



A novel bacterial foraging optimization algorithm for feature selection

Yu-Peng Chen^{a,b}, Ying Li^{a,b}, Gang Wang^{a,b,*}, Yue-Feng Zheng^{a,b}, Qian Xu^{a,b}, Jia-Hao Fan^{a,b}, Xue-Ting Cui^{a,b}

^a College of Computer Science and Technology, Jilin University, Changchun 130012, PR China

^b Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun, PR China



ARTICLE INFO

Article history:

Received 2 April 2016

Revised 6 April 2017

Accepted 7 April 2017

Available online 8 April 2017

Keywords:

Bacterial foraging optimization algorithm

Feature selection

Classification

Support vector machine

ABSTRACT

Bacterial foraging optimization (BFO) algorithm is a new swarming intelligent method, which has a satisfactory performance in solving the continuous optimization problem based on the chemotaxis, swarming, reproduction and elimination-dispersal steps. However, BFO algorithm is rarely used to deal with feature selection problem. In this paper, we propose two novel BFO algorithms, which are named as adaptive chemotaxis bacterial foraging optimization algorithm (ACBFO) and improved swarming and elimination-dispersal bacterial foraging optimization algorithm (ISEDBFO) respectively. Two improvements are presented in ACBFO. On the one hand, in order to solve the discrete problem, data structure of each bacterium is redefined to establish the mapping relationship between the bacterium and the feature subset. On the other hand, an adaptive method for evaluating the importance of features is designed. Therefore the primary features in feature subset are preserved. ISEDBFO is proposed based on ACBFO. ISEDBFO algorithm also includes two modifications. First, with the aim of describing the nature of cell to cell attraction-repulsion relationship more accurately, swarming representation is improved by means of introducing the hyperbolic tangent function. Second, in order to retain the primary features of eliminated bacteria, roulette technique is applied to the elimination-dispersal phase.

In this study, ACBFO and ISEDBFO are tested with 10 public data sets of UCI. The performance of the proposed methods is compared with particle swarm optimization based, genetic algorithm based, simulated annealing based, ant lion optimization based, binary bat algorithm based and cuckoo search based approaches. The experimental results demonstrate that the average classification accuracy of the proposed algorithms is nearly 3 percentage points higher than other tested methods. Furthermore, the improved algorithms reduce the length of the feature subset by almost 3 in comparison to other methods. In addition, the modified methods achieve excellent performance on wilcoxon signed-rank test and sensitivity-specificity test. In conclusion, the novel BFO algorithms can provide important support for the expert and intelligent systems.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction

In machine learning and statistics, dimensionality reduction or dimension reduction is the process of reducing the number of variables under consideration or discovering compact representations of high-dimensional data. Dimensionality reduction can be divided into feature extraction and feature selection (Guyon, Gunn, Nikravesh, & Zadeh, 2006; Guyon et al., 2007; Jain & Zongker, 1997; Kudo & Sklansky, 2000). Feature extraction involves reduc-

ing the amount of resources required to describe a large set of data. A feature selection algorithm can be seen as the combination of a search technique for proposing new feature subsets, along with an evaluation measure which scores the different feature subsets. Feature selection algorithms are divided into three main categories which are wrappers, filters and embedded methods. Wrapper methods use a predictive model to score feature subsets. Each new subset is used to train a model, which is tested on a hold-out set (Apollonia, Leguizamóna, & Albab, 2015; Kohavi & John, 1997; Vignoloa, Milonea, & Scharcanskib, 2013). Filter methods use a proxy measure instead of the error rate to score a feature subset (Hsu, Hsieh, & Lu, 2011). Embedded methods are a catch-all group of techniques which perform feature selection as part of the model construction process (Peralta, & Soto, 2014). However, in traditional statistics, the most popular form of feature selection is metaheuristic algorithm, which is a wrapper technique

* Corresponding author at: College of Computer Science and Technology, Jilin University, Changchun 130012, PR China.

E-mail addresses: yeppchan@hotmail.com (Y.-P. Chen), liying.jlu.cs@gmail.com (Y. Li), wanggang.jlu@gmail.com (G. Wang), honest_zjf@hotmail.com (Y.-F. Zheng), quincyxu@hotmail.com (Q. Xu), jihanfan@hotmail.com (J.-H. Fan), X.T.Cui@hotmail.com (X.-T. Cui).

(Bonabeau, Dorigo, & Theraulaz, 1999; Camazine et al., 2003; Kudo & Sklansky, 2000; Tahir & Smith, 2010; Wanga, Hedarb, Wanga, & Mac, 2012). Common metaheuristic algorithms include particle swarm optimization (PSO) (Kennedy & Eberhart, 1995), genetic algorithm (GA) (Goldberg, 1989), simulated annealing (SA) (Kirkpatrick, Gelatt, & Vecchi, 1983), ant lion optimization (ALO) (Mirjalili, 2015), binary bat algorithm (BBA) (Nakamura et al., 2012) and cuckoo search (CS) (Yang & Deb, 2014). A few algorithms mentioned above have achieved successful results on feature selection problem. The current research status is described as follows:

Huang and Dun (2008) proposed a novel PSO-SVM model that hybridized the PSO and support vector machines (SVMs) to improve the classification accuracy with a small and appropriate feature subset. This optimization mechanism combined the discrete PSO with the continuous-valued PSO to simultaneously optimize the input feature subset selection and the SVM kernel parameter setting. Wang, Yang, Teng, Xia, and Jensen (2007) presented a new feature selection mechanism, where one investigated how PSO can be applied to find optimal feature subsets or rough sets. Huang and Wang (2006) designed a new GA approach for feature selection and parameters optimization. The objective of this research was to simultaneously optimize the parameters and feature subset without degrading the SVM classification accuracy. Zhu and Guan (2004) integrated relative importance factor (RIF) into the modular GA based scheme by employing it in finding a suitable feature subset for each class module. This new feature selection technique was designed to find less relevant features in the input domain of each class module. Lin, Lee, Chen, and Tseng (2008) developed a SA approach for parameter determination and feature selection in the SVM, termed SA-SVM. The aim of this approach was to obtain the better parameter values while also finding a subset of features that did not degrade the SVM classification accuracy. Meiri and Zahavi (2006) employed the SA approach for specifying a large-scale linear regression model. Rodrigues et al. (2014) presented a wrapper feature selection approach based on BA and Optimum-Path Forest (OPF), in which they modeled the problem of feature selection as a binary-based optimization technique, guided by BA using the OPF accuracy over a validating set as the fitness function to be maximized. Nakamura et al. (2012) proposed a new nature-inspired feature selection technique based on the bats behavior. This wrapper approach combined the power of exploration of the bats together with the speed of the Optimum-Path Forest classifier to find the set of features that maximizes the accuracy in a validating set.

BFO algorithm mimics the foraging strategy of *Escherichia coli* bacteria (Passino, 2002). The core idea of this strategy is to find the maximum nutrition in the unit time. The chemotaxis operation mimics the searching movement by taking small steps. Tumbling, moving and swimming are important phases for chemotaxis operation. Swarming process realizes the cell to cell communication. BFO algorithm has been effectively applied to solve real world continuous optimization problem. To the best of our knowledge, few studies have been successfully applied to feature selection problem. Panda and Naik (2015) designed a modified bacterial foraging optimization algorithm called adaptive crossover BFO algorithm, which incorporated adaptive chemotaxis and also inherited the crossover mechanism of genetic algorithm. The proposed algorithm was used for finding optimal principal components for dimension reduction in linear discriminated analysis (LDA) based face recognition. Jakhar, Kaur, and Singh (2011) proposed a novel feature selection algorithm based on BFO. The algorithm was applied to coefficients extracted by discrete cosine transforms (DCT).

Unfortunately, three problems are not considered in the studies mentioned above. Firstly, all features are selected randomly by using the same probability. Therefore the primary features cannot be easily selected into the feature subset. Secondly, original swarming formula cannot accurately reveal the cell to cell attraction-

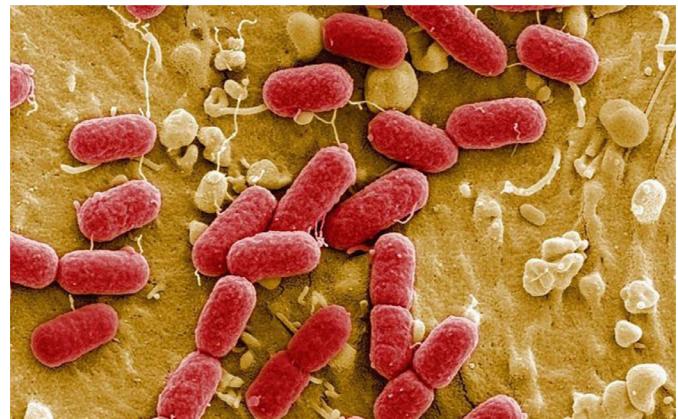


Fig. 1. *E. coli* bacteria.

repulsion relationship. Thus the bacteria are too close to each other in the searching process. Thirdly, the primary features of the eliminated bacteria are not retained in elimination-dispersal stage. This approach seriously degrades the efficiency of searching optimal feature subset.

In this paper, three corresponding improvements are proposed for solving the problems mentioned above. The first improvement is presented in ACBFO. The second and third improvements are applied to ISEDBFO.

Firstly, each bacterium is constructed in a format of binary data structure. Moreover every feature is mapped to a selected probability. Therefore the primary features can be easily selected into the feature subset. Secondly, hyperbolic tangent function is introduced to improve the swarming equation. The modified equation can strengthen the ability of bacterial dispersion. As a result, the optimal feature searching range is extended. Finally, the roulette method is used to preserve the primary features of the eliminated bacteria. Thereupon, the capability of searching optimal feature subset is enhanced by applying this approach.

The rest of paper is organized as follows: Section 2 introduces the basic concept of BFO algorithm. Section 3 presents the improved ACBFO and ISEDBFO algorithms and their implementation. Section 4 describes the experimental results and analysis on feature selection. Finally, Section 5 draws some conclusions for this paper.

2. Basic concept of BFO

Classical BFO algorithm describes the *E. coli* foraging process. This method includes chemotaxis, swarming, reproduction and elimination-dispersal operations. Fig. 1 presents the *E. coli* bacteria under a microscope. The parameters of classical BFO are showed in Table 1.

The detailed contents of these four operations are introduced as follows:

2.1. Chemotaxis

Bacterial foraging strategy is simulated by this phase. In the first place, bacteria will tumble to change present direction for a period of time. After that, bacteria are going to move a step size. If bacteria find rich nutrients and then they will keep swimming in the same direction. Theoretically, we suppose θ is the position of a bacterium and $\theta^i(j, k, l)$ means the i th bacterium in the j th chemotaxis, k th reproduction, l th elimination-dispersal procedure.

Table 1
Parameters of classical BFO.

Parameter	Description
p	Number of features
S	Bacterial number
N_{re}	Number of reproductive steps
N_{ed}	Number of elimination-dispersal steps
N_c	Number of chemotactic steps
N_s	Number of swimming steps
θ	A bacterium on the optimization domain
θ^i	The i th bacterium position
θ_f^i	The f th component of the θ^i
$\theta^i(j, k, l)$	The i th bacterium in the j th chemotaxis, k th reproduction, l th elimination-dispersal procedure
J_{last}	Fitness value of last time swarming
$C(i)$	A step size in the direction of the tumble for the i th bacterium
P_{ed}	Elimination probability
$\Delta(i)$	A random value on $[-1, 1]$ for the i th bacterium
$d_{attract}$ and $w_{attract}$	Coefficients of attraction-repulsion
$h_{repellant}$ and $w_{repellant}$	Coefficients of repulsion-repulsion

The definition of tumbling is described as follows:

$$\varphi(i) = \frac{\Delta(i)}{\sqrt{\Delta(i)^T \Delta(i)}} \quad (1)$$

where $\Delta(i), i = 1, 2, \dots, S$ is a random vector. Each element $\Delta_m(i), m = 1, 2, \dots, p$ of $\Delta(i)$ is a random number between -1 and 1 . S is the number of bacteria. The expression for bacterial position updating is defined as:

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i)\varphi(i) \quad (2)$$

where $C(i), i = 1, 2, \dots, S$ is a moving step size in the swimming phase.

