

Research Article

Improved Bat Algorithm Based on Multipopulation Strategy of Island Model for Solving Global Function Optimization Problem

Sha-Sha Guo,¹ Jie-Sheng Wang ^{1,2} and Xiao-Xu Ma¹

¹School of Electronic and Information Engineering, University of Science & Technology Liaoning, Anshan 114044, China

²School of International Finance and Banking, University of Science & Technology Liaoning, Anshan 114044, China

Correspondence should be addressed to Jie-Sheng Wang; wang_jiesheng@126.com

Received 24 February 2019; Revised 3 July 2019; Accepted 8 July 2019; Published 18 August 2019

Academic Editor: Amparo Alonso-Betanzos

Copyright © 2019 Sha-Sha Guo et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The bat algorithm (BA) is a heuristic algorithm that globally optimizes by simulating the bat echolocation behavior. In order to improve the search performance and further improve the convergence speed and optimization precision of the bat algorithm, an improved algorithm based on chaotic map is introduced, and the improved bat algorithm of Levy flight search strategy and contraction factor is proposed. The optimal chaotic map operator is selected based on the simulation experiments results. Then, a multipopulation parallel bat algorithm based on the island model is proposed. Finally, the typical test functions are used to carry out the simulation experiments. The simulation results show that the proposed improved algorithm can effectively improve the convergence speed and optimization accuracy.

1. Introduction

Optimization is the selection of the best elements for a particular set of criteria from a range of effective choices, which shows many different advantages and disadvantages in terms of computational efficiency and global optimization probability, but has a wide range of applications in industry and scientific research [1]. Function optimization proposes a formal framework for modeling and solving a series of specific problems, giving an “objective” function that takes a parameter as input, and the goal is to find the value of the combined parameter to return the “best” value. This framework is abstract enough that various problems can be interpreted as “function optimization” problems [2]. However, traditional function optimization is only used to solve some small-scale problems, which are often not applicable in practice. Therefore, people put their eyes on nature, which provides rich models to solve these problems (such as fireflies, bats, and ants). By simulating the natural biological systems, many swarm intelligent optimization algorithms are proposed to solve the application problems with nontraditional methods [3]. Many swarm intelligent optimization algorithms have been proposed, such as particle swarm optimization

(PSO) algorithm [4], ant colony optimization (ACO) algorithm [5], bat algorithm (BA) [6], social learning optimization (SLO) algorithm [7], chicken swarm optimization (CSO) algorithm [8], firefly algorithm (FA) [9], and so on.

The bat algorithm (BA) is a heuristic search algorithm proposed by Professor Yang in 2010 based on swarm intelligence [10]. The BA algorithm has many advantages, such as simplicity, fewer parameters, robustness, ease of implementation, and so on. Therefore, due to its obvious superiority, BA has been applied in various application fields, such as the optimal independent micro-smart grid, multi-objective function optimization based on artificial neural network model, economic scheduling problem, economic load scheduling for wind power generation system, fault diagnosis on low-speed rolling bearing, and the optimization of echo state network [11–15]. However, some research reports show that with the increase of problem dimensions, its performance may decline and its exploration capacity may become poor, so it is almost impossible to converge to the global optimal solution. In order to overcome this shortcoming, many scholars have studied and applied this algorithm and proposed corresponding improvement strategies.

An elite crossover binary bat algorithm proposed in [16], the algorithm using the cross mechanism in the elite strategy and genetic algorithm, according to the certain proportion selection in the bat group of elite individuals to crossover, will be a child of the merit of bats group and the parent group of mixed, guarantee the diversity of bats and good sex, and improve the global search ability. Zhou et al. [17] applied the cloud model to bat algorithm. Starting from the living and predation characteristics of bats, the echolocation model is reconstructed, and the transformation theory of cloud model is used to describe the qualitative concept: bats are close to prey. Fister et al. [18] developed the hybrid self-adaptive bat algorithm (HSABA) on the basis of self-adaptive bat algorithm (SABA). The HSABA was combined with the local search heuristic algorithm. The hybrid algorithm was crossbred with different DE strategies and used as the local search heuristic algorithm to improve the current optimal solution, and the solution group was pointed to a better area in the search space. Qu et al. [19] integrated the weed growth, reproduction, spatial diffusion, and competition mechanism of the invasive weed algorithm into the bat algorithm and dynamically adjusts the standard deviation of the weed spatial diffusion operator so that the algorithm not only increases the global search ability but also improves the local search ability. Wu et al. [20] proposed an elite crossover binary bat algorithm. Based on the crossover mechanism of elite strategy and genetic algorithm, this algorithm selects elite individuals in the bat group to cross over according to a certain proportion and then makes a mixed selection between the daughter bat group and parent bat group so as to ensure the diversity and excellence of the bat group and improve the global search ability.

In this paper, a multipopulation parallel bat algorithm based on island model is proposed and applied to three strategies including chaos [21], Levy flight search [22], and contraction factor. Simulation results show that the proposed algorithm is more effective than single chaos, Levy flight search, and contraction factor in terms of convergence speed and accuracy.

2. Bat Algorithm

2.1. Basic Principle of Bat Algorithm. The bat algorithm (BA) is a new swarm intelligence optimization algorithm, which simulates the foraging behavior of bats. Its principle is to use the bat's advanced echolocation capability [10]. Echolocation is a kind of sonar: the bat (the main small bat) emits a loud and short pulse sound. When the sound hits an object, the echo will return to their ears in a short period of time; bats receive and detect the position of the prey in this way. In order to simulate the foraging process of bats, the biological mechanism of the bat algorithm is described as follows. All bats adopt echolocation to detect distances, and the method used to identify obstacles and prey is difficult to understand. Based on the variable length waves λ , loudness A_0 , and fixed frequency f_{\min} , the bat searches for the prey with the velocity V_i at the position X_i . The bat adjusts the pulse wavelength

according to the distance between itself and the prey. On the other hand, when it is close to the prey, the frequency of the transmission $r \in (0, 1)$ will also be adjusted. The loudness changes from the maximum value A_0 to the minimum value A_{\min} in the searching process.

BA's development takes advantage of existing algorithms and other interesting features, inspired by the wonderful behavior of miniature bat echolocation. Based on these assumptions, this algorithm generates a set of solutions in a random manner and then uses the loop search to find the optimal solution. During this period, the local search is adopted. That is to say that around the optimal solution, the local solution is generated by random flight and produces a global optimal solution. For bats, if their foraging space belongs to the d -dimension at the $t - 1$ moment, the position of the bat i is X_i^{t-1} , the flight velocity is V_i^{t-1} , and the current global optimal position is X^* . So, the position and flight velocity of bat i at time t can be updated by

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

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X^*)f_i, \quad (2)$$

$$X_i^t = X_i^{t-1} + V_i^t, \quad (3)$$

where the minimum frequency of the sound waves generated by the bat is f_{\min} and the maximum frequency is f_{\max} . β is a uniformly distributed random number located in the scope $[0, 1]$.

