functions, at the early stage of the search, the population gathered slowly, which increases the probability of globe search and alleviates premature of the algorithm. At the late stages of the search, the probability of local search will increase and the convergence speed will accelerate.

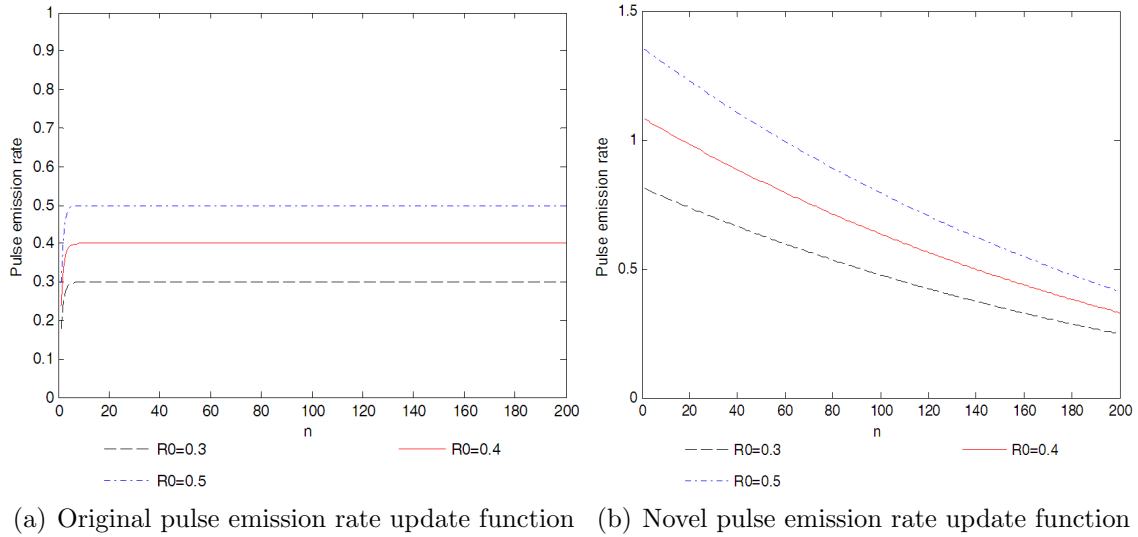


FIGURE 1. The curves of two different pulse emission rate functions

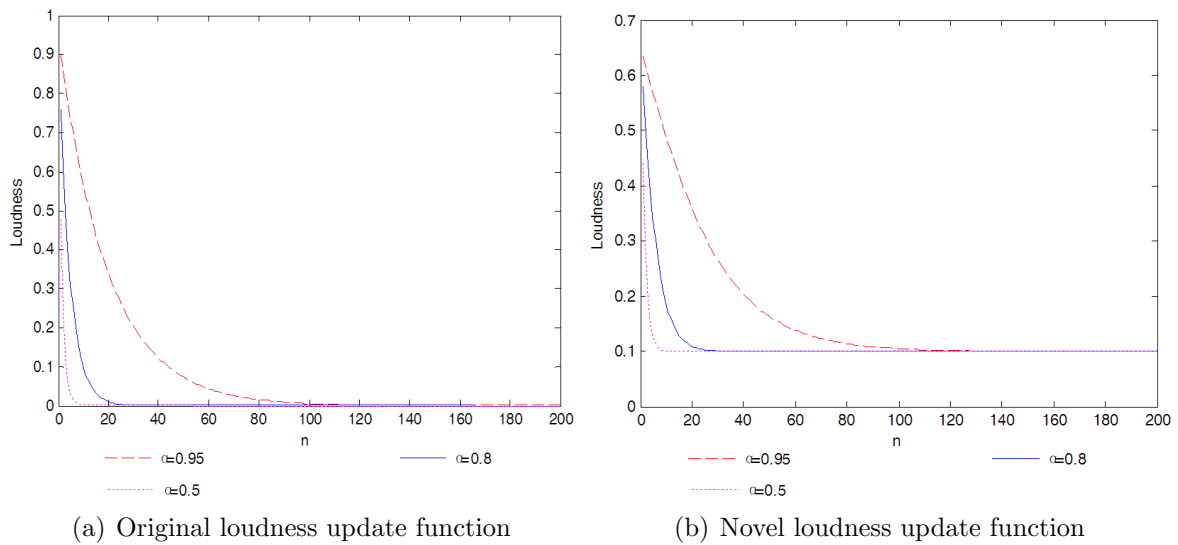


FIGURE 2. The curves of two different pulse loudness functions

3.2. Improved velocity and location update function. To enhance the diversity of population to improve the chance to find global optima and effectively improve convergence speed, this paper takes the optimal value of current iteration as one parameter of the novel velocity update function to reduce the strength that bat individuals are attracted by the global optimal solution (the best bat). The improved velocity update strategy is effective to avoid the solutions gather to the global optimal solution quickly and take the bat algorithm fall into local optimum. The novel velocity function is designed as follows:

$$V_n(i) = V_{n-1}(i) + [X_n(i) - (1 - c \cdot \varepsilon)X^* - (1 - (1 - c) \cdot \varepsilon) \cdot X_b^*] \cdot f(i) \quad (13)$$

where n is the current iteration number, X^* is the global optimal value, $\varepsilon \in [0, 1]$ is a random number and $c \in [0.1, 0.5]$ is an adjustable coefficient. X_b^* is the optimal value of current generation:

$$X_b^* = \min[f(X_{new}(i))] \quad (14)$$

For location update function, to maintain the diversity of population to improve the chance to find global optima and improve convergence speed effectively, we use the following strategy:

$$X_i(n+1) = rand \cdot X_i(n) + (1 - rand)V_i(n) \cdot N(1, 1) \quad (15)$$

where $rand(\cdot)$ is random number between $[0, 1]$. $N(1, 1)$ is *Gauss* disruption operator. The first aspect comes from itself position information. The second component is consisted in the disruption aspect.

