IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received March 4, 2020, accepted April 24, 2020, date of publication April 28, 2020, date of current version May 13, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991091

Data Clustering Method Based on Improved Bat Algorithm With Six Convergence Factors and Local Search Operators

L. F. ZHU[1], J. S. WANG[1,2], (Member, IEEE), H. Y. WANG[1], S. S. GUO[1], M. W. GUO[1], AND W. XIE[1]

[1]School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan 114051, China
[2]National Financial Security and System Equipment Engineering Research Center, University of Science and Technology Liaoning, Anshan 114051, China

Corresponding author: J. S. Wang (wang_jiesheng@126.com)

This work was supported in part by the Project by National Natural Science Foundation of China under Grant 21576127, in part by the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province under Grant 2017FWDF10, and in part by the Project by Liaoning Provincial Natural Science Foundation of China under Grant 20180550700.

ABSTRACT Clustering as an unsupervised learning method is a process of dividing a data object or observation object into a subset, that is to classify the data through observation learning instead of example learning without the guidance of the prior class label information. Bat algorithm (BA) is a swarm intelligence optimization algorithm inspired by bat's ultrasonic echo localization foraging behavior, but it has the disadvantages of being easily trapped into local minima and not being highly accurate. So an improved bat algorithm was proposed. In the global search, a Gaussian-like convergence factor is added, and five different convergence factors are proposed to improve the global optimization ability of the algorithm. In the local search, the hunting mechanism of the whale optimization algorithm (WOA) and the sine position updating strategy are adopted to improve the local optimization ability of the algorithm. This paper compares the clustering effect of the improved bat algorithm with bat algorithm, flower pollination algorithm (FPA), harmony search (HS) algorithm, whale optimization algorithm and particle swarm optimization (PSO) algorithm on seven real data sets under six different convergence factors. The simulation results show that the clustering effect of the improved bat algorithm is superior to other intelligent optimization algorithms.

INDEX TERMS Clustering, bat algorithm, convergence factor, hunting mechanism, sinusoidal position update strategy.

I. INTRODUCTION

At present, swarm intelligence algorithms based on bionics have attracted people's attention. People have successfully applied the inspiration obtained from the biological world to the solution of practical problems, and proposed a series of meta-heuristic swarm intelligence algorithms based on biological behavior. For example, the whale optimization algorithm (WOA) based on whale predation [1], the particle swarm optimization (PSO) algorithm based on the swarm behavior of birds and fish swarms [2], the harmony search (HS) algorithm based on the behavior of simulated musical instruments [3], bee colony algorithm (BCA) [4], artificial flower pollination algorithm (FPA) [5] for self-pollination and cross-pollination of flowers in nature, the gray wolf optimizer (GWO) [6] and so on. The bat algorithm (BA) is a swarm intelligence algorithm proposed by Prof. Yang in 2010 based on the foraging behavior of bat ultrasonic echo localization [7]. It has been widely used due to its features of few parameters, simple model and easy coding. However, like other random searching algorithms, it has the disadvantages of easy premature convergence and low convergence accuracy, especially in the face of high-dimensional data.

Guo proposed an improved bat algorithm based on multiple swarm strategies and chaotic bat swarm algorithm so as to improve the convergence speed and accuracy of the bat algorithm. The chaos factor and the second-order oscillation mechanism are introduced to improve the update speed and dynamic parameter mechanism of the system [8]. Zhu *et al.* designed new pulse emissivity, loudness, velocity, and

The associate editor coordinating the review of this manuscript and approving it for publication was Jenny Mahoney.

position update functions to avoid premature convergence, and designed a new one-dimensional perturbed local search strategy to improve the efficiency and accuracy of local search [9]. Yuan proposed an improved bat algorithm based on weighted method to solve the multi-objective optimal power flow problem, and the experimental results shown its effectiveness [10]. Meng introduced the bat habitat selection and its adaptive compensation method to the Doppler effect into the basic BA, and proposed a new bat algorithm (NBA), which was experimentally verified with BA and other algorithms to show its effectiveness [11]. Yaseen proposed a hybrid optimization algorithm based on the bat algorithm and particle swarm optimization algorithm, that is, the hybrid bat swarm algorithm, whose main idea is to improve BA by using PSO algorithm in parallel to replace the suboptimal solution generated by BA. This algorithm effectively speeds up the convergence speed of the algorithm and avoids the local optimal trapping due to the existence of BA [12]. Selim enhances the local and global search characteristics of the Bat algorithm through three different methods. In order to verify the performance of the enhanced bat algorithm (EBA), the practical problems of standard test functions and constraints are used, and the results prove that EBA is better than standard BA [13]. Miodragović introduced the Bat family to expand to continuously repeat the process of finding the optimal solution by including a loop search in the solution area. For each bat in each family, perform a fine search according to Levy-flight to find an improved solution until the given constraints are met [14]. An improved adaptive bat algorithm (SABA) was proposed, which has adaptive step control and mutation mechanism. This step control mechanism uses two frequencies to adapt to the step size used for global search and local search. This mutation mechanism can improve the algorithm's ability to avoid local optimization [15]. A bat algorithm based on iterative local search and stochastic inertia weight (ILSSIWBA) is proposed [16]. A new local search algorithm, iterative local search (ILS), is proposed, which makes ILSSIWBA have a strong ability to jump out of local optimal solutions. A new weight update method, random inertia weight method, is also proposed, and the pulse rate and loudness are improved to improve the balance performance of global search and local search. Al-Betar Applied the island model strategy to the bat algorithm to enhance the algorithm's ability to control the concept of diversity [17]. Sensitivity analysis of the main parameters of the island bat algorithm was conducted, and their influence on convergence was studied. The comparison with other algorithms on the benchmark function was very successful. A binary cooperative bat search algorithm (BCBA) was proposed [18]. Different from the original bat search algorithm, in the cooperative bat search algorithm (CBA), a consensus term is added to the speed equation of the original bat search algorithm. By comparing with the four binary algorithms in the literature, a numerical explanation is provided to prove the superior performance of BCBA. A chaotic enhanced bat algorithm is proposed to solve the global optimization problem [19]. The proposed method controls the steps of chaotic mapping through thresholds and uses velocity inertia weights to synchronize the speed of the agent. These mechanisms are designed to immediately improve the stability and convergence speed of the bat algorithm. Ylidizdan originally proposed an advanced modified BA (MBA) algorithm, and then proposed a hybrid system (MBADE), which includes the use of MBA in combination with DE to further increase development potential and provide excellence in various test problem clusters Performance. Compared with published data of existing algorithms, the developed hybrid system shows better performance than standard BA in all test problem sets and produces more acceptable results [20]. Hong proposed a chaotic and efficient bat algorithm based on chaos, niche search, and evolutionary mechanisms to optimize the parameters of a mixed kernel support vector regression model [21]. In order to overcome the low search capability of the bat algorithm and the premature convergence may occur, Chakri introduced directional echo localization in the standard bat algorithm to enhance its detection and development capabilities [22]. In addition to this directional echo localization, three other improvements are embedded in the standard bat algorithm to improve its performance. In order to improve the search ability of the bat algorithm, an improved bat algorithm based on the covariance adaptive evolution process is proposed [23]. The information contained in the covariance adaptive evolution diversifies the search direction and sampling distribution of the population, which is of great benefit to the search process. Dhar proposed an image threshold segmentation method based on interval fuzzy set (IT2FS) and proposed an improved bat algorithm, which improved the calculation efficiency of threshold technology [24].

As an unsupervised learning method, clustering does not need prior class labeling information to classify data through observation learning rather than example learning [25]. Clustering is the process of dividing a data object or an observation object into subsets. Each subset is also a cluster. The purpose of clustering is to make the objects in the cluster similar to each other, and the objects between the clusters different from each other. The swarm intelligence optimization algorithm has good optimization ability, and the clustering problem can also be regarded as an optimization problem to find the optimal clustering center in the solution space. The combination of different clustering centers constitutes the solution space of the clustering problem. The goal of clustering is to find the clustering center that optimally divides the data in the solution space. The optimization mechanism of the swarm intelligence algorithms is used to enable individuals to continuously move in the solution space to find a better combination of clustering centers. Therefore, the swarm intelligence optimization algorithm is an efficient way to solve the clustering problem. Kuo proposed a dynamic clustering method based on particle swarm algorithm and genetic algorithm. This algorithm realizes automatic clustering of data by detecting the data without pre-specifying the number of clusters [26]. Yang proposed a Chinese text clustering optimization algorithm based on

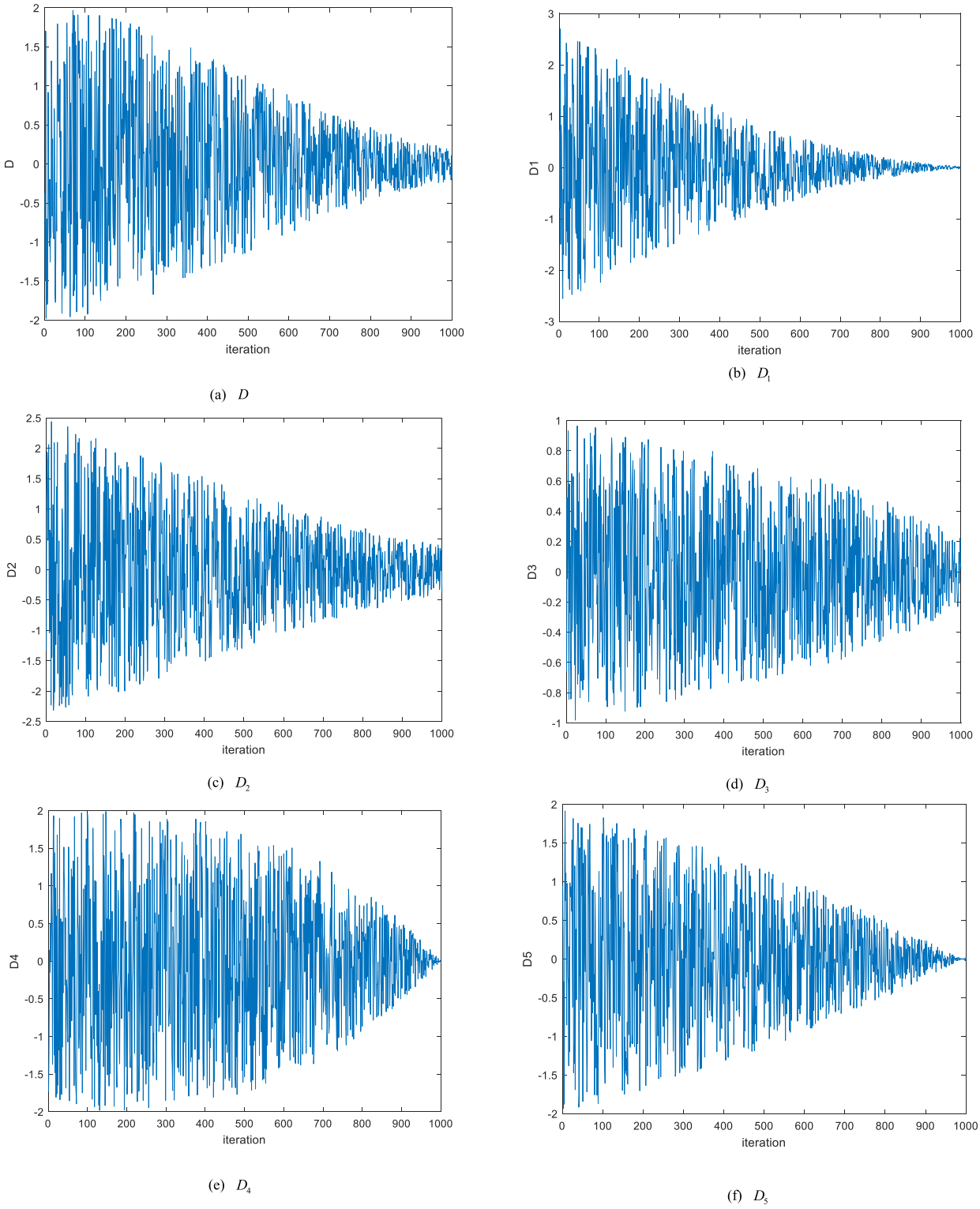
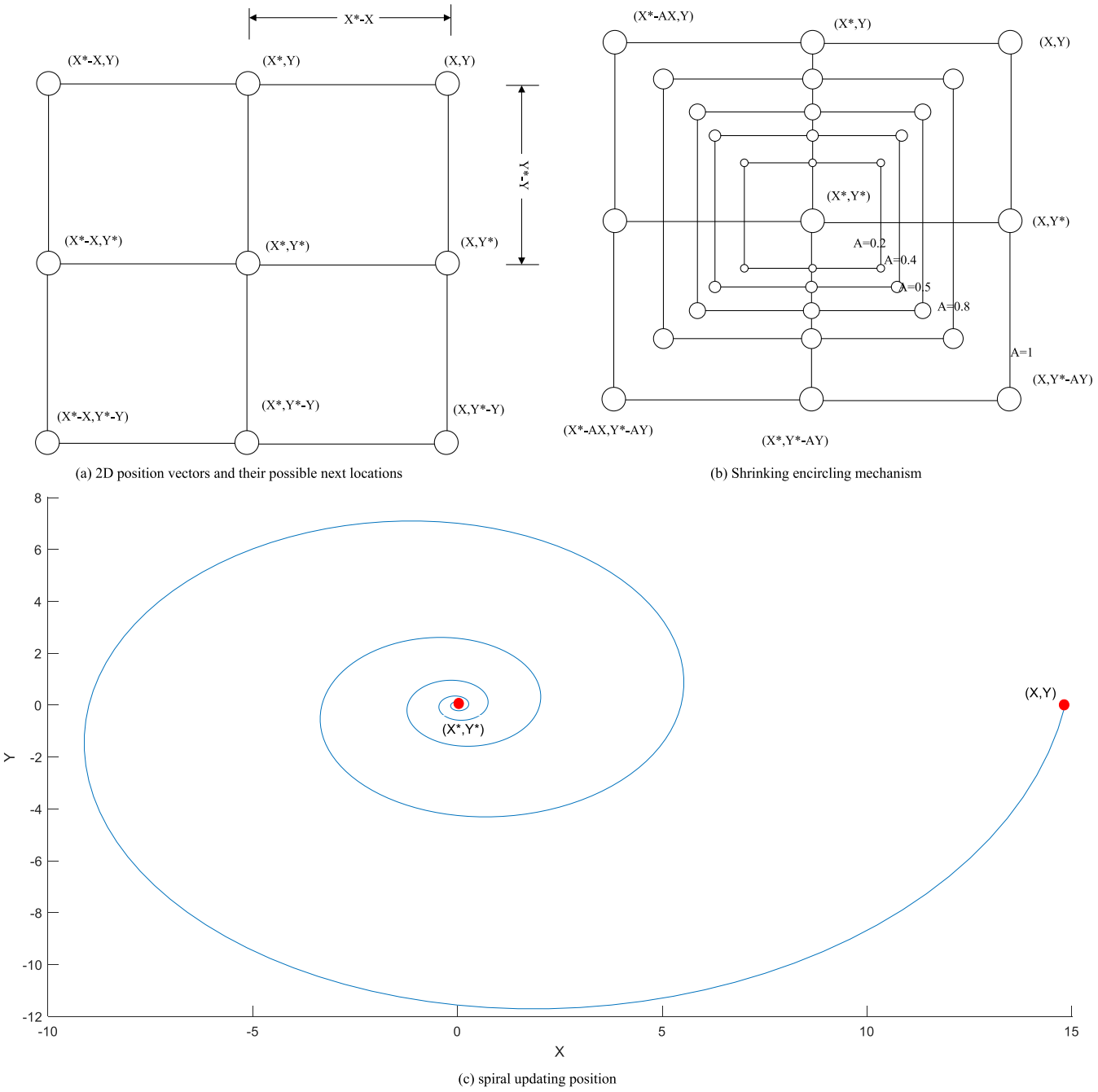