In the initial setting process, the frequency of the bat's emitted sound waves is uniformly distributed in $[f_{\min}, f_{\max}]$. The corresponding frequency is obtained according to equation (1), and then the local search is carried out according to equations (2) and (3). The bat randomly walks according to the optimal solution, and the new solution is generated by the following equation:

$$X_{\text{new}} = X_{\text{old}} + \varepsilon \bar{A}(t), \quad (4)$$

where ε is a random number located in $[-1, 1]$, X_{old} represents the solution selected from the current optimal solutions in a random manner, and $\bar{A}(t)$ refers to the average loudness that the bat produces when the number of iterations is t .

By analyzing the loudness A_i and rate r_i of the bat pulse emission, it is found that the update rule can be described as follows. If the bat is aware of the presence of the prey, it will reduce the response of its pulsed emission and increase its pulse emission rate. The loudness A_i and rate r_i of the bat launch pulse are updated by the following equations:

$$r_i^{A_i^{t+1}} = \alpha A_i^t, \quad (5)$$

$$r_i^{t+1} = r_i^t [1 + \exp \gamma t], \quad (6)$$

where r_i^0 is the initial rate and A_i^0 is the initial loudness, which are all randomly chosen. α and γ are constants ($0 < \alpha < 1$, $\gamma > 0$).

TABLE 1: Simulation test functions.

Function	Expression	Range	Minimum value
F_1	$f_1(x) = \sum_{i=1}^d x_i^2$	[-100, 100]	0
F_2	$f_2(x) = \sum_{i=1}^n (x_i^2/4000) - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	[-600, 600]	0
F_3	$f_3(x) = 20 + e - 20e^{(-0.2\sqrt{1/n}\sum_{i=1}^n x_i^2)} - e^{((1/m)\sum_{i=1}^n \cos(cx_i))}$	[-32, 32]	0
F_4	$f_4(x) = -\sum_{i=1}^d \sin(x_i) [\sin(ix_i^2/\pi)]^{2m}, m = 10$	[0, π]	0
F_5	$f_5(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.12, 5.12]	0
F_6	$f_6(x) = \sum_{i=1}^d \sum_{j=1}^i x_j^2$	[-65.536, 65.536]	0
F_7	$f_7(x) = (x_1 - 1)^2 + \sum_{i=2}^d i(2x_i^2 - x_{i-1})^2$	[-5, 10]	0
F_8	$f_8(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^n u(x_i, 5, 100, 4)u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	[-50, 50]	0
F_9	$f_9(x) = \sum_{i=1}^{11} [a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}]^2$	[-5, 5]	0.00030
F_{10}	$f_{10}(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij}(x_j - p_{ij})^2)$	[0, 1]	-3.32
F_{11}	$f_{11}(x) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	[0, 10]	-10.536

2.2. *Pseudocode of the Bat Algorithm.* The pseudocode of the bat algorithm is described as follows [18]:

Input: Bat population $x_i = (x_{i1}, \dots, x_{iD})^T$ for $i = 1, \dots, \text{Np}, \text{MAX_FE}$.

Output: The best population x_{best} and its corresponding value $f_{\text{min}} = \min(f(x))$.

```

init_bat ();
eval = evaluate the new population;
f_min = find_best_solution (x_best); {initialization}
while termination_condition_not_meet do
  for i = 1 to Np do
    y = generate_new_solution (x_i);
  if rand (0, 1) > r then
    y = improve_the_best_solution (x_i);
  end if {local search}
  if f_new = evaluate_new_solution (y);
  eval = eval + 1;
  if f_new ≤ f_i and N (0, 1) < A then
    x_i = y; f_i = f_new
  end if {simulated annealing}
  f_min = find the best solution (x_best);
end for
end while

```

3. Improved Bat Algorithm

3.1. *Bat Algorithm Based on Chaotic Mapping.* The chaotic motion is a highly unstable motion in a deterministic system that is limited to a finite phase space. Chaos is a form of aperiodic motion, which is unique and extensive in nonlinear systems [23]. Chaotic systems are very common in natural systems and social systems, which has a complex, random, and accurate characteristic [24]. By analyzing the

logistic equation, it is known as the most typical of chaotic systems:

$$S_{k+1} = \mu S_k (1 - S_k), \quad (7)$$

where μ is a constant and $S \in (0, 1)$. So, the determined sequence S_0, S_1, \dots , can be obtained, and the system is a chaotic system.

There are 10 typical kinds of chaotic mapping [25]. An optimal chaotic mapping method is selected by carrying out the simulation experiments on the test functions. In order to verify the performance of various improved algorithms, a total of seven functions were selected for simulation experiments. The name, expression, and domain range of these functions are shown in Table 1.

In this section, the first six functions are adopted to carry out the simulation experiments. The simulation results of BA based on the different chaotic mappings is shown in Figure 1, and the performance comparison results are listed in Table 2. By considering the fluctuating feature of the chaotic mapping and the influence of the initial value, the initial point in all chaotic mappings is set as 0.7.

It can be seen from the simulation results on six function optimization problems that the convergence speed and optimization ability of the piecewise chaotic mapping are the best, the optimal value can be found, and the fluctuation is relatively small. Although the optimal value is not obtained in function F_6 , the overall trend deviation is not large. So, a chaotic bat algorithm (CBA) is proposed based on the piecewise chaotic mapping shown in Table 3.

It can be seen from the simulation results on six function optimization problems that the convergence speed and optimization ability of the piecewise chaotic mapping are the best, the optimal value can be found, and the fluctuation is relatively small. Although the optimal value is not obtained in functions F_1 and F_6 , the overall trend deviation is not large. So, a chaotic bat algorithm (CBA) is proposed based on the piecewise chaotic mapping shown in Table 3.

