IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 25, 2019, accepted February 11, 2019, date of publication February 18, 2019, date of current version March 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900078

Binary Differential Evolution Based on Individual Entropy for Feature Subset Optimization

TAO LI, HONGBIN DONG, AND JING SUN
College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Corresponding author: Hongbin Dong (donghongbin@hrbeu.edu.cn)

This work was supported in part by the National Science Foundation of China under Grant 61472095 and Grant 61502116, and in part by the Heilongjiang Provincial Education Department Key Laboratory of Intelligent Education and Information Engineering.

ABSTRACT The high dimensionality of data brings great challenges to the classification accuracy and complexity of the algorithm. Feature selection technology can improve the classification performance of the algorithm effectively. In this paper, a novel binary differential evolution based on individual entropy (BDIE) is proposed. First, the individual entropy method is constructed to quantify the diversity of the population, and the relationship between population diversity and convergence is analyzed. Then, the objective function based on individual entropy is designed to evaluate the feature subset. A new binary mutation strategy is proposed, and it can effectively search the global optimal solution. In order to validate the BDIE, the datasets with different sizes and the classifiers of different types are used for testing. In addition, the well-known algorithms are introduced for comparison. The experimental results show that the proposed algorithm can effectively improve the classification performance and reduce the time cost without increasing the size of the feature subset.

INDEX TERMS Feature selection, differential evolution, optimization algorithm, evaluation criterion.

I. INTRODUCTION

Feature selection (FS) is an effective data processing technology in data mining and machine learning. The purpose of FS is to find the smaller size of feature subset and ensure that the performance of the algorithm model is not reduced [1]. It is clear that the data usually contains many irrelevant features or redundant features which may leads to the phenomenon of over-fitting and limits the generalization ability of algorithm. While FS can remove the irrelevant features or redundant features effectively to reduce the dimensionality of data. Most notably, the reduced feature subset can significantly improve the speed of algorithm, enhance the interpretability of the model and prevent over-fitting. Therefore, feature selection has been studied and applied to pattern recognition, task decision and other fields effectively.

Obtaining the ideal subset of features is a difficult task because of the arbitrary combination of features. In essence, FS is a kind of NP-hard combination optimization problem. With the dimensionality of data increasing, the number of feature subsets increases exponentially [2]. Hence, it is essential to explore the efficient search mechanism to address feature selection problem. There are many search techniques are designed to handle feature selection, such as complete search, heuristic search and random search. Among them, the complete search methods cost a considerable amount of computation time, especially for high-dimensional data. Heuristic algorithm can get the convergent solution, but it is easy to fall into local optimum or premature. However, the methods based random search consider both local optimum and global optimal solution simultaneously, and the better solution can be obtained. Hence, the evolutionary computation (EC) algorithms based on population randomization have been performed to search for the optimal feature subset.

Feature selection methods are generally divided into three types: wrappers, filters and embedded. Because most evolutionary algorithms usually combine specific classifier to select feature subsets, so the kind of evolutionary algorithm based on random strategy is used as evolving wrapper-evolution method [3]. While the evolution algorithm without classifier is called filter-evolution method. Currently, many researchers have studied the EC-based approaches for feature selection, such as genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO) and differential evolution (DE). Among the existing approaches, genetic algorithm (GA) should be the most popular optimization algorithm in evolutionary algorithm. For example, considering the effect of chromosome length on the search process, an adapted version of GA with variable length representation scheme was developed by Yahya *et al.* [4].

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

In addition, the hybrid GA with neural networks is presented to identify an optimum feature subset [5], which is based on feature ranking with fast algorithms and initial population with enhancements. A number of improved PSO algorithms have been applied to optimize the feature subset. In order to design the optimal individual update strategy, Xue *et al.* [6] proposed a new global optimal individual updating mechanism, which significantly improves the performance of the PSO. While other scholars put forward the combination of different evolutionary algorithms, such as a novel hybrid feature selection algorithm with a new local search strategy [7] that is embedded in the PSO to select the less correlated and salient feature subset. Besides, there are many typical feature selection approaches based on ACO. For instance, ant colony optimization was applied in unsupervised probabilistic feature selection [8] and the inter-feature information is utilized to evaluate the similarity between features for selecting the optimal feature subset. A new study proposed a novel swarm based hybrid algorithm [9], which combines the characteristics of ant colony optimization and artificial bee colony algorithms to optimize feature selection, and it performs well both the dimensionality reduction and the classification accuracy.

Differential evolution is arguably one of the efficient and powerful population-based random search techniques for global optimization [10]. Compare with GA, PSO and ACO algorithm [11], [12], the easy to operate of DE make it widely applied in many scientific and engineering fields, especially in the optimization of continuous search domain [13]. Many researchers have explored DE to improve the performance of the algorithm in terms of accuracy, convergence speed and stability. Since DE is usually an effective tool for solving continuous problems, so it is expert in optimizing the parameters of the learning model. As a combinatorial optimization problem, feature selection is a discrete problem. Thus, the continuous DE should be improved to suit for the feature subset optimization. The binary differential evolution was proposed by Pampara *et al.* [14] and the trigonometric function is utilized to generate 0-1 strings, realizing the transformation of floating point variables into binary forms. Combining artificial bee colony optimization technique with differential evolution algorithm for feature selection [15] and the hybrid method improves run-time performance and accuracy of the classifier. Moreover, a supervised feature selection technique guided by self-adaptive differential evolution for feature subset generation was developed [2]. The proposed method shows promising results compared to others in terms of overall classification accuracy and Kappa coefficient.