3.3. Novel local search method. Bat algorithm uses full perturbation method in Formula (5) for local search. To generate a new solution, each vector of the current optimal solution will change. The search efficiency is low and the fixed search step-size leads to poor search accuracy. In order to improve the efficiency and accuracy of local search, this paper introduces a novel local search method with reducing step-size searching strategy. The novel local search formula is designed as follows:

$$\begin{cases} j = f_{ix}(M * \theta) + 1 \\ \Phi = \{\phi_1, \phi_2, \dots, \phi_M\}, \quad \phi_k = \begin{cases} 0, & k \neq j \\ 1, & k = j \end{cases} \\ S = ((N - n)/N)^\tau \\ X_{new}(i) = X^* + \Phi \left(\frac{1}{2} - \theta \right) S \bar{A}_n \end{cases} \quad (16)$$

where $f_{ix}(\cdot)$ is round toward zero, $\theta \in [0, 1]$ is a random number, M is the dimension of the solution, S is step-size, N is the max iterations, n is the current iteration number, \bar{A}_n is the average pulse loudness of the current bat population and $\tau \in [3, 7]$ is an adjustable coefficient.

It can be seen from Formula (16), the method to generate new solutions is changed and the efficiency of local search will also change. And with the increase of iteration number, the step-size S will gradually decrease and the search for details will gradually increase.

3.4. Novel population moving mechanism. In bat algorithm, the capability of escaping from local optimum is very weak, and it is difficult to continue searching global optimal solution once it falls into local optimum. To improve the global search ability of swarm intelligence algorithm, the methods mainly include Gauss and Cauchy perturbation, adaptive mutation, lévy flight, catfish effect and so on. Although these methods have certain effect, the implementation is complex [23-27]. In order to improve the global search ability of bat algorithm, this paper presents a novel and simple population moving mechanism. In order to implement this novel population moving mechanism, we need to modify Step 3 of basic bat algorithm. The modified Step 3 is as follows.

Step 3: Define a parameter L . (L is the number of generation that optimal fitness value of the optimization problem remains unchanged, when the optimal fitness is changed, the value of L is 0.) If $L > 3$, do the following:

$$X_n(i) = X_{new}(i) \quad (17)$$

Here the new algorithm will unconditionally accept new solutions generated in Step 2. If ($r > R_n(i)$ & $L < 6$), generate new solutions according to Formula (16).

The novel population moving mechanism can make the gathered bats spread out automatically when the algorithm is trapped in a local optimum and the algorithm will regain global search ability.

4. Simulation Results and Analysis. In order to illustrate the difference between BA and the proposed IBA algorithm, the two algorithms are tested by *Griewank*.

The 3D image of the function is shown in Figure 3.

The function is a multimodal function with multiple local optimal points, and due to the correlation between variables, it is very hard to obtain the global optimal solution. When the algorithms are tested, the fitness is the value of each function. The experimental parameters are set as follows by the several experimental tests: the number of population $M = 25$, max iterations $N = 100$, initial pulse loudness $A_0 = 0.95$ and pulse emission rate $R_0 = 0.4$, pulse frequency range $f \in [-2, 2]$, initial loudness attenuation coefficient $\alpha = 0.9$, pulse emission rate increase coefficient $\gamma = 0.95$, optimal range $[-10, 10]$, velocity range $[-2, 2]$. In IBA algorithm, adjustable coefficients $c = 0.3$, $\xi = 0.5$, $\lambda = 0.7$ and $\tau = 3$. The test results are shown in Figures 4 and 5.

It can be seen from Figure 4 and Figure 5, for BA in the early stage of the search, the solutions (bats) gather quickly, and the algorithm lost global search ability when it is trapped in local optimum. In the later stage of the search, the aggregation of the solutions (bats) is poor, which is not conducive to the improvement of searching accuracy. However, for the improved IBA, in the early stage of the search, aggregation rate of the solutions (bats) is alleviated, which is helpful to improve the global search ability. When