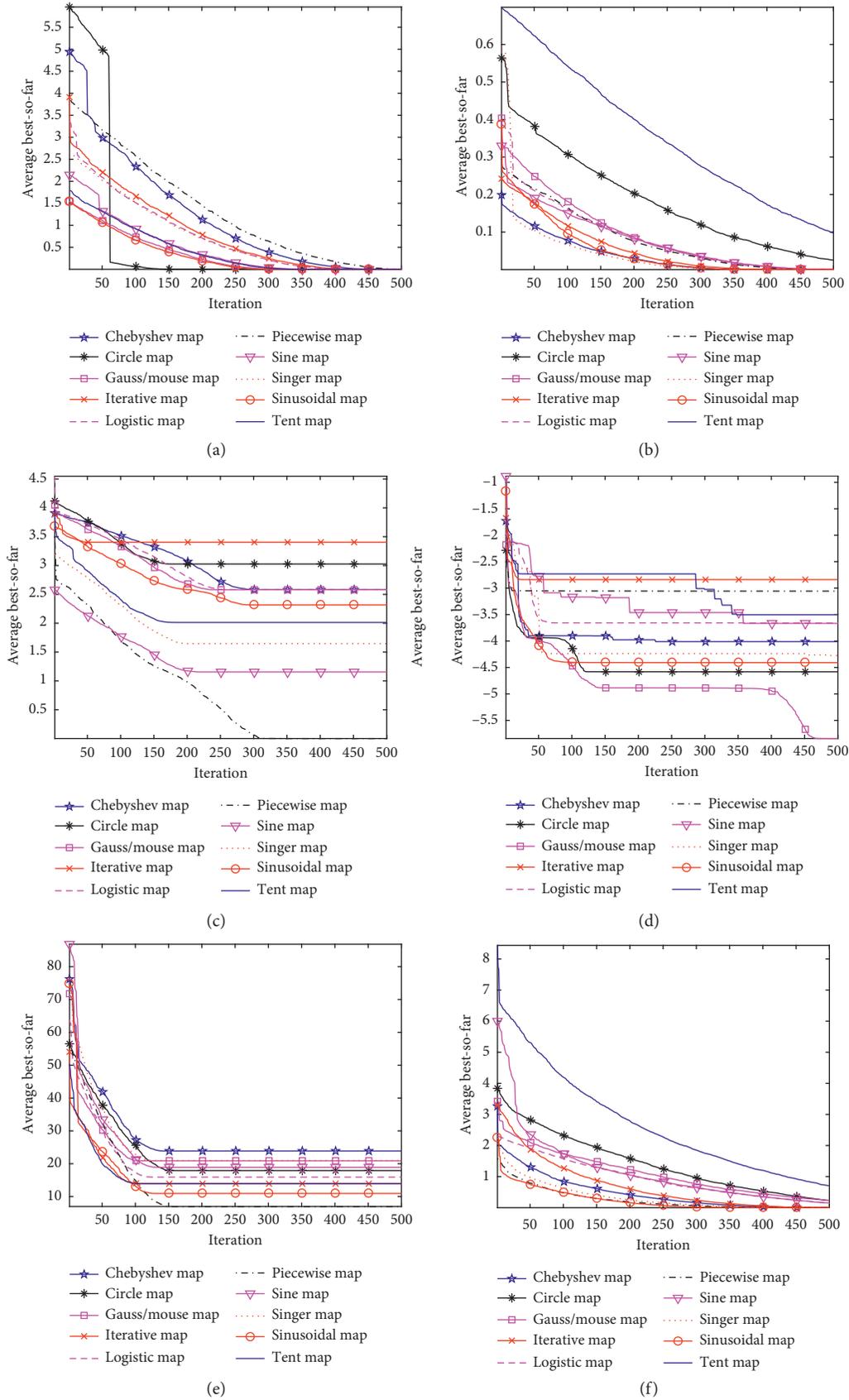


FIGURE 1: Convergence curves under ten chaotic mappings. (a) F_1 . (b) F_2 . (c) F_3 . (d) F_4 . (e) F_5 . (f) F_6 .

TABLE 2: Simulation results on function optimization problems under different chaotic mappings.

Function	Chaotic mapping	Avg.	Best	Std.
F_1	Chebyshev map	2.46E-03	6.07E-07	7.37E-03
	Circle map	4.08E-03	9.77E-07	1.12E-02
	Gauss/mouse map	1.77E-04	6.33E-07	5.25E-04
	Iterative map	1.92E-02	1.22E-06	5.76E-02
	Logistic map	1.74E-06	1.10E-06	5.28E-07
	Piecewise map	1.91E-06	5.70E-07	7.10E-07
	Sine map	1.60E-02	9.53E-07	4.76E-02
	Singer map	3.38E-02	1.14E-06	1.01E-01
	Sinusoidal map	8.43E-03	1.07E-06	2.53E-02
	Tent map	2.25E-06	1.21E-06	7.34E-07
F_2	Chebyshev map	1.22E-03	1.67E-07	3.01E-03
	Circle map	2.06E-03	1.27E-07	5.76E-03
	Gauss/mouse map	6.44E-03	1.28E-07	1.47E-02
	Iterative map	3.04E-03	1.28E-07	6.60E-03
	Logistic map	7.70E-05	8.32E-08	1.66E-04
	Piecewise map	2.28E-06	1.33E-07	6.04E-06
	Sine map	3.75E-03	4.52E-08	1.12E-02
	Singer map	8.91E-03	1.16E-07	1.33E-02
	Sinusoidal map	4.62E-03	2.06E-07	9.24E-03
	Tent map	9.79E-04	8.08E-08	2.58E-03
F_3	Chebyshev map	2.11E+00	1.16E+00	5.77E-01
	Circle map	2.13E+00	1.16E+00	9.04E-01
	Gauss/mouse map	2.01E+00	1.42E-03	7.21E-01
	Iterative map	2.24E+00	1.16E+00	7.99E-01
	Logistic map	2.23E+00	1.91E-03	9.84E-01
	Piecewise map	1.89E+00	1.59E-03	6.80E-01
	Sine map	2.41E+00	1.16E+00	6.13E-01
	Singer map	1.95E+00	1.64E-03	7.85E-01
	Sinusoidal map	2.22E+00	1.76E-03	9.29E-01
	Tent map	1.90E+00	1.31E-03	7.28E-01
F_4	Chebyshev map	-4.3381	-3.4831	0.4727
	Circle map	-4.2815	-2.8800	0.9298
	Gauss/mouse map	-4.4270	-3.3986	0.7517
	Iterative map	-4.3976	-3.2347	0.7557
	Logistic map	-4.6527	-3.5362	0.9810
	Piecewise map	-3.6912	-3.3934	0.1789
	Sine map	-4.1871	-3.3551	0.5061
	Singer map	-3.9540	-3.2770	0.5729
	Sinusoidal map	-4.1621	-2.6385	0.7561
	Tent map	-4.0751	-2.6231	0.9906
F_5	Chebyshev map	12.3377	6.9650	4.9388
	Circle map	14.6261	6.9650	5.0547
	Gauss/mouse map	13.4322	5.9700	5.1170
	Iterative map	11.6412	4.9750	4.6038
	Logistic map	12.0392	3.9801	4.3472
	Piecewise map	9.6514	6.9649	1.6074
	Sine map	13.1337	4.9750	5.5895
	Singer map	14.0292	7.9600	3.9410
	Sinusoidal map	13.0342	6.9649	6.4242
	Tent map	13.5317	6.9650	5.2686
F_6	Chebyshev map	2.64E-01	3.69E-06	2.98E-01
	Circle map	5.17E-01	1.08E-05	8.15E-01
	Gauss/mouse map	1.84E-01	4.49E-05	1.45E-01
	Iterative map	1.92E-01	1.64E-05	1.69E-01
	Logistic map	3.22E-01	2.76E-03	3.84E-01
	Piecewise map	1.49E-01	6.80E-06	1.90E-01
	Sine map	1.75E-01	2.25E-06	2.58E-01
	Singer map	2.17E-01	2.46E-06	2.55E-01
	Sinusoidal map	1.19E-01	2.12E-06	1.72E-01
	Tent map	2.02E-01	7.22E-06	2.21E-01

TABLE 3: Expression of piecewise map.