2.2. Swarming

Cell to cell signaling is simulated by this behavior. If bacteria find high amounts of nutrients, they will release chemical substances to attract other bacteria. If they are in danger, they will tend to repel each other. This social behavior is modeled as follows:

$$Jcc(\theta, \theta^i(j, k, l)) = \sum_{i=1}^S \left[-d_{attract} \exp \left(-w_{attract} \sum_{f=1}^p (\theta_f - \theta_f^i)^2 \right) \right] \\ + \sum_{i=1}^S \left[h_{repellant} \exp \left(-w_{repellant} \sum_{f=1}^p (\theta_f - \theta_f^i)^2 \right) \right] \quad (3)$$

where $\theta = [\theta_1, \dots, \theta_p]^T$ is a bacterium on the optimization domain and θ_f^i is the f th component of the i th bacterium position θ^i (for convenience we omit some of the indices). $Jcc(\theta, \theta^i(j, k, l))$ is the cell to cell communication value which will be added to the result of fitness function in the chemotaxis phase j ; p is the number of problem dimension; S is the number of bacteria; $d_{attract}$, $w_{attract}$, $h_{repellant}$, $w_{repellant}$ are different coefficients which represent the strength of attraction or repulsion. Considering the swarming effect, the fitness value of i th bacterium is described as follows:

$$J(i, j, k, l) = J(i, j, k, l) + Jcc(\theta, \theta^i(j, k, l)) \quad (4)$$

2.3. Reproduction

After N_c chemotactic steps, a reproduction step is taken. We assume that S is a positive even integer. S_r is the number of population members who have had sufficient nutrients so that they will

reproduce (split in two) with no mutations.

$$S_r = \frac{S}{2} \quad (5)$$

The health of a bacterium represents its accumulated cost. Higher accumulated cost means that a bacterium did not get as many nutrients during its lifetime of foraging. Therefore this bacterium is not healthy and unlikely to reproduce. Bacteria are sorted in descending order according to their health in this stage, and then the S_r least healthy bacteria die and the other S_r healthiest bacteria each split into two bacteria, which are placed at the same location.

2.4. Elimination-dispersal

Some bacteria die because of the adverse environment, e.g. rising of the temperature may kill a group of bacteria in a certain range. This process is simulated by the dispersal of some bacteria with a small probability P_{ed} . Simultaneously, some new bacteria are randomly generated for replacement. The flowchart of standard BFO algorithm is modeled as follows (Fig. 2):

2.5. Support vector machines

SVMs are a core machine learning technology. They have strong theoretical foundations and excellent empirical successes. They have been applied to tasks such as handwritten digit recognition, object recognition, and text classification. SVMs use a linear model to implement nonlinear class boundaries through some non-linear mapping input vectors into a high-dimensional feature space. In the new space, an optimal separating hyperplane is constructed. Thus, SVM is known as the algorithm that finds a special kind of linear model, the maximum margin hyperplane. The maximum margin hyperplane gives the maximum separation between decision classes. The training examples that are closest to the maximum margin hyperplane are called support vectors. All other training examples are irrelevant to define the binary class boundaries (Akay, 2009; Alonso-Atienza et al., 2012; Neumann, Schnörr, & Steidl, 2005).

The application of SVM involves two steps. The first step is to choose the appropriate kernel function. The second step is to train the kernel function with a data set. Thereby the selection of kernel function is the key to the SVM classifier.

In this paper, LIBSVM is applied to classification operation (Glasmachers & Igel, 2008). LIBSVM supports four kinds of kernel functions, which are linear kernel function (Eq. (6)), polynomial kernel function (Eq. (7)), RBF kernel function (Eq. (8)) and sigmoid kernel function (Eq. (9)). The expressions of these kernel functions are described as follows:

$$K(x_i, x_j) = x_i^T x_j \quad (6)$$

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0 \quad (7)$$

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0 \quad (8)$$

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r) \quad (9)$$

where γ , r and d are kernel parameters. For the purpose of solving feature selection problem, the original fitness function of BFO is replaced by the SVM classifier. Thereupon, the feature subset with the highest accuracy is the solution that we want to search.

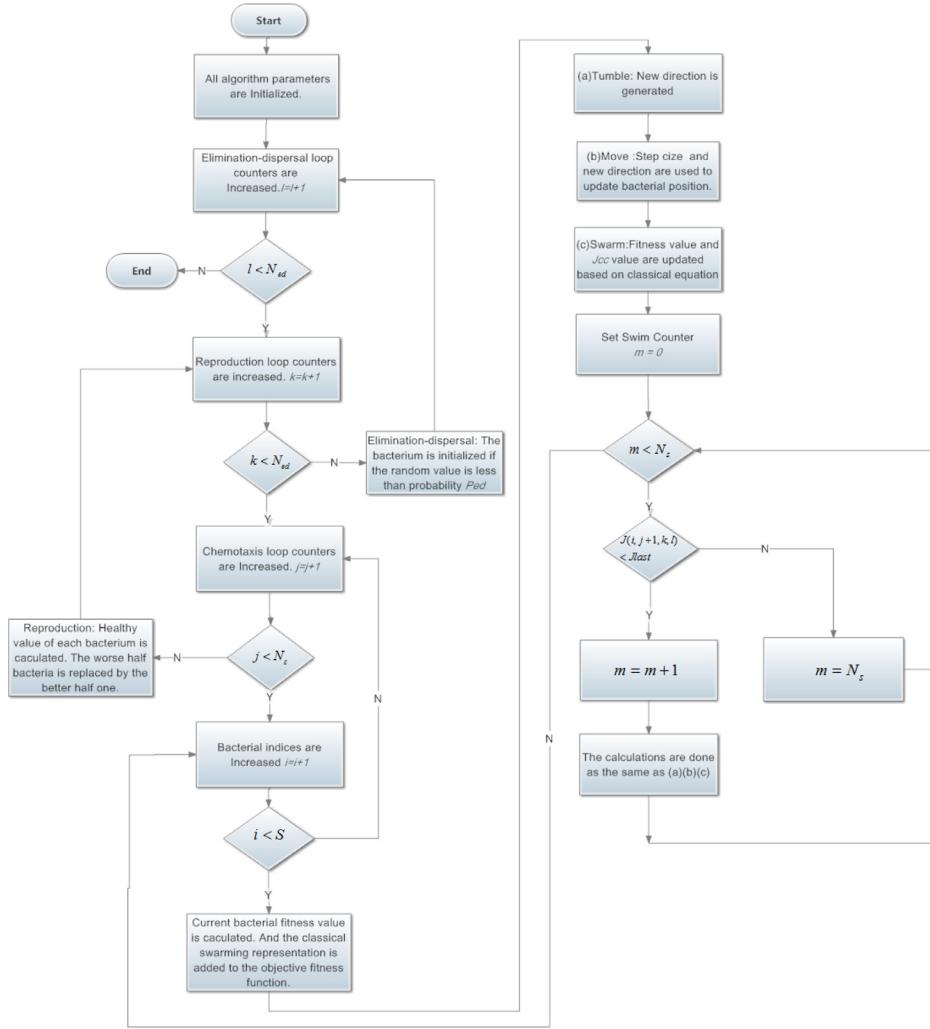


Fig. 2. Flowchart of standard BFO.

Table 2
Improved parameters of ACBFO.

Parameter	Description
$\Delta(f)$	A random value on $[0, 1]$ for the f th feature
$C(f)$	Selected probability of the f th feature
α	The initial value of $C(f)$
γ	A coefficient variation for $C(f)$

3. Improved BFO algorithms

3.1. The improvements of ACBFO

In this section two improvements of ACBFO are introduced. First, in order to make the ACBFO algorithm solve the discrete problem, data structure of each bacterium is redefined to establish the relationship between the bacterium and the feature subset. Second, an adaptive method is designed for evaluating the importance of features. Table 2 shows the improved parameters for ACBFO.

3.1.1. Binary data structure

In order to solve the discrete problem based on BFO algorithm, the encoding concept of GA is applied to classical BFO algorithm. Each bacterium is constructed in a format of binary data structure. We assume that the number of total features in a data set is p .

Meanwhile the number of dimensions in a bacterium is the same as the number of features. Furthermore, the value of each dimension means that the corresponding feature is selected or not. Value 1 means this feature is selected and value 0 is just the opposite. In this way, the mapping between the data set and the bacterium is established. Fig. 3 shows the improved data structure of ACBFO method.

3.1.2. Improved chemotaxis step

In the original BFO method, $\Delta(i), i = 1, 2, \dots, S$ is a random vector whose value lies in the scope $[-1, 1]$. C is defined as a constant step size. These two parameters are used for updating bacteria position. However they are not suitable for feature selection problem.

In order to solve the problems mentioned above, the parameters $\Delta(i)$ and C are improved. $\Delta(i)$ is redefined as $\Delta(f), f = 1, 2, \dots, p$ representing a group of random values between 0 and 1. Moreover, the value of $\Delta(f)$ is generated after tumbling. At the same time, variable C is redefined as a vector $C(f), f = 1, 2, \dots, p$ storing the selected probability value of each feature. If $C(f)$ approaches to 1, then the f th feature is more likely to be selected. In other words, the level of importance for this feature is high.

The improved bacteria position updating formula is showed in Eq. (10):

$$\theta_f^i(j+1, k, l) = \begin{cases} 1 & \Delta(f) < C(f) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

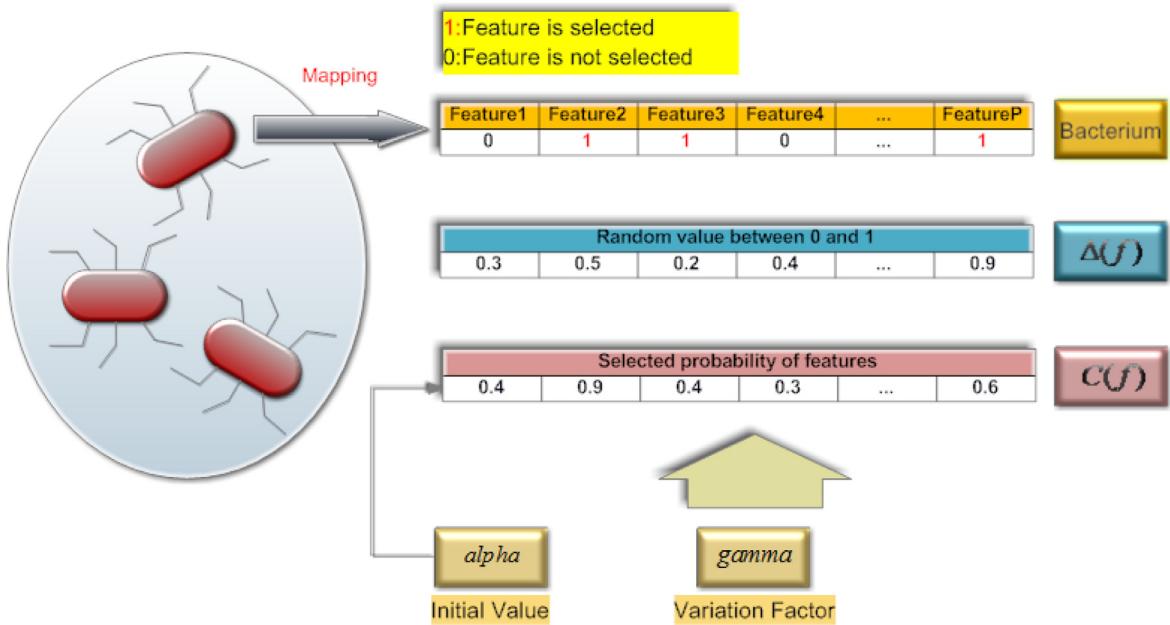


Fig. 3. Improved data structure of ACBFO method.

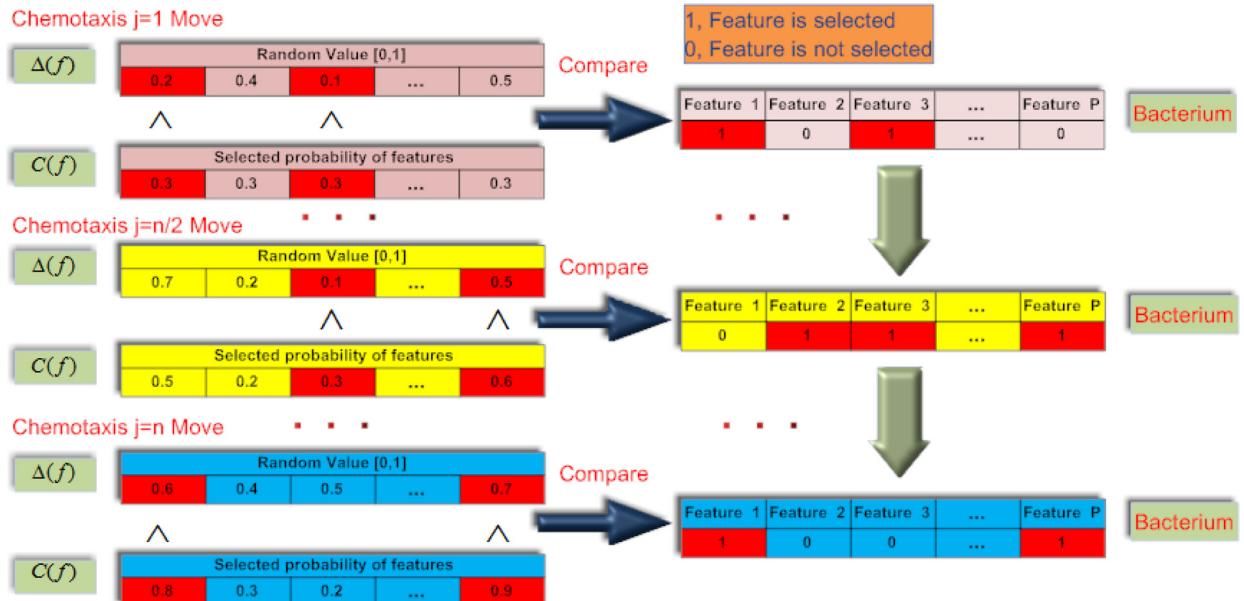


Fig. 4. Improved BFO Chemotaxis process.

where $\theta = [\theta_1, \dots, \theta_p]^T$ is a bacterium on the optimization domain and θ_f^i is the f th component of the i th bacterium position θ^i (for convenience we omit some of the indices). $\theta_f^i(j+1, k, l)$ means the f th dimension of i th bacterium in the $j+1$ th chemotaxis, k th reproduction, l th elimination-dispersal procedure. p means the number of features.