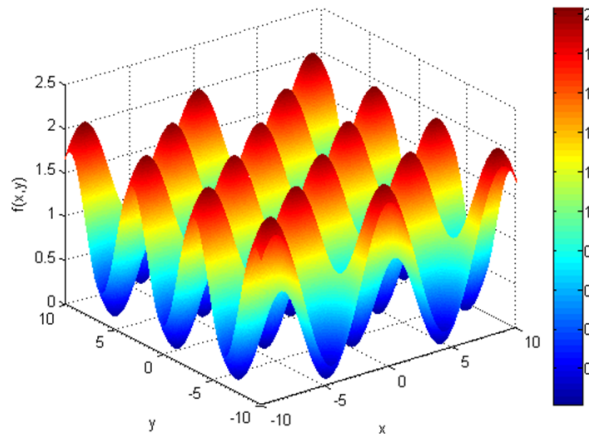


FIGURE 3. 3D image of test function

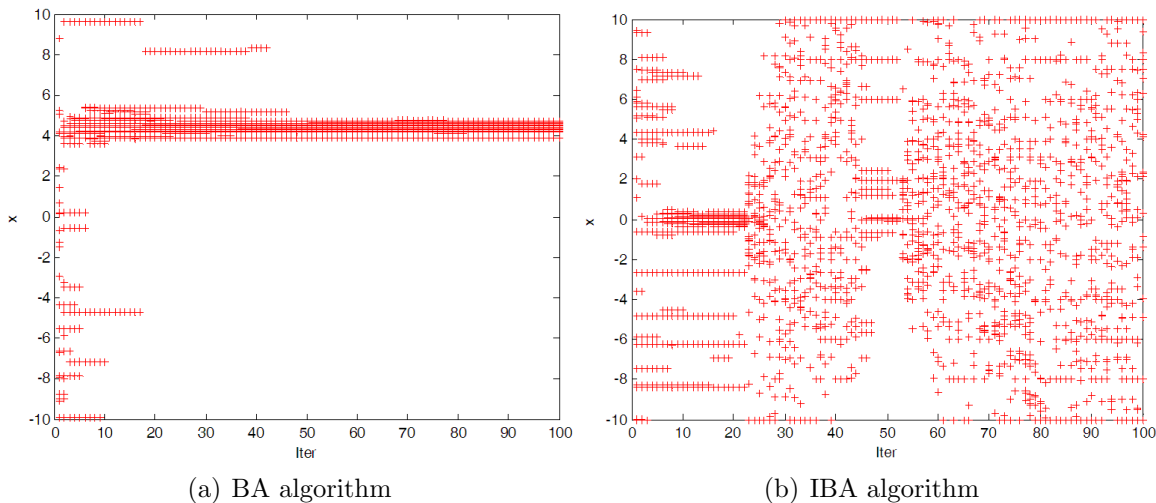


FIGURE 4. Distribution of solution x with increasing number of iterations

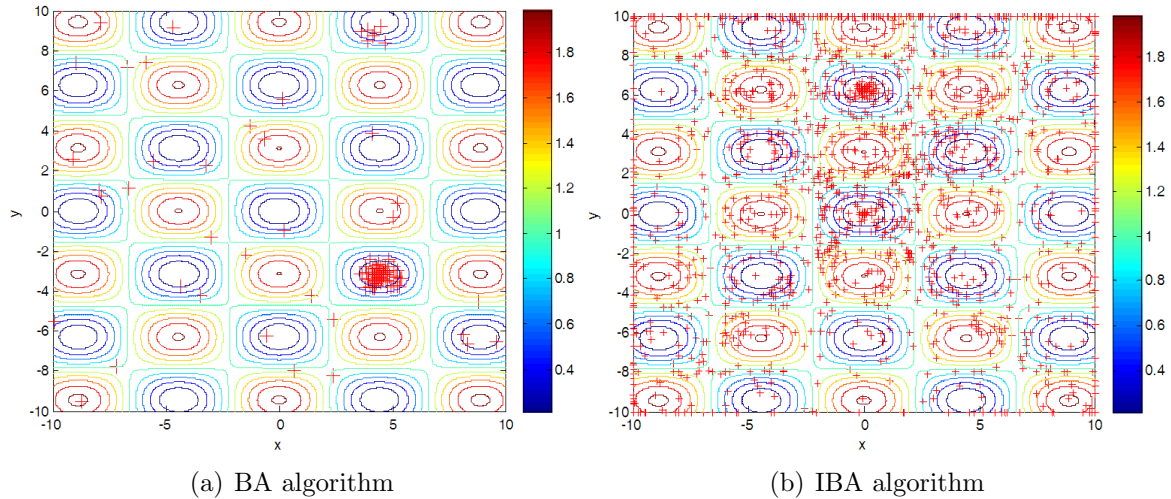


FIGURE 5. Location map of solution (x, y)

TABLE 1. Six test functions

Function	Function name	Expression	Dimension	Domain
f_1	<i>Sphere</i>	$f_1(x) = \sum_{i=1}^d x_i^2$	20	$[-100, 100]$
f_2	<i>Rosenbrock</i>	$f_2(x) = \sum_{i=1}^{d-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	20	$[-2.08, 2.08]$
f_3	<i>Griewank</i>	$f_3(x) = \frac{1}{4000} \left(\sum_{i=1}^d (x_i^2) - \left(\prod_{i=1}^d \cos\left(\frac{x_i}{\sqrt{i}}\right) \right) + 1 \right)$	20	$[-300, 300]$
f_4	<i>Rastrigrin</i>	$f_4(x) = \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i) + 10)$	20	$[-1.25, 1.25]$
f_5	<i>Sumsquares</i>	$f_5(x) = \sum_{i=1}^d i x_i^2$	20	$[-10, 10]$
f_6	<i>Michalewicz</i>	$f_6(x) = - \sum_{i=1}^d \sin(x_i) \left[\sin\left(\frac{i x_i^2}{\pi}\right) \right]^{2m}$, $m = 10$	20	$[0, \pi]$

the algorithm falls into local optimum, the population can be dispersed automatically and jump out of local optimum, so IBA has strong global search ability.

From Figure 5, we can see that the distribution of solutions of IBA is more extensive than bat algorithm, so the probability of obtaining global optimal solution is enhanced. Therefore, the novel population moving mechanism can effectively improve the global search ability of the improved algorithm. Secondly, in order to test the optimization ability of IBA for high-dimensional functions, six typical functions in [28] are chosen to test the performance of the algorithm. The six functions are listed in Table 1.

Sphere function is a simple unimodal function. *Rosenbrock* function is an inseparable single mode function, and its global extreme value is in steep valleys. For most search algorithms, it is difficult to acquire the right search direction within the canyon. *Griewank* function is a multimodal function with multiple local optimal points, and due to the correlation between variables, it is very hard to obtain the global optimal solution. *Rastrigin* function is a typical inseparable multimodal function, and in its searching domain, there are a large number of local minimum values, so it is difficult to obtain the global optimum. *Michalewicz* function has $d!$ local extreme values.

To illustrate the detail of the distribution of points during the search process, we choose three test functions as an example. One is *Sphere* function (f_1), which is a unimodal function. The other two are *Rastrigin* (f_4) and *Michalewicz* (f_6) functions, which are multimodal function. BA and IBA are used to solve the three minimization problems. The distributions of the points in BA and IBA during the search process are observed and shown in Figures 6-8 (Max iterations $N = 500$).

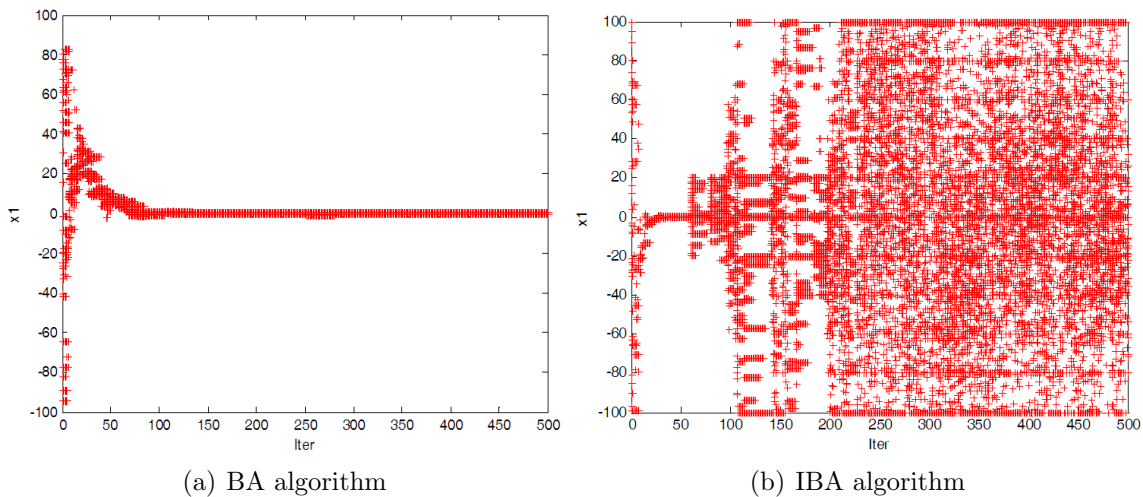


FIGURE 6. Distribution of solution x_1 with increasing number of iterations in f_1