Chaotic mapping	Expression	Range
Piecewise map	$x_{i+1} = \begin{cases} x_i/P & [0, P) \\ x_i - P/0.5 - P & [P, 0.5) \\ 1 - P - x_i/0.5 - P & [0.5, 1 - P) \\ 1 - x_i/P & [1 - P, 1) \\ P = 0.4 \end{cases}$	(0, 1)

3.2. *Bat Algorithm Based on Levy Flight Search Strategy.* The study found that Levy's flight behavior in nature is based on the ideal way for food seekers to find food in an unfocused and unpredictable environment, which includes short-range exploratory bounce and occasional long walks. Viswanathan et al. [26] studied the foraging behavior of the albatross and found the same flight path as the Levy flight. Reynolds et al. [27, 28] observed the foraging trajectory of bees and fruit flies and found that the flight trajectory also shows the characteristics of Levy's flight, and even human behavior is similar to the existence of Levy's flight behavior. A series of studies have confirmed that Levy's flight behavior is the best searching strategy for N independent explorers when the target position is in a random state and the distribution is relatively loose. A Levy flight-based bat algorithm (LBA) is proposed in this paper.

From a mathematical point of view, Levy flight behavior reflects a class of non-Gaussian stochastic processes, whose steady increments obey the stable distribution of Levy and whose flight path simulation is shown in Figure 2. It can be seen from Figure 2 that it can jump a lot in Levy's searching process and change direction many times, so it can make an individual bat effectively avoid being bound by local attractions and expand the searching space.

Combined with the bat's echolocation feature, it helps to significantly and effectively improve the performance of the bat algorithm. Therefore, the improved algorithm replaces equation (3) with the following equation:

$$X_i^t = X_i^{t-1} + \text{levy} X(X_i^{t-1} - X^*) + V_i^t. \quad (8)$$

The Levy flight is used to replace the local searching for the optimal position of an individual bat, which generates a larger matching and optimization iterations in the global search process so as to make the situation of falling into local optimum improve and also make the convergence accuracy of the algorithm improve.

3.3. *Bat Algorithm Based on Shrink Factor.* For the bat algorithm, since the individual is flying toward the optimal solution during optimization, there is usually a problem of early convergence of the bats, so it is difficult to obtain better optimization results. In order to avoid the premature convergence problem of bat algorithm and make the individual converge to the global optimal solution quickly, a contraction factor is proposed to realize the shrink factor bat algorithm (SBA), which not only maintains the diversity of the population but also improves the convergence efficiency.

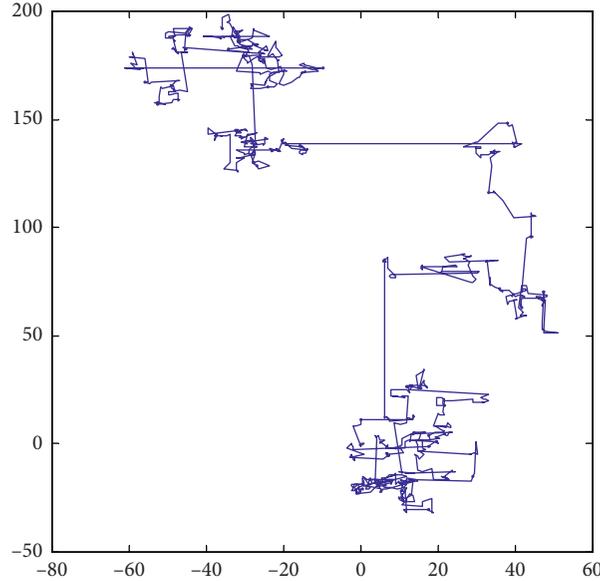


FIGURE 2: Levy flight tracks.

The proposed SBA can be realized by the following equations:

$$k = \frac{2}{|2 - d - \sqrt{d^2 - 4d}|}, \quad (9)$$

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X_*) f_i \cdot k, \quad (10)$$

where d is a constant, and the value of the d in this experiment is 10.

3.4. Multipopulation Parallel Bat Algorithm Based on Island Model. The algorithm based on the parallel model has the following two characteristics. The first is to break down a group into multiple groups by realizing a divide-and-conquer approach. The second is to control and manage the information exchange among the subgroups. From the perspective of parallel algorithms, this structural difference produces three parallel population models: master-slave parallel model, island model, and adjacency model [29]. The adjacency model and the island model belong to the decomposition parallel scheme, which divides the entire group into several subgroups. Each subgroup is distributed on its own processor for subgroup evolution, and each processor exchanges information at the appropriate time. The adjacency model is also known as a fine-grained model, with only one individual in each subgroup. The island model, also known as the coarse-grained model, has multiple subgroups on each processor. The coarse-grained models are easy to implement and can be simulated on networks or stand-alone systems without parallel computers, so coarse-grained models are the most commonly used in parallel algorithms [30].

In this paper, an island multipopulation parallel bat algorithm (IBA) is proposed by adopting the parallel optimization scheme and introducing a centralized information

migration strategy. The entire population is divided into many subgroups. Each subgroup performs a global search only on the island, and the suitability of each individual in the island is calculated and evaluated to produce the best individual in the island. The entire evolution of the island is realized by using a separate subprocess to reduce the degree of coupling. Each subprocess uses a centralized migration strategy to periodically send the best individual in the island to the main process to form the main process, and the main process selects the global best individual to from the entire population so as to broadcast to the subprocess, which will force the subpopulation to perform the global most excellent evolution. The flow chart of the algorithm is shown in Figure 3.

4. Simulation Experiment and Result Analysis

The multipopulation idea based on the island model is integrated into the other three improved bat algorithms (SBA, LBA, and CBA) to form the island multipopulation chaotic bat algorithm (CBAS), the island multipopulation Levy flight bat algorithm (LBAS), and the island multipopulation shrink factor bat algorithm (SBAS). The seven algorithms (IBA, SBA, LBA, CBA, CBAS, LBAS, and SBAS) were adopted to carry out the simulation experiments on twelve typical test functions ($F_1, F_2, F_3, F_5, F_6, F_7, F_8, F_9, F_{10}$, and F_{11}) shown in Table 1. The performance of the algorithms was evaluated by counting the optimal value, the average values, and the convergence curves of the corresponding functions in 10 runs. The parameter settings of the algorithm are shown in Table 4.

The performance comparison results are listed in Table 5, and the function convergence curves are shown in Figure 4.

It can be seen from the simulation results that the overall searching ability of IBA, SBAS, LBAS, and CBAS is better than that of the original algorithms (BA, SBA, LBA, and CBA), which have less volatility and relatively stable performance.