Based on the above analysis, we find that most of the existing DE algorithms mainly improve their performance from three aspects: individual coding, evaluation criteria and search strategy. However, there are still significant issues require further survey:

1) Individual evaluation without considering the effect of diversity on the evolution process. Population diversity not only affects the convergence rate of the algorithm, but also affects the quality of the global optimal solution. Therefore, it is necessary to analyze the diversity of each individual. For feature selection, each individual represents a candidate feature subset. According to the evaluation criteria of feature subset, we hope that the selected subset has fewer features and better classification ability, and the algorithm can converge the optimal solution quickly. Therefore, how to quantitatively judge individual diversity is critical for the population evolution.

2) The discrete operators satisfying the closed condition need to be developed. We all know that discrete and continuous evolutionary algorithms can transform each other by encoding strategies. However, the discretization process may need other auxiliary operations for feature selection. It not only increases the time cost, but also causes instability of the algorithm.

3) Adaptive parameter selection strategy needs further study. The results are subjective because of the artificial setting of parameters. Choosing different parameters according to the evolution process is of great significance to improve the performance of the algorithm. Therefore, how to select reasonable parameter is also an important problem when using evolutionary algorithm to optimize feature subset.

In this paper, a novel binary differential evolution based on individual entropy is proposed for feature selection. It can obtain the optimal feature subset with superior classification performance. The rest of the paper is organized as follows: In Section II, related works on DE algorithm are reviewed. Then, in Section III, the individual entropy method is constructed, and the effect of individual diversity on feature subset is discussed. Moreover, a detailed description of the proposed methodology is presented in Section IV. In Section V, the experimental results and comparisons with other algorithms are given. Finally, the paper is summarized in Section VI.

II. RELATED WORK

When using evolutionary algorithm to handle feature selection problem, the common way to pose feature selection as an minimization (or maximization) problem is to optimize the evaluation index. In this paper, the feature subset is measured by minimizing the objective function. Suppose $X = [x_1, x_2, \cdots, x_n]$ is the parameter vector to be optimized, where x is is a real number. If the objective function $f: \Omega \subseteq R^D \rightarrow R$, then the local minimum f_{min} can be defined as

$$f_{min}(x_l) = \{x_l | \|x - x_l\| < \epsilon \Rightarrow f_{min}(x_l) \leq f(x), \exists \epsilon > 0, \forall x \in \Omega\} \quad (1)$$

where $\|\cdot\|$ indicates any p-norm distance measure and ϵ is a minimum. (1) gives the idea of solving most of the optimization problems including the feature combination optimization problem. The difference is to redesign the new individual representation and evaluation function in the feature combination.

Differential algorithm is a popular algorithm for finding the minimum objective value. The core idea of DE algorithm

is that it uses the difference strategy to randomly obtain the difference individual, then it extends the search region by mutation operation. Finally, the optimal individual (solution) is obtained according to the evaluation function after a certain number of iterations [16], [17]. Differential evolution algorithm can be divided into continuous DE and discrete DE according to the task requirements. At present, many variations of the difference algorithm have been proposed. It can be found that the existing algorithms mainly focus on parameter setting and mutation strategy.

In the DE algorithm, there are three important control parameter named population size N, scaling factor F and crossover rate p respectively. In the process of optimization, the selection of parameter may influences the optimization performance of the DE. Based on the parameter setting mechanism, the selection of parameter can be divided into constant parameter and adaptive parameter. In classical DE algorithm [10], the range of N is $[5D, 10D]$, where D is the dimension of the individual. F and p are set to 0.5 and 0.1 respectively. According to the different dimensions of the problem, we need to choose a reasonable range of parameters. Therefore, some scholars quantify the relationship between dimensions and parameters to determine reasonable constant parameter values [18]–[21]. Another method is the adaptive parameter, which automatically adjusts the size of the parameter based on the feedback information of the evolution process. This type of methodes [22]–[25] can effectively select the appropriate parameters to control the performance of the algorithm in the search space. In this paper, we tend to construct reasonable strategies to adaptively select more appropriate parameters. In addition, the diversity of the population is one of the key factors to find the optimal individual. Thus, the relationship between parameter and population diversity should been considered during the evolutionary process. Let $X = \{X_1, X_2, \cdots, X_N\}$ be current population, $Y = \{Y_1, Y_2, \cdots, Y_N\}$ the population after mutation operator, and $Z = \{Z_1, Z_2, \cdots, Z_N\}$ the population obtained by the crossover operator. The relationship between population and parameters can be expressed as follows

$$E(Var(Y)) = (2F^2 + \frac{(N-1)}{N})Var(X) \tag{2}$$

$$E(Var(Z)) = (2F^2 p + \frac{2p}{N} + \frac{p^2}{N} + 1)Var(X) \tag{3}$$

where the $E()$ and the $Var()$ represent the mean and variance of the population respectively. The detailed proof of the process can refer to [26]. It can provide the basis for the selection of parameters and control the diversity of the population. However, only the relationship between population diversity and parameters is described qualitatively, but the variation of individual diversity and its quantitative measurement method are not further studied.

Mutation strategy is an important operator in DE algorithm. It not only affects the quality of the generated individuals, but also correlates the convergence rate of the algorithm. DE usually contains three types of mutation strategies: DE/rand, DE/best, DE/current [13], [27], [28]. DE/best strategy can accelerate the convergence speed of the algorithm, but it is easy to fall into local optimum because it has been searching around the best individual. DE/current strategy can keep the diversity of the algorithm well, but the convergence speed of the algorithm is reduced due to the lack of guidance from the best individual. However, DE/rand can be seen as a compromise between DE/current strategy and DE/best strategy. Hence, we propose a new mutation strategy called DE/current&best. In this way, the diversity of the algorithm can be guaranteed and the convergence speed of the algorithm can be accelerated.