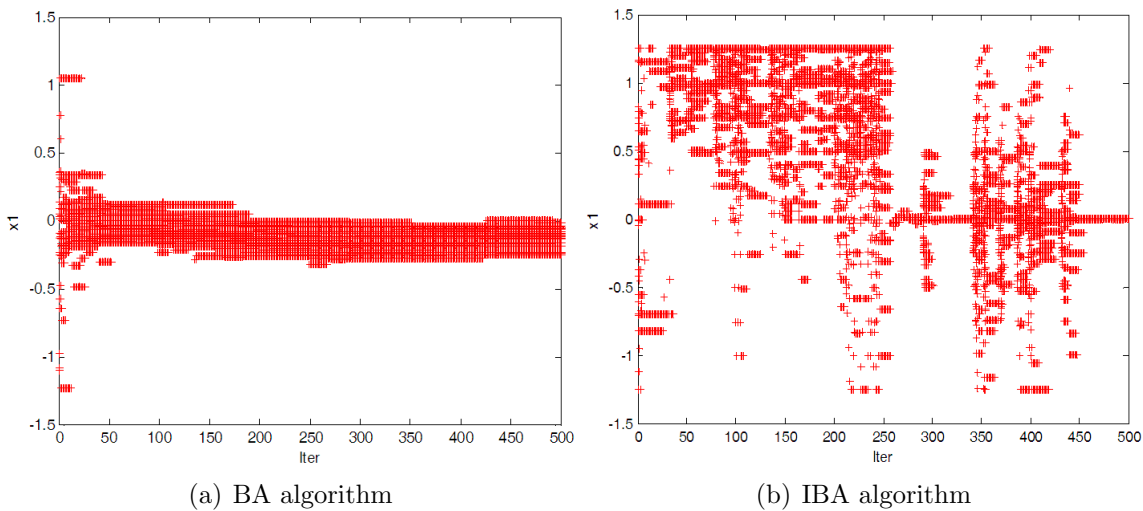


FIGURE 7. Distribution of solution x_1 with increasing number of iterations in f_4

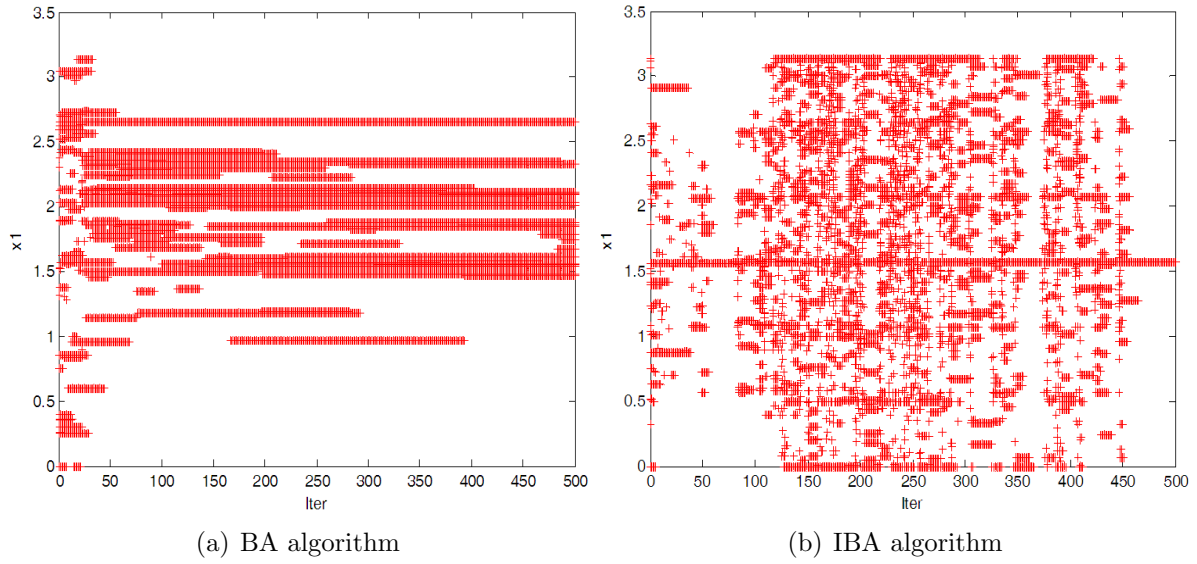


FIGURE 8. Distribution of solution x_1 with increasing number of iterations in f_6