FIGURE 1. Tendency of six convergence factors with the number of iterations.

hybrid differential evolution optimization and invasive weed optimization. Experimental results show that the method has better performance [27]. The bee mating optimization algorithm was applied to clustering and got good results [28]. In order to overcome the disadvantages of K-means method that is highly dependent on the initial solution and easily fall into the local optimum, a flower pollination algorithm with bee pollination was proposed [29]. An improved differential evolution (DE) algorithm was proposed by utilizing Archimedean spiral, Mantegna Levy flight and neighborhood



(a) 2D position vectors and their possible next locations

(b) Shrinking encircling mechanism

(c) spiral updating position

FIGURE 2. **Principle of whale algorithm.**

search (NS). These strategies achieved good efficiency in convergence speed and better local and global search [30]. In order to solve the problem that the EM algorithm with the Gaussian model is very sensitive to the initial value, a robust Gaussian mixture model EM clustering algorithm is proposed, which is robust to initialization and different cluster capacities, and can automatically obtain the optimal number of clusters [31]. A K-means clustering method based on the shuffled leap frog algorithm (SFLKmeans) was proposed, which is compared with other heuristic algorithms (such as GAK, SA, TS, and ACO) on multiple simulated and real data sets. The results show that the algorithm has better performance [32].

This paper proposes an improved bat algorithm to solve cluster optimization problems. In the global search stage, a convergence factor with the Gaussian function form is added, and on the basis of this, five different convergence factors are proposed to improve the algorithm's global optimization capability. The local search is added with the whale optimization algorithm's hunting mechanism and the sine position updating strategy in order to improve the local exploration ability of the algorithm. The improved bat algorithm,

TABLE 1. Benchmark functions.

Function	Dim	Range	f_{min}
$F_1(x)=\sum_{i=1}^{n}x_i^2$	30	[-100,100]	0
$F_2(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	30	[-10,10]	0
$F_3(x)=\sum_{i=1}^{n}\left(\sum_{j=1}^{i}x_j\right)^2$	30	[-100,100]	0
$F_4(x)=\sum_{i=1}^{D}\lvert x_i\sin(x_i)+0.1x_i\rvert$	30	[-10,10]	0
$F_5(x)=\sum_{i=1}^{D}x_i^6\left(2+\sin\frac{1}{x_i}\right)$	30	[-1,1]	0
$F_6(x)=\sum_{i=0}^{D}\frac{x_i^2}{4000}-\prod\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	30	[-100,100]	0
$F_7(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos x_1+10$	2	[-5,5]	-0.398
$F_8(x)=-\sum_{i=1}^{5}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	4	[0,10]	-10.1532
$F_9(x)=-\sum_{i=1}^{7}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	4	[0,10]	-10.4028
$F_{10}(x)=-\sum_{i=1}^{10}\left[(X-a_i)(X-a_i)^T+c_i\right]^{-1}$	4	[0,10]	-10.5363

bat algorithm, flower pollination algorithm (FPA), harmony search algorithm, whale optimization algorithm and particle swarm optimization algorithm are adopted to perform clustering experiments on seven real data sets to verify the effectiveness of the proposed algorithm.

II. BAT ALGORITHM

Bats use echolocation technology to detect prey, avoid obstacles, and find habitat in dark surroundings. It can emit very loud pulses and listen to echoes that bounce back from surrounding objects. Based on the time and intensity of the echoes to the ears, it can determine the direction and position of the object. It can also issue pulses of different properties according to the characteristics of the target prey or obstacle. The frequency of sound waves emitted by bats is usually in the range of 25-100 kHz. Each sound wave emission usually lasts a few thousandths of a second (5-20 ms), and a miniature bat emits sound waves about 10-20 times per second. When hunting for prey, bats emit sonic pulses about 200 times per second. Bats make loud sounds up to 110 dB, which can change from the loudest when hunting for prey to the silence when approaching the prey. The bat detects the distance and orientation of the target, the type of the prey, and the speed of the prey [5] through the time difference between the time when the bat emits and receives the echo. If the echo localization characteristics of bats is studied in an idealized way, it can be more easy to simulate the bat algorithm. In analyzing the bat algorithm, the following approximately idealized rules are adopted.

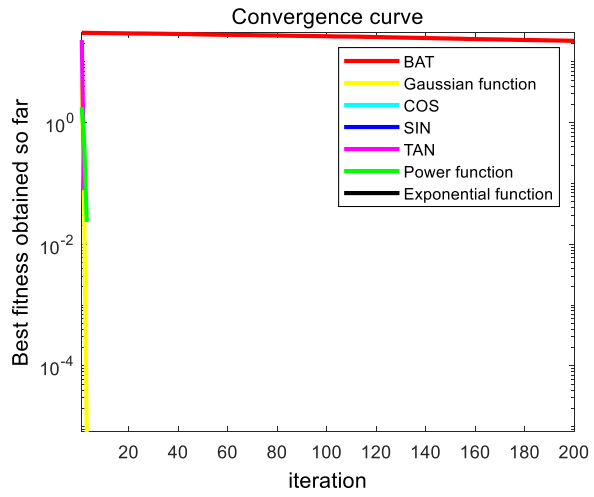
1) All bats adopt echolocation to sense distance, and they also know the difference between food / prey and background obstacles in some magical way.

2) The bats fly randomly at position x_i at speed v_i. They can automatically adjust the frequency (wavelength) of the emitted pulses and adjust the pulse emission rate $r \in [0, 1]$ according to the proximity of the target.

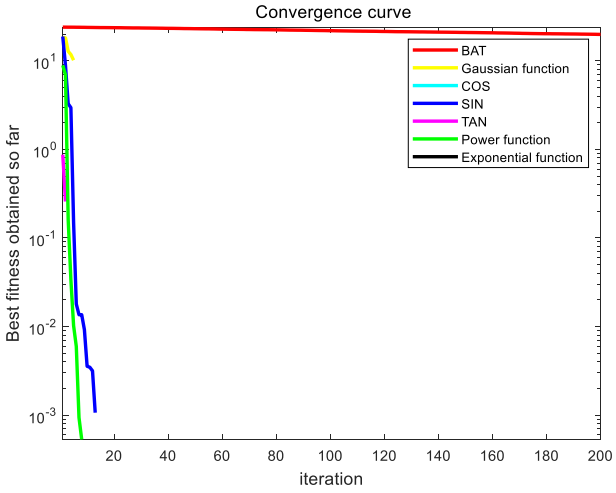
3) Although the loudness can be changed in many ways, we assume that the loudness changes from a large (positive) value A_0 to a minimum value A_{min}.

In the process of simulating the bat algorithm, it is assumed that the search space of the bat has D dimension, and the update rules of the position x_i^t and speed v_i^t of each bat in each generation are given by Eq. (1)-(3).

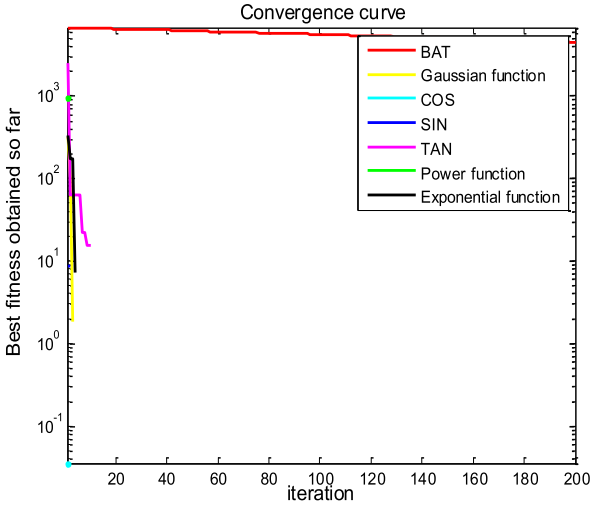
$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{1}$$
$$v_i^{t+1} = v_i^t + (x_i^t - x_*)f_i \tag{2}$$
$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{3}$$