III. PROPOSED INDIVIDUAL ENTROPY

In order to study the diversity of DE, the difference of individual is analyzed in the paper. The difference is considered from the following two aspects. One is the internal difference of individuals, the other is the external difference of individuals. For feature selection problem, there is complex interaction among features. Because an individually redundant (relevant) feature may become relevant (redundant or irrelevant) feature when it combines with other features. Hence, the internal difference of individuals is used to measure the relationship among features. For evolution of DE, we expect to obtain global optimal solution with stable and fast convergence property. If the diversity of the population is better, it means that the population has broad search space and slow down the convergence rate. On the contrary, if the population has a poor diversity, it speeds the convergence of the algorithm, but there may exist search stagnation or even can not find the global optimal solution. So we utilize external difference of individuals to measure the diversity of the population. Based on this, the individual entropy is presented.

Definition 1: Suppose individual $X = (x_1, x_2, \cdots, x_m)$, $X = seq_1 \cup seq_2 \cup, \cdots, \cup seq_n$, where seq_j is the subsequence of X and it satisfies $\forall seq_a \cap \forall seq_b = \oslash$, where $a \neq b$. k_j is the number of element in the seq_j, so $k_1 + k_2 + \cdots + k_n = m$. For sub-sequence $seq_j = (x_i, x_{i+1}, \cdots, x_{i+k_j-1})$, if $x_i = x_{i+1} = \cdots = x_{i+k_j-1}$, $0 < j < n+1$, $1 < i < m - k_j + 1$, then the individual entropy of X is defined as

$$IE(X) = -\sum_{j=1}^{n} \frac{|seq_j|}{k_j} log_2 \frac{|seq_j|}{k_j} \tag{4}$$

Inference 1: An individual $X = (x_1, x_2, \cdots, x_m)$, if $x_1 = x_2 =, \cdots, = x_m$, then the individual entropy of X reaches to the minimum 0. If $x_1 \neq x_2 \neq, \cdots, \neq x_m$, the individual entropy of X reaches to the maximum $log_2 m$.

Definition 2: Suppose population $P = (X_1, X_2, \cdots, X_{|P|})$, the average entropy $E(P) = \frac{\sum_{i=1}^{|P|} IE(X_i)}{|P|}$, so the population entropy of P is defined as

$$PE(P) = \sum_{i=1}^{|P|} (IE(X_i) - E(P))^2 \tag{5}$$

In order to describe the calculation process of individual entropy more clearly, a case is given in this section. Suppose a randomly generated population $P = \{X_1, X_2, X_3, X_4, X_5\}$, which contains five individuals and each individual contains ten binary value. Obviously, the value 1 represents the selected feature and the value 0 represents the non-selected feature. Thus, each individual represents a candidate features subset.

$$P = \begin{bmatrix} X_1 : 1110000111 \\ X_2 : 0010101100 \\ X_3 : 1001001111 \\ X_4 : 1111100000 \\ X_5 : 0011111000 \end{bmatrix} \tag{6}$$

Firstly, the entropy values of the X_1, X_2, X_3, X_4, X_5 are calculated by (4), and the process is shown as follows

$$IE(X_1) = -(\frac{3}{6}log_2\frac{3}{6} + \frac{4}{4}log_2\frac{4}{4} + \frac{3}{6}log_2\frac{3}{6}) = 1 \tag{7}$$

$$\begin{aligned} IE(X_2) = &-(\frac{2}{6}log_2\frac{2}{6} + \frac{1}{4}log_2\frac{1}{4} + \frac{1}{6}log_2\frac{1}{6} + \frac{1}{4}log_{20}\frac{1}{4} \\ &+\frac{1}{6}log_2\frac{1}{6} + \frac{2}{4}log_2\frac{2}{4} + \frac{2}{6}log_2\frac{2}{6}) = 3.4183 \end{aligned} \tag{8}$$

$$\begin{aligned} IE(X_3) = &-(\frac{1}{5}log_2\frac{1}{5} + \frac{2}{4}log_2\frac{2}{4} + \frac{1}{5}log_2\frac{1}{5} \\ &+\frac{2}{4}log_2\frac{2}{4} + \frac{3}{5}log_2\frac{3}{5}) = 2.3710 \end{aligned} \tag{9}$$

$$IE(X_4) = -(\frac{5}{5}log_2\frac{5}{5} + \frac{5}{5}log_2\frac{5}{5}) = 0 \tag{10}$$

$$IE(X_5) = -(\frac{2}{5}log_2\frac{2}{5} + \frac{5}{5}log_2\frac{5}{5} + \frac{3}{5}log_2\frac{3}{5}) = 0.9710 \tag{11}$$

Then, the individuals are arranged in descending order according to the entropy value, so we can obtain X_2, X_3, X_1, X_5, X_4. From the distribution of the individual, it is obvious that the randomicity of X_2 is the best and the randomicity of X_4 is the worst. The result of observation is consistent with our calculation results. In other words, the proposed individual entropy can measure individual diversity effectively.

Further, the average value of the individual entropy is calculated as $E(P) = 1.5521$. So the entropy of the whole population is calculated as $PE(P) = 7.2048$. It is clear that (4) depicts the differences within an individual. The larger the value of individual entropy means that we consider more the combination of features, which helps to explore local optimal solutions. While (5) reflects the differences among individuals. The larger the value of population entropy means that we consider a broader feature subset space, which helps to exploit global optimal solutions. Therefore, the diversity measurement method proposed in this paper can effectively monitor the evolution process of evolutionary algorithm.

IV. PROPOSED BINARY DIFFERENTIAL EVOLUTION BASED ON INDIVIDUAL ENTROPY

In this section, the design process of each operator in the proposed algorithm is described in detail. It mainly contains the following basic operations: population initialization strategy, individual evaluation function, new mutation operator, adaptive crossover operator and individual selection strategy.