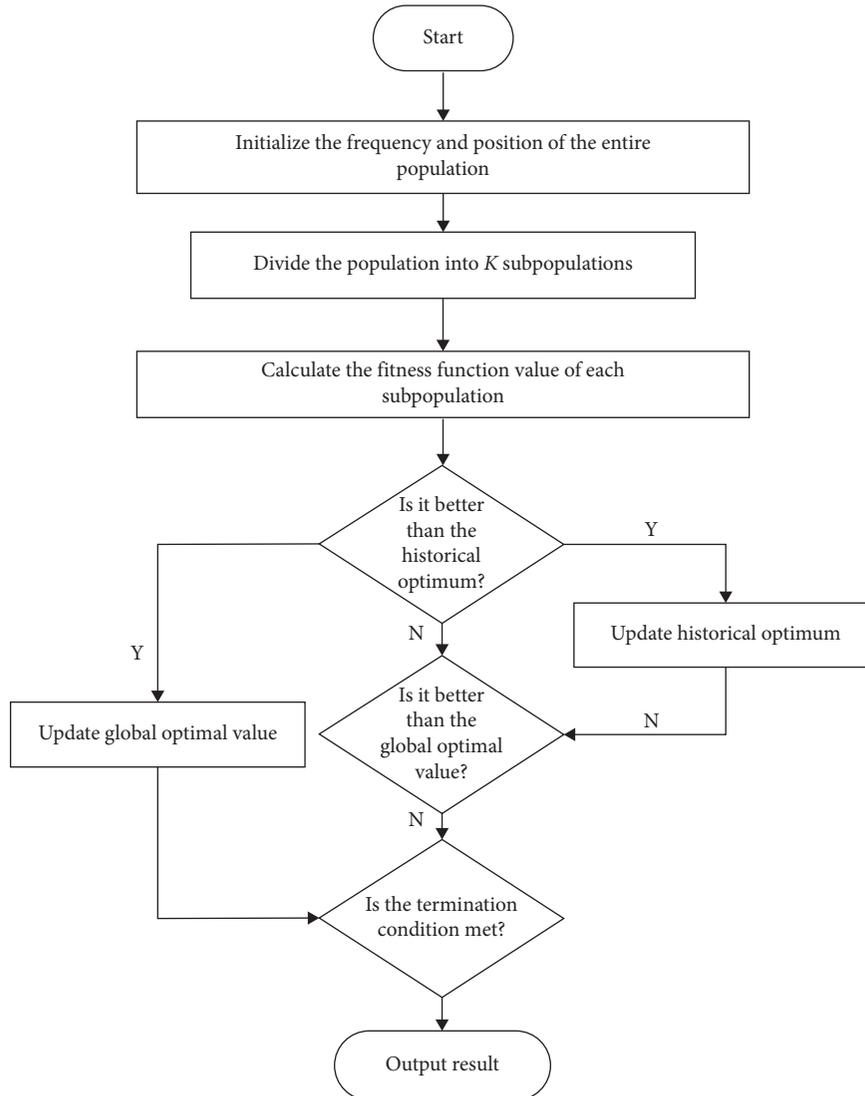


FIGURE 3: Bat algorithm flow chart based on island model.

TABLE 4: Parameter settings of the algorithm.

Name of parameter	Parameter values
Population size	$n = 30$
Maximum number of iterations	$\text{Max}T = 500$
Loudness	$A = 0.5$
Rate	$r = 0.5$
Maximum frequency	$f_{\max} = 2$
Minimum frequency	$f_{\min} = 0$

From the perspective of a single function, most algorithms corresponding to each function are different, which proves that the seven different algorithms have different optimization capabilities for the same problem. SBAS obtains the optimal value for test functions F_1 , F_6 , and F_7 . IBA obtains the optimal value for test function F_3 . LBA obtains the optimal value for functions F_5 and F_2 . All seven optimization algorithms get the global optimal value for test function F_3 . It can be seen from the comparison of the convergence curves of the six test functions that overall, the convergence speed of the

multipopulation algorithms (IBA, SBAS, and LBAS) is faster than that of the original algorithms. However, the multipopulation algorithm CBAS is slower than the original algorithms. For the test functions F_1 and F_2 , SBAS converges faster. For test functions F_6 and F_7 , LBAS converges faster. For function F_3 , CBAS converges faster. For function F_5 , IBA algorithm converges faster. The convergence rate of the function is relatively larger at an early stage, and there is a delay phenomenon in the later stage although the speed is slowed down.

It can be seen from the simulation results that the overall searching ability of IBA, SBAS, LBAS, and CBAS is better than that of the original algorithms (BA, SBA, LBA, and CBA), which have less volatility and relatively stable performance. From the perspective of a single function, most algorithms corresponding to each function are different, which proves that the seven different algorithms have different optimization capabilities for the same problem. SBAS obtains the optimal value for test functions F_2 , F_5 , F_9 , F_{12} , and F_{13} . SBA obtains the optimal value for test functions F_3

TABLE 5: Performance comparison results under seven algorithms.

Function	Optimization method	Optimal solution	Average	Standard deviation
F_1	IBA	$3.4364E-08$	$3.3842E-01$	$6.7955E-01$
	SBA	$3.0580E-09$	$3.0762E-08$	$3.0405E-08$
	LBA	$4.4310E-08$	$1.0789E-07$	$5.6368E-08$
	CBA	$2.9439E-20$	$1.3670E-15$	$3.1453E-15$
	SBAS	$4.2826E-09$	$9.0180E-03$	$2.7054E-02$
	LBAS	$9.1300E-09$	$4.8765E-08$	$2.9546E-08$
	CBAS	$3.3146E-09$	$2.0426E-08$	$1.4284E-08$
F_2	IBA	$6.6230E-08$	$5.9714E-02$	$1.7914E-01$
	SBA	$7.3645E-08$	$8.5002E-03$	$1.2582E-02$
	LBA	$2.1114E-07$	$9.9413E-03$	$2.4116E-02$
	CBA	$2.2553E-08$	$1.9696E-08$	$1.3041E-08$
	SBAS	$3.3806E-08$	$6.1948E-08$	$1.8490E-08$
	LBAS	$6.3657E-08$	$1.0919E-07$	$2.4972E-08$
	CBAS	$1.8559E-08$	$6.6582E-02$	$1.9975E-01$
F_3	IBA	2.3169	2.9481	0.4250
	SBA	0.0009	1.9313	0.8860
	LBA	1.1552	2.1514	0.6486
	CBA	1.1551	2.3774	0.5599
	SBAS	2.0133	2.3912	0.3715
	LBAS	2.3169	2.8544	0.4626
	CBAS	1.6462	2.7592	0.7613
F_5	IBA	5.9699	15.5719	13.7849
	SBA	3.9899	9.9497	5.6459
	LBA	7.9598	13.4321	3.3074
	CBA	3.9798	12.8350	6.8421
	SBAS	3.9800	7.2633	2.3566
	LBAS	5.9699	8.5568	1.7909
	CBAS	4.9748	19.5561	16.6536
F_6	IBA	$3.8782E-08$	$4.4456E-01$	$9.5610E-01$
	SBA	$5.8307E-09$	$1.1123E-04$	$2.9158E-04$
	LBA	$8.6362E-08$	$1.1877E-02$	$2.3393E-02$
	CBA	$1.0664E-19$	$3.4219E-17$	$6.0237E-17$
	SBAS	$1.3461E-08$	$5.0505E-08$	$2.2872E-08$
	LBAS	$2.8347E-08$	$1.7624E-01$	$5.2873E-01$
	CBAS	$3.2492E-09$	$2.5676E-08$	$1.5720E-08$
F_7	IBA	-0.0916	-0.8837	0.3961
	SBA	-1.0833	-1.0833	0.0000
	LBA	-1.0833	-1.0833	0.0000
	CBA	-1.0833	-1.0833	0.0000
	SBAS	-0.6009	-1.0315	0.2153
	LBAS	1.0416	-0.7575	0.8996
	CBAS	0.2604	-0.8475	0.5539
F_8	IBA	$3.8153E-09$	$1.3534E-01$	$1.2480E-01$
	SBA	$1.8240E-10$	$2.9037E-02$	$5.6482E-02$
	LBA	$1.3140E-08$	$2.0437E-02$	$2.1852E-02$
	CBA	$2.1564E-31$	$1.9632E-02$	$2.5663E-02$
	SBAS	$1.2837E-10$	$2.4524E-02$	$2.0517E-02$
	LBAS	$1.4992E-09$	$1.3792E-01$	$1.4911E-01$
	CBAS	$1.3498E-32$	$1.3242E-01$	$1.9408E-01$
F_9	IBA	0.0003	0.0090	0.0159
	SBA	0.0003	0.0016	0.0026
	LBA	0.0003	0.0033	0.0073
	CBA	0.0003	0.0006	0.0003
	SBAS	0.0003	0.0003	$5.6425E-05$
	LBAS	0.0005	0.0028	0.0029
	CBAS	0.0003	0.0007	0.0005