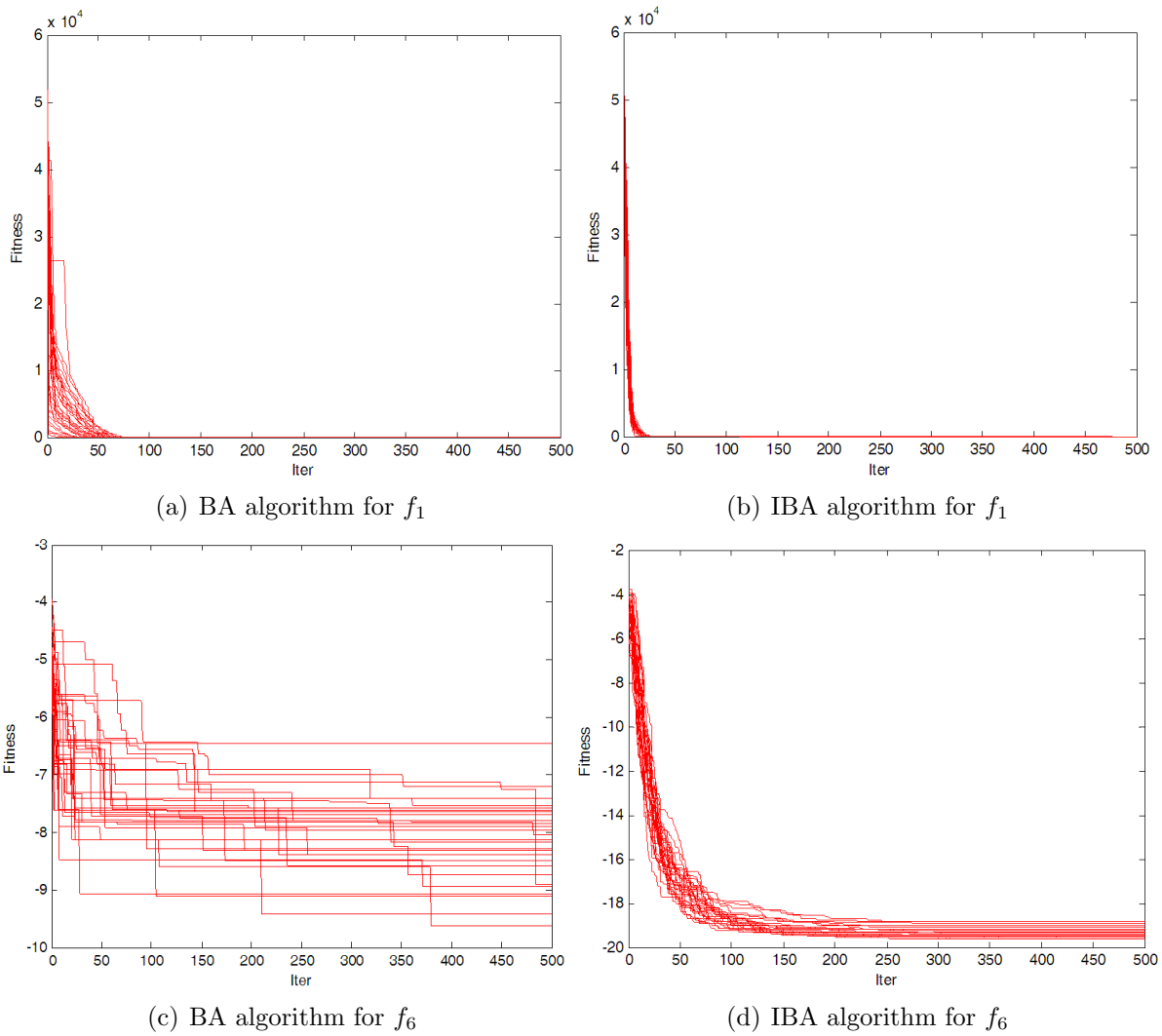


FIGURE 9. Convergence curves of different functions

They are plotted from the initialization to the final iteration. It can be seen from Figures 6-8, for 20-dimensional functions, whether in solving the unimodal function or the multimodal function, IBA produces a wide range of points than BA, and gathers more bats near the optimal solution. Therefore, IBA may have stronger ability to lead the bats escape from the local minima, and the global search ability is enhanced greatly.

To test the convergence rate and stability of IBA, the algorithm runs independently 30 times, Figure 9 shows the convergence rate and stability comparison between IBA and BA. For typical unimodal function f_1 , IBA has better convergence rate and stability than bat algorithm. The novel global and local search method improves the search efficiency of bat algorithm. For typical multimodal function f_6 , which is extremely difficult to obtain the optimal solution, IBA exhibits excellent convergence speed, stability and ability of jumping out from local optimum. Therefore, the IBA in this paper is successful, and the performance of IBA is much better than BA.

To prove the advantages of the IBA algorithm, and compare with other algorithms, such as dBA [29], RC-SSCS [30] and BA, we did another test. In the test, the experimental parameters are set as follows: optimal range $[X_{\min}, X_{\max}]$ of each function is shown in Table 2. For f_1, f_3, f_4 , velocity range $V = [X_{\min}/5, X_{\max}/5]$, and for f_6 , velocity range $V = [-X_{\max}/5, X_{\max}/5]$, the maximum iteration number is 1000, the population size is 30, and the other parameters of dBA and RC-SSCS are selected according to [29,30], respectively. Each algorithm runs independently 30 times and Figure 10 graphically shows the average fitness convergence curve of the above algorithms. From Figure 10, we can see that, for unimodal and multimodal function, the performance of IBA is superior to other algorithms. IBA has better convergence rate than dBA, RC-SSCS and BA. Specially, for multimodal function f_6 , which is extremely difficult to obtain the optimal solution, IBA exhibits excellent ability of global convergence. Therefore, IBA proposed in this paper is successful.

TABLE 2. Comparison of numerical testing results

Function	Dim.	Method	Best	Worst	Mean
f_1	20	BA	0.4324	0.6134	0.5324
		RC-SSCS	3.6450e-012	3.0061e-005	1.7847e-008
		IBA	1.3465e-020	4.4746e-011	6.2245e-013
f_2	20	BA	8.4223	43.4570	16.4371
		RC-SSCS	3.3432	14.2079	10.8277
		IBA	3.9532e-006	18.2510	1.7556
f_3	20	BA	0.0232	2.1724	0.2346
		RC-SSCS	1.0616e-011	0.1954	0.0454
		IBA	3.0918e-019	0.0662	0.0079
f_4	20	BA	7.8686e-08	2.5466e-05	4.5579e-06
		RC-SSCS	0.6950	10.3400	5.8859
		IBA	1.6527e-010	9.9496	2.6865
f_5	20	BA	4.0631	8.2127	6.1827
		RC-SSCS	2.4677e-009	8.4555e-004	5.6853e-006
		IBA	3.7815e-020	7.6555e-007	5.7873e-009
f_6	20	BA	-18.7220	-17.9451	-18.3535
		RC-SSCS	-8.2248	-5.1578	-6.8579
		IBA	-19.5643	-19.0417	-19.3370