A. POPULATION INITIALIZATION STRATEGY

Opposition-based Learning (OL) [29] is an efficient way for obtaining the better solution for the next generation iteration. During the process of OL, it not only evaluates the optimal solution, but also evaluates the solution in the opposite direction. Inspired by the idea of OL, we proposed an improved strategy called local opposition-based Learning (LOL).

Definition 3: Suppose an individual $I = (x_1, x_2, \cdots, x_n)$, where $x_i \in \{0, 1\}, i = 1, 2, \cdots, n$. The I_1 and I_2 are the local opposition individual of individual I. Then the I_1 and I_2 are defined as $I_1 = (\overline{x}_1, \overline{x}_2, \cdots, \overline{x}_{\lfloor \frac{n}{k} \rfloor}, x_{\lfloor \frac{n}{k} \rfloor+1}, x_{\lfloor \frac{n}{k} \rfloor+2}, \cdots, x_n)$ and $I_2 = (x_1, x_2, \cdots, x_{\lfloor \frac{n}{k} \rfloor}, \overline{x}_{\lfloor \frac{n}{k} \rfloor+1}, \overline{x}_{\lfloor \frac{n}{k} \rfloor+2}, \cdots, \overline{x}_n)$.

Where k is the factor that control the size of opposition and the k is set to $n/2$ in the paper. Base on definition 3, we can infer that a randomly generated population P can produce two new initial populations P_1 and P_2. A candidate solution $I = (x_1, x_2, \cdots, x_n)$, $I \in P$, $\overline{I}$ is the local opposition of I. For $\forall \overline{I} \in \{P_1, P_2\}$, if $f(\overline{I}) < f(I)$, then we select $\overline{I}$ instead of I, else, we still select the I as the candidate solution. The $f()$ is evaluation function of candidate solution. As a result, we get the new initial population $\bar{P}$.

B. INDIVIDUAL EVALUATION FUNCTION

It is clear that samples belonging to the same class have stronger similarity, while samples belonging to different classes are more discriminating. For feature selection, we should select the feature subset that make sure the distance of samples within-class is smaller and distance of samples between-class is larger. Traditional evaluation criteria often ignores the influence of individual diversity on searching for optimal solution. In order to fully develop individual potential, the individual entropy is utilized as one of the evaluation indicators. Hence, the new individual evaluation function is presented.

$$IEF(X) = \alpha * \frac{DS}{DL} + (1 - \alpha) * \frac{1}{IE(X)} \tag{12}$$

where the between-class distance $DL = \sum_{i=1}^{c} p_i(\frac{1}{n_i}\sum_{j=1}^{n_i} x_j^i - m)(\frac{1}{n_i}\sum_{j=1}^{n_i} x_j^i - m)^T$, and the within-class distance $DS = \sum_{i=1}^{c}\frac{p_i}{n_i}\sum_{j=1}^{n_i}(\frac{1}{n_i}\sum_{j=1}^{n_i} x_j^i - x_j^i)(\frac{1}{n_i}\sum_{j=1}^{n_i} x_j^i - x_j^i)^T$. Beside, the x_j^i is the jth feature vector value in the ith class, and the average feature vector value $m = \sum_{i=1}^{c}\frac{p_i}{n_i}\sum_{k=1}^{n_i} x_k^i$. While the $IE(X)$ is the individual entropy of X and the weight coefficient α is set to 0.8 in the paper. So our proposed fitness function can not only effectively measure the classification ability of feature subset, but also balance the exploration and exploitation of the algorithm.

C. NEW MUTATION OPERATOR

Because feature selection is a discrete problem, and the new individual generated by traditional binary mutation operator usually do not satisfy the closeness. In order to tackle the issue, a new non-parameter binary mutation operator is designed in this paper. For the gth iteration initial individual $X_i = \{x_{i,1,g}, x_{i,2,g}, \cdots, x_{i,j,g}\}$, and the mutation individual $V_i = \{v_{i,1,g}, v_{i,2,g}, \cdots, v_{i,j,g}\}$, then the $v_{i,j,g}$ is calculated by

$$v_{i,j,g} = \begin{cases} x_{r_0,j,g} + (-1)^{x_{r_0,j,g}} * |x_{r_1,j,g} - x_{r_2,j,g}|, \; if \;\; \theta < 0.5 \\ x_{r_b,j,g} + (-1)^{x_{r_b,j,g}} * |x_{r_1,j,g} - x_{r_2,j,g}|, \; otherwise \end{cases} \tag{13}$$

where $r_0, r_1, r_2 \in 1, 2, \cdots, n$, and $r_0 \neq r_1 \neq r_2$, the r_b is the index of the best individual in the current population. While the j is the dimension of feature variables. We expect to increase individual diversity as much as possible in the early stage of evolution and reduce diversity as much as possible in the later stage of evolution so that the algorithm converges gradually to the optimal solution. So, the evolution process is divided into two stages according to the iterative process. If $\theta = \frac{g}{g_{max}} < 0.5$, then three individuals are selected randomly to implement mutation operations to explore more individuals that may become optimal solutions. If $\theta = \frac{g}{g_{max}} \geq 0.5$, then the individual with best fitness value should be inherited to ensure the algorithm can finally converges to the optimal solution.