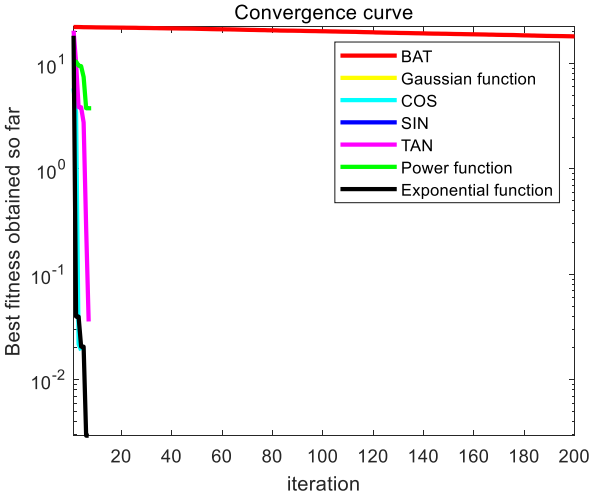
(a) F_1



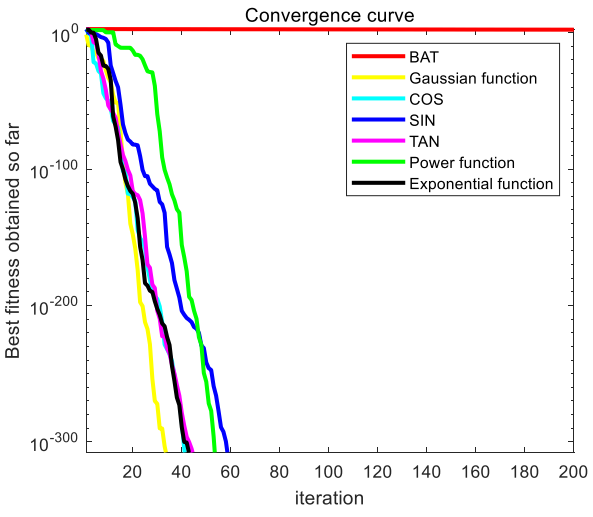
(b) F_2



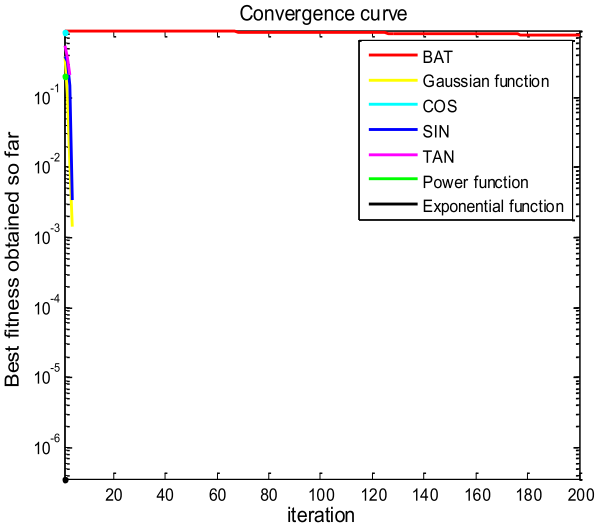
(c) F_3



(d) F_4

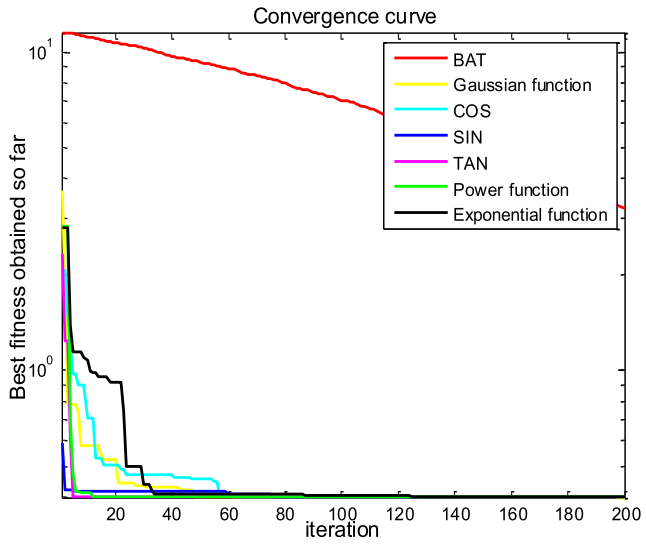


(e) F_5

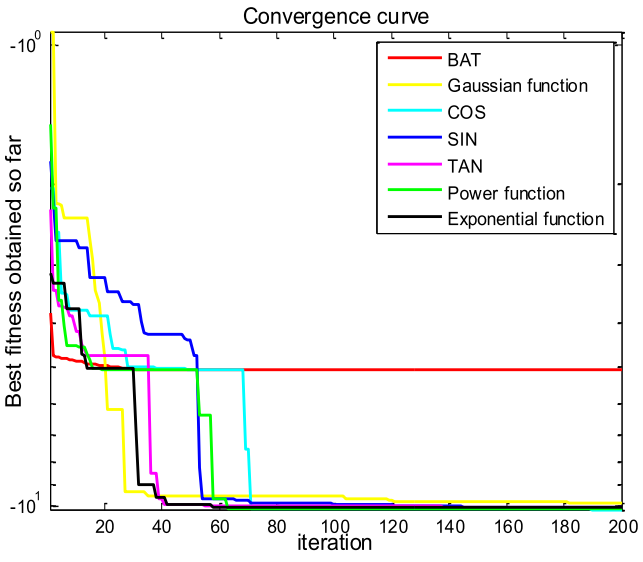


(f) F_6

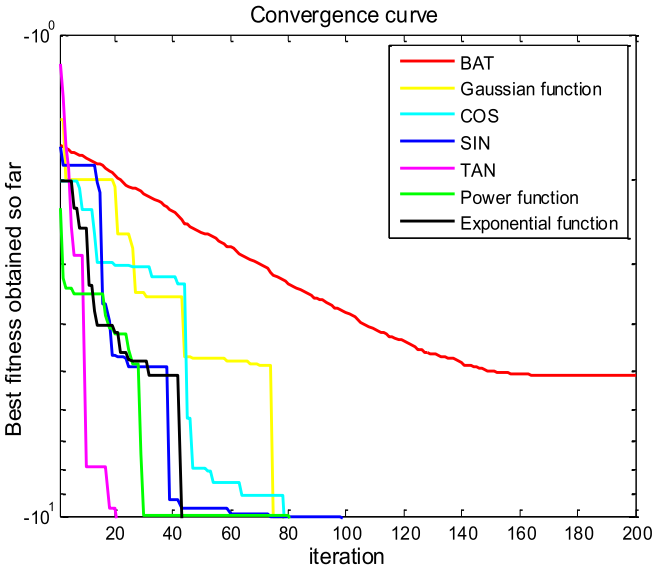
FIGURE 3. The convergence curves of test functions.



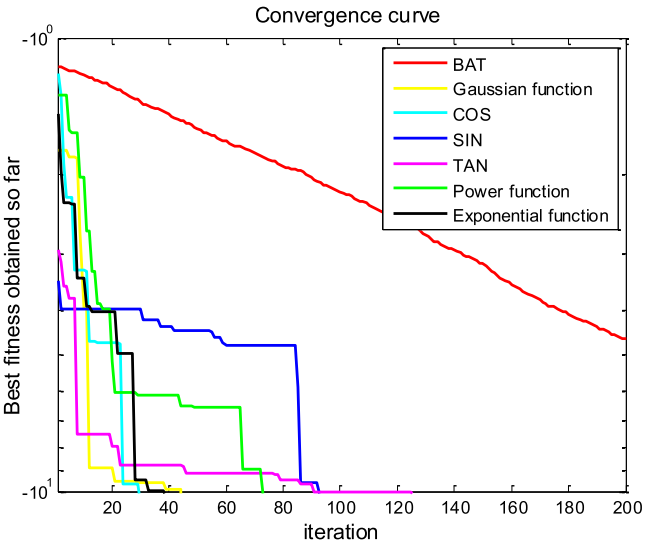
(g) F_7



(h) F_8



(i) F_9



(j) F_{10}

FIGURE 3. ***(Continued.)*** **The convergence curves of test functions.**

where x_* is the current global optimal solution, $\beta \in [0, 1]$ is a random number, f_i is the sonic frequency of the bat, which is located between $[f_{\min}, f_{\max}]$.

For the local search, once a solution is selected among the current best solutions, a local random walk is used to locally generate a new solution for each bat.

$$X_{new} = X_{old} + \varepsilon A^t \tag{4}$$

where, $\varepsilon \in [-1, 1]$ is a random number and A^t is the average loudness of the entire population in the same generation.

Assume that once a bat finds its prey, it will gradually reduce the loudness of its pulse emission, while increasing its pulse emission rate. The loudness A_i and rate r_i of the bat's transmitted pulse are adjusted according to Eq. (5) and (6).

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

$$r_i^{t+1} = r_i^0[1 - \exp(-\gamma t)] \tag{6}$$

where, $\alpha \in (0, 1)$ is the acoustic loudness attenuation coefficient, $\gamma > 0$ is the pulse frequency enhancement coefficient and r_i^0 is the initial pulse frequency of bat i.

Based on the above analysis, the procedure of the basic bat algorithm are summarized as follows:

Step 1: Parameter initialization. Bat population size m, number of iterations N, objective function $f(X)$, bat position $X_i (i = 1, 2, \ldots, m)$ and velocity V_i, sound wave frequency f_i, sound wave loudness A_i and frequency r_i.

TABLE 2. Performance comparison of test functions.

Function	Algorithms	Ave	Std	Best
F_1	BA	22.09544	2.044470461	18.2533
	Gaussian function	0	0	0
	COS	0	0	0
	SIN	0	0	0
	TAN	0	0	0
	Power function	0	0	0
	Exponential function	0	0	0
F_2	BA	21.11127	1.545386276	19.0233
	Gaussian function	0	0	0
	COS	0	0	0
	SIN	0	0	0
	TAN	0	0	0
	Power function	0	0	0
	Exponential function	0	0	0
F_3	BA	4452.05054	673.1625098	3722.7102
	Gaussian function	0	0	0
	COS	0	0	0
	SIN	0	0	0
	TAN	0	0	0
	Power function	0	0	0
	Exponential function	0	0	0
F_4	BA	19.02234	1.795302775	17.042
	Gaussian function	0	0	0
	COS	0	0	0
	SIN	0	0	0
	TAN	0	0	0
	Power function	0	0	0
	Exponential function	0	0	0
F_5	BA	149.21996	63.43608098	71.3751
	Gaussian function	0	0	0
	COS	0	0	0
	SIN	0	0	0
	TAN	0	0	0
	Power function	0	0	0
	Exponential function	0	0	0
F_6	BA	0.696051	0.058614024	0.56834
	Gaussian function	0	0	0
	COS	0	0	0
	SIN	0	0	0
	TAN	0	0	0
	Power function	0	0	0
	Exponential function	0	0	0

TABLE 2. *(Continued.)* **Performance comparison of test functions.**

	BA	5.278884945	1.658645969	3.052770239
	Gaussian function	0.397970532	8.11581E-05	0.397899505
	COS	0.397992875	0.000172664	0.397887854
F_7	SIN	0.398010682	0.000141005	0.397887709
	TAN	0.398079839	0.000413837	0.397891848
	Power function	0.397964283	0.000143716	0.39788799
	Exponential function	0.398087671	0.000292095	0.397888196
	BA	-4.96833907	0.260556089	-5.055196417
	Gaussian function	-10.0085458	0.106534784	-10.15319508
	COS	-10.0732211	0.066945373	-10.1531989
F_8	SIN	-10.088054	0.068583808	-10.15287198
	TAN	-10.0326081	0.081456578	-10.13924711
	Power function	-10.0815214	0.117508162	-10.15248422
	Exponential function	-10.0411562	0.10671847	-10.15194163
	BA	-5.04637201	0.075981768	-5.087669633
	Gaussian function	-10.3115955	0.08935731	-10.40202032
	COS	-10.3231276	0.083909035	-10.39886069
F_9	SIN	-10.3259565	0.084189927	-10.40287555
	TAN	-10.2774173	0.086792038	-10.3813266
	Power function	-10.3084416	0.120546578	-10.40240108
	Exponential function	-10.3594008	0.036349254	-10.40278264
	BA	-5.01277447	0.165000083	-5.128479498
	Gaussian function	-10.4361153	0.10615844	-10.53616337
	COS	-10.4042077	0.100165156	-10.53603845
F_{10}	SIN	-10.4525379	0.08737822	-10.53572043
	TAN	-10.419019	0.077123397	-10.52717148
	Power function	-10.4309141	0.069653546	-10.50484706
	Exponential function	-10.4716982	0.062746755	-10.53612628

Step 2: Find the optimal bat position x_* in the current population, and update the speed and position according to Eq. (1)-(3).

Step 3: Generate a random number $rand1$ located in the scope [0, 1]. If $rand1 > r_i$, choose an optimal individual among the best bats, and then generate a local solution by Eq. (4) near the selected optimal individual, otherwise update the bat position according to Eq. (3).

Step 4: Generate a random number $rand2$ located in the scope [0, 1]. If $rand1 < A_i$, and the fitness of the objective function is better than the new solution in Step 3, then accept this position. Adjust A_i (decrease) and r_i (increase) according to Eq. (5)-(6).

Step 5: Sort the fitness values of all individuals in the population and find the current best x_*.

Step 6: Repeat Step (1)-(4) to determine whether the maximum number of iterations is met, and then output the global optimal value.

III. IMPROVED BAT ALGORITHM

The bat algorithm relies on the mutual cooperation and interaction between bat individuals. There is no mutation mechanism for individuals within the population. Once the local optimal value is found, it will fall into it and affect other individuals to move closer to it, which will cause the algorithm to prematurely converge, and it will also greatly reduce the diversity of the population. Aiming at the shortcomings of the basic bat algorithm, such as easy to fall into local extreme values, low optimization accuracy, and slow convergence speed in the later stages of the algorithm, this paper introduces

TABLE 3. Parameter settings for each algorithm.

Algorithm	Main parameters Settings
BA	Particle number $n=20$, $f_{\max}=2$, $f_{\min}=0$, α=0.5, γ=0.5, r_i^0=0.001
Gaussian function	Particle number $n=20$, $f_{\max}=2$, $f_{\min}=0$, α=0.5, γ=0.5, r_i^0=0.001
COS	Particle number $n=20$, $f_{\max}=2$, $f_{\min}=0$, α=0.5, γ=0.5, r_i^0=0.001
SIN	Particle number $n=20$, $f_{\max}=2$, $f_{\min}=0$, α=0.5, γ=0.5, $r_i^0=0.001$
TAN	Particle number $n=20$, $f_{\max}=2$, $f_{\min}=0$, α=0.5, γ=0.5, $r_i^0=0.001$
Power function	Particle number $n=20$, $f_{\max}=2$, $f_{\min}=0$, α=0.5, γ=0.5, $r_i^0=0.001$
Exponential function	Particle number $n=20$, $f_{\max}=2$, $f_{\min}=0$, α=0.5, γ=0.5, $r_i^0=0.001$
FPA	Particle number $n=20$, $p=0.8$
Harmony	Particle number $n=20$, Harmony memory considering rate $HMCR_{\max}=0.95$, $HMCR_{\min}=0.06$, Pitch adjusting rate $PAR_{\max}=0.95$, $PAR_{\min}=0.35$, Tuning bandwidth $BW_{\max}=0.1$, $BW_{\min}=0.01$
WOA	Particle number $n=20$
PSO	Particle number $n=20$, Learning factor $c_1=c_2=2$, Inertia weight $\omega_{\max}=0.9$, $\omega_{\min}=0.2$
WEGWO	Particle number $n=20$,
CPSO	Particle number $n=20$, Learning factor $c_1=c_2=2$, Inertia weight $\omega_{\max}=0.9$, $\omega_{\min}=0.2$
PSO_SA	Particle number $n=20$, Learning factor $c_1=c_2=2$, Inertia weight $\omega_{\max}=0.9$, $\omega_{\min}=0.2$, Initial temperature $T=100$

a non-linear mutation factor in the speed update equation in the global search phase. It keeps the bat population highly diverse, thereby enhancing the global exploration ability of the algorithm. At the same time, the position update equation is also changed during the local search stage. The narrowing and enclosing mechanism in the whale optimization algorithm and the sine position updating strategy in the sine and cosine search algorithm are adopted to improve the deep exploration ability of the algorithm.