At the start of the algorithm, a constant value α between 0 and 1 is assigned to $C(f)$. Because we assume that the level of importance for total features is very low, thus the initial value of $C(f)$ (α) is close to 0. In every chemotaxis step, $C(f)$ is compared with the corresponding random value $\Delta(f)$. If $\Delta(f)$ is smaller than $C(f)$, then the corresponding f th dimension value of current bacterium is changed to 1. In other words, the f th feature of current bacterium is selected. Otherwise the corresponding f th dimension value of current bacterium is changed to 0. It means that

the f th feature of current bacterium is not selected. Finally, we get the selected features at every chemotaxis step. The improved BFO chemotaxis process is illustrated in Fig. 4.

The flowchart of ACBFO (Fig. 5) is described as follows:

We introduce a variable γ which is between 0 and 1 to adjust the selected probability of each feature. The formula for γ is given in Eq. (11).

$$\gamma = \frac{1 - \alpha}{(S \times N_c \times N_{re} \times N_{ed}) \times \cos(\alpha)} \quad (11)$$

where α is the initial value for $C(f)$; S is the total number of bacteria; N_c is the iteration number of chemotaxis step; N_{re} is the number of reproductive steps; N_{ed} is the cycling times of elimination-dispersal phase; $S \times N_c \times N_{re} \times N_{ed}$ represents the total number of iterations.

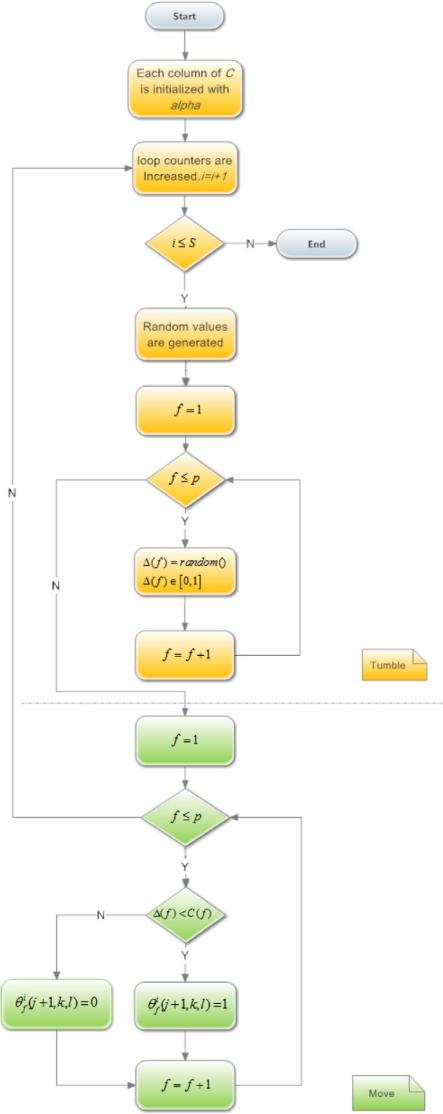


Fig. 5. Improved chemotaxis flowchart.

Because the initial value of $C(f)$ is α and the upper limit of $C(f)$ is 1, therefore the maximum increasing range of $C(f)$ is $1 - \alpha$. Moreover, we divide $1 - \alpha$ by $S \times N_c \times N_{re} \times N_{ed}$ to get maximum increasing range of each operation step. In particular, in order to increase the value of γ , $\cos(\alpha)$ is applied to the Eq. (11).

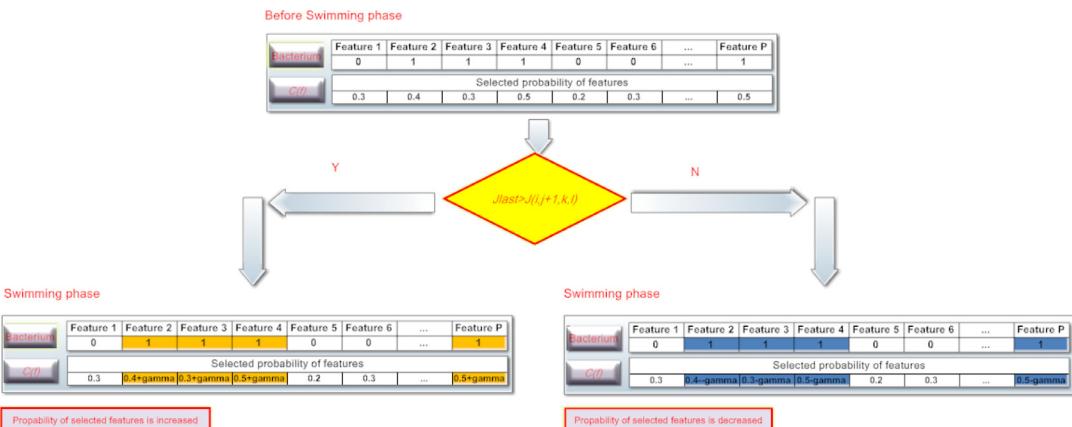


Fig. 6. Improved BFO swimming process.

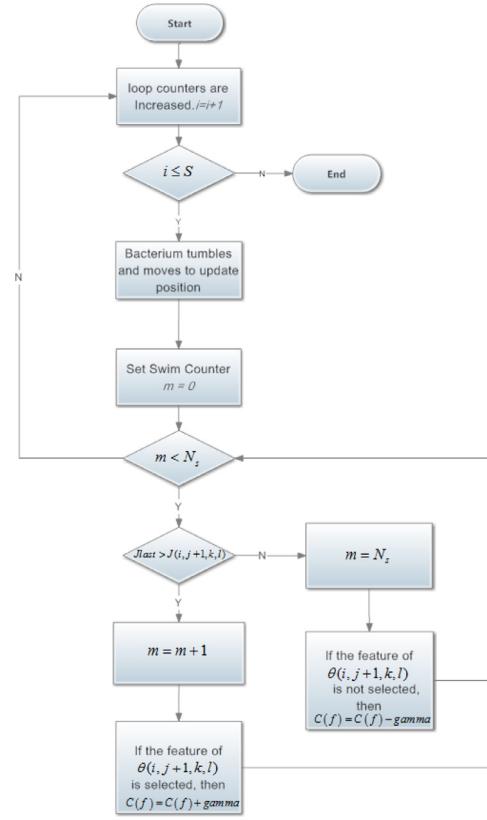


Fig. 7. Improved BFO swimming flowchart.

In the bacterial swimming stage, if current fitness value of a bacterium is less than the local optimal value, then this result means the selected feature is important. Furthermore, the probability of selected feature is increased by variable γ . On the contrary, the selected feature is not important. Therefore the probability of selected feature is decreased by variable γ . Finally, the features with high selected probability are the important ones. In this way, the importance of bacteria is evaluated. The detailed formula is described in Eq. (12).

$$C(f) = \begin{cases} C(f) + \gamma & \text{if } J_{last} > J(i, j+1, k, l) \text{ and } \theta_f^i(j+1, k, l) = 1 \\ C(f) - \gamma & \text{if } J_{last} \leq J(i, j+1, k, l) \text{ and } \theta_f^i(j+1, k, l) = 1 \end{cases} \quad (12)$$

where $J(i, j+1, k, l)$ is the i th bacterium at $j+1$ th chemotaxis stage, k th reproduction phase, l th elimination-dispersal period fit-

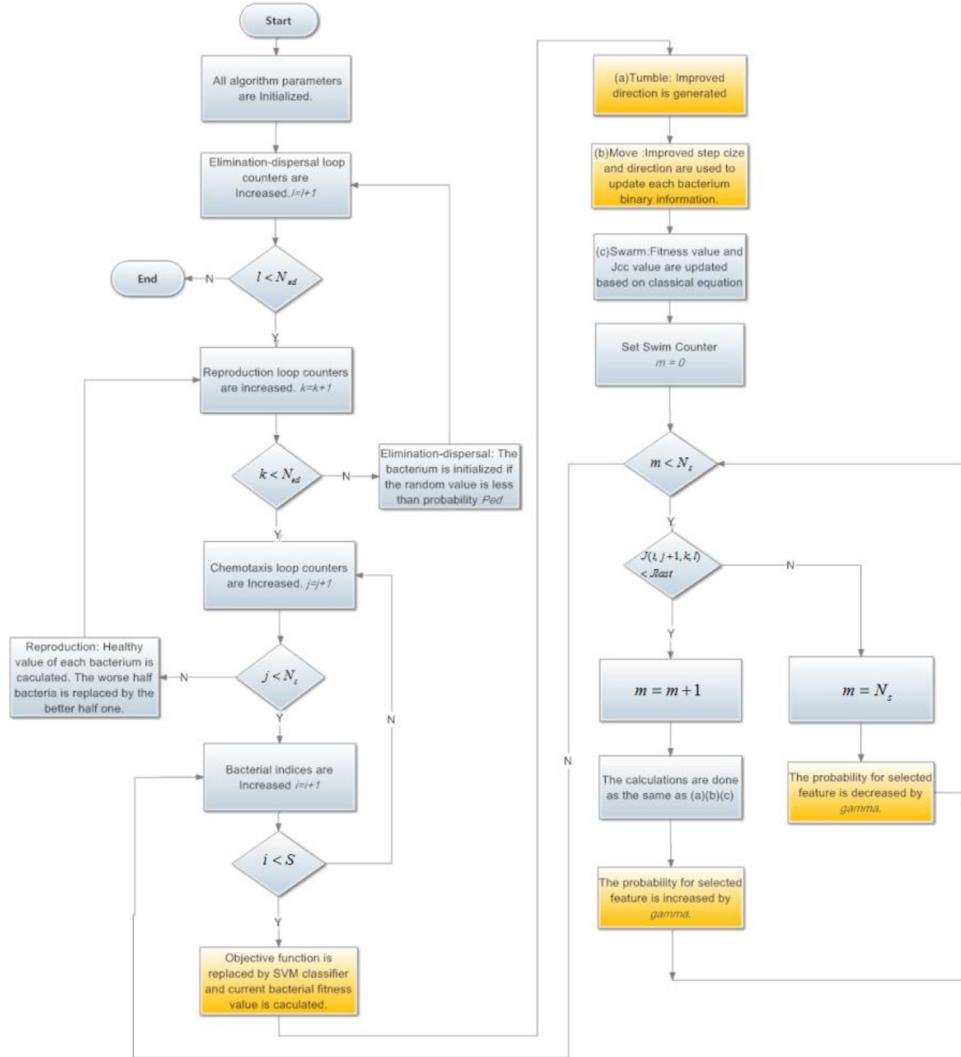


Fig. 8. Flowchart of ACBFO.

ness value; J_{last} is the local optimal fitness value; definition of $\theta_f^i(j+1, k, l)$ refers to Eq. (10). The improved BFO swimming process and flowchart are showed as follows (Figs. 6 and 7):

The flowchart of ACBFO algorithm is illustrated in Fig. 8. The yellow blocks represent the improvements of ACBFO.

3.1.3. Time computation complexity analysis of ACBFO

The time computation complexity of ACBFO algorithm can be presented as $O(L \times (K \times (J \times S \times (M + N_1) + P) + Q))$. In this expression, L means iteration number for elimination-dispersal; K represents reproduction iteration time; J describes the number of chemotaxis activity; S is the total bacterial number; M is the calculation time for SVM model training and prediction; N_1 represents the time complexity for chemotaxis operations based on evaluation method; P expresses the calculation time for reproduction process; Q implies the computation time for elimination-dispersal module.

The time computation complexity of BFO algorithm can be presented as $O(L \times (K \times (J \times S \times M + P) + Q))$. We can see that the computational complexity of ACBFO is higher than BFO. However, because N_1 only contains addition and subtraction operations according to Eq. (12), therefore ACBFO does not significant increase the time computation complexity.