D. ADAPTIVE CROSSOVER OPERATOR

Adaptive crossover factor is important for evolutionary process. When the performance of parent individuals is better, the smaller value of crossover factor should be selected to obtain more information by offspring individuals from parent individuals. If the fitness value of parent individual is worse, more new offspring individuals should be produced by the lager crossover factor. Hence, we present an adaptive mechanism based on individual fitness for obtaining the reasonable factor. The adaptive crossover factor $CF(X_i)$ is shown as

$$CF(X_i) = \frac{IEF(X_i) - IEF_l(X) + \mu}{IEF_u(X) - IEF_l(X) + \mu} \tag{14}$$

where the maximum and minimum fitness value of all individuals are expressed by $IEF_u(X)$ and $IEF_l(X)$ respectively. In order to prevent the crossover factor to be 0, the parameter μ is introduced and it is the absolute value of the difference between the minimum fitness value and the second minimum. Note that we select a random number from a normal distribution by $\eta = randn(CF(X_i), 0.1)$ [20]. So the crossover operator is presented as follows

$$u_{i,j,g} = \begin{cases} v_{i,j,g}, \; if \;\; (rand_j[0,1) \leq \eta \;\; or \;\; (j = j_{rand}) \\ x_{i,j,g}, \; otherwise \end{cases} \tag{15}$$

where j_{rand} is the dimension selected randomly from an individual. Hence, we can infer that the selection of crossover factor is associate with the fitness value of individual. In other words, the number of inherit element depends on the fitness difference of individuals.

E. INDIVIDUAL SELECTION STRATEGY

For generate the new population, we should compare the offspring individual with the corresponding parent individual, and the individual with better fitness value should inherited to the next generation. So, the individual selection strategy can be presented as

$$x_{i,j,g+1} = \begin{cases} u_{i,j,g}, \; if \;\; IEF(u_{i,j,g}) < IEF(x_{i,j,g}) \\ x_{i,j,g}, \; otherwise \end{cases} \tag{16}$$

where $u_{i,j,g}$ is the ith offspring individual in the gth iteration. If the fitness value of new individual yield the corresponding parent individual, then $u_{i,j,g}$ is set to $x_{i,j,g+1}$. Otherwise the current individual will be inherited in the next generation.

F. COMPLETE PROCEDURE OF THE PROPOSED BDIE

Based on the above analysis, a new approach for feature selection using binary differential evolution based on individual entropy is proposed. The main ideas of BDIE are as follows:

1) The initialize population is generated by local opposition-based learning, and it can improve the quality of individual population and avoid the unpredictable convergence rate caused by pure random initialization.
2) Consider the effect of individual diversity on population evolution process, the individual entropy is integrated into the fitness function. By (12), it can be learn that the proposed individual evaluation function can measure the classification quality of feature subset and also can monitor individual evolution process.
3) A two stage binary mutation operator is presented. The purpose of adopting this approach is to ensure that the more potential individuals are discovered in the early stage of evolution and the individuals converge to the optimal individual gradually in the later stage.
4) Adaptive crossover operator is adopted to generate the new individual. The crossover factor is selected according to the fitness value.
5) The fitness function is utilized to select the candidate individual (feature subsets), and make sure the superior individuals can be retained to the next generation. Repeat the process until the number of iterations reaches the required value.

Hence, the pseudo-code of binary differential evolution based on individual entropy for feature subset optimization is illustrated in Algorithm 1.

The time complexity of the proposed algorithm is analyzed in this paper and it contains five basic operations process. In the first stage (lines 2-4), we initialize the population based on local opposition-based learning. The process needs to compare the fitness values of three different populations, so the time complexity of the process is $O(3 \times P_{size})$. In the second stage (lines 7-11), we calculate individual fitness values and sort them in descending order, so the the time complexity is $O(P_{size})$. In the third stage (lines 12-20), the mutation operation is executed and the corresponding

Algorithm 1 Binary Differential Evolution Based on Individual Entropy(BDIE)

Input: Data set $DS = (x_1, x_2, \cdots, x_n, y)$, population size P_{size}, individual length H_{size}, iterations number g.
Output: Optimized feature subset S

1: $S_n = DS(\frac{(v-v_{min})}{(v_{max}-v_{min})})$ // Normalizing the data set.
2: $P = randerr(P_{size}, H_{size})$
3: Generate P_1 and P_2 by the LOL.
4: $\bar{P}$ is the initial population. // Generate the initial population by local opposition-based learning.
5: $g = 0$ // Initialize iteration number.
6: **While** maximum number of iterations is not meet g
7: **for** i =1 to P_{size} **do**
8: $F_{set} = S_n(:, find(X(i,:) == 1))$
9: $Fit(i) = IEF(X(i))$ // Calculate fitness of individual.
10: $Rank(Fit(i))$
11: **end for**
12: **for** i =1 to P_{size} **do**
13: **for** j =1 to H_{size} **do**
14: **if** $\theta < 0.5$ **then**
15: $v_{i,j,g} = x_{r_0,j,g} + (-1)^{x_{r_0,j,g}} * |x_{r_1,j,g} - x_{r_2,j,g}|$
16: **else**
17: $v_{i,j,g} = x_{r_b,j,g} + (-1)^{x_{r_0,j,g}} * |x_{r_1,j,g} - x_{r_2,j,g}|$
18: **end if**
19: **end for**
20: **end for**
21: **for** i =1 to P_{size} **do**
22: **for** j =1 to H_{size} **do**
23: **if** $(rand_j[0, 1) \leq \eta \ \ or \ \ (j = j_{rand})$ **then**
24: $u_{i,j,g} = v_{i,j,g}$
25: **else**
26: $u_{i,j,g} = x_{i,j,g}$
27: **end if**
28: **end for**
29: **end for**
30: **for** i =1 to P_{size} **do**
31: **if** $IEF(u_{i,j,g}) \leq IEF(x_{i,j,g})$ **then**
32: $x_{i,j,g+1} = u_{i,j,g}$
33: **else**
34: $x_{i,j,g+1} = x_{i,j,g}$
35: **end if**
36: **end for**
37: $fit_{best} = IEF(x_{b,j,g})$
38: $g = g + 1$
39: **end While**

time complexity is $O(P_{size} \times H_{size})$. In the fourth stage (lines 21-29), the adaptive crossover operator to generate offspring individual and the time complexity is $O(P_{size} \times H_{size})$. In the fifth stage (lines 30-36), selecting the best individual (feature subset) according to the selection strategy and the time complexity is $O(P_{size})$. Because of the number of iteration is g, so the total time complexity is $O(3 \times P_{size} + 2 \times g \times P_{size} + 2 \times g \times P_{size} \times H_{size})$, and the final time complexity is expressed as $O(P_{size}(3 + 2g + 2gH_{size}))$.