A. GLOBAL SEARCHING BASED ON CONVERGENCE FACTORS

During the global search stage, the bat mainly updates its position by relying on its corresponding speed value as its moving step, so as to keep approaching the prey [33]. It can be seen from the speed update Eq. (2) of the bat algorithm that $x_i^t - x_*$ has an important effect on the speed update method, that is to say it has an important effect on the bat's moving step size. $x_i^t - x_*$ is the distance from the i-th bat at the t generation to the current optimal position. The bat will be constrained by this distance in the global search, and it will not be able to swing the bat population well for the global exploration. Therefore, the global optimization ability of the algorithm is reduced. So the speed update Eq. (2) determines the global exploration ability of the bat population. In order to enhance the global search ability of the algorithm, this paper adds a non-linear mutation factor D to Eq. (2), which can be described as:

$$v_i^{t+1} = v_i^t + (x_i^t - x_*) f_i D \tag{7}$$

$$D = 2ca - c \tag{8}$$

TABLE 4. Results obtained by different algorithms on Iris dataset.

	F-measure	ARI	Accuracy
Gaussian function	0.916±0.0214	0.773±0.0303	92.167±1.751
COS	0.94±0.0139	0.8315±0.0349	94.0275±1.385
SIN	0.9302±0.0197	0.8047±0.0319	93.0538±1.2514
TAN	0.9243±0.0339	0.7955±0.0347	92.5113±2.3871
Power function	**0.9504±0.0273**	**0.8555±0.0739**	**95.058±2.7165**
Exponential function	0.941±0.0251	0.8328±0.0661	94.1159±2.498
BAT	0.8118±0.0647	0.6244±0.0921	88.667±3.5588
FPA	0.9077±0.0254	0.7528±0.0564	90.834±2.4991
Harmony	0.841±0.0427	0.6404±0.0572	87.779±2.8325
WOA	0.8893±0.0343	0.7204±0.0417	89.168±3.1575
PSO	0.8848±0.0321	0.7068±0.0624	88.612±3.1575
WEGWO	0.9015±0.0314	0.743±0.0725	90.279±3.1059
CPSO	0.9022±0.0185	0.7386±0.0439	90.279±1.8626
PSO_SA	0.9159±0.0231	0.7717±0.0459	91.668±2.1508

The obtained speed update strategy non-linearly expands the search range and ensures the diversity of the population, thereby increasing the global search capability of the bat algorithm. a is a random number between [0, 1], and c is calculated by:

$$c = 2\exp(-(1.5t)^2/Maxiter^2) \tag{9}$$

where, t is the current number of iterations, $Maxiter$ is the maximum number of iterations.

The convergence factor D decreases gradually as the number of iterations increases. At the beginning of the iteration, D has a lower attenuation degree and can move with a larger amplitude, which can better find the global optimal solution. In the later iterations, the degree of attenuation of D increases, and the range of movement decreases, which can more accurately find the optimal solution and balance the exploitation and exploration capabilities during global search. The decay behavior of the convergence factor D with the number of iterations is shown in Fig. 1 (a). It can be seen from the Eq. (9) that the expression of c belongs to the Gaussian function. Therefore, this paper proposes non-linear factors with the expressions of cosine, sine, tangent, power function and exponential function. The formulas for the five convergence factors are described as follows. The convergence factor D_1 with the cosine form is defined as:

$$D_1 = 10(c_1 a - 0.5c_1) \tag{10}$$

where c_1 can be calculated by:

$$c_1 = 1 - \cos({}^{t}/_{Maxiter} - 0.35\pi) \tag{11}$$

The convergence factor D_2 with the sine form is defined as:

$$D_2 = 5(c_2 a - 0.5c_2) \tag{12}$$

where c_2 can be calculated by:

$$c_2 = 1 + \sin({}^{t}/_{Maxiter} + \pi) \tag{13}$$

The convergence factor D_3 with the tangent form is defined as:

$$D_3 = 2(c_3 a - 0.5c_3) \tag{14}$$

where c_3 can be calculated by:

$$c_3 = 2 - \tan({}^{t}/_{Maxiter}) \tag{15}$$

The convergence factor D_4 with the power function form is defined as:

$$D_4 = 4(c_4 a - 0.5c_4) \tag{16}$$

where c_4 can be calculated by:

$$c_4 = 1 - ({}^{t}/_{Maxiter})^3 \tag{17}$$

TABLE 5. Results obtained by different algorithms on wine dataset.

	F-measure	ARI	Accuracy
Gaussian function	0.7228±0.0229	0.3859±0.023	72.252±1.4802
COS	0.7496±0.007	0.4226±0.0084	74.2088±0.9833
SIN	0.7326±0.0222	0.3942±0.0359	72.7693±2.2819
TAN	**0.7503±0.038**	**0.4356±0.0199**	**74.7143±1.0516**
Power function	0.749±0.0378	0.4316±0.0599	74.6473±2.4032
Exponential function	0.7345±0.0209	0.4099±0.0435	73.0735±2.1553
BAT	0.6417±0.077	0.2614±0.1144	63.238±10.3962
FPA	0.689±0.0319	0.3443±0.0456	68.31±3.3471
Harmony	0.7056±0.0297	0.3574±0.0562	69.858±3.3447
WOA	0.6828±0.0243	0.3083±0.0417	67.324±2.94
PSO	0.6947±0.0092	0.3295±0.0202	68.449±1.0725
WEGWO	0.7201±0.0313	0.3652±0.0608	71.267±3.0999
CPSO	0.7226±0.0309	0.3814±0.0627	71.407±3.4526
PSO_SA	0.7189±0.0314	0.3736±0.0547	71.266±3.4037

The convergence factor D_5 with the exponential function form is defined as:

$$D_5 = 4(c_5 a - 0.5c_5) \tag{18}$$

where c_5 can be calculated by:

$$c_5 = 2 - \exp(^{0.7t}/_{Maxiter}) \tag{19}$$

The movement trend of above convergence factors with the increase of iterations is shown in Fig.1.

B. LOCAL SEARCHING BASED ON HUNTING MECHANISM AND SINUSOIDAL POSITION UPDATING STRATEGY

In the local search stage, it is considered that the bat algorithm adopts the complete perturbation method in Eq. (4) for local search.

In order to generate a new solution, each vector of the current optimal solution will change, so the search efficiency is low and the search accuracy is poor. Therefore, in this paper, the shrinking enclosing mechanism in the whale optimization algorithm and the sine position update strategy in the sine and cosine algorithm are combined to enhance the local search ability of the bat algorithm.

1) HUNTING MECHANISM

The whales use the bubble net attack method (exploitation stage), which includes two methods of reducing the surrounding mechanism and updating the position by the spiral.

a: REDUCING ORBITING MECHANISM

The WOA assumes that the current best candidate solution is the target prey or near the optimal solution. After the best search agent is defined, other search agents will try to update their positions to the best search agent, this strategy is expressed as follows.

$$D = \left|C \cdot X^*(t) - X(t)\right| \tag{20}$$

$$X(t+1) = X^*(t) - A \cdot D \tag{21}$$

A and C are calculated by:

$$A = 2ar - a \tag{22}$$

$$C = 2r \tag{23}$$

where, a decreases linearly from 2 to 0, and r is a random number between [0, 1].

Fig.2 (a) illustrates the principle on the two-dimension WOA. The location (X, Y) of the search agent can be updated based on the location of the current best record (X^*, Y^*). By adjusting the values of the A and C vectors, different positions around the best agent can be achieved based on the current position. The same concept can be extended to an n-dimensional search space, and the search agent will move around the best solution obtained so far in the hypercube. The fluctuation range of A also decreases as a decreases. a decreases from 2 to 0 during the iteration. The value range of A is a random value in $[-a, a]$. Set a random value for A

TABLE 6. Results obtained by different algorithms on bupa dataset.

	F-measure	ARI	Accuracy
Gaussian function	0.6564±0.0135	0.0926±0.0641	57.463±2.3676
COS	0.6413±0.0217	0.0881±0.0342	64.2513±2.3925
SIN	0.6587±0.0131	0.1027±0.0042	65.8315±1.1458
TAN	0.6568±0.013	0.0966±0.0298	65.7522±1.3779
Power function	0.6605±0.0295	0.0991±0.037	66.0114±2.0636
Exponential function	**0.6672±0.018**	**0.1086±0.0341**	**66.7179±1.5484**
BAT	0.5879±0.0386	-0.0016±0.0084	47.03±4.0142
FPA	0.5837±0.0305	-0.0037±0.0061	52.319±3.3941
Harmony	0.5805±0.026	-0.0027±0.0068	52.608±3.3707
WOA	0.5981±0.0284	-0.0044±0.0089	53.334±2.9018
PSO	0.5753±0.014	0.0062±0.0084	55.58±2.4032
WEGWO	0.5983±0.0276	0.012±0.0295	54.058±6.1049
CPSO	0.6033±0.0351	0.0244±0.0176	58.985±2.4297
PSO_SA	0.5836±0.0281	0.0184±0.0236	57.029±4.0218

in [1,1], that is to say the new location of the search agent when $|A| \leq 1$ can be defined anywhere between the original location of the agent and the location of the current best agent. Fig.2 (b) shows all possible positions from (X, Y) to (X^*, Y^*), and 0 to 1 can achieve these positions in a two-dimensional space.

b: POSITION SPIRAL UPDATING METHOD

As shown in Fig.2 (c), this method first calculates the distance between the whale at (X, Y) and the prey at (X^*, Y^*), then create a spiral equation between the whale and the prey's position, and simulate the spiral motion of the humpback whale, which is described as follows.

$$X(t+1) = D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t) \tag{24}$$

where, $D' = |X^*(t) - X(t)|$ is the distance from the whale i to the prey (the best solution currently obtained), b is a constant defining the logarithmic spiral shape, and l is a random number in $[-1, 1]$.

The humpback whale swims around its prey in a narrow circle, while swimming along a spiral path. To simulate this simultaneous behavior, it is assumed that there is a 50% probability that a choice can be made between the reduced enclosing mechanism and the spiral model to update the position of the whales. The mathematical model is described as follows.

$$X(t+1) = \begin{cases} X^*(t) - A \cdot D, p < 0.5 \\ D' \cdot e^{bl} \cdot \cos(2\pi l) + X^*(t), p \geq 0.5 \end{cases} \tag{25}$$

where, p is a random number between [0, 1].

In the searching for prey (exploration stage), the same method based on the change of vector A can also be used to find prey (exploration). In fact, humpback whales carry out the random searches based on each other's location. Therefore, a random value A greater than or less than 1 is used to force the search agent away from the reference whale. When $|A| > 1$, exploration is emphasizes by combining with WOA algorithm for global search. Its mathematical model is expressed as follows.

$$D = |C \cdot X_{rand} - X| \tag{26}$$

$$X(t+1) = X_{rand} - A \cdot D \tag{27}$$

where, X_{rand} is a randomly selected position vector (random whale) from the current population.

2) SINUSOIDAL POSITION UPDATING STRATEGY

The sine and cosine algorithm uses simple mathematical functions (sine and cosine functions) to explore and use the space between the two solutions to design an optimization algorithm in order to find a better solution. Its position

TABLE 7. Results obtained by different algorithms on seed dataset.

	F-measure	ARI	Accuracy
Gaussian function	0.9069±0.005	0.7428±0.0087	88.929±0.9905
COS	0.9061±0.0134	0.7367±0.0224	90.5629±1.9119
SIN	**0.9105±0.0134**	**0.7511±0.0214**	**91.04±1.5711**
TAN	0.8917±0.0167	0.7029±0.0433	89.1388±1.5096
Power function	0.8977±0.0151	0.7169±0.0371	89.7363±1.5096
Exponential function	0.8999±0.0163	0.7243±0.0286	89.97±1.1346
BAT	0.7694±0.0706	0.5236±0.1201	85.593±2.4097
FPA	0.8661±0.0211	0.6415±0.0465	86.546±2.1348
Harmony	0.8716±0.0112	0.6562±0.027	87.142±1.1702
WOA	0.8515±0.0201	0.6344±0.0357	85.474±1.7519
PSO	0.8768±0.0136	0.6656±0.0343	87.621±1.432
WEGWO	0.8398±0.0303	0.6155±0.0545	84.521±2.8173
CPSO	0.887±0.0209	0.6941±0.049	88.692±2.0825
PSO_SA	0.8957±0.0161	0.7121±0.043	89.526±1.6691

updating principle can be expressed as:

$$X_i^{t+1} = \begin{cases} X_i^t + r_1 \sin(r_2) \times \left| r_3 P_i^t - X_i^t \right|, & r_4 < 0.5 \\ X_i^t + r_1 \cos(r_2) \times \left| r_3 P_i^t - X_i^t \right|, & r_4 \geq 0.5 \end{cases} \quad (28)$$

where, X_i^{t+1} is the position of the current solution in the i-th dimension at the t-th iteration, P_i^t is the position of the end point of the i-th dimension, and $r_1 = 2(1 - t/Maxiter)$, r_2 is a random number between $[0, 2pi]$, r_3 is a random number between $[0, 2]$, and r_4 is a random number between $[0, 1]$.