3.2. The improvements of ISEDBFO

In this section two improvements of ISEDBFO are introduced. First, swarming representation is improved by means of introducing the hyperbolic tangent function. Second, in order to retain the primary features of eliminated bacteria, roulette technique is applied to the elimination-dispersal phase.

3.2.1. Modified swarming representation

The swarming characteristic of *E. coli* bacteria is described in the classical paper (Passino, 2002). Bacteria release some type of chemical reagent to attract other bacteria to come together when they find the high amount of nutrients. At the same time, bacteria also release repellent chemical reagent to prevent bacteria from gathering too close. However, the original swarming equation cannot describe the cell to cell attraction-repulsion relationship accurately when bacteria gathered at the same point. Furthermore, the irrationality of the original swarming equation is proved as follows. We suppose that a bacterium swims from position A to position B, and the distance between the bacterium and the bacteria group is reduced. Thereby the following inequality is deduced.

$$\sum_{i=1}^S \sum_{f=1}^p (\theta_f^A - \theta_f^i)^2 > \sum_{i=1}^S \sum_{f=1}^p (\theta_f^B - \theta_f^i)^2 \quad (13)$$

Inequality (13) is multiplied by attractant width $-w_{attract}$.

$$\sum_{i=1}^S \left(-w_{attract} \sum_{f=1}^p (\theta_f^A - \theta_f^i)^2 \right) < \sum_{i=1}^S \left(-w_{attract} \sum_{f=1}^p (\theta_f^B - \theta_f^i)^2 \right) \quad (14)$$

Because $\exp(x)$ function is a monotonically increasing function, then inequality (14) is transformed as follows:

$$\begin{aligned} & \sum_{i=1}^S \exp \left(-w_{attract} \sum_{f=1}^p (\theta_f^A - \theta_f^i)^2 \right) \\ & < \sum_{i=1}^S \exp \left(-w_{attract} \sum_{f=1}^p (\theta_f^B - \theta_f^i)^2 \right) \end{aligned} \quad (15)$$

Inequality (15) is multiplied by attractant depth $-d_{attract}$.

$$\begin{aligned} & \sum_{i=1}^S \left[-d_{attract} \exp \left(-w_{attract} \sum_{f=1}^p (\theta_f^A - \theta_f^i)^2 \right) \right] \\ & > \sum_{i=1}^S \left[-d_{attract} \exp \left(-w_{attract} \sum_{f=1}^p (\theta_f^B - \theta_f^i)^2 \right) \right] \end{aligned} \quad (16)$$

The attraction is decreased when bacteria come close together according to inequality (16). In the same way, repellent related inequality is derived based on Eq. (13).

$$\Rightarrow \sum_{i=1}^S \left(-w_{repellant} \sum_{f=1}^p (\theta_f^A - \theta_f^i)^2 \right) < \sum_{i=1}^S \left(-w_{repellant} \sum_{f=1}^p (\theta_f^B - \theta_f^i)^2 \right) \quad (17)$$

$$\begin{aligned} & \Rightarrow \sum_{i=1}^S \exp \left(-w_{repellant} \sum_{f=1}^p (\theta_f^A - \theta_f^i)^2 \right) \\ & < \sum_{i=1}^S \exp \left(-w_{repellant} \sum_{f=1}^p (\theta_f^B - \theta_f^i)^2 \right) \end{aligned} \quad (18)$$

$$\begin{aligned} & \Rightarrow \sum_{i=1}^S \left[h_{repellant} \exp \left(-w_{repellant} \sum_{f=1}^p (\theta_f^A - \theta_f^i)^2 \right) \right] \\ & < \sum_{i=1}^S \left[h_{repellant} \exp \left(-w_{repellant} \sum_{f=1}^p (\theta_f^B - \theta_f^i)^2 \right) \right] \end{aligned} \quad (19)$$

Repulsion is increased after the bacterium close to the bacteria group according to inequality (19). In classical BFO algorithm, the conditions $w_{repellant} = 50w_{attract}$ and $d_{attract} = h_{repellant}$ are satisfied. The distance between the bacterium and the bacteria group is proportional to the attraction and inversely proportional to the repulsion. Attraction is dominant in the whole chemotaxis process. We suppose that all bacteria are gathered at the same point, which means $\sum_{f=1}^p (\theta_f - \theta_f^i)^2 = 0$. Following results are deduced based on Eq. (3).

$$\Rightarrow \sum_{i=1}^S [-d_{attract} \exp(0)] + \sum_{i=1}^S [h_{repellant} \exp(0)] \quad (20)$$

$$\begin{aligned} & \Rightarrow \sum_{i=1}^S [-d_{attract} + h_{repellant}] \\ & \Rightarrow 0 \end{aligned} \quad (21)$$

Apparently, the results of the above deduction are not completely consistent with the real bacterial activity. Especially, the

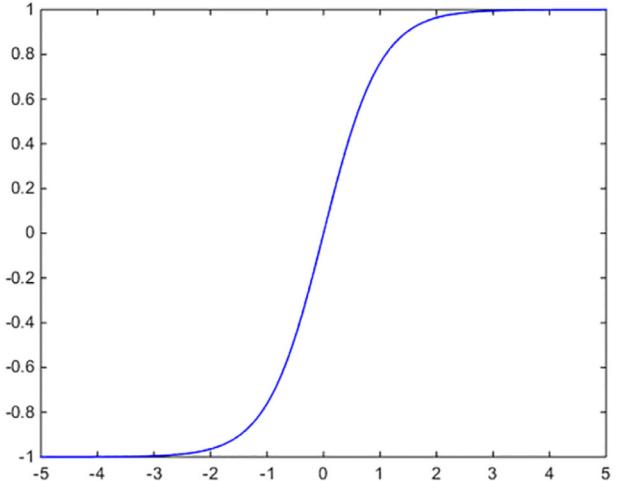


Fig. 9. $\text{Tanh}(x)$ function.

attraction and repulsion should not be equal when bacteria come together. In other words, the repulsion should be far greater than the attraction at this point. According to the original design, bacteria will be easily trapped into local optimal value. In this paper, the improved swarming equation Eq. (22) overcomes the above problems. This formula is not only more suitable for describing the behavior of bacteria, but also to improve the efficiency of searching optimal value.

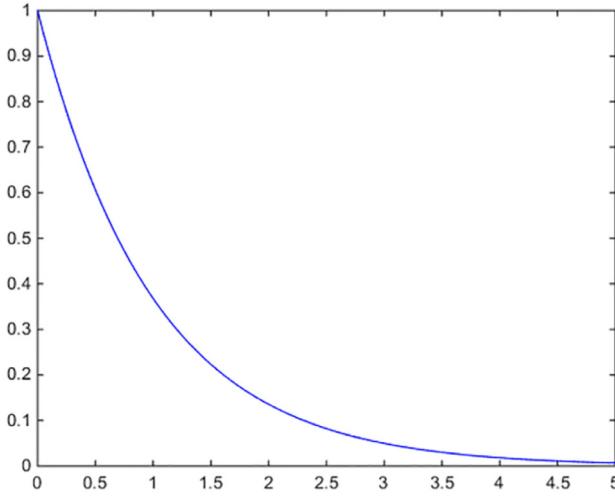
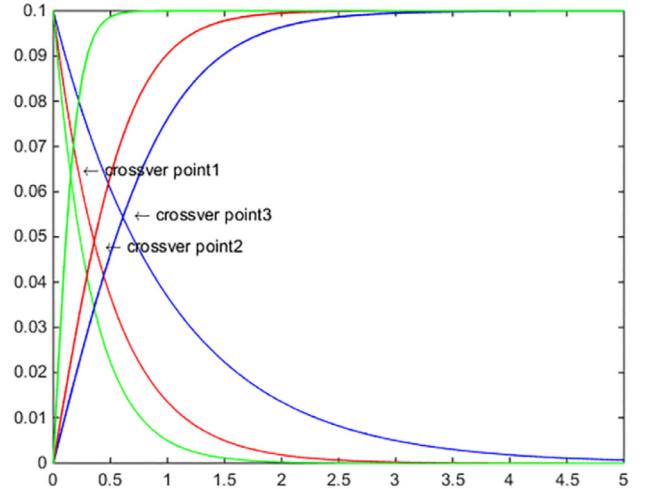
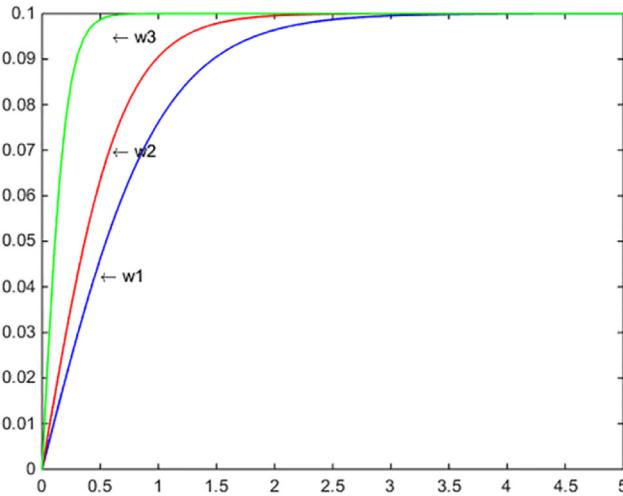
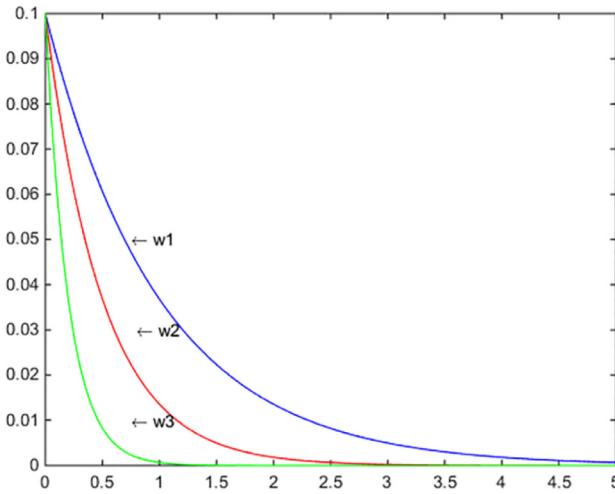
$$\begin{aligned} Jcc(\theta, \theta^i(j, k, l)) = & \sum_{i=1}^S [d_{attract} \tanh(w_{attract} \|\theta - \theta^i\|_2)] \\ & - \sum_{i=1}^S [h_{repellant} \exp(-w_{repellant} \|\theta - \theta^i\|_2)] \end{aligned} \quad (22)$$

where $Jcc(\theta, \theta^i(j, k, l))$ is the cell to cell communication value which will be added to the result of fitness function in the chemotaxis phase j ; S is the number of bacteria; $d_{attract}$, $w_{attract}$, $h_{repellant}$ and $w_{repellant}$ are different factors which represent the strength of attraction and repulsion. In order to limit the swarming effect in a reasonable range, the original equation $\sum_{f=1}^p (\theta_f - \theta_f^i)^2$ is replaced by formula $\|\theta - \theta^i\|_2$. At the same time, hyperbolic function $\tanh(x)$ is introduced to express attraction part more accurately. The definition of hyperbolic function $\tanh(x)$ is given in Eq. (23).

$$\tanh(x) = \frac{\exp(2x) - 1}{\exp(2x) + 1} \quad (23)$$

The graph of function $\tanh(x)$ is showed in Fig. 9. In order to make the attraction positive, $[0, +\infty)$ is chosen as the domain of definition for function $\tanh(x)$. On the one hand, the value of attraction is proportional to the distance between bacteria. On the other hand, this force decreases when bacteria gradually come to close. In particular, the attractive force equals to zero when they swarm to one point. In conclusion, this design is fitted to the biology phenomenon.

The graph of function $\exp(-x)$ is described in Fig. 10. To achieve the purpose of keeping the attraction and repulsion in the same range, domain of definition for function $\exp(-x)$ is defined in $[0, +\infty)$. Furthermore, the value of repulsion is inverse proportional to the distance between cells. That is to say, this force rises when distance among bacteria gradually come to close. Particularly, the repulsive force becomes the maximum when bacteria gather at the same point.

Fig. 10. $\exp(-x)$ function in the range.Fig. 13. Crossover points between $\tanh(x)$ and $\exp(-x)$.Fig. 11. The influence of coefficients to function $\tanh(x)$.Fig. 12. The influence of coefficients to function $\exp(-x)$.

The influence of coefficients $d_{attract}$ and $w_{attract}$ on hyperbolic function $\tanh(x)$ is illustrated in Fig. 11, where $d_{attract}$ is the depth of attractant and $w_{attract}$ is the width of attractant. The value of $d_{attract}$ is 0.1 and $w_{attract}$ is assigned to variables $w1$, $w2$, $w3$ separately on function $d_{attract}\tanh(w_{attract}x)$, where $w3 > w2 > w1 > 0$.