V. EXPERIMENTAL DESIGN AND RESULT ANALYSIS

A. DATASET AND PARAMETER SETUP

In order to test the performance of the proposed algorithm, we carried out the BDIE and the compared algorithm using matlab R2014b and WEKA 3.8.0. All the simulations are performed on a computer with the Intel Core i5-3470, 3.20 GHz CPU and 2 GB RAM. The low-dimensional data and high-dimensional data two types of data sets are adopted to verify the effectiveness of the algorithm. Among them, the low-dimensional data contains Wine, Lymph, WDBC, Ionophere, SPECTF and Sonar six data sets. The high-dimensional data set includes Musk, GLRC, Colon, SRBCT, Leukemia-2 and Leukemia-3 six data sets. The number of samples (min = 62, max = 569) and the number of features (min = 13, max = 7129) are different depending on the size of the data sets. In addition, these data sets are mainly collected from diverse fields such as chemistry, medical, biological information, sound and atmosphere. More information of the benchmark data sets are available from the UCI repository. The details of the data sets are shown in Table 1.

TABLE 1. Description of used data sets.

ID	Dataset	Instances	Features	Types	Areas	Class
1	Wine	178	13	Continuous&Integer	Chemistry	3
2	Lymph	148	18	Integer	Medical	4
3	WDBC	569	30	Continuous	Biology	2
4	Ionosphere	351	34	Continuous&Integer	Atmosphere	2
5	SPECTF	267	44	Integer	Medical	2
6	Sonar	208	60	Continuous	Voice	2
7	Musk	476	166	Integer	Chemistry	2
8	GLRC	76	698	Continuous	Medical	3
9	Colon	62	2000	Continuous	Bioinformatics	2
10	SRBCT	83	2308	Continuous	Bioinformatics	4
11	Leukemia-2	72	7219	Integer	Bioinformatics	2
12	Leukemia-3	72	7219	Integer	Bioinformatics	3

In the BDIE, the population size is set to 20, and the individual size is adaptively adjusted according to the size of the data set dimension. In other words, the length of the individual is consistent with the dimension of the data set, which allows each feature have the same probability of being selected. For more clearly show the evolution process of individual and population in the search process, the number of iterations of the algorithms is set to 500. Additionally, in order to make the results of the experiment more statistically significant, the algorithms are executed 10 times independently. More importantly, the filter-based methods and the population-based methods are introduced to compare with the proposed algorithm in the paper. Their basic ideas and parameter settings are described below.

1) The importance of feature is measured by its relevance to the class label in the ReliefF [30]. Generally, the greater the importance value of the feature, the stronger the classification ability of the feature. The algorithm is efficient, but it does not consider the redundancy between features, which makes the classification accuracy unsatisfactory.
2) Minimum Redundancy Maximum Correlation (mRMR) [31] is a filtered feature selection method. The core idea of mRMR is to maximize the correlation

between feature and categorical variable, and to minimize the redundancy between features. Mutual information is used in the algorithm to calculate correlation and redundancy.

3) Sparse Multinomial Logistic Regression via Bayesian L1 Regularization (SBMLR) [32] is the multinomial logistic regression method incorporating bayesian regularization using a Laplace prior to select the most discriminative features. The algorithm mainly achieves feature sparsity through L1 regular terms to reduce the computational expense.
4) MoDEFS [33] is the unsupervised feature selection algorithm using an improved differential evolution technique. The average standard deviation of the selected feature subset, the average dissimilarity of the selected features, and the average similarity of non-selected features with respect to their first nearest neighbor selected features are considered to optimize the feature subset.
5) Genetic algorithm (GA) [34] is the binary version for feature selection. For individual coding, we use a binary vector to represent a chromosome. Where 0 means the feature is not selected and 1 represents the feature is selected. The crossover factor and the variation factor are set to 0.9 and 0.1, respectively. In addition, the elitist selection strategy is used in the algorithm.
6) Ant colony optimization (ACO) [35] can be modified for feature selection. The search space of the method can be regarded as a complete graph. The nodes in the graph represent the original feature set, and the reciprocal of the similarity value between the features is used as heuristic information to guide the ants to search for the optimal feature subset. Where the number of ants is set to 10 and the pheromone evaporation rate of ants is set to 0.05.
7) Particle swarm algorithm (PSO) [36]. Its core idea is to use each particle as a subset of candidate features, calculate and update the particle position information through distance, and finally obtained the approximate the optimal feature subset by iterative. The inertia weight coefficient is set to 0.5, and the acceleration factors both c_1 and c_2 are set to 2.
8) Difference algorithm (DE) [17] is also a population-based optimization algorithm. For the feature selection problem, the initial vector is used as a candidate feature subset, and the feature subset is updated by the mutation operation and the cross operation. The distance between samples within and between classes is calculated as the objective function to evaluate the quality of feature subsets. The mutation rate and crossover rate are 0.9 and 0.1, respectively.