3) LOCAL SEARCH BASED ON HUNTING MECHANISM AND SINUSOIDAL POSITION UPDATING STRATEGY

By combining the whale optimization algorithm's miniaturization surrounding mechanism and the sinusoidal position updating strategy of the sines and cosines optimization algorithm. This specific strategy can be expressed as:

$$x_{new} = \begin{cases} x_{old} + r_1 \sin(r_2) \left| r_3 x_* - x_{old} \right|, & |A| \geq 1 \\ x_* - A \cdot D, & |A| < 1 \end{cases} \quad (29)$$

When $|A| < 1$, the new position (X, Y) can be updated based on the position of the current best record (X^*, Y^*). The position can be updated around the current optimal solution, so that the position can be better explored and updated. When $|A| \geq 1$, the sine position update principle is adopted to expand the search range and balance the global search and local search capabilities more effectively.

C. PSEUDO CODE OF IMPROVED BAT ALGORITHM

The pseudo-code based on the improved bat algorithm is described as follows.

Initialize the bat population x_i and $v_i (i = 1, 2, \ldots, n)$
Initializes pulse frequency f_i, pulse rates r_i, and loudness A_i.
while ($t <$ *Max* number of iterations)
Generate new solutions by adjusting frequency, and updating velocities and locations/solutions [equations (1), (7) and (3)]
  if ($rand > r_i$)
  Select a solution among the best solutions
  Update the formula according to formula (29)
  end if
  if ($rand < A_i \& f(x_i) < f(x_*)$)
  Accept the new solutions
  Increase r_i and reduce A_i
  end if
Rank the bats and find the current best x_*
$t = t + 1$
end while

According to the above pseudo-code, the time complexity of the algorithm is $O(\log n)$. The inner nested algorithm needs to loop all individuals, so the time complexity is $O(n \log n)$.

TABLE 8. Results obtained by different algorithms on heartstatlog dataset.

	F-measure	ARI	Accuracy
Gaussian function	**0.6933±0.0148**	**0.1452±0.0107**	64.167±2.2658
COS	0.6345±0.0121	0.0698±0.008	64.09±1.2193
SIN	0.643±0.0303	0.0758±0.0519	64.5223±1.4741
TAN	0.6647±0.0128	0.0979±0.0222	66.4994±1.3791
Power function	0.6636±0.0327	0.103±0.0451	66.7483±1.968
Exponential function	0.6714±0.0247	0.1115±0.0452	**67.2563±2.0233**
BAT	0.6166±0.0327	0.0204±0.0345	51.39±8.5377
FPA	0.5878±0.0302	0.0202±0.0229	58.806±3.0739
Harmony	0.6031±0.0259	0.0333±0.0228	58.358±7.013
WOA	0.6082±0.0333	0.0332±0.0357	60.148±3.8962
PSO	0.5991±0.0264	0.0283±0.0182	59.999±2.6531
WEGWO	0.6138±0.0353	0.0333±0.0358	59.253±5.6659
CPSO	0.6162±0.0465	0.0484±0.0383	61.493±4.9391
PSO_SA	0.6155±0.0318	0.0426±0.0332	61.493±3.1879

When formula (29) is calculated, the position is updated for each dimension of each individual, so its time complexity is $O(n^2 \log n)$. So the overall time complexity is $O(\log n + n \log n + n^2 \log n)$, which is $O(n^2 \log n)$.

D. GLOBAL SEARCHING BASED ON CONVERGENCE FACTORS

In this section, the numerical efficiency of the improved algorithm proposed in this paper is verified by solving 10 mathematical optimization problems. The expressions of the ten benchmark functions are shown in Table 1. In order to prove the superiority of the algorithm from various aspects, the test functions are divided into three groups of functions, one is the unimodal function F1 $\sim$ F3 [34], and the unimodal function has only one global optimal solution. One is the multimodal function F4 $\sim$ F6 [35]. Multimodal functions have more than extreme points, so multimodal functions have local optimal values; the last combination function is F7 $\sim$ F10 [34]. The combination function is formed by rotating, shifting, and offsetting various benchmark test functions. These functions have lower dimensions and fewer local optimal values. Both single-mode and multi-modal functions are tested in 30 dimensions, and compound functions are tested in their respective dimensions. Ten simulation experiments were performed on these three test functions, and more comprehensive test results were obtained by running 200 generations each time. The parameters of each algorithm are set to the number of populations $n = 30$; and other parameters are set: $f_{\max} = 2, f_{\max} = 2, \alpha = 0.5, \gamma = 0.5, r_i^0 = 0.001$.

It can be seen from the simulation results shown in Fig. 2 and Table 2 that the improved six bat algorithms are superior to the original bat algorithms. For the results of the unimodal and multimodal functions, the improved bat algorithm finds an optimal value of 0 every time, but the bat algorithm falls into a local optimum. For a fixed-dimension test function, the improved algorithm shows its superior performance. It can also be seen from the above figure that the convergence rate of the improved bat algorithm has been greatly improved, and the results are stable after multiple experiments. Thus we can say that the improved bat algorithm improves the convergence speed and convergence accuracy of the original bat algorithm.

IV. DATA CLUSTERING METHOD BASED ON IMPROVED BAT ALGORITHM

Based on unsupervised learning, a clustering method is proposed to divide objects into groups or classes. In unsupervised technology, the training data set is first grouped based only on the numerical information in the data (the cluster center), and then matched to the class. The adopted data set contains class information for each data. Therefore, the main goal is to find the center of the cluster by minimizing the objective function (the sum of the distance of the pattern from its center).

TABLE 9. Results obtained by different algorithms on WDBC dataset.

	F-measure	ARI	Accuracy
Gaussian function	0.9053±0.0109	0.6591±0.0214	90.569±0.9361
COS	0.9054±0.008	0.6616±0.0114	90.8035±0.6128
SIN	0.9114±0.0078	0.6801±0.0201	91.3738±0.7355
TAN	0.9056±0.0129	0.6609±0.0158	90.7716±1.0355
Power function	0.9102±0.0115	0.6763±0.0367	91.2606±1.0815
Exponential function	**0.9343±0.0145**	**0.7572±0.0323**	**93.5458±1.0975**
BAT	0.7097±0.0861	0.0919±0.2488	84.518±3.8631
FPA	0.8776±0.0129	0.576±0.037	88.169±1.1709
Harmony	0.8748±0.0181	0.5694±0.0478	87.957±1.5587
WOA	0.8849±0.0187	0.5978±0.026	88.874±0.8225
PSO	0.8727±0.0112	0.564±0.0303	87.817±0.9493
WEGWO	0.8952±0.0155	0.6283±0.0476	89.791±1.4859
CPSO	0.8736±0.0196	0.5672±0.0558	87.889±1.7764
PSO_SA	0.8764±0.0175	0.5756±0.0492	88.169±1.5689

The purpose of clustering is to minimize the objective function given N patterns [36]:

$$J(K) = \sum_{k=1}^{K} \sum_{i \in c_k} d(x_i - c_k) \tag{30}$$

where, K is the number of clusters, d is the Euclidean distance, $c_k (k = 1, 2, \ldots, K)$ is the center of the K-th cluster, and $x_i (i = 1, 2, \ldots, N)$ is the data of the K-th cluster.

Clustering is to assign the patterns in the data to the cluster, so that the patterns in a cluster are similar based on a certain similarity measure. The most common measurement method is distance measurement. This paper uses the Euclidean distance between the minimized data center and the data set belonging to the center as the objective function [37], [38]:

$$obj_i = \sum_{j=1}^{D_{Train}} d(c_i, x_j^{Bl(c_i)}) \tag{31}$$

where, $i = 1, \ldots, K$, D_{Train} is the number of training data sets, c_i is the i-th cluster center, Bl is the instance to which c_i belongs, and $x_j^{Bl(c_i)}$ is the training data matrix belonging to cluster i.

In this paper, the clustering center is the decision variable. The objective function shown in Eq. (31) is minimized to obtain the optimal clustering center. 75% of the data are randomly selected in the data set as training set so as to obtain the optimal clustering center, and then tested the remaining 25% of the data (test set) to obtain the accuracy of the clustering result. The F-measure and ARI indexes below classify all the data according to the optimal clustering center obtained from the training set to evaluate the clustering effect. The specific procedure of the clustering algorithm are described as follows.

Step 1: Initial population.

Step 2: Input each cluster data.

Step 3: 75% of each type of data was randomly selected as training data.

Step 4: The fitness value is calculated according to the objective function, and the fitness value of the small value is denoted as $f_{\min}$ and its corresponding global optimal position.

Step 5: In the iterative process, the training data were trained according to the improved bat algorithm and the population location was updated.

Step 6: Calculate the fitness value of the updated position after each iteration, and compare the minimum fitness value with $f_{\min}$. If less than $f_{\min}$, update the minimum fitness value and the optimal location, otherwise continue the iteration process.

Step 7: At the end of the iteration, the final global optimal position is obtained, which is the optimal cluster center.

Step 8: Repeat Step (4)-(7) to determine whether the maximum number of iterations is met, and then output the global optimal value.

TABLE 10. Results obtained by different algorithms on cencer dataset.

	F-measure	ARI	Accuracy
Gaussian function	0.9711±0.0052	0.8866±0.0112	96.133±0.4017
COS	0.968±0.0036	0.8749±0.0224	96.8011±0.8297
SIN	0.971±0.0114	0.8866±0.0236	97.1106±0.6261
TAN	0.9629±0.0072	0.856±0.0222	96.298±0.7144
Power function	0.9694±0.0111	0.8808±0.0415	96.9587±0.5487
Exponential function	**0.9725±0.0136**	**0.8923±0.031**	**97.2494±0.749**
BAT	0.6648±0.076	0.0521±0.1646	90.257±3.5089
FPA	0.8864±0.0531	0.6149±0.1451	89.298±4.5364
Harmony	0.9357±0.0097	0.7593±0.0333	93.686±0.9368
WOA	0.9494±0.0076	0.8064±0.0274	94.973±0.7495
PSO	0.9524±0.0082	0.8171±0.0295	95.265±0.8049
WEGWO	0.9502±0.0106	0.8089±0.0386	95.031±1.0537
CPSO	0.959±0.0079	0.8411±0.0291	95.907±0.7856
PSO_SA	0.9554±0.008	0.828±0.0295	95.558±0.7931

Step 9: Repeat steps (2)-(8) to find the optimal clustering center for the next cluster of data.

Step 10: The data sets are classified according to the distance from each data to each clustering center.

A. EVALUATION INDEX

There are three methods for testing the clustering effect: F-Measure, adjusted Rand index, and accuracy.

1) F-MEASURE

The F-Measure represents the harmonic mean between the accuracy and recall of the clustering of all classes [39]. Given the number of samples n_i in the known class i, the number of samples n_j in the cluster j, and the number of samples n_{ij} in the cluster j belonging to the known class i, the accuracy can be defined as:

$$p(i,j) = \frac{n_{ij}}{n_j} \tag{32}$$

The recall rate can be defined as:

$$r(i,j) = \frac{n_{ij}}{n_i} \tag{33}$$

Then the overall F-Measure of the data set can be defined as:

$$F = \sum_i \frac{n_i}{n} MAX_j \{F(i,j)\} \tag{34}$$

$$F(i,j) = \frac{(b^2+1)(p(i,j) \times r(i,j))}{b^2(p(i,j) + r(i,j))} \tag{35}$$

where, $b = 1$. The value range of the F-measure is [0, 1]. The larger the value, the better the clustering effect.