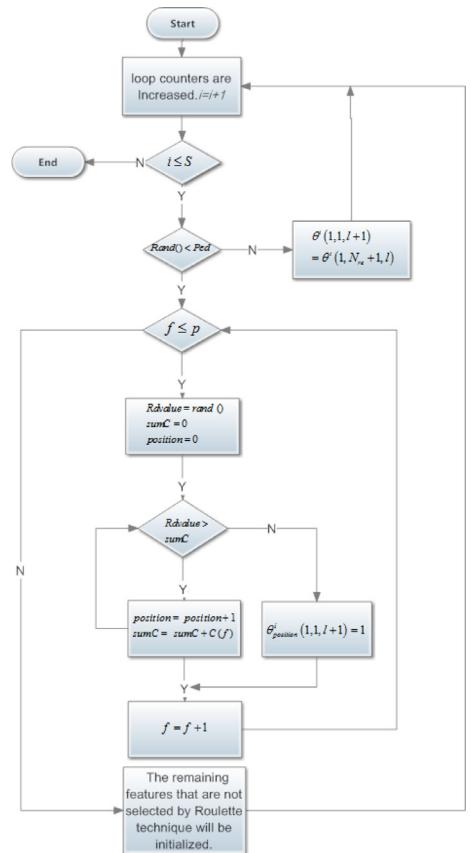


Fig. 14. Flowchart of modified elimination-dispersal stage.

The experiment result shows that the slope of the function $d_{attract}\tanh(w_{attract}x)$ is raised when the value of $w_{attract}$ increases.

The effect of coefficients $h_{repellant}$ and $w_{repellant}$ on function $\exp(-x)$ is described in Fig. 12, where $h_{repellant}$ is the height of repellent and $w_{repellant}$ is the width of repellent. The value of $h_{repellant}$ is 0.1 and $w_{repellant}$ is assigned to variables $w1$, $w2$, $w3$ separately on function $d_{repellant}\exp(-w_{repellant}x)$, where $w3 > w2 > w1 > 0$. The experiment result shows that the slope of the function $d_{repellant}\exp(-w_{repellant}x)$ is raised when the value of $w_{repellant}$ increases.

The combined changed process of attraction and repulsion is depicted in Fig. 13. Crossover point is the position where attraction

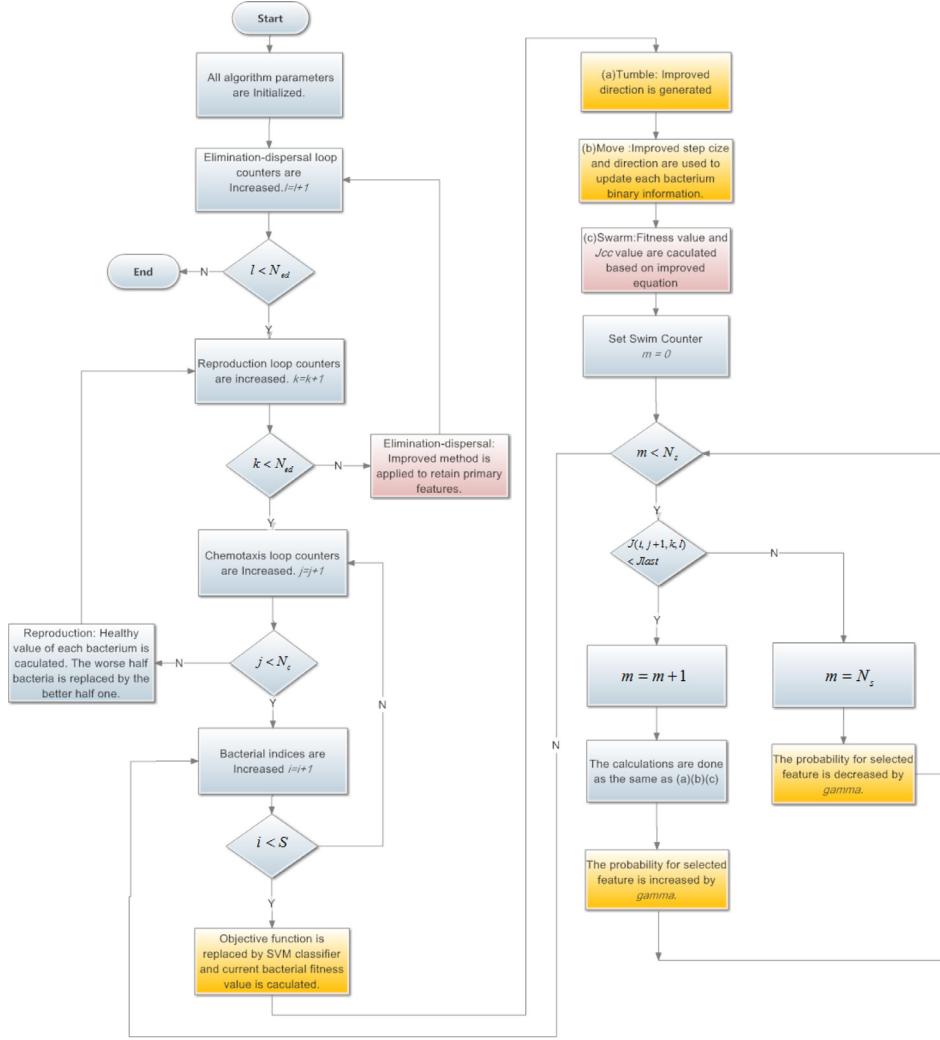


Fig. 15. Flowchart of ISEDBFO.

equals to repulsion when the distance between bacteria changes from large to small. Obviously, the repulsion is dominant when the cell to cell distance between 0 and crossover point. Thus bacteria can keep a proper distance between each other when bacteria are too close. Through experiment and analysis, the ability of searching optimal value is improved when coefficients w_{attrat} and $w_{repellant}$ are assigned to 5 and 10 respectively. In Fig. 13, because the red curve uses the recommended coefficients, therefore the performance of the red curve is the best in the three curves.

3.2.2. Modified elimination-dispersal stage

Every bacterium is mapped to a random value between 0 and 1 in the original elimination-dispersal stage. The bacterium is eliminated if the corresponding random value is less than a small probability P_{ed} . Nevertheless this approach can kill the potential optimal bacteria. In this paper, the roulette method is introduced to preserve the primary features of the eliminated bacteria based on feature selected probability. That is to say, high probability features are more likely to be retained in each iteration of the roulette method. The remaining features are initialized. Thereupon the search efficiency can be greatly improved. In addition, the biological diversity is guaranteed. The flowchart of modified elimination-dispersal stage is showed as follows (Fig. 14):

The flowchart of ISEDBFO algorithm is illustrated in Fig. 15, where the purple blocks are the improved parts comparing with ACBFO.

3.2.3. Time computation complexity analysis of ISEDBFO

$O(L \times (K \times (J \times S \times (M + N_2) + P) + Q + R))$ represents the time computation complexity of ISEDBFO algorithm. In this formula, L means iteration number for elimination-dispersal; K represents reproduction iteration time; J describes the number of chemotaxis activity; S is the total bacterial number; M is the calculation time for SVM model training and prediction; N_2 represents the time complexity for chemotaxis operations based on improved swarming strategy; P expresses the calculation time for reproduction process; Q implies the computation time for elimination-dispersal module; R represents the computation complexity for roulette process.

$O(L \times (K \times (J \times S \times (M + N_1) + P) + Q))$ is the time computation complexity of ACBFO. We can see that the computational complexity of ISEDBFO is higher than ACBFO. In other words, the performance of ISEDBFO is promoted while increasing the time computation complexity.

3.3. Theoretical comparison to other approaches

Global optimal feature subset searching ability of ACBFO is enhanced by the improvements. Especially, because the improved

Table 3
Experimental data sets information.

Data sets	No.of classes	No.of instances	No.of features
Australian	2	690	14
Bupa liver	2	345	6
Cleveland heart	2	303	13
Diabetes	2	768	8
German	2	1000	24
Ionosphere	2	351	34
Sonar	2	208	60
Vowel	10	990	13
Thyroid	3	215	5
Yeast	10	1484	8

chemotaxis step can evaluate the importance of each feature, thus the optimal feature subset is easier to obtain. The modified swarming representation of ISEDBFO can strengthen the ability of bacterial dispersion. Therefore, the optimal feature searching range is extended. Finally, the modified elimination-dispersal stage of ISEDBFO is more likely to obtain the high probability features. However, the classical BFO method does not have the capacity of evaluating features. Simultaneously, the original swarming equation cannot describe cell to cell attraction-repulsion relationship accurately when bacteria gathered at the same point. Thereafter the optimal feature searching process of classical BFO is blind and the bacteria are easily trapped into local optimal value when these bacteria come close together.

Because the chemotaxis and elimination-dispersal operations of improved BFO methods contain many random parameters, therefore the proposed methods have strong exploration ability. In other words, the modified methods can search a wider space than other metaheuristic methods. In conclusion, the proposed methods are easier to find the optimal solution than other methods. However, because the time computation complexity of improved BFO methods is higher than metaheuristic methods, thereby the optimal feature subset searching time for improved BFO methods is higher than other metaheuristic methods. Reducing time computation complexity of improved BFO methods is our prospect of study.

4. Experiment results and analysis

4.1. Data sets introduction

The experimental data sets are chosen from the UCI machine learning database (Blake, 1998). The performance of the proposed methods is evaluated based on following data sets, Australian, Breast cancer, Bupa liver, Cleveland heart, Diabetes, German, Ionosphere, Sonar, Wine and Thyroid. The detailed information of these data sets is described in Table 3.

4.2. Algorithm parameter setting

In the following comparative experiments, the initial parameters and realization ways of PSO, GA, BBA, SA refer to Uler, Murat, and Chinnam (2011) Huang and Wang (2006) and Nakamura et al. (2012) and Debuse and Rayward-Smith (1997) respectively. In order to solve feature selection problem, ALO and CS are modified in binary structure. For ALO, the position range of ant and ant lion is distributed between 0 and 1. If the position value is bigger than 0.5, then the corresponding feature is selected. Otherwise, the feature is not selected. For CS, the value of every dimension in each nest is randomly set to 0 or 1. If the random value equals to 1, then the corresponding feature is selected. Otherwise, the feature is not selected. The detailed parameter values of each algorithm are described as follows (Tables 4–6, 8–11):

Table 4
Detailed parameter values for ACBFO.

Parameter	Description	Value
S	Bacterial number	30
N_{re}	Number of reproductive steps	5
N_{ed}	Number of elimination-dispersal steps	2
N_c	Number of chemotactic steps	10
N_s	Number of swimming steps	4
α	The initial value of $C(f)$	0.2
P_{ed}	Elimination probability	0.25
$d_{attract}=h_{repellant}$	Coefficients of attraction-repulsion	0.1
$w_{attract}=w_{repellant}$	Coefficients of attraction-repulsion	0.2

Table 5
Detailed parameter values for ISEDBFO.

Parameter	Description	Value
S	Bacterial number	30
N_{re}	Number of reproductive steps	5
N_{ed}	Number of elimination-dispersal steps	2
N_c	Number of chemotactic steps	10
N_s	Number of swimming steps	4
α	The initial value of $C(f)$	0.2
P_{ed}	Elimination probability	0.25
$d_{attract}=h_{repellant}$	Coefficients of attraction-repulsion	0.1
$w_{attract}$	Coefficients of attraction	5
$w_{repellant}$	Coefficients of repulsion	10

Table 6
Detailed parameter values for PSO.

Parameter	Description	Value
N	Number of particles	30
w_{max}	Maximum value of weight	0.9
w_{min}	Minimum value of weight	0.4
C_1 and C_2	Coefficients	2
i	Maximum number of iterations	100

In order to ensure the experiment results stability, we repeat each algorithm for 3 times and average the results. In this research, fitness function is replaced by SVM classifier except for Table 17. The feature subset with the best classification accuracy is the solution to the improved algorithms. The radial base function (RBF) is used as the kernel function of the SVM model. Penalty parameter C and RBF parameter γ are chosen by the grid search method.

10-fold cross validation technique is used to test the mentioned above algorithms (Dietterich, 1998). Sensitivity and specificity are statistical measures of the performance of a binary classification test (Kane et al., 2000). Sensitivity measures the proportion of positives that are correctly identified. Specificity measures the proportion of negatives that are correctly identified.

4.3. Results and discussion

Comparative experiments are designed to show the superiority of the improved ACBFO and ISEDBFO algorithms. Because the important purpose of feature selection is to select the optimal feature subset with the highest classification accuracy, so in the first comparative experiment, the average classification accuracy via PSO based, GA based, SA based, ALO based, BBA based and CS based approaches is calculated with ten public UCI data sets. The detailed classification accuracy is showed in Table 12.