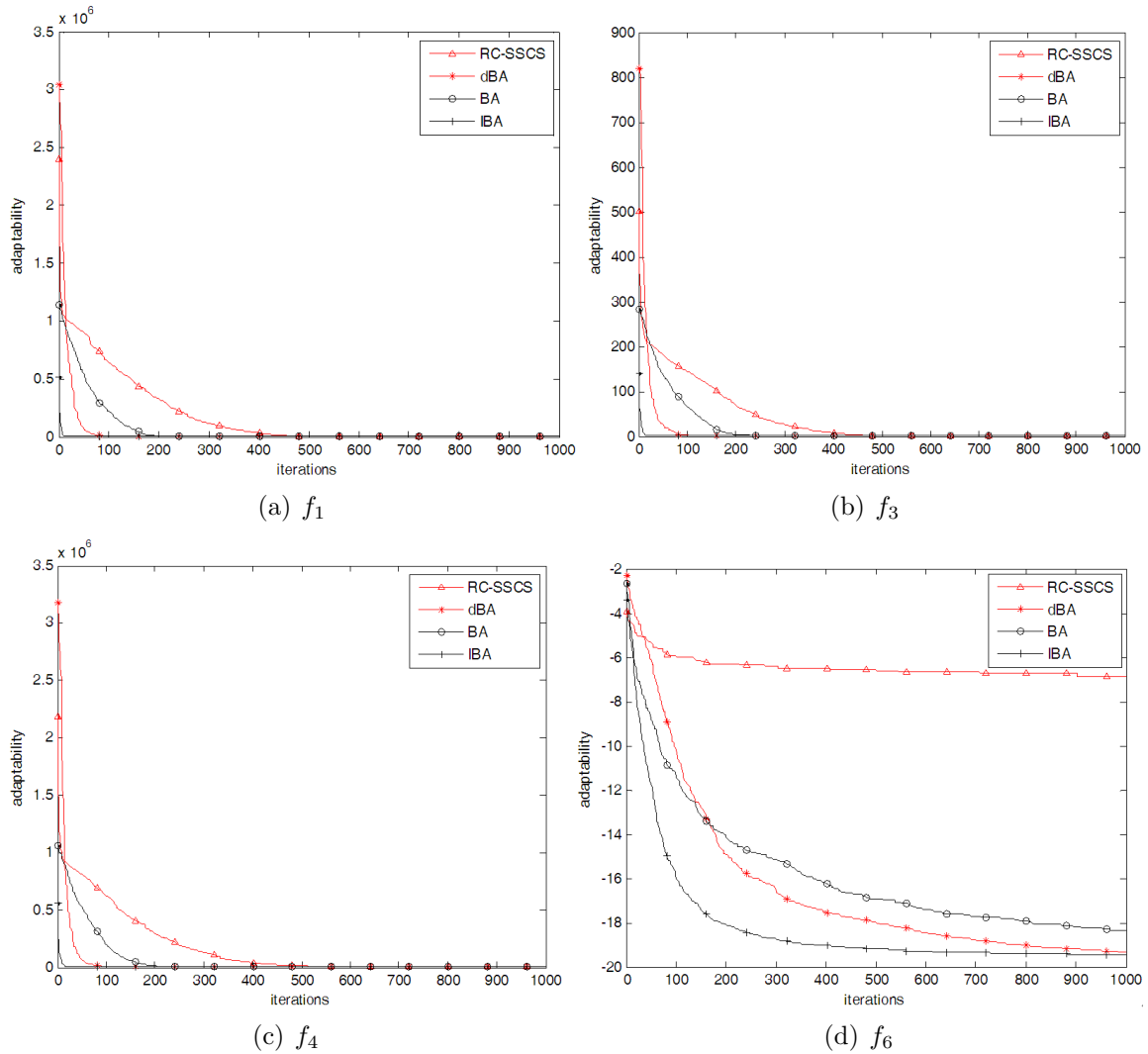


FIGURE 10. Convergence comparisons on the objective function

The numerical statistics results of BA, IBA and RC-SSCS are shown in Table 2. It can be seen from Table 2, IBA algorithm is much better than BA and RC-SSCS for functions f_1, f_3, f_4, f_5 and f_6 . The results of [30] have shown that the performance of RC-SSCS is better than ABC, PSO, CS, GCS, and SSCS. So it is also better than ABC, PSO, CS, GCS and SSCS in [29]. Especially, for function f_6 , the worst value of IBA is much better than the best value of RC-SSCS. For function f_4 , the best and mean value of IBA is much better than RC-SSCS. The worst value is little worse than RC-SSCS algorithm, but much better than ABC, PSO, CS, GCS and SSCS algorithm in [30]. Therefore, for 20-dimensional function, IBA has better search performance than other algorithms in [30].

To further test the performance of IBA algorithm for high-dimensional functions, the dimension of functions $f_1 \sim f_6$ is set to 25, 30, 40, 50 and 100 respectively. When the algorithm is tested, the algorithm runs independently 30 times, the number of iterations is 1000, and the test results are listed in Table 3. Figure 11 shows box diagrams of the 30 test results. The column numbers 1, 2, 3, 4, 5 in the figures stand for 25, 30, 40, 50 and 100 dimensions, respectively. It can be seen from Table 3, the algorithm has good performance for functions f_1, f_2, f_3, f_5 and f_6 . The performance of the algorithm does not decrease much with the increase of dimension of the test functions. For function f_4 ,

TABLE 3. Comparison of numerical testing results for different dimensions

Function	Dim	Best	Worst	Mean	Var
f_1	25	0	0	0	0
	30	0	0	0	0
	40	0	1.6040e-010	5.3466e-012	8.5757e-022
	50	0	1.5492e-010	5.1639e-012	7.9998e-022
	100	0	7.0721e-007	9.6809e-008	2.6947e-014
f_2	25	3.2422e-007	23.1863	1.3509	27.0121
	30	5.8745e-007	3.9866	0.1330	0.5298
	40	2.1625e-006	0.0049	2.4696e-004	8.1298e-007
	50	1.9493e-006	4.1822e-004	6.9046e-005	9.9132e-009
	100	4.3345e-005	0.0046	0.0012	1.5241e-006
f_3	25	0	5.4734e-014	1.8245e-015	9.9860e-029
	30	0	0.1226	0.0041	5.0122e-004
	40	0	8.7823e-012	3.9709e-013	2.8344e-024
	50	0	0.0320	0.0035	8.4693e-005
	100	0	0.0025	8.1815e-005	2.0080e-007
f_4	25	0	14.9244	3.2502	24.8464
	30	0	15.9193	4.6763	42.7485
	40	0	21.8891	11.1435	81.4758
	50	0	26.8639	10.4802	109.2852
	100	0	69.6473	37.2133	496.7891
f_5	25	0	0	0	0
	30	0	0	0	0
	40	0	2.6463e-010	8.8211e-012	2.3343e-021
	50	0	2.6427e-008	1.9662e-009	3.8962e-017
	100	0	1.2748e-005	1.9995e-006	1.5601e-011
f_6	25	-24.6272	-23.1281	-24.3815	0.1005
	30	-29.6101	-28.6523	-29.2947	0.0818
	40	-39.5878	-38.4248	-39.0931	0.0974
	50	-49.2887	-47.2241	-48.5742	0.2479
	100	-95.3727	-91.3507	-93.4289	0.9080

the performance of the algorithm is not perfect, but it does not deteriorate rapidly with the increase of the dimension of the function and the results are acceptable. Therefore, the IBA algorithm proposed in this paper can optimize high-dimensional functions with different dimensions.