2) ADJUSTED RAND INDEX

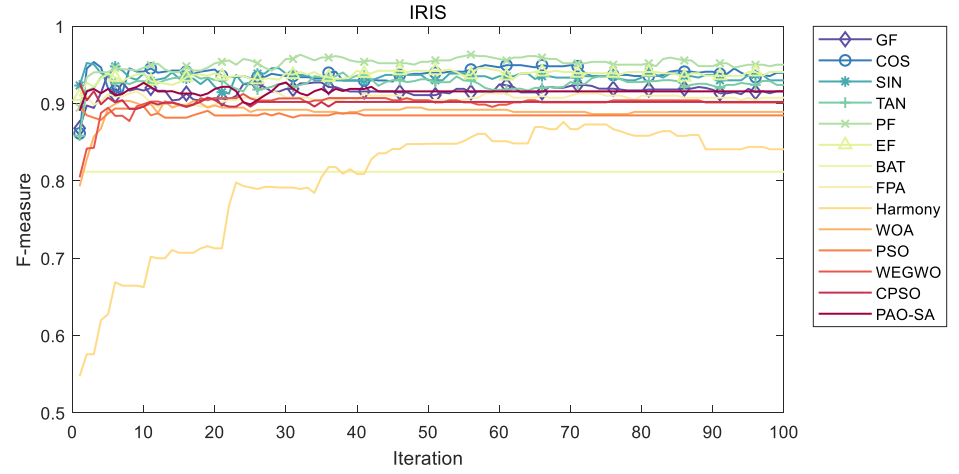
The Rand index (RI) [40] needs to provide the actual category information C. Assuming that K is the clustering result, a represents the logarithm of elements of the same category in both C and K, and b represents both C and K is the logarithm of the elements in different categories, the Rand index is defined as:

$$RI = \frac{a+b}{C_2^{n_{samples}}} \tag{36}$$

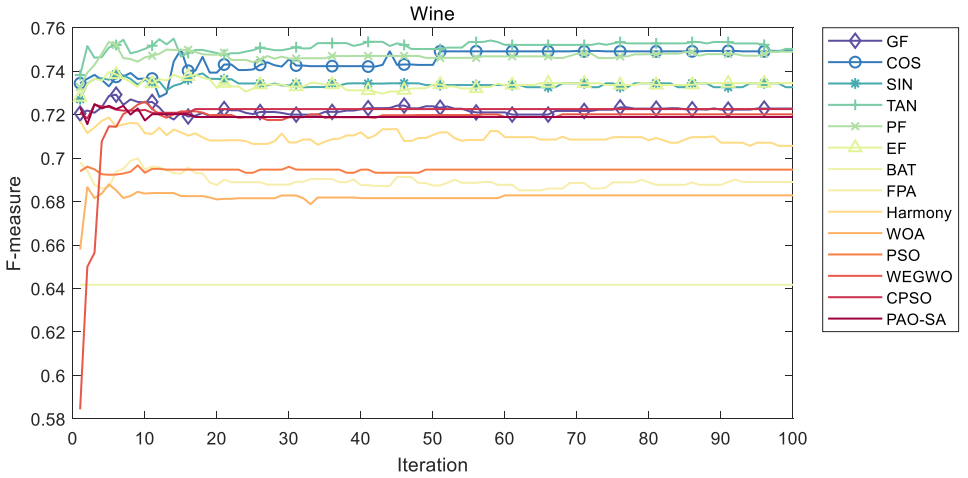
where, $C_2^{n_{samples}}$ is the total number of element pairs that can be composed in the data set. The value range of RI is $[0, 1]$. A larger value means that the clustering result is more consistent with the real situation.

For random results, RI does not guarantee a score close to zero. In order to achieve "in the case where the clustering results are randomly generated, the index should be close to zero", an adjusted rand index (ARI) [41] was proposed, which has a higher degree of discrimination.

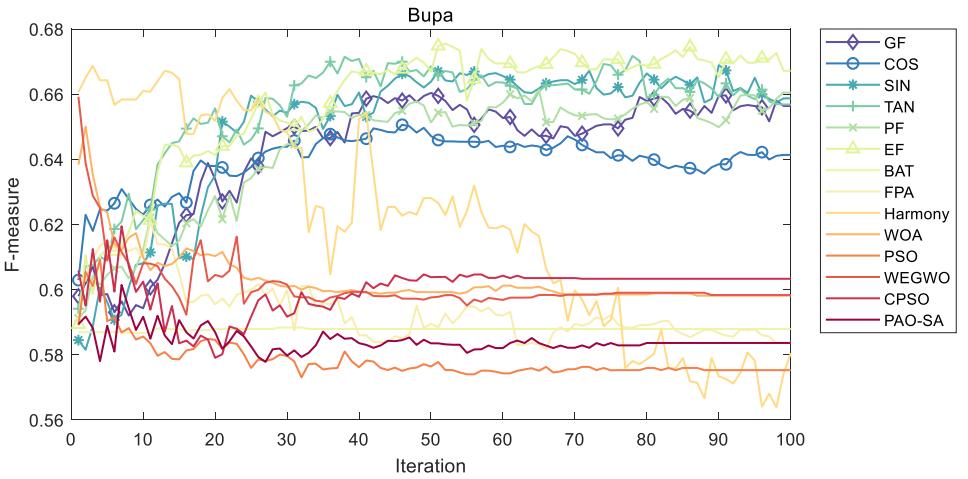
$$ARI = \frac{RI - E[RI]}{\max(RI) - E[RI]} \tag{37}$$



(a) IRIS



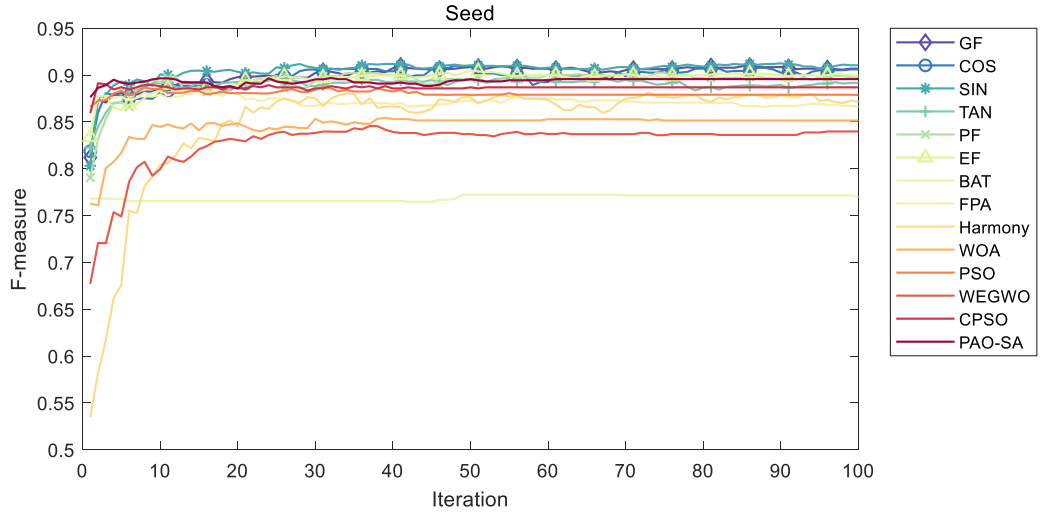
(b) Wine



(c) Bupa

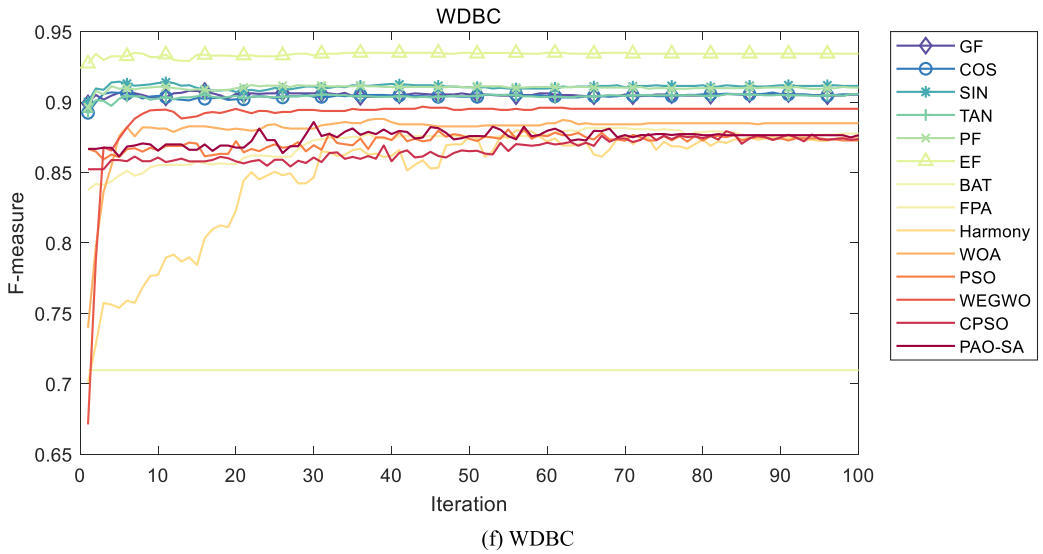
FIGURE 4. F-measures of different data sets under different algorithms.

The value range of *ARI* is $[-1, 1]$. A larger value means that the clustering result is more consistent with the real situation. In a broad sense, *ARI* measures how well the two data distributions fit.



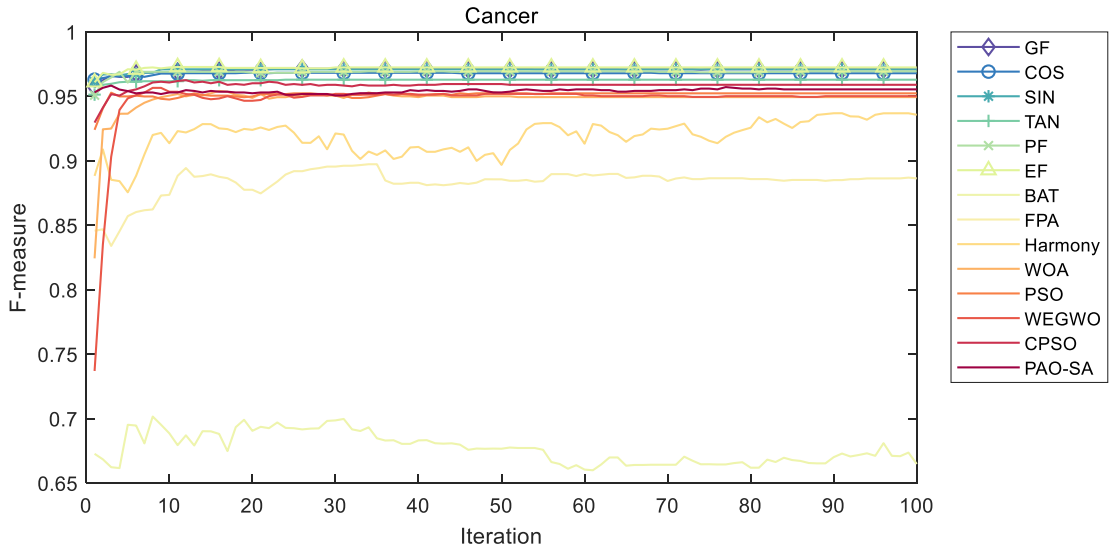
(d) Seed

(e) Heartstatlog



(f) WDBC

FIGURE 4. *(Continued.)* **F-measures of different data sets under different algorithms.**



(g) Cancer

FIGURE 4. ***(Continued.)*** **F-measures of different data sets under different algorithms.**

3) ACCURACY

The definition of Accuracy is defined as the ratio of the number of correctly classified samples to the total number of samples for a given test data set, that is to say the loss function is the accuracy on the test data set when the loss is 0-1 [42].

B. DATA SETS

7 kinds of real data in UCI library are adopted to perform clustering experiments, which are Iris, Wine, Bupa, Seeds, Heartstatlog, WDBC, and Wisconsin breast cancer.

(1) Iris (N = 150, d = 4, K = 3) is the most widely used data set, which can be divided into three types of iris plants. Each type contains 50 data, a total of 150 four-dimensional attribute data sets. Properties include sepal length, sepal width, petal length, and petal width. Two types of data are highly overlapping, and the other is linearly separable from the other two types.

(2) Wine (N = 178, d = 13, K = 3) data are the result of chemical analysis of wines from the same region of Italy but from three different varieties. It analyzes and determines the amount of 13 ingredients in each of the three wines.

(3) BUPA (N = 345, d = 6, K = 2). The BUPA liver disorders data set contains 2 types of 345 data with a total of 7 attributes, each of which represents a record of a male individual. The first five variables are blood tests and are considered sensitive to liver disease that can be caused by excessive drinking.

(4) Seeds (N = 210, d = 7, K = 3) includes seeds belonging to three different wheat varieties: Kama, Rosa, and Canadian wheat, each of which is randomly selected from 70 elements.

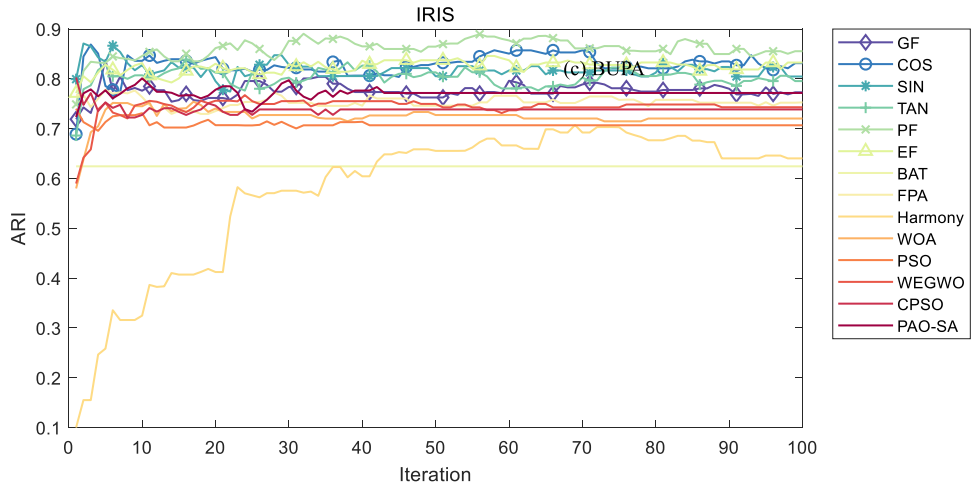
(5) Heartstatlog (N = 270, d = 13, K = 2). This data set is a heart disease database and contains 270 data with 13 attributes.