TABLE 5: Continued.

Function	Optimization method	Optimal solution	Average	Standard deviation
F_{10}	IBA	-3.3220	-3.0019	0.9602
	SBA	-3.3220	-2.8942	0.6939
	LBA	-3.3220	-3.0492	0.7143
	CBA	-3.2031	-3.2507	0.0582
	SBAS	-3.2031	-3.2863	0.0545
	LBAS	-3.2031	-3.2863	0.0545
	CBAS	-3.2031	-3.2982	0.0476
F_{11}	IBA	-5.1285	-4.2011	1.8551
	SBA	-5.1285	-5.1285	3.6134E-07
	LBA	-5.1285	-4.8311	0.8920
	CBA	-5.1285	-5.1285	4.5201E-15
	SBAS	-5.1285	-5.1285	1.5732E-07
	LBAS	-5.1285	-2.8510	1.9806
	CBAS	-5.1285	-5.1285	5.7559E-08

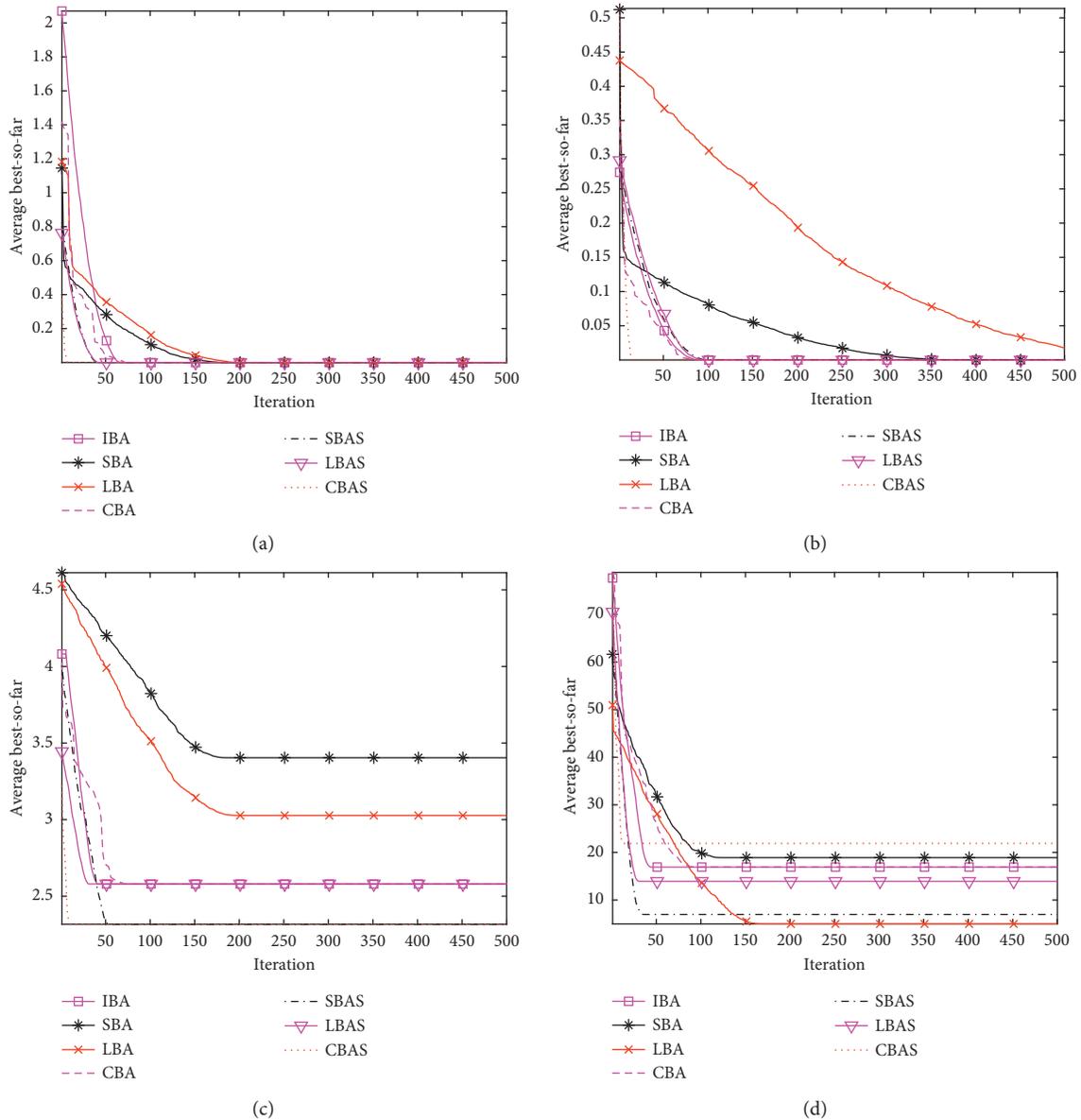
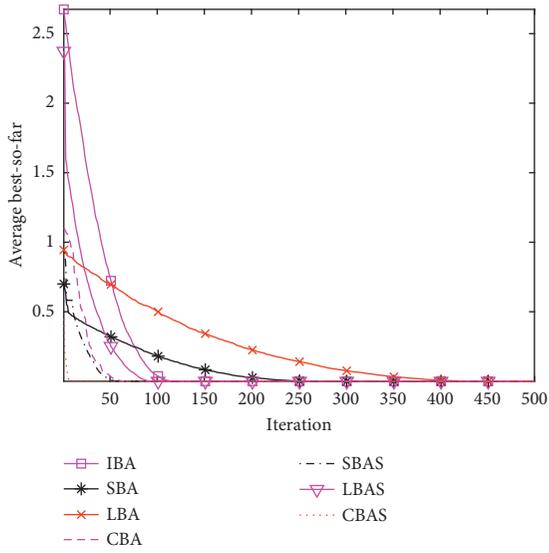
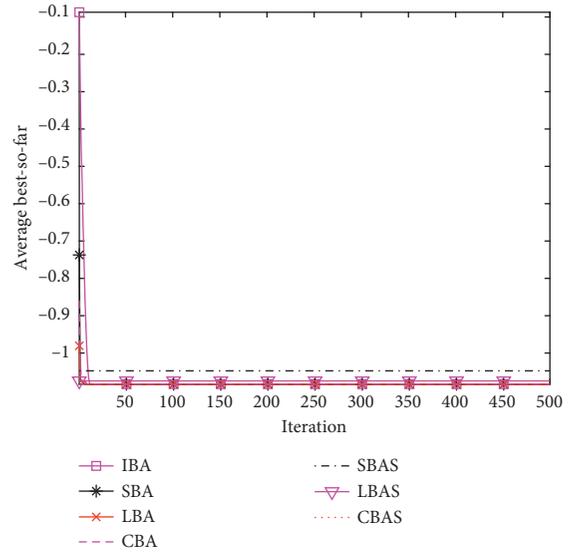