For the fairness of comparison, the number of features selected by the filter-based algorithms is consistent with the results obtained by the BDIE. Their classification performance is compared with the same feature subset size. Moreover, the population size of the evolution algorithms mentioned is set to 20.

In order to verify the classification ability of the optimized feature subset, eight different classifiers including Naïve Bayes (NB), Logistic Regress (LR), Support Vector Machine (SVM), K-Nearest Neighbor (KNN), AdaBoost (AB), Decision Table (DT), C4.5 and Random Forest (RF) have been applied to evaluate the feature selection methods. These classifiers are available from WEKA software package [37].

B. PERFORMANCE OF BDIE

1) FITNESS CONVERGENCE CURVE OF THE BDIE

Figure 1 shows the fitness value convergence curve of the algorithm after 500 iterations on 12 data sets. BDIE_FA, BDIE_FB, and BDIE_FW respectively represent the average fitness value, the best fitness value and the worst fitness value of the proposed algorithm in 10 runs. It can be observed that the BDIE can converge smoothly to the optimal value in basically all cases. It seems that the appropriate fitness function can alleviate the local optimization shortcoming of algorithm. According to the distribution characteristics of the sample, the minimum internal distance and the maximum distance between classes are taken into account, which is to ensure that the feature subset has better classification ability. On the other hand, in order to speed up the convergence of proposed algorithm and obtain the superior candidate individuals, we introduce individual diversity metrics to find the potential candidate individuals in the search space as much as possible, and guide the population to the optimal individuals direction to ensure the global optimal.

2) CONVERGENCE AND DIVERSITY OF THE BDIE

In order to analyze the population diversity in the process of feature selection, we characterize individual diversity as the difference of the distribution of features in this paper. Here a new convergence calculation method [38] is adopted to quantify the convergence process of the algorithm. As mentioned earlier, there are complex interrelationships between features, where correlation and redundancy are important factors influencing the effect of feature selection. When a related feature (redundant feature) is combined with other features, the feature may become a redundant feature (related feature). Searching for feature combinations as much as possible is useful for searching for the optimal feature subset. Therefore, quantifying individual diversity is important for searching for feasible solutions.

Figure 2 exposes the trend of diversity and convergence curve of the algorithm in the process of iteration. It can be seen that the diversity value of the BDIE on the 11 data sets (excluding the GLRC data set) has larger fluctuations in the first half, but it gradually stabilizes in the latter half. It can be explained by the new mutation strategy in the paper. In the first half, the DE/current mutation strategy is adopted that can exploit fully the potential excellent feature subsets. While in the latter half, we use the DE/best mutation strategy,

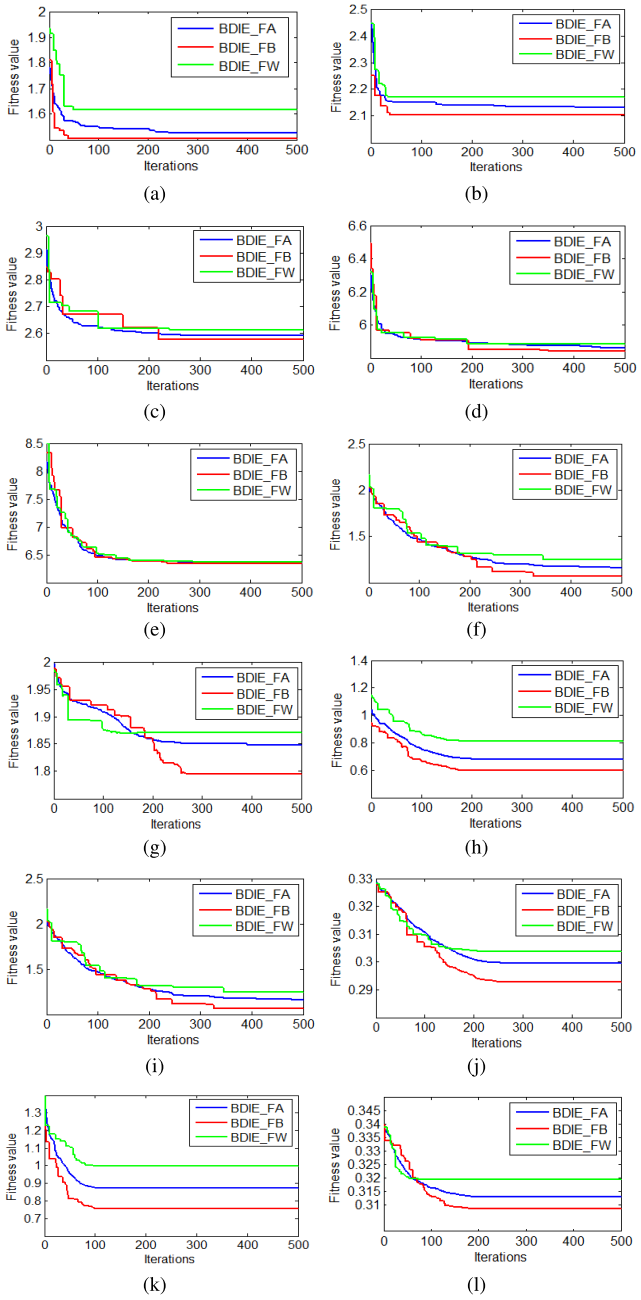


FIGURE 1. The fitness value and convergence curve obtained by IDBE for 12 datasets: (a) Wine, (b) Lymph, (c) WDBC, (d) Ionosphere, (e) SPECTF, (f) Sonar, (g) MUSK, (h) GLRC, (i) Colon, (j) SRBCT, (k) leukemia-2, (l) leukemia-3.