(6) WDBC (N = 569, d = 32, K = 2). Wisconsin Diagnostic Breast Cancer is a diagnostic breast cancer data set and contains 569 data sets of 2 types with 32 attributes. Features were calculated from digital images of fine needle aspiration (FNA) of breast masses.

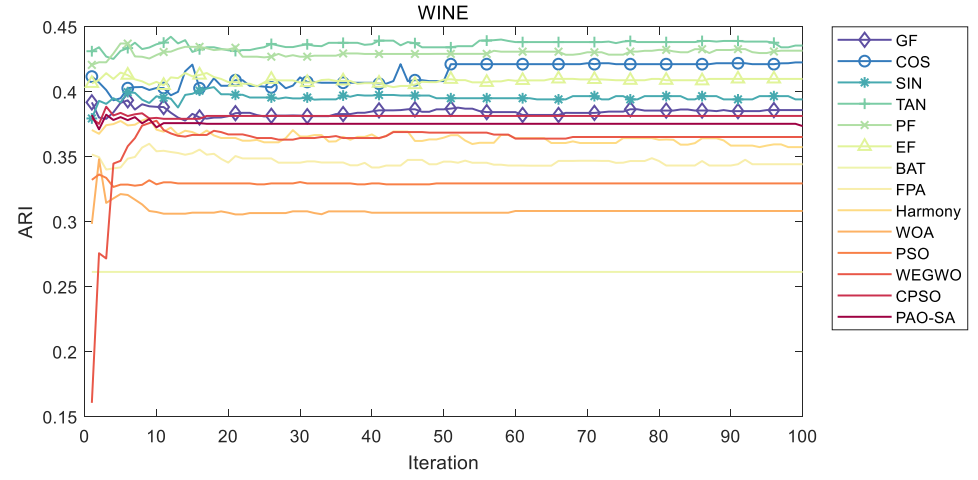
(7) Wisconsin breast cancer (N = 683, d = 9, K = 2), which consists of 683 objects characterized by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. There are two categories in the data: malignant (444 objects) and benign (239 objects).

C. SIMULATION EXPERIMENT AND RESULT ANALYSIS

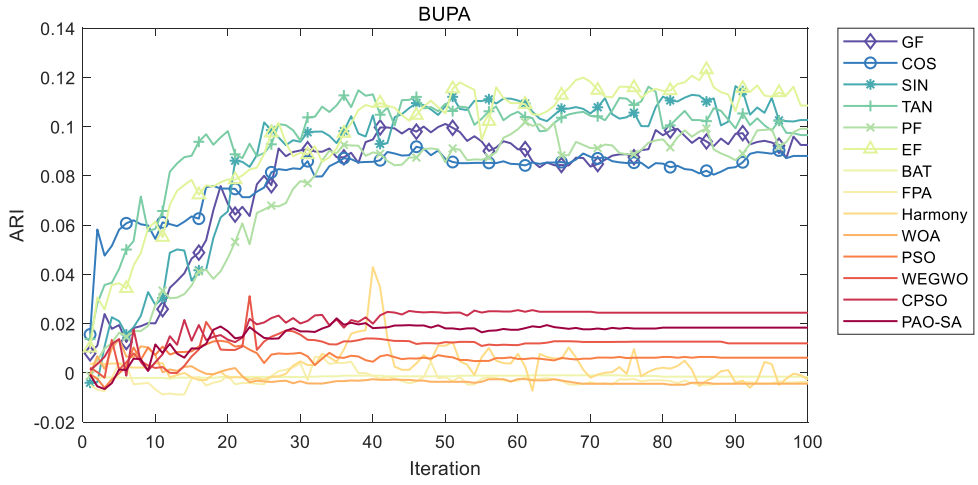
The five improved bat algorithms are compared with the original bat algorithm, the flower pollination algorithm (FPA) [5], the harmony search algorithm (Harmony) [3], the whale optimization algorithm (WOA) [1], the particle swarm optimization (PSO) algorithm [2], a clustering algorithm combining whale algorithm and grey Wolf optimizer (WEGWO) [43], Chaotic particle swarm optimization (CPSO) [44] and a hybrid PSO and SA clustering algorithm (PSO_SA) [45]. The main parameter settings of each algorithm are shown in Table 3. Clustering experiments were performed by using seven real data set from the UCI database, namely Iris, Wine, Bupa, Seeds, Heartstatlog, WDBC and Wisconsin breast cancer. The effectiveness of stochastic algorithms depends to a large extent on the choice of initial solution, so all algorithms in this paper take randomly generated initial solutions, and for each data set, the algorithm is executed ten times to perform their own validity tests. The clustering results were evaluated by using F-measure, ARI, and Accuracy performance indicators. The running results are shown in Table 4-10.



(a) IRIS



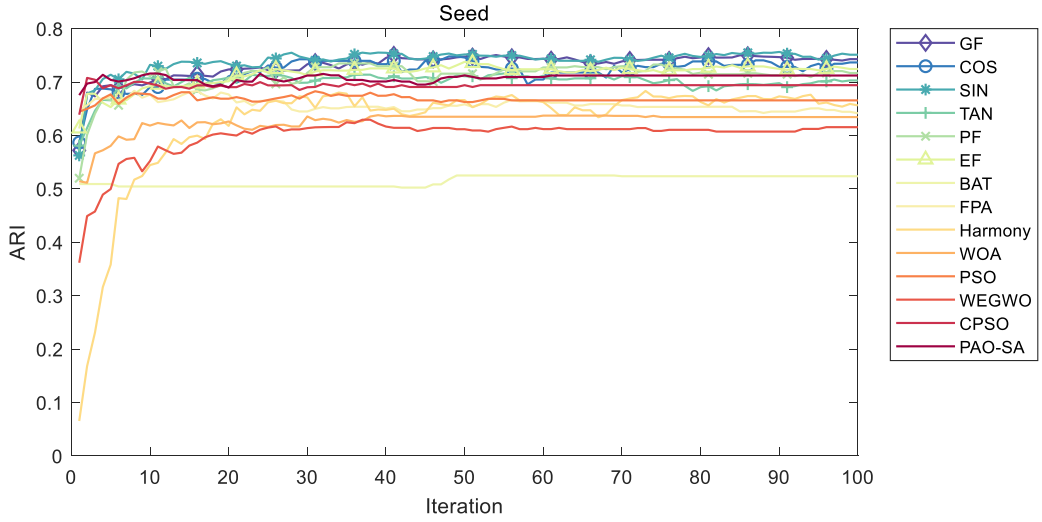
(b) WINE



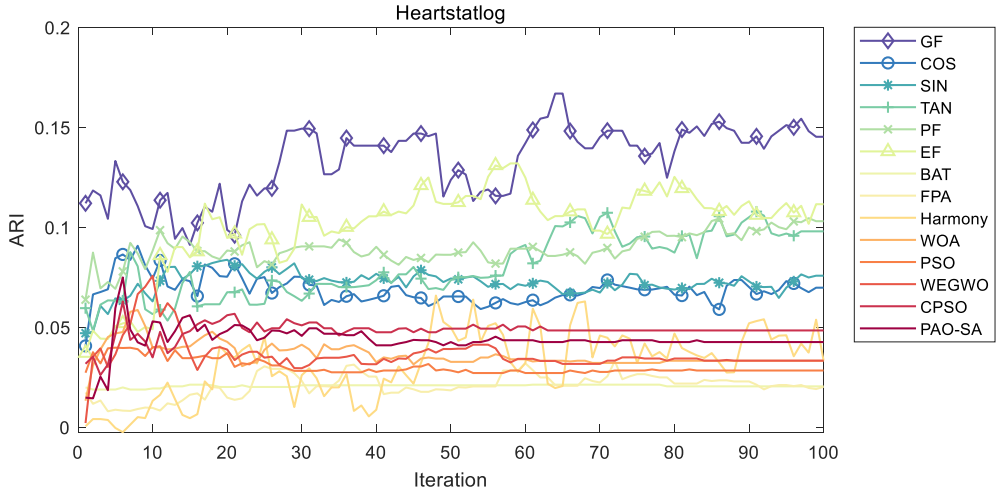
(c) BUPA

FIGURE 5. ARI on different data sets under different algorithm.

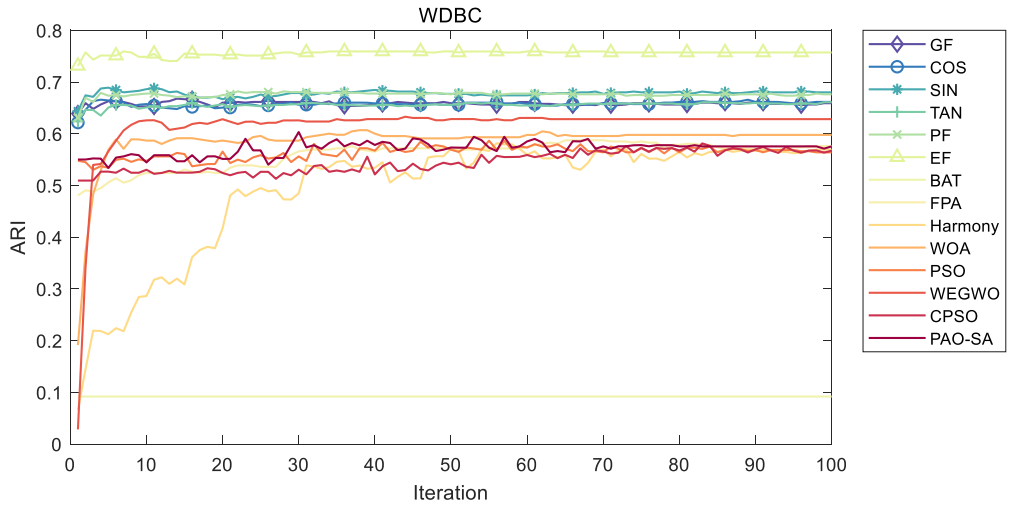
It can be seen from Table 4 that the five proposed improved schemes are better than the original bat algorithm in clustering effect and accuracy, and have a certain improvement in stability. Compared with the seven swarm intelligent optimization algorithms, the improved bat algorithm has improved the clustering effect and accuracy, but there are some shortcomings in the stability of the clustering algorithm, such as bats based on the power function and



(d) Seed

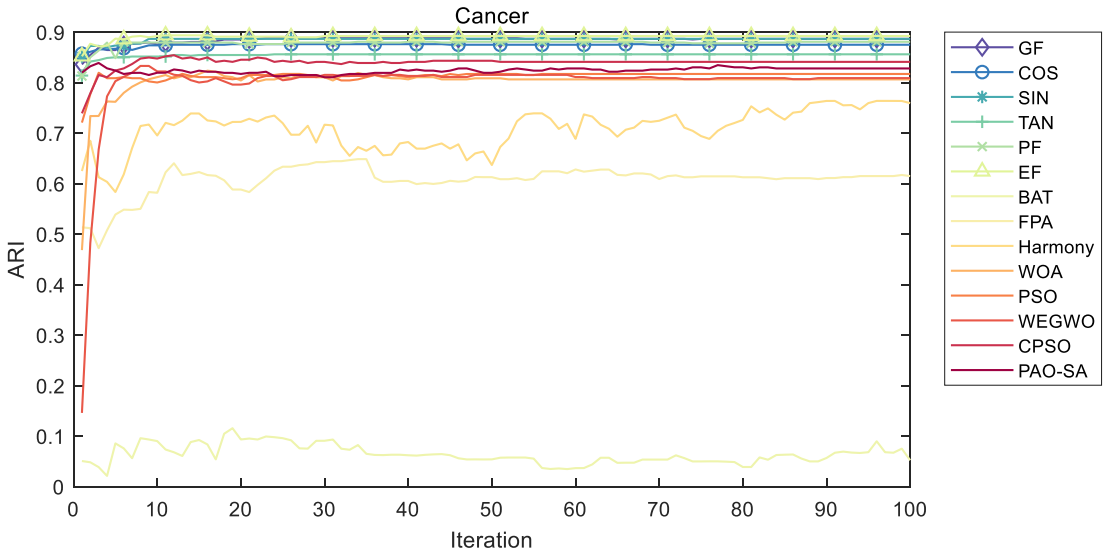


(e) Heartstatlog



(f) WDBC

FIGURE 5. ***(Continued.)*** **ARI on different data sets under different algorithm.**



(g) Cancer

FIGURE 5. *(Continued.)* **ARI on different data sets under different algorithm.**

exponential function in the Iris data set. The ARI index of the algorithm is larger than the standard deviation of FPA, Harmony, WOA, PSO, CPSO, PAO_SA, which shows that the stability of the algorithm is insufficient. However, in the comparison of three indicators, the six improved algorithms have better clustering results than the other five typical algorithms, and the best of them is the bat algorithm based on the power function. Seen from Table 5 on the Wine data set, the six improved BAT algorithms are better than BAT, FPA, Harmony, WOA, WEGWO, CPSO and PSO_SA in clustering results and stability, among which the improved BAT algorithm based on tangent function has the best effect. PSO algorithm is superior to the partially improved BAT algorithm in terms of stability, but in terms of clustering effect, the six improved methods are better than BAT, FPA, Harmony, WOA and PSO algorithm, which also reflects the superiority of our improved BAT algorithm to some extent. In the Table 6 about Bupa dataset, the accuracy of the six improved algorithms is higher than that of the typical eight swarm intelligent optimization algorithms. The ARI index results are 0.0926, 0.0881, 0.1027, 0.0966, 0.0991, 0.1086 compared to -0.0016, -0.0037, -0.0027, -0.0044, 0.0062, 0.012, 0.0244, 0.0184 of BAT, FPA, Harmony, WOA, PSO, WEGWO, CPSO, PSO_SA algorithm. It can be seen that the clustering effect has been significantly improved. F-measure also show the effectiveness of the six improved bat algorithms in terms of results and stability. The Table 7 on Seed data set shows that PSO_SA algorithm is slightly superior to the improved algorithm based on tangent function in accuracy, f-measure and ARI, but the other five improved algorithms are all superior to other algorithms. However, it is slightly insufficient in stability. Although Harmony and PSO algorithm are slightly inferior in the comparison of indicators, they are indeed more stable than the improved bat algorithm. This shows that the improved bat algorithm is not stable enough in the clustering process. It can be seen from Table 8 on the Heartstatlog data set that the improved bat algorithm based on the Gaussian function has the best effect on the F-measure and ARI indicators is, while the highest accuracy of clustering is the improved bat algorithm based on the exponential function. It can be seen that high clustering accuracy does not mean that it has a good clustering effect. Seen from Table 9 and 10, the improved bat algorithm based on the exponential function has the best effect on both Wdbc and Cencer datasets, but the stability of the F-measure and ARI of the improved bat algorithm based on the exponential function are not minimal. Compared with the original bat algorithm, the performance of the improved bat algorithm has been greatly improved, but for other swarm intelligent optimization algorithms, the stability is still slightly insufficient.