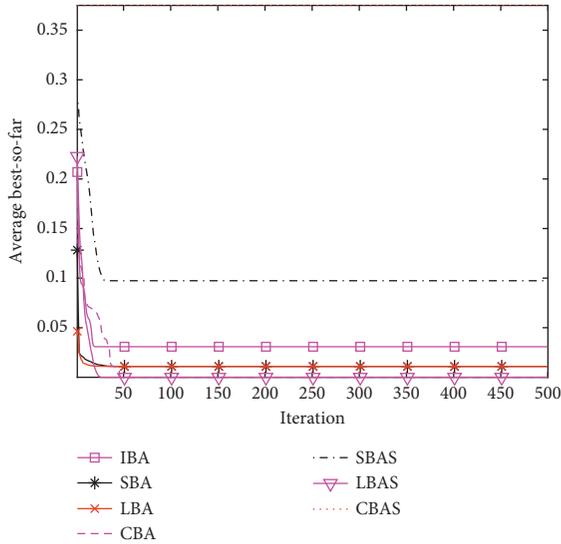
FIGURE 4: Continued.



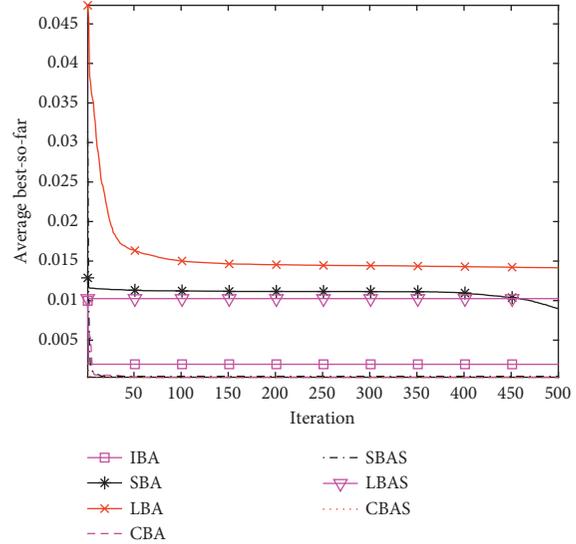
(e)



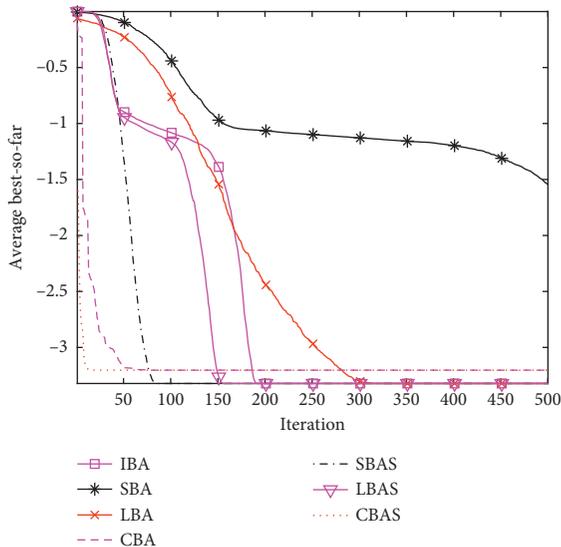
(f)



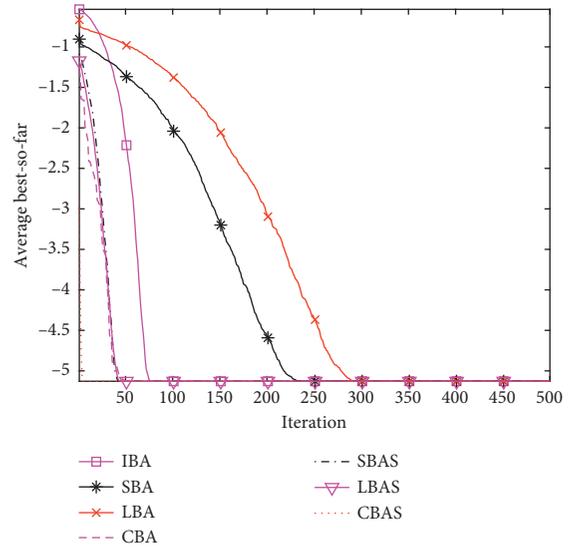
(g)



(h)



(i)



(j)

FIGURE 4: Convergence curves for typical function optimization problems under seven algorithms. (a) F_1 . (b) F_2 . (c) F_3 . (d) F_5 . (e) F_6 . (f) F_7 . (g) F_8 . (h) F_9 . (i) F_{10} . (j) F_{11} .

and F_8 . LBAS obtains the optimal value for test functions F_1 , F_7 , and F_{12} . It can be seen from the comparison of the convergence curves of the ten test functions that overall, the convergence speed of the multi-population algorithms (IBA, SBAS, and CBAS) is faster than that of the original algorithms. For the test functions F_5 and F_{10} , SBAS converges faster. For the test functions F_8 , LBAS converges faster. For the test functions F_1, F_2, F_3, F_6, F_9 , and F_{11} , CBAS converges faster. The convergence rate of the function is relatively larger at an early stage, and there is a delay phenomenon in the later stage although the speed is slowed down.

5. Conclusions

Based on the concept of chaotic mapping, the optimal chaotic mapping is selected to produce the chaotic bat algorithm. The LBA algorithm is proposed by adopting the Levy flight searching strategy. The shrink factor is proposed to realize SBA algorithm, and the multipopulation parallel bat algorithm (IBA) based on island is proposed. The performance is compared by using seven test functions under seven optimization algorithms (IBA, SBA, LBA, CBA, SBAS, LBAS, and CBAS). The comparison results show that the multipopulation algorithms (IBA, SBAS, LBAS, and CBAS) are better than the original algorithms and have relatively stable performance in both the optimization ability and the convergence speed. The experiment proves that the improved algorithm has better optimization precision and convergence speed, which can make up for the deficiency of the original algorithm.