It can be seen from Figure 11, for function f_1 , with the increase of dimension, the performance of IBA is stable. When the dimension is 100, the performance is litter worse. However, the worst value is below $7.5e-07$ and it is still a good result. For function f_2 , with the increase of dimension, the performance of IBA becomes better, when the dimension is 100, the result is perfect. For function f_3 , the performance of IBA is stable in different dimensions. The algorithm can get the optimal solution with a large probability. For function f_4 , with the increase of dimension, the performance of IBA becomes worse. For function f_5 , with the increase of dimension, the performance of IBA is stable. When the dimension is 100, the performance becomes worse, there are some isolated data points. However, the worst value is below $1.5e-05$ and it is still a good result. For function f_6 , the algorithm has good stability.

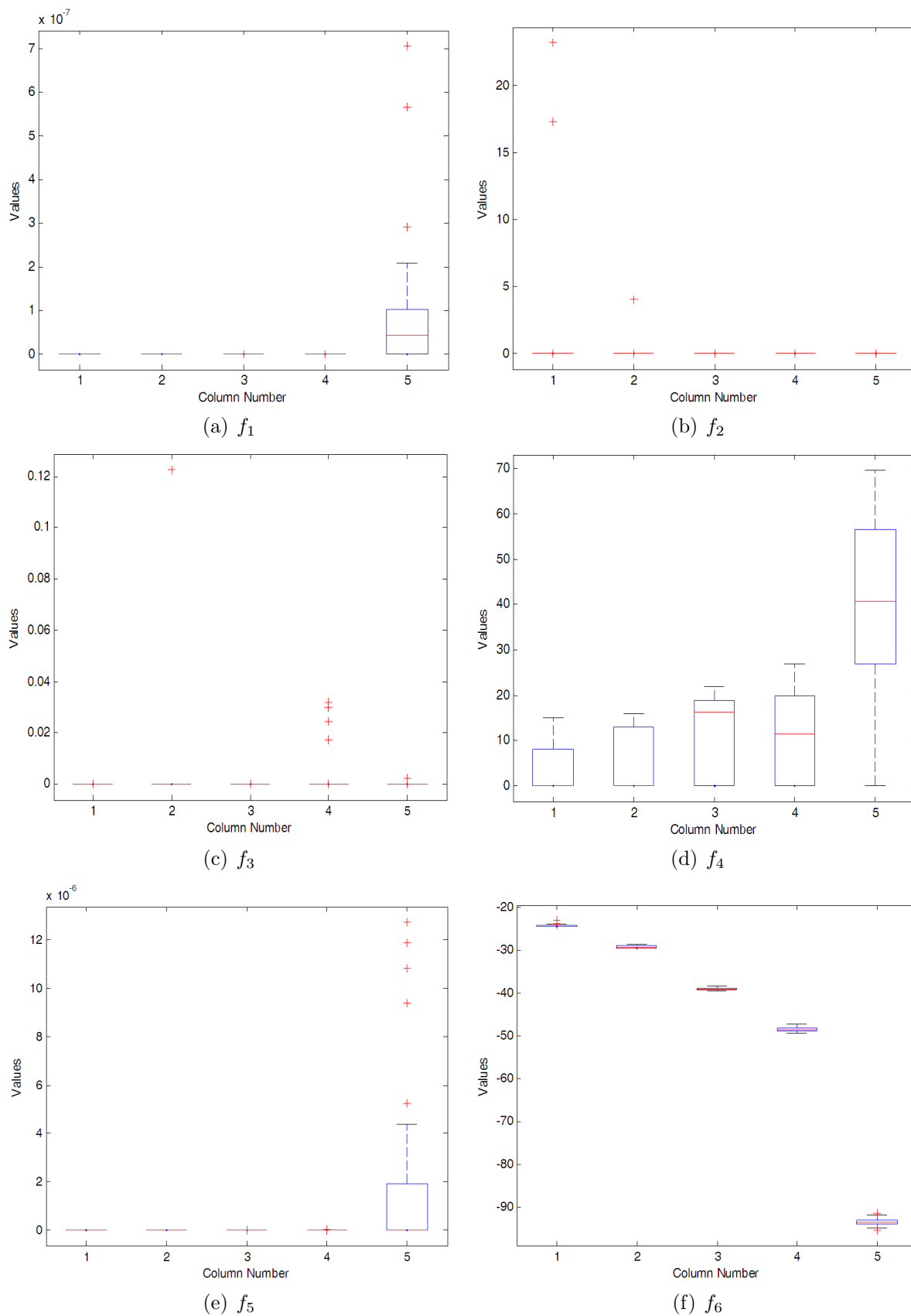


FIGURE 11. Box diagram of function

5. Conclusion. This paper proposed an improved bat algorithm (IBA) with reducing step-size local searching strategy and novel population moving mechanism. The IBA solved the problem of local optimization problem in bat algorithm and improved the search performance. The simulation results of typical functions show that IBA has better convergence velocity, searching precision and stability than BA. The algorithm is also better than dBA, ABC, PSO, CS, GCS, SSCS and RC-SSCS. It has a good performance for high-dimensional functions with different dimensions. In future, the algorithm could be extended to solve more real-world optimization problems and the novel population moving mechanism will be extended to other swarm intelligence algorithms.