Fig. 4 and Fig. 5 are the convergence trends of F-Measure and ARI indicators in 100 iterations of 11 algorithms on different data sets. The curve obtained is the average of ten runs. Fig. 5 is a comprehensive display of the accuracy of each algorithm in different data sets. Simulation results show that the average of F-Measure, ARI and Accuracy of this algorithm is better than other algorithms. This indicates that these clusters are well separated in space. The simulation results in the table show that the hybrid evolution algorithm converges to the global optimum with a small standard deviation, and naturally concludes that the six improved bat algorithms are a feasible and robust data clustering technique.

V. CONCLUSION

The clustering problem is a very important problem that has attracted the attention of many researchers. Among them,

the meta-heuristic swarm intelligence algorithms have more and more applications in clustering because of its good optimization ability to effectively find the optimal clustering centers. The bat algorithm has the disadvantages of being easily trapped into local minima, and the optimization precision is not high. By improving the bat algorithm, this paper effectively improves the global optimization and local optimization capabilities of the algorithm so that it can better solve the clustering problems. The algorithm has been implemented and tested on several known real data sets, and the results obtained are encouraging. Many random search algorithms have the disadvantage of unstable searching. The improved algorithm in this paper also has this problem, and more work is needed in the future. In general, the algorithm proposed in this paper has high precision and low standard deviation, so the improved bat algorithm can be applied to the case of a known number of clusters.

REFERENCES

[1] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, May 2016.

[2] J. Kennedy and R. Eberhart, "Particle swarm optimisation," in *Proc. IEEE Int. Conf. Neural Netw.*, Piscataway, NJ, USA, vol. 4, Nov. 1995, pp. 1942–1948.

[3] R. Enayatifar, M. Yousefi, A. H. Abdullah, and A. N. Darus, "LAHS: A novel harmony search algorithm based on learning automata," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 18, no. 12, pp. 3481–3497, Dec. 2013.

[4] D. Karaboga, "Artificial bee colony algorithm," *Scholarpedia*, vol. 5, no. 3, p. 6915, 2010.

[5] X. S. Yang, "Flower pollination algorithm for global optimization," in *Proc. Int. Conf. Unconventional Comput. Natural Comput.*, Vol. 7445, Dec. 2012, pp. 240–249.

[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[7] X. S. Yang, "A new metaheuristic bat-inspired algorithm," *Comput. Knowl. Technol.*, vol. 284, pp. 65–74, Oct. 2010.

[8] S.-S. Guo, J.-S. Wang, and X.-X. Ma, "Improved bat algorithm based on multipopulation strategy of island model for solving global function optimization problem," *Comput. Intell. Neurosci.*, vol. 2019, pp. 1–12, Aug. 2019.

[9] H. B. Zhu, Y. K. Wang, and Y. Zhang, "Improved Bat algorithm with novel search mechanism and one-dimensional perturbation local search strategy," *Int. J. Innov. Comput., Inf. Control*, vol. 14, no. 5, pp. 1877–1892, Oct. 2018.

[10] Y. Yuan, X. Wu, P. Wang, and X. Yuan, "Application of improved bat algorithm in optimal power flow problem," *Appl. Intell.*, vol. 48, no. 8, pp. 2304–2314, Aug. 2018.

[11] X.-B. Meng, X. Z. Gao, Y. Liu, and H. Zhang, "A novel bat algorithm with habitat selection and Doppler effect in echoes for optimization," *Expert Syst. Appl.*, vol. 42, nos. 17–18, pp. 6350–6364, Oct. 2015.

[12] Z. M. Yaseen, M. F. Allawi, H. Karami, M. Ehteram, S. Farzin, A. N. Ahmed, S. B. Koting, N. S. Mohd, W. Z. B. Jaafar, H. A. Afan, and A. El-Shafie, "A hybrid bat–swarm algorithm for optimizing dam and reservoir operation," *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8807–8821, Dec. 2019.

[13] S. Yılmaz and E. U. Küçüksille, "A new modification approach on bat algorithm for solving optimization problems," *Appl. Soft Comput.*, vol. 28, pp. 259–275, Mar. 2015.

[14] G. R. Miodragović and R. R. Bulatović, "Loop bat family algorithm (Loop BFA) for constrained optimization," *J. Mech. Sci. Technol.*, vol. 29, no. 8, pp. 3329–3341, Aug. 2015.

[15] S. Lyu, Z. Li, Y. Huang, J. Wang, and J. Hu, "Improved self-adaptive bat algorithm with step-control and mutation mechanisms," *J. Comput. Sci.*, vol. 30, pp. 65–78, Jan. 2019.

[16] C. Gan, W. Cao, M. Wu, and X. Chen, "A new bat algorithm based on iterative local search and stochastic inertia weight," *Expert Syst. Appl.*, vol. 104, pp. 202–212, Aug. 2018.

[17] M. A. Al-Betar and M. A. Awadallah, "Island bat algorithm for optimization," *Expert Syst. Appl.*, vol. 107, pp. 126–145, Oct. 2018.

[18] H. Zhang, "A binary cooperative bat algorithm based optimal topology design of leader–Follower consensus," *ISA Trans.*, vol. 96, pp. 51–59, Jan. 2020.

[19] H. Yu, N. Zhao, P. Wang, H. Chen, and C. Li, "Chaos-enhanced synchronized bat optimizer," *Appl. Math. Model.*, vol. 77, pp. 1201–1215, Jan. 2020.

[20] G. Yildizdan and Ö. K. Baykan, "A novel modified bat algorithm hybridizing by differential evolution algorithm," *Expert Syst. Appl.*, vol. 141, Mar. 2020, Art. no. 112949.

[21] H. Wei-Chiang, L. Ming-Wei, G. Jing, and Z. Yang, "Novel chaotic bat algorithm for forecasting complex motion of floating platforms," *Appl. Math. Model.*, vol. 72, pp. 425–443, Aug. 2019.

[22] A. Chakri, R. Khelif, M. Benouaret, and X.-S. Yang, "New directional bat algorithm for continuous optimization problems," *Expert Syst. Appl.*, vol. 69, pp. 159–175, Mar. 2017.

[23] X. Shan and H. Cheng, "Modified bat algorithm based on covariance adaptive evolution for global optimization problems," *Soft Comput.*, vol. 22, no. 16, pp. 5215–5230, Aug. 2018.

[24] S. Dhar and M. K. Kundu, "A novel method for image thresholding using interval type-2 fuzzy set and Bat algorithm," *Appl. Soft Comput.*, vol. 63, pp. 154–166, Feb. 2018.

[25] L. F. Zhu, J. S. Wang, and H. Y. Wang, "A novel clustering validity function of FCM clustering algorithm," *IEEE Access*, vol. 7, pp. 152289–152315, 2019.

[26] R. J. Kuo, Y. J. Syu, Z.-Y. Chen, and F. C. Tien, "Integration of particle swarm optimization and genetic algorithm for dynamic clustering," *Inf. Sci.*, vol. 195, pp. 124–140, Jul. 2012.

[27] L. P. Yang, F. Z. Wang, and C. M. Fan, "A text clustering algorithm based on Weedsand differential optimization," *Int. J. Database Theory Appl.*, vol. 9, no. 12, pp. 121–130, Dec. 2016.

[28] M. Fathian, B. Amiri, and A. Maroosi, "Application of honey-bee mating optimization algorithm on clustering," *Appl. Math. Comput.*, vol. 190, no. 2, pp. 1502–1513, Jul. 2007.

[29] R. Wang, Y. Zhou, and S. Qiao, *Flower Pollination Algorithm With Bee Pollinator for Cluster Analysis*. Amsterdam, The Netherlands: Elsevier, 2016.

[30] O. Tarkhaneh and I. Moser, "An improved differential evolution algorithm using Archimedean spiral and neighborhood search based mutation approach for cluster analysis," *Future Gener. Comput. Syst.*, vol. 101, pp. 921–939, Dec. 2019.

[31] M.-S. Yang, C.-Y. Lai, and C.-Y. Lin, "A robust EM clustering algorithm for Gaussian mixture models," *Pattern Recognit.*, vol. 45, no. 11, pp. 3950–3961, Nov. 2012.

[32] B. Amiri, M. Fathian, and A. Maroosi, "Application of shuffled frog-leaping algorithm on clustering," *Appl. Math. Comput.*, vol. 45, nos. 1–2, pp. 199–209, 2009.

[33] H. Y. Ji, "Improved bat algorithm and its application in robot path planning," M.S. thesis, Graduate School, Henan Univ., Kaifeng, China, 2019, pp. 13–14.

[34] Z. Hongguo, Z. Changwen, H. Xiaohui, and L. Xiang, "Path planner for unmanned aerial vehicles based on modified PSO algorithm," in *Proc. Int. Conf. Inf. Autom.*, Jun. 2008, pp. 541–544.

[35] J. J. Liang, P. N. Suganthan, and K. Deb, "Novel composition test functions for numerical global optimization," in *Proc. IEEE Swarm Intell. Symposium SIS*, Jun. 2005, pp. 68–75.

[36] Y. Marinakis, M. Marinaki, M. Doumpos, N. Matsatsinis, C. Zopounidis, "A hybrid stochastic genetic-GRASP algorithm for clustering analysis," *Oper. Res.*, vol. 8, no. 1, pp. 33–46, 2008.

[37] J. Senthilnath, S. N. Omkar, and V. Mani, "Clustering using firefly algorithm: Performance study," *Swarm Evol. Comput.*, vol. 1, no. 3, pp. 164–171, Sep. 2011.

[38] J. Senthilnath, S. Kulkarni, J. A. Benediktsson, and X. S. Yang, "A novel approach for multispectral satellite image classification based on the bat algorithm," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 4, pp. 599–603, Apr. 2016.

[39] A. Dalli, "Adaptation of the F-measure to cluster based lexicon quality evaluation," in *Proc. EACL Workshop Eval. Initiatives Natural Lang. Process. Eval. Methods, Metrics Resour. Reusable? Evalinitiatives*, 2003, pp. 51–56.

[40] W. M. Rand,"Objective criteria for the evaluation of clustering methods," *J. Amer. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, Dec. 1971.

[41] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, 1985.

[42] H. Li, *Statistical Learning Method*. Beijing, China: Tsinghua Univ. Press,2012.

[43] A. N. Jadhav and N. Gomathi, "Kernel-based exponential grey wolf optimizer for rapid centroid estimation in data clustering," *Jurnal Teknologi*, vol. 78, no. 11, pp. 65–74, Apr. 2016.
[44] L.-Y. Chuang, C.-J. Hsiao, and C.-H. Yang, "Chaotic particle swarm optimization for data clustering," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 14555–14563, Nov. 2011.
[45] T. Niknam, B. Amiri, J. Olamaei, and A. Arefi, "An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering," *J. Zhejiang Univ.-Sci. A*, vol. 10, no. 4, pp. 512–519, Apr. 2009.

L. F. ZHU is currently pursuing the master's degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China.

J. S. WANG (Member, IEEE) received the B.Sc. and M.Sc. degrees in control science from the University of Science and Technology Liaoning, China, in 1999 and 2002, respectively, and the Ph.D. degree in control science from the Dalian University of Technology, China, in 2006. He is currently a Professor and a Master's Supervisor with the School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interests include modeling of complex industry process, intelligent control, and computer integrated manufacturing.

H. Y. WANG is currently pursuing the bachelor's degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China.

S. S. GUO is currently pursuing the master's degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China.

M. W. GUO is currently pursuing the master's degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China.

W. XIE is currently pursuing the master's degree with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, China.