From Table 12, we can see that ACBFO achieves the highest average classification accuracy in Diabetes and Sonar data sets. Simultaneously, ISEDBFO obtains excellent performance in the rest data sets. Therefore, it is obvious that the performance of ISEDBFO is better than ACBFO. Furthermore, the average classification accuracy of ACBFO and ISEDBFO is higher than other 6 metaheuristic algorithms on all ten public UCI data sets.

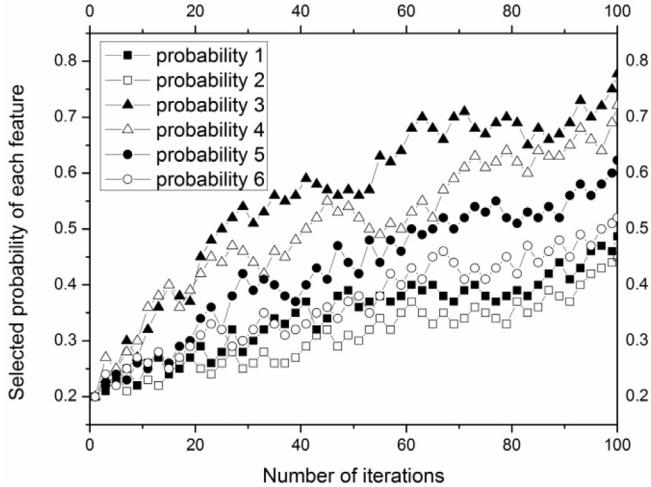


Fig. 16. Selected probability of each feature via ACBFO on Bupa liver data set.

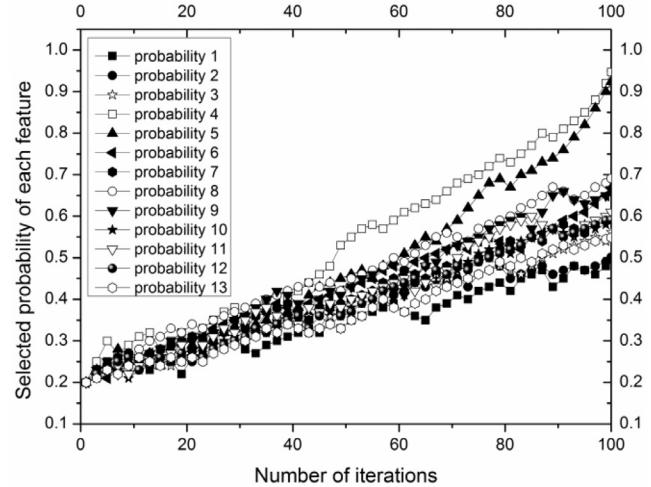


Fig. 17. Selected probability of each feature via ACBFO on Vowel data set.

The improved chemotaxis step of ACBFO and ISEDBFO algorithms promotes the ability of searching optimal feature subset. Because the importance of each feature is reflected by the feature selected probability based on improved chemotaxis step, therefore, the primary features are easily selected into feature subset. Moreover, the modified elimination-dispersal stage overcomes the drawback that the ACBFO algorithm is relatively easy to trap into local optimal feature subset. Because the primary features are kept in the eliminated bacteria, thereupon the speed of constructing optimal feature subset is accelerated.

From Table 13, we can see that ACBFO achieves the shortest average numbers of selected features in Diabetes and Thyroid data sets. At the same time, ISEDBFO wins ACBFO in the rest data sets. Moreover, the average number of selected features for ACBFO and ISEDBFO is better than other 6 metaheuristic algorithms on all ten public UCI data sets. Table 13 shows that the improved chemotaxis step is able to search for more extensive feature space. In other words, the modified swarming representation has a good effect on the dimension reduction. Simultaneously, the attraction and repulsion between bacteria is kept in a reasonable range. As a result, the ISEDBFO method can obtain the optimal feature subset with high probability.

In order to present the performance of improved chemotaxis step, the changing trend of ACBFO vector $C(f)$ is illustrated in Figs. 16 and 17 on Bupa liver and Vowel data sets respectively. The selected probability of each feature is calculated in the improved chemotaxis step of ACBFO. In the first phase, because the selected probability of each feature is unknown at the beginning, thus the selected probability of each feature is initialized with the constant small value α . In the second phase, we can see that the fluctuation of selected probability is small in whole iterative process. The reason is that the increment or decrement variable γ of selected probability is a relatively small value. In conclusion, the selected probability can filter some unexpected situation and truly reflect the importance of each feature. In the third phase, we get the final selected probability value of each feature. From Fig. 16, we can see that the order set of feature selected probability is 3-4-5-6-1-2 which represents the third feature is the most important feature and the second feature is the least important one. The solution of ACBFO on Bupa liver data set is 1-3-4-5-6. In other words, the primary features make a huge contribution to the optimal feature subset. From Fig. 17, we can see that the selected probability of each feature is very close before 50 iterations. Moreover, the selected probability of fourth feature is significantly better than other ones after 50 iterations. At last, we get the order set of feature se-

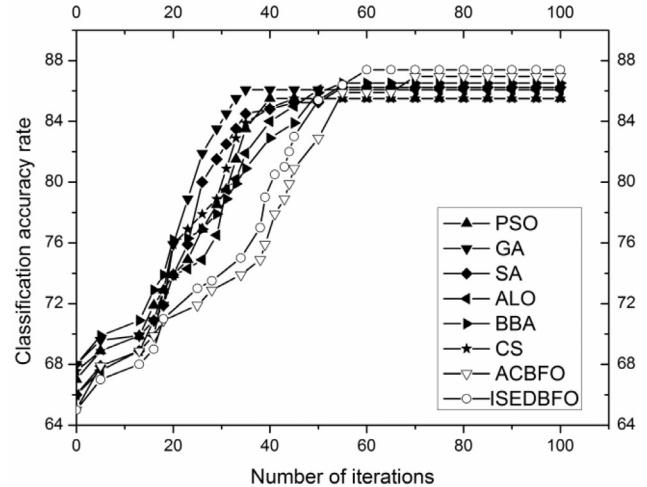


Fig. 18. Classification accuracy via eight methods on Australian data set.

lected probability which is 4-5-8-6-9-11-10-7-12-3-13-2-1. At the same time, the solution of ISEDBFO on Vowel data set is 4-5-6-8-9-10-11. As a result, constructing optimal feature subset based on feature selected probability is very effective.

The classification accuracy of eight algorithms based on each experimental data set is shown in Figs. 18–27 respectively. We take Figs. 21 and 27 for example to illustrate the changing process of average classification accuracy. From Fig. 21, we can see that ALO reaches the highest classification accuracy before 20 iterations. The algorithms except ACBFO and ISEDBFO obtain the local optimal feature subset before 50 iterations. Simultaneously, the classification accuracy of ACBFO and ISEDBFO is lower than the rest algorithms before 50 iterations. Moreover, ACBFO and ISEDBFO get the highest classification accuracy after 50 iterations. Especially, ACBFO wins ISEDBFO at the end of iteration. The Fig. 27 shows that the classification accuracy of ACBFO and ISEDBFO is lower than other metaheuristic algorithms before 30 iterations. Moreover, the performance of ACBFO is better than ISEDBFO before 30 iterations. The algorithms except ACBFO and ISEDBFO reach the highest classification accuracy before 60 iterations. In particular, ISEDBFO wins ACBFO after 60 iterations. Finally, it is obvious that the performance of ISEDBFO and ACBFO is better than other metaheuristic algorithms.

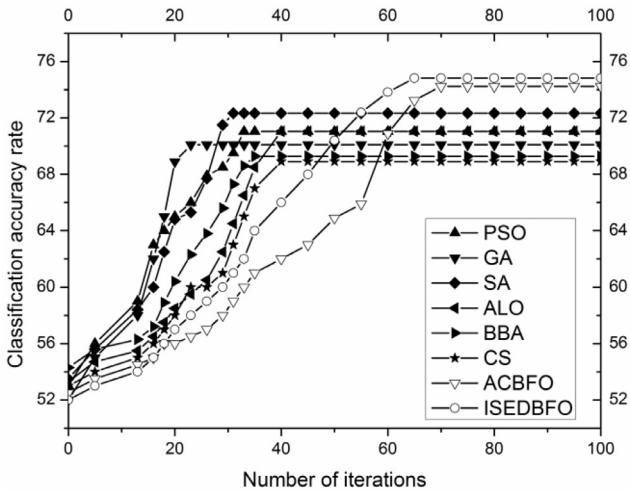


Fig. 19. Classification accuracy via eight methods on Bupa Liver data set.

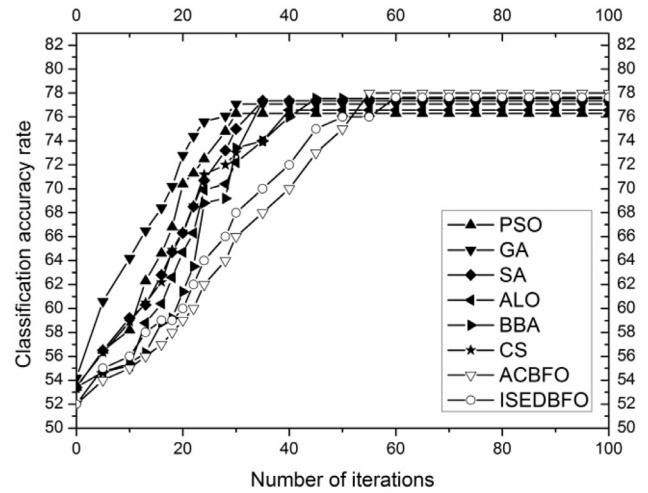


Fig. 21. Classification accuracy via eight methods on Diabetes data set.

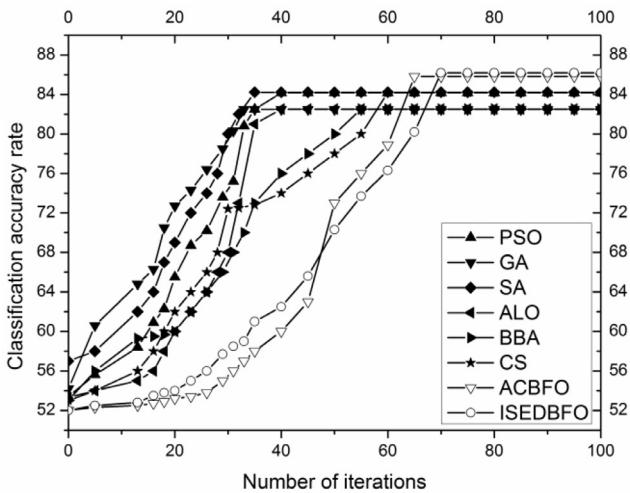


Fig. 20. Classification accuracy via eight methods on Cleveland Heart data set.

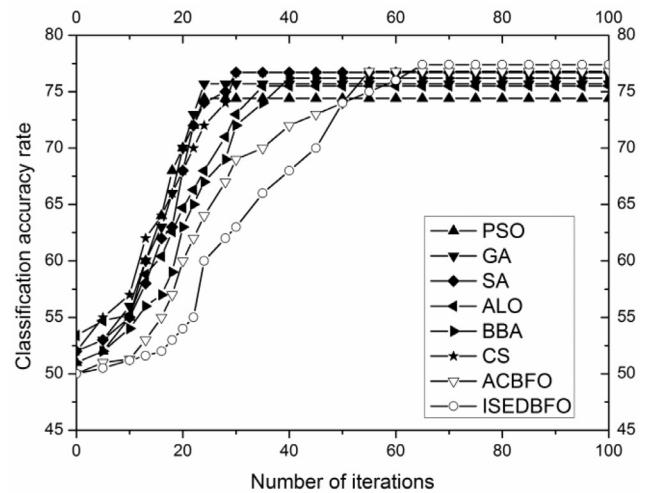


Fig. 22. Classification accuracy via eight methods on German data set.

At the beginning of the improved algorithms, because the initial feature selected probability α is close to 0, thus the convergence speed of ACBFO and ISEDBFO is slower than other 6 algorithms. Specially, if the classification accuracy of current bacterium is higher than local best classification accuracy in the swimming phase, then the selected probability of the primary features for current bacterium is increased. In other words, the importance of the features is evaluated with the increasing of iteration. Because the improved swarming step can make the bacteria more dispersed in the process of searching optimal feature subset, therefore the local optimum can be skipped and the global optimum can be obtained in the final phase. The improved swarming step of ISEDBFO has a great contribution to the experimental result.