Acknowledgment. This work is supported by National Science Foundation of China under Grant No. 61473054. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

- [1] D. Rodrigues, L. A. M. Pereira, R. Y. M. Nakamura et al., A wrapper approach for feature selection based on bat algorithm and optimum-path forest, *Expert Systems with Applications*, vol.41, no.5, pp.2250-2258, 2014.
- [2] X. Yang, Bat algorithm for multi-objective optimisation, *International Journal of Bio-Inspired Computation*, vol.3, no.5, pp.267-274, 2012.
- [3] X. S. Yang, A new metaheuristic bat-inspired algorithm, *Computer Knowledge & Technology*, vol.284, pp.65-74, 2010.
- [4] X. Yang and A. H. Gandomi, Bat algorithm: A novel approach for global engineering optimization, *Engineering Computations*, vol.29, no.5, pp.464-483, 2013.
- [5] L. R. Kumar, K. P. Padmanaban, S. G. Kumar et al., Design and optimization of concurrent tolerance in mechanical assemblies using bat algorithm, *Journal of Mechanical Science & Technology*, vol.30, no.6, pp.2601-2614, 2016.
- [6] M. A. Salam, Comparative study between FPA, BA, MCS, ABC, and PSO algorithms in training and optimizing of LS-SVM for stock market prediction, *International Journal of Advanced Research in Computer Science*, vol.5, no.18, pp.35-45, 2015.
- [7] K. Khan and A. Sahai, A comparison of BA, GA, PSO, BP and LM for training feed forward neural networks in e-learning context, *International Journal of Intelligent Systems & Applications*, vol.4, no.7, 2012.
- [8] A. H. Gandomi, X.-S. Yang, A. H. Alavi and S. Talatahari, Bat algorithm for constrained optimization tasks, *Neural Computing & Applications*, vol.22, no.6, pp.1239-1255, 2013.
- [9] M. R. Ramli, Z. A. Abas, M. I. Desa et al., Enhanced convergence of bat algorithm based on dimensional and inertia weight factor, *Journal of King Saud University – Computer and Information Sciences*, 2018.
- [10] I. Fister Jr., S. Fong, J. Brest et al., A novel hybrid self-adaptive bat algorithm, *The Scientific World Journal*, vol.2014, 2014.
- [11] S. Pravesjit, A hybrid bat algorithm with natural-inspired algorithms for continuous optimization problem, *Artificial Life & Robotics*, vol.21, no.1, pp.1-8, 2015.
- [12] S. Yilmaz and E. U. Küçükşille, A new modification approach on bat algorithm for solving optimization problems, *Applied Soft Computing*, vol.28, no.5, pp.259-275, 2015.
- [13] N. M. Nawi, M. Z. Rehman, A. Khan et al., A modified bat algorithm based on Gaussian distribution for solving optimization problem, *Journal of Computational & Theoretical Nanoscience*, vol.13, no.1, 2015.
- [14] X. S. He, W. J. Ding and X. S. Yang, Bat algorithm based on simulated annealing and Gaussian perturbations, *Neural Computing and Applications*, vol.25, no.2, pp.459-468, 2014.
- [15] Y. Zhou, Q. Luo, J. Xie et al., A hybrid bat algorithm with path relinking for the capacitated vehicle routing problem, *Mathematical Problems in Engineering*, vol.2013, no.3, pp.831-842, 2013.
- [16] Y. Zhou, J. Xie, L. Li et al., Cloud model bat algorithm, *The Scientific World Journal*, vol.2014, no.4, 2014.
- [17] B. Bahmani-Firouzi and R. Azizipanah-Abarghooee, Optimal sizing of battery energy storage for micro-grid operation management using a new improved bat algorithm, *International Journal of Electrical Power & Energy Systems*, vol.56, no.3, pp.42-54, 2014.

- [18] A. Alihodzic and M. Tuba, Improved bat algorithm applied to multilevel image thresholding, *The Scientific World Journal*, vol.2014, no.3, 2014.
- [19] M. L. Gao, J. Shen, L. J. Yin et al., A novel visual tracking method using bat algorithm, *Neurocomputing*, vol.177, pp.612-619, 2016.
- [20] I. Fister, S. Rauter, X. S. Yang et al., Planning the sports training sessions with the bat algorithm, *Neurocomputing*, vol.149, pp.993-1002, 2015.
- [21] E. Osaba, X. S. Yang, F. Diaz et al., An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems, *Engineering Applications of Artificial Intelligence*, vol.48, pp.59-71, 2016.
- [22] G. G. Wang, H. C. E. Chu and S. Mirjalili, Three-dimensional path planning for UCAV using an improved bat algorithm, *Aerospace Science & Technology*, vol.49, pp.231-238, 2016.
- [23] M. Tahar, BAT algorithm for cryptanalysis of feistel cryptosystems, *Biochimica Et Biophysica Acta*, vol.3, no.2, p.82, 2015.
- [24] K. S. Kumar and S. Naveen, Power system reconfiguration and loss minimization for a distribution systems using "Catfish PSO" algorithm, *Frontiers in Energy*, vol.8, no.4, pp.434-442, 2014.
- [25] L. Zhang, Y. Xu, G. Xu and S. Gong, A catfish effect inspired harmony search algorithm for optimization, *International Journal of Nonlinear Sciences and Numerical Simulation*, vol.14, no.6, pp.413-422, 2016.
- [26] X. S. Yang and S. Deb, Cuckoo search via Lévy flights, *Mathematics*, pp.210-214, 2010.
- [27] Q. Qin, S. Cheng, Q. Zhang et al., Multiple strategies based orthogonal design particle swarm optimizer for numerical optimization, *Computers & Operations Research*, vol.60, pp.91-110, 2015.
- [28] V. K. Kamboj, S. K. Bath and J. S. Dhillon, Implementation of hybrid harmony search/random search algorithm for single area unit commitment problem, *International Journal of Electrical Power & Energy Systems*, vol.77, pp.228-249, 2016.
- [29] A. Chakri, R. Khelif, M. Benouaret et al., New directional bat algorithm for continuous optimization problems, *Expert Systems with Applications*, vol.69, pp.159-175, 2017.
- [30] J. S. Wang, S. X. Li and J. D. Song, Cuckoo search algorithm based on repeat-cycle asymptotic self-learning and self-evolving disturbance for function optimization, *Computational Intelligence & Neuroscience*, 2015.