Data Availability

There are no data available for this paper.

Conflicts of Interest

The authors declare no conflicts of interest.

Authors' Contributions

Sha-Sha Guo participated in the draft writing and critical revision of this paper; Jie-Sheng Wang participated in the concept, design, and interpretation of results and commented on the manuscript; and Xiao-Xu Ma participated in the data collection, analysis, and algorithm simulation.

Acknowledgments

This work was supported by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (grant no. 2017FWDF10) and the project by Liaoning Provincial Natural Science Foundation of China (grant no. 20180550700).

References

- [1] G.-G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang, "Chaotic krill herd algorithm," *Information Sciences*, vol. 274, no. 274, pp. 17–34, 2014.
- [2] N. Bell and B. J. Oommen, "A novel abstraction for swarm intelligence: particle field optimization," *Autonomous Agents and Multi-Agent Systems*, vol. 31, no. 2, pp. 362–385, 2017.
- [3] X.-S. Yang, S. Deb, S. Fong, X. He, and Y.-X. Zhao, "From swarm intelligence to metaheuristics: nature-inspired optimization algorithms," *Computer*, vol. 49, no. 9, pp. 52–59, 2016.
- [4] L. Wang, H. Geng, P. Liu et al., "Particle swarm optimization based dictionary learning for remote sensing big data," *Knowledge-Based Systems*, vol. 79, pp. 43–50, 2015.
- [5] Y. Gheraibia, K. Djafri, and H. Krimou, "Ant colony algorithm for automotive safety integrity level allocation," *Applied Intelligence*, vol. 48, no. 3, pp. 555–569, 2018.
- [6] B. R. Adarsh, T. Raghunathan, T. Jayabarathi, and X.-S. Yang, "Economic dispatch using chaotic bat algorithm," *Energy*, vol. 96, pp. 666–675, 2016.
- [7] F. Huang, X. Li, S. Zhang, and J. Zhang, "Harmonious genetic clustering," *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 199–214, 2018.
- [8] D. Wu, S. Xu, and F. Kong, "Convergence analysis and improvement of the chicken swarm optimization algorithm," *IEEE Access*, vol. 4, pp. 9400–9412, 2016.
- [9] L. Olatomiwa, S. Mekhilef, S. Shamshirband, K. Mohammadi, D. Petković, and C. Sudheer, "A support vector machine-firefly algorithm-based model for global solar radiation prediction," *Solar Energy*, vol. 115, pp. 632–644, 2015.
- [10] X.-S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, pp. 65–74, Springer, Berlin, Germany, 2010.
- [11] N. S. Jaddi, S. Abdullah, and A. R. Hamdan, "Multi-population cooperative bat algorithm-based optimization of artificial neural network model," *Information Sciences*, vol. 294, pp. 628–644, 2015.
- [12] Y. Duan and M. Zhao, "Bat algorithm on economic load dispatch in wind power system," *Electrical Engineering*, vol. 2, 2015.
- [13] D. Zhao and Y. He, "Chaotic binary bat algorithm for analog test point selection," *Analog Integrated Circuits and Signal Processing*, vol. 84, no. 2, pp. 201–214, 2015.
- [14] M. Kang, J. Kim, and J.-M. Kim, "Reliable fault diagnosis for incipient low-speed bearings using fault feature analysis based on a binary bat algorithm," *Information Sciences*, vol. 294, pp. 423–438, 2015.
- [15] H. Wang and X. Yan, "Optimizing the echo state network with a binary particle swarm optimization algorithm," *Knowledge-Based Systems*, vol. 86, pp. 182–193, 2015.
- [16] H. H. Xiao and Y. M. Duan, "Research and application of improved bat algorithm based on DE algorithm," *Computer Simulation*, vol. 31, no. 1, 2014.
- [17] Y. Zhou, J. Xie, L. Li, and M. Ma, "Cloud model bat algorithm," *Scientific World Journal*, vol. 2014, Article ID 237102, 11 pages, 2014.
- [18] I. Fister, S. Fong, J. Brest, and I. Fister, "A novel hybrid self-adaptive bat algorithm," *Scientific World Journal*, vol. 2014, Article ID 709738, 12 pages, 2014.
- [19] C. W. Qu, Y. M. Fu, and Y. S. Hou, "A bat algorithm based on the fusion of invasive weed operators," *Computer Application and Software*, vol. 32, no. 4, pp. 243–246, 2015.
- [20] C. C. Wu, Y. Z. He, Y. Y. Chen, X. Liu, and X. Cai, "Elite crossover binary bat algorithm for 0-1 knapsack problem," *Application Research of Computers*, vol. 6, no. 6, 2017.
- [21] Y. D. Ma, W. D. Wang, and Q. Wen, "Transmission network planning based on bat algorithm with chaotic search strategy,"

- Power System Protection and Control*, vol. 43, no. 15, pp. 17–21, 2015.
- [22] H. Shi, W. L. Wang, Y. J. Li et al., “Bat algorithm based on Levy flight feature and its localization application in WSN,” *Journal of Sensing Technology*, vol. 28, no. 6, 2015.
- [23] P. Cai and Z. Z. Yuan, “Hopf bifurcation and chaos control in a new chaotic system via hybrid control strategy,” *Chinese Journal of Physics*, vol. 55, no. 1, pp. 64–70, 2017.
- [24] S. J. Huang, P. H. Gu, W. F. Su, X.-Z. Liu, and T.-Y. Tai, “Application of flower pollination algorithm for placement of distribution transformers in a low-voltage grid,” in *Proceedings of the IEEE International Conference on Industrial Technology*, pp. 1280–1285, IEEE, Seville, Spain, March, 2015.
- [25] S. Saremi, S. Mirjalili, and A. Lewis, “Biogeography-based optimisation with chaos,” *Neural Computing and Applications*, vol. 25, no. 5, pp. 1077–1097, 2014.
- [26] G. M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E. J. Murphy, P. A. Prince, and H. E. Stanley, “Lévy flight search patterns of wandering albatrosses,” *Nature*, vol. 381, no. 6581, pp. 413–415, 1996.
- [27] A. M. Reynolds, A. D. Smith, D. R. Reynolds, N. L. Carreck, and J. L. Osborne, “Honeybees perform optimal scale-free searching flights when attempting to locate a food source,” *Journal of Experimental Biology*, vol. 210, no. 21, pp. 3763–3770, 2007.
- [28] A. M. Reynolds and M. A. Frye, “Free-flight odor tracking in drosophila is consistent with an optimal intermittent scale-free search,” *PLoS One*, vol. 2, no. 4, p. e354, 2007.
- [29] E. Cantú-Paz and D. E. Goldberg, “Efficient parallel genetic algorithms: theory and practice,” *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2–4, pp. 221–238, 2000.
- [30] E. Alba and J. M. Troya, “Analyzing synchronous and asynchronous parallel distributed genetic algorithms,” *Future Generation Computer Systems*, vol. 17, no. 4, pp. 451–465, 2001.



Hindawi

Submit your manuscripts at
www.hindawi.com