Sensitivity and specificity are statistical measures, which are suitable for binary classification test. However because Vowel, Thyroid and Yeast are not binary classification data sets, therefore the sensitivity and specificity of each algorithm are tested only on seven UCI binary data sets. From Table 14, we can see that ACBFO achieves the highest sensitivity on Diabetes and Sonar data sets. Simultaneously, ISEDBFO obtains the highest sensitivity in the rest. Moreover, the performance of ISEDBFO is better than ACBFO. In particular, the ACBFO and ISEDBFO win other 6 metaheuristic algorithms. Table 15 shows that BBA reaches the highest specificity in Australian data set. At the same time, SA gets the highest speci-

ficity in German data set. ACBFO achieves the highest specificity in Cleveland heart data set. ISEDBFO acquires the highest specificity in Bupa liver, Diabetes, Ionosphere and Sonar data sets. From the results we can see that the improved chemotaxis step significantly promotes the proportion of correctly identified positive. Therefore, the improved chemotaxis step makes a huge contribution to ACBFO and ISEDBFO in sensitivity and specificity.

The wilcoxon signed-rank test is proposed by Frank Wilcoxon as a non-parametric statistical hypothesis test (Conover, 1973). This method is applied to compare two related samples. We can decide whether the corresponding data population distributions are identical based on this method. In this paper, the wilcoxon signed-rank test is executed by SPSS software. The data information in Tables 16 and 17 is the result of applying SPSS software. In Table 16, seven pairs of wilcoxon signed-rank tests are made on Bupa liver data. From Table 16, with the significant level 0.05, we can see that the performance of ACBFO is better than PSO, GA, SA, ALO, BBA and CS algorithms. Furthermore, ISEDBFO wins ACBFO. In other words, the each fold classification accuracy of ISEDBFO has the most significant difference. The descriptive statistics of eight algorithms are described in Table 17. Descriptive statistics are statistics that quantitatively describe or summarize features of some data information. The descriptive statistics of eight algorithms is described in Table 17. N means the processing times. Mean measures the central ten-

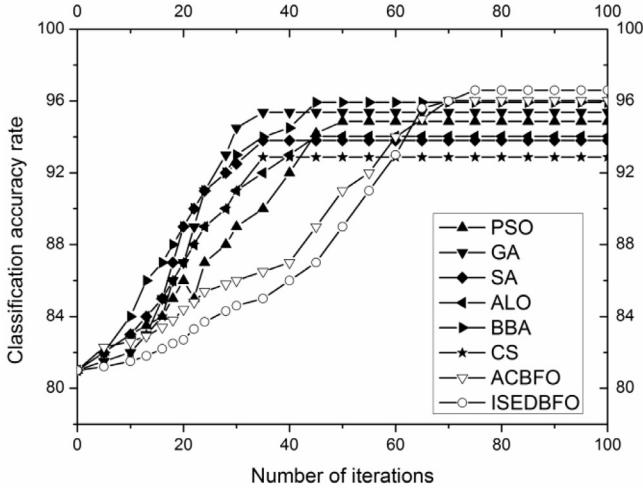


Fig. 23. Classification accuracy via eight methods on Ionosphere data set.

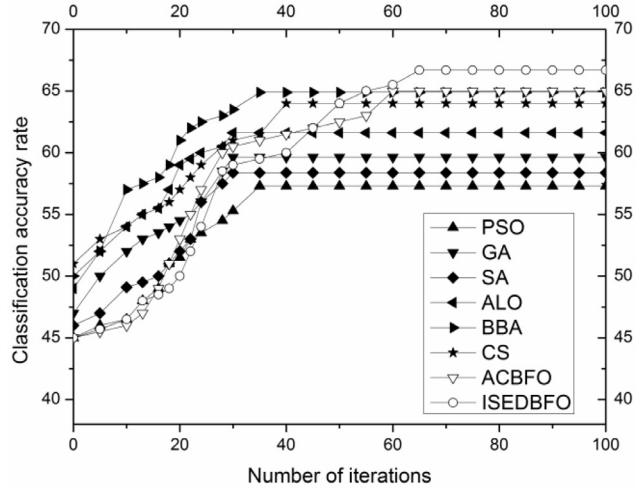


Fig. 25. Classification accuracy via eight methods on Vowel data set.

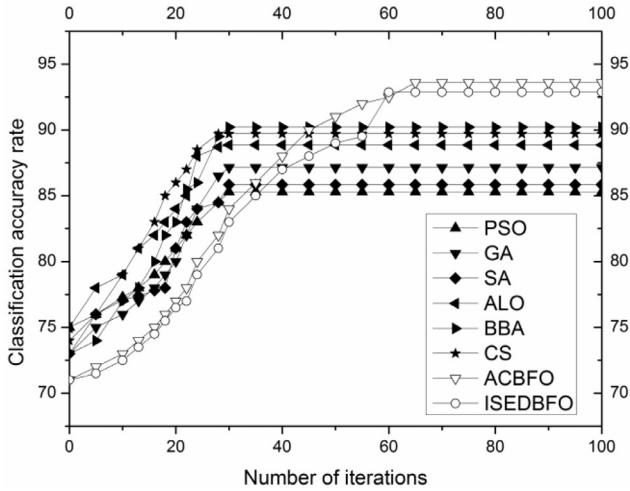


Fig. 24. Classification accuracy via eight methods on Sonar data set.

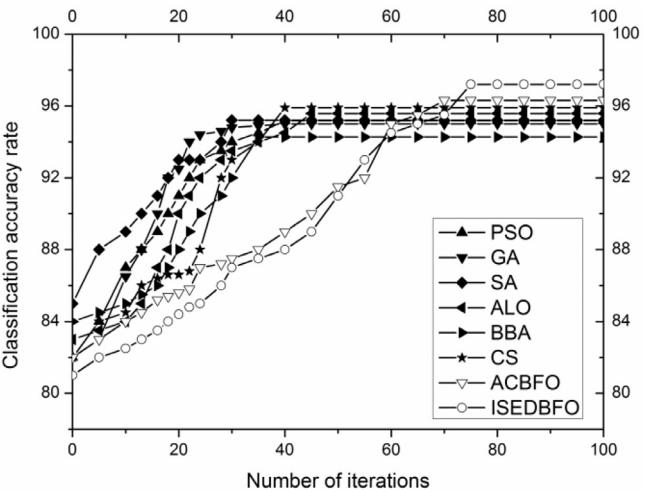


Fig. 26. Classification accuracy via eight methods on Thyroid data set.

Table 7
Detailed parameter values for GA.

Parameter	Description	Value
N	Number of chromosomes	30
P_c	Crossover probability	0.7
P_m	Mutation probability	0.02
i	Maximum number of iterations	100

Table 8
Detailed parameter values for SA.

Parameter	Description	Value
T_0	Initial temperature	0.8
T_f	Stop temperature	0.8^{30}
f	Cooling factor	0.8
i	Maximum number of iterations	100

dency of the data set. The standard deviation is a measure that is used to quantify the amount of variation or dispersion of a set of data values. A low standard deviation indicates that the data points tend to be close to the mean (also called the expected value) of the set, while a high standard deviation indicates that the data points are spread out over a wider range of values. From Table 7, we can see that the mean value of ISEDBFO and ACBFO is higher than the rest algorithms. Moreover, ISEDBFO wins ACBFO. The results mean

Table 9
Detailed parameter values for BBA.

Parameter	Description	Value
N	Number of bats	30
L	Loudness	1.5
P	Pulse rate	0.5
Q_{\min}	Maximum value of frequency	0
Q_{\max}	Minimum value of frequency	1
i	Maximum number of iterations	100

Table 10
Detailed parameter values for ALO.

Parameter	Description	Value
N	Number of ants	30
V_{\max}	Maximum of all variables	0
V_{\min}	Minimum of all variables	1
i	Maximum number of iterations	100

the central tendency the ISEDBFO is the best. However, standard deviation of ISEDBFO is the highest. It shows that the few individual values of ISEDBFO are not ideal.

Ten public data sets of UCI are tested by SVM, K-Nearest Neighbor (KNN), Decision Tree (DT), Naive Bayesian (NB), Random Forest (RF) and Feed-forward Neural Net (FNN) classifiers respectively.

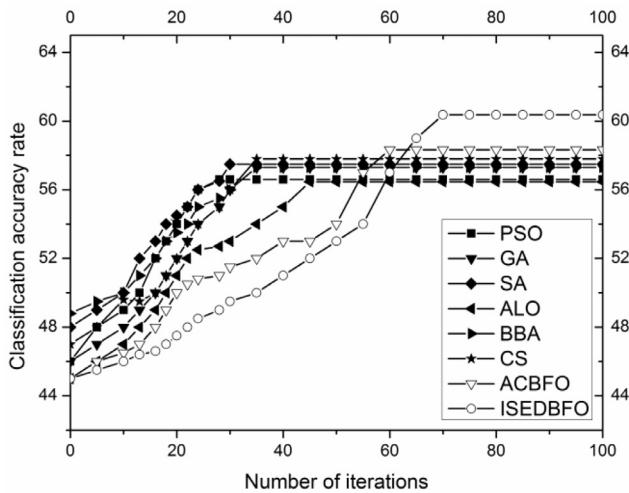


Fig. 27. Classification accuracy via eight methods on Yeast data set.

Table 11

Detailed parameter values for CS.

Parameter	Description	Value
<i>N</i>	Number of nests	30
<i>R</i>	Discovery rate of alien eggs/solution	0.25
<i>C</i>	Levy exponent and coefficient	1.5
<i>i</i>	Maximum number of iterations	100

Table 12

Average classification accuracy via eight algorithms on each data set. Bold value denotes the best value.

Data set	PSO	GA	SA	ALO	BBA	CS	ACBFO	ISEDBFO
Australian	85.5	86.1	86.2	86.1	86.5	85.4	86.9	87.3
Bupa liver	71.1	70.1	72.3	71.2	69.2	68.9	74.2	74.8
Cleveland heart	84.1	82.4	84.2	82.5	82.5	84.1	85.8	86.1
Diabetes	76.2	77.1	77.3	76.5	77.5	77.3	77.9	77.6
German	74.4	75.7	76.7	75.5	76.2	76.7	76.8	77.4
Ionosphere	94.8	95.3	93.7	94.1	95.9	92.8	96.2	96.6
Sonar	85.2	87.1	85.8	88.8	90.2	89.7	93.5	92.8
Vowel	57.2	59.6	58.3	61.6	64.8	63.9	64.9	66.6
Thyroid	95.1	95.1	95.2	95.5	94.2	95.9	96.3	97.2
Yeast	56.5	57.3	57.4	61.4	60.3	62.7	63.3	65.3

From Table 18, we can see that ACBFO based on SVM classifier reaches the highest average classification accuracy in Diabetes and Sonar data sets. Simultaneously, ISEDBFO based on SVM classifier obtains the highest average classification accuracy in Australian, Bupa Liver, Cleveland Heart, German, Ionosphere, Vowel, Thyroid and Yeast data sets. Furthermore, test results have shown that ACBFO and ISEDBFO applying with SVM classifier can get good performance on the most of data sets. As a result, the SVM is the most suitable classifier for ACBFO and ISEDBFO.

Through the above experimental results we can see that the improved BFO methods not only promote the average classification accuracy, but also reduce the average numbers of selected features in most of the tested data sets of UCI. In other words, the improved chemotaxis step of ACBFO and ISEDBFO algorithms enhances the ability of searching optimal feature subset. Moreover, the changing trend test of vector $C(f)$ means that the evaluation results of important features are very close to the optimal feature subsets. In a word, constructing optimal feature subset based on feature selected probability is very effective. Simultaneously, sensitivity and specificity test results represent that the improved chemotaxis step significantly promotes the proportion of correctly identified positive. Finally, wilcoxon signed-rank test and descriptive statistics show that the data points are spread out over a wider range of values. In conclusion, through the experimental results we can see that the performance of ACBFO is significantly strengthened.

5. Conclusion

In this work, we propose two novel BFO algorithms, which are named as ACBFO and ISEDBFO respectively. The data structure and chemotaxis process are improved for solving discrete problem in ACBFO. Additionally, the cell to cell attraction-repulsion representation and elimination process are modified in ISEDBFO. From the experimental results, we can see that the improved algorithms can evaluate the importance of features and select the primary features more easily in every iteration. Moreover, the convergence speed of the improved algorithms is accelerated in the second half the whole iteration. Simultaneously, hyperbolic function is introduced to avoid the premature convergence of the bacteria, and guide the bacteria to find a better solution. Especially, the performance of elimination process is promoted by using the roulette method. Finally, experimental results presented that the proposed methods are significantly better than other six metaheuristic algorithms.

The improved BFO algorithms still have great potential to develop on the feature selection problem. Therefore we plan to further optimize improved BFO algorithms in the following aspects: Firstly, because classical SVM method is applied to ACBFO and ISEDBFO, thus we will study SVM method in depth for promoting the performance of fitness function. Secondly, chemotaxis, swarming and elimination-dispersal operations have been improved in this paper. Thus the reproduction step will be our next research emphasis. Thirdly, in order to promote the convergence rate, we will pay much attention to reduce time computation complexity of the improved BFO methods. Fourthly, because the improved BFO methods are only suitable for single object optimization problem in this paper, so the improved BFO methods will be further optimized to adapt to multi-objective combinatorial optimization problem. Finally, we will research on combining improved BFO methods with other metaheuristic methods to promote the ability of searching optimal feature subset.

Table 13

Average number of selected features via eight algorithms on each data set. Bold value denotes the best value.

Data set	Original features	PSO	GA	SA	ALO	BBA	CS	ACBFO	ISEDBFO
Australian	14	9.8	9.0	9.7	9.3	10.1	9.5	8.6	8.2
Bupa liver	6	5.9	5.8	5.6	5.8	5.7	5.6	5.5	5.4
Cleveland heart	13	8.5	8.7	9.2	8.1	7.9	8.3	7.2	6.9
Diabetes	8	6.3	6.6	5.5	5.1	4.8	5.4	4.2	4.6
German	24	16.8	15.7	14.3	13.9	15.2	14.8	13.1	12.3
Ionosphere	34	19.2	19.5	18.9	17.3	18.2	17.8	16.8	16.1
Sonar	60	29.4	27.7	28.4	28.1	30.0	27.2	26.1	25.4
Vowel	13	9.2	8.8	8.0	7.4	8.1	7.8	6.9	6.5
Thyroid	5	4.1	4.3	3.4	4.0	3.6	3.7	2.8	3.0
Yeast	8	6.6	5.7	6.2	5.0	5.3	5.1	4.8	4.6

Table 14
Sensitivity via eight algorithms on each data set. Bold value denotes the best value.

Data set	PSO	GA	SA	ALO	BBA	CS	ACBFO	ISEDBFO
Australian	0.79	0.84	0.87	0.83	0.84	0.87	0.87	0.88
Bupa liver	0.51	0.44	0.52	0.49	0.48	0.57	0.53	0.58
Cleveland heart	0.84	0.83	0.58	0.53	0.81	0.81	0.81	0.85
Diabetes	0.51	0.53	0.55	0.56	0.53	0.53	0.59	0.54
German	0.86	0.82	0.54	0.54	0.83	0.82	0.85	0.87
Ionosphere	0.92	0.92	0.91	0.88	0.93	0.93	0.95	0.96
Sonar	0.62	0.66	0.77	0.74	0.73	0.66	0.82	0.79

Table 15
Specificity via eight algorithms on each data set. Bold value denotes the best value.

Data set	PSO	GA	SA	ALO	BBA	CS	ACBFO	ISEDBFO
Australian	0.83	0.85	0.86	0.86	0.88	0.58	0.83	0.87
Bupa liver	0.84	0.84	0.87	0.86	0.81	0.84	0.81	0.88
Cleveland heart	0.79	0.81	0.82	0.84	0.75	0.77	0.85	0.77
Diabetes	0.88	0.88	0.77	0.77	0.87	0.88	0.88	0.89
German	0.45	0.42	0.69	0.68	0.33	0.49	0.34	0.45
Ionosphere	0.62	0.87	0.42	0.44	0.89	0.70	0.85	0.87
Sonar	0.82	0.65	0.87	0.88	0.70	0.67	0.80	0.89

Table 16
Comparison based on wilcoxon signed-rank test.

Algorithm 1	ACBFO	ACBFO	ACBFO	ACBFO	ACBFO	ACBFO	ISEDBFO
Algorithm 2	PSO	GA	SA	ALO	BBA	CS	ACBFO
Z Value	-2.724 ^b	-2.609 ^b	-2.616 ^b	-2.556 ^b	-2.583 ^b	-2.518 ^b	-1.501 ^a
P Value	0.005	0.008	0.007	0.010	0.012	0.015	0.024

^a Based on negative ranks.

^b Based on positive ranks.

Table 17
Descriptive statistics of eight algorithms on each data set.

Algorithm	Descriptive statistics				Percentiles			
	N	Mean	Std. deviation	Minimum	Maximum	25th	50th	75th
PSO	10	71.0334	4.88044	65.71	78.57	65.7143	70.8068	75.0000
GA	10	70.0000	6.06092	62.86	80.00	65.0000	68.5741	73.5714
SA	10	72.3333	7.46014	62.86	83.33	65.7143	70.0000	80.7143
ALO	10	71.0476	8.20744	57.14	85.71	65.0000	71.4285	75.7143
BA	10	69.2857	4.94758	62.82	77.14	64.9235	69.8728	73.5513
CS	10	69.9048	4.65073	62.14	77.86	66.7072	70.6429	72.3357
ACBFO	10	74.2353	8.41313	64.11	84.19	64.8353	75.2213	81.6750
ISEDBFO	10	74.8235	10.06902	61.76	88.24	65.0000	75.7143	85.2941

Table 18

Average classification accuracy via six classifiers on each data set. Bold value denotes the best value.

Data set	ACBFO	SVM	ACBFO	KNN	ACBFO	DT	ACBFO	RF	ACBFO	ISEDBFO	DT	ISEDBFO	NB	ISEDBFO	RF	ISEDBFO	FNN
Australian	86.9	86.3	85.7	85.4	86.5	85.8	87.3	86.8	87.2	86.9	87.1	86.2					
Bupa Liver	74.2	73.5	64.3	73.6	71.8	72.4	74.8	74.4	65.4	74.4	72.6	73.7					
Cleveland Heart	85.8	83.4	85.1	85.4	85.1	85.5	86.1	83.1	85.9	84.2	85.4	85.7					
Diabetes	77.9	77.2	76.4	75.3	76.4	76.4	77.6	76.1	77.3	77.1	76.9	77.4					
German	76.8	75.1	72.7	74.5	75.8	76.2	77.4	74.7	73.7	73.5	76.2	76.5					
Ionosphere	96.2	88.3	91.7	92.1	94.8	95.8	96.6	90.6	92.9	93.2	95.4	95.9					
Sonar	93.5	88.3	89.2	88.3	89.7	91.1	92.8	87.2	90.8	90.4	90.2	92.5					
Vowel	64.9	63.7	63.3	61.3	65.8	63.7	66.6	65.8	64.4	62.5	66.1	64.7					
Thyroid	96.3	96.6	95.5	96.4	95.7	95.9	97.2	97.1	96.2	97.1	96.5	96.3					
Yeast	63.3	59.3	60.4	63.1	64.2	63.8	65.3	62.3	61.5	63.4	64.8	64.1					

Acknowledgment

This research is supported by the National Natural Science Foundation of China (NSFC) under Grant nos. of 61602206.

References

- Akay, M. F. (2009). Support vector machines combined with feature selection for breast cancer diagnosis. *Expert Systems with Applications*, 36(2), 3240–3247.
- Alonso-Atienza, F., Rojo-Álvarez, J. L., Rosado-Muñoz, A., Vinagre, J. J., García-Alberola, A., & Camps-Valls, G. (2012). Feature selection using support vector ma-
- chines and bootstrap methods for ventricular fibrillation detection. *Expert Systems with Applications*, 39(2), 1956–1967.
- Apollonia, J., Leguizamón, G., & Albab, E. (2015). Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments. *Applied Soft Computing*, 38, 922–932.
- Blake, C. (1998). Uci repository of machine learning databases. *Neural Information Processing Systems*.
- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. New York: Oxford University Press.
- Camazine, S., Deneubourg, J. L., Franks, N. R., Sneyd, J., Theraulaz, G., & Bonabeau, E. (2003). *Self-organization in biological systems*. Princeton: Princeton University Press.

- Conover, W. J. (1973). On methods of handling ties in the wilcoxon signed-rank test. *Journal of the American Statistical Association*, 68(344), 985–988.
- Debuse, J. C. W., & Rayward-Smith, V. J. (1997). Feature subset selection within a simulated annealing data mining algorithm. *Journal of Intelligent Information Systems*, 957–981.
- Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7), 1895–1923.
- Glasmachers, T., & Igel, C. (2008). Second-order smo improves svm online and active learning. *Neural Computation*, 20(2), 374–382.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimisation and machine learning*. New York: Addison Wesley Longman Publishing Co., Inc.
- Guyon, I., Gunn, S., Nikravesh, M., & Zadeh, L. A. (2006). *Feature extraction: Foundations and applications (studies in fuzziness and soft computing)*. Secaucus, NJ, USA: Springer Verlag, New York, Inc.
- Guyon, I., Li, J., Mader, T., Pletscher, P. A., Schneider, G., & Uhr, M. (2007). Competitive baseline methods set new standards for the NIPS 2003 feature selection benchmark. *Pattern Recognition Letters*, 28(12), 1438–1444.
- Hsu, H. H., Hsieh, C. W., & Lu, M. D. (2011). Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications*, 38(7), 8144–8150.
- Huang, C. L., & Dun, J. F. (2008). A distributed PSO-SVM hybrid system with feature selection and parameter optimization. *Applied Soft Computing*, 8(4), 1381–1391.
- Huang, C. L., & Wang, C. J. (2006). A GA-based feature selection and parameters optimization for support vector machines. *Expert Systems with Applications*, 31(2), 231–240.
- Jain, A., & Zongker, D. (1997). Feature selection: Evaluation, application, and small sample performance. *IEEE transactions on pattern analysis and machine intelligence*, 19(2), 153–158.
- Jakhar, R., Kaur, N., & Singh, R. (2011). Face recognition using bacteria foraging optimization-based selected features. *International Journal of Advanced Computer Science and Applications, Special Issue on Artificial Intelligence*, 106–111.
- Kane, M. D., Jatkoe, T. A., Stumpf, C. R., Lu, J., Thomas, J. D., & Madore, S. J. (2000). Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays. *Nucleic Acids Research*, 28(22), 4552–4557.
- Kennedy, J. F., & Eberhart, R. C. (1995). Particle swarm optimization. In *Proceedings of IEEE international conference of neural network*, Perth, Australia (pp. 1942–1948).
- Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671–680.
- Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2), 273–324.
- Kudo, M., & Sklansky, J. (2000). Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1), 25–41.
- Lin, S. W., Lee, Z. J., Chen, S. C., & Tseng, T. Y. (2008). Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied Soft Computing*, 8(4), 1505–1512.
- Meiri, R., & Zahavi, J. (2006). Using simulated annealing to optimize the feature selection problem in marketing applications. *European Journal of Operational Research*, 171(3), 842–858.
- Mirjalili, S. (2015). The Ant Lion Optimizer. *Advances in Engineering Software*, 83(6), 80–98.
- Nakamura, R. Y. M., Pereira, L. A. M., Costa, K. A., Rodrigues, D., Papa, J. P., & Yang, X.-S. (2012). BBA: A binary bat algorithm for feature selection. In *Conference on graphics, patterns & images* (pp. 291–297).
- Neumann, J., Schnörr, C., & Steidl, G. (2005). Combined SVM-based feature selection and classification. *Machine Learning*, 61(1-3), 129–150.
- Panda, R., & Naik, M. K. (2015). A novel adaptive crossover bacterial foraging optimization algorithm for linear discriminant analysis based face recognition. *Applied Soft Computing*, 30(C), 722–736.
- Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control System Magazine*, 22(3), 52–67.
- Peralta, B., & Soto, A. (2014). Embedded local feature selection within mixture of experts. *Information Sciences*, 269(8), 176–187.
- Rodrigues, D., LAM Pereira, L. A. M., Nakamura, R. Y. M., Costa, K. A. P., Yang, X. S., Souzac, A. N., et al. (2014). A wrapper approach for feature selection based on bat algorithm and optimum-path forest. *Expert Systems with Applications*, 41(5), 2250–2258.
- Tahir, M. A., & Smith, J. (2010). Creating diverse nearest-neighbour ensembles using simultaneous metaheuristic feature selection. *Pattern Recognition Letters*, 31(11), 1470–1480.
- Unler, A., Murat, A., & Chinnam, R. B. (2011). mr²PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences*, 181(20), 4625–4641.
- Vignola, L. D., Milonea, D. H., & Scharcanski, J. (2013). Feature selection for face recognition based on multi-objective evolutionary wrappers. *Expert Systems with Applications*, 40(13), 5077–5084.
- Wanga, J., Hedarb, A. R., Wanga, S. Y., & Mac, J. (2012). Rough set and scatter search metaheuristic based feature selection for credit scoring. *Expert Systems with Applications*, 39(6), 6123–6128.
- Wang, X. Y., Yang, J., Teng, X. L., Xia, W. J., & Jensen, R. (2007). Feature selection based on rough sets and particle swarm optimization. *Pattern Recognition Letters*, 28(4), 459–471.
- Yang, X. S., & Deb, S. (2014). Cuckoo search: Recent advances and applications. *Neural Computing & Applications*, 24(1), 169–174.
- Zhu, F. M., & Guan, S. (2004). Feature selection for modular GA-based classification. *Applied Soft Computing*, 4(4), 381–393.