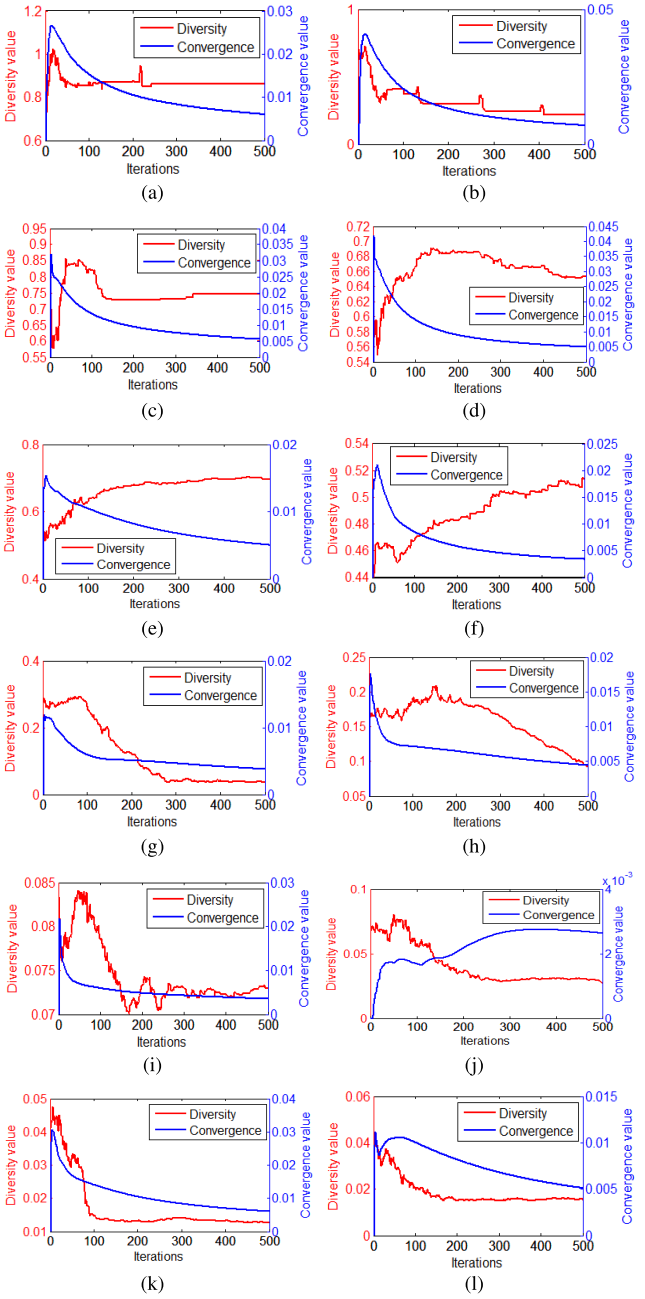


FIGURE 2. The diversity and convergence curve of BDIE for 12 datasets: (a) Wine, (b) Lymph, (c) WDBC, (d) Ionosphere, (e) SPECTF, (f) Sonar, (g) MUSK, (h) GLRC, (i) Colon, (j) SRBCT, (k) leukemia-2, (l) leukemia-3.

so that the outstanding individuals in the parent population are inherited into the offspring. Therefore, the population can converge to the optimal individual quickly. The final convergence value of the algorithm on nine data sets is less than 0.5, such as Lymph, SPECTF, Sonar, MUSK, GLRC, Colon, SRBCT, Leukemia-2 and Lemkemia-3. While the convergence values on the other three data sets are also less than 1. It means that the proposed algorithm can find an optimal (approximate) feature subset. In addition, diversity and convergence usually correspond to the global exploration ability and local development ability. And we can also infer that the mutation strategy of the proposed algorithm can well balance the diversity and convergence of the algorithm.

3) CLASSIFICATION PERFORMANCE OF THE BDIE

Table 2 and Table 3 report the classification accuracy of different algorithms on the corresponding classifier for each data set. The AVG and STD represent the average classification accuracy and standard deviation of the algorithm on different classifiers respectively. The RANK is the ranking of different feature selection algorithms according to the value of

TABLE 2. The classification (%) performance comparison between BDIE and filter-based algorithms.

Dataset	Algo.	NB	LR	SVM	KNN	AB	DT	C4.5	RF	AVG	STD	RANK
Wine	ReliefF	94.38	97.19	97.75	96.07	91.57	85.96	94.38	97.19	94.31	3.69	2
	mRMR	94.38	96.07	97.75	96.07	90.45	88.76	89.89	97.75	93.89	3.42	4
	SBMLR	96.63	94.94	97.75	96.63	90.45	85.96	93.82	97.75	94.24	3.87	3
	IMoDEFS	95.33	94.39	97.64	96.38	90.73	84.85	93.33	97.19	93.73	3.96	5
	BDIE	95.51	96.07	96.63	95.51	89.89	88.76	94.38	97.75	**94.31**	3.03	**1**
Lymph	ReliefF	82.43	81.08	83.11	82.43	69.59	76.35	75.68	83.11	79.22	4.58	**1**
	mRMR	81.76	77.70	79.73	75.00	69.59	75.68	73.65	79.73	76.60	3.69	5
	SBMLR	81.08	81.08	85.81	77.70	68.92	76.35	75.68	84.46	78.89	5.08	2
	IMoDEFS	80.35	81.32	85.46	79.21	67.78	75.65	75.68	84.46	78.74	5.32	3
	BDIE	81.76	79.73	84.46	82.43	66.22	72.30	75.67	83.78	78.29	6.00	4
WDBC	ReliefF	92.03	93.48	92.03	92.03	91.30	86.96	86.96	93.48	91.03	2.45	2
	mRMR	88.41	90.58	92.75	89.86	90.58	90.58	90.58	91.30	90.58	1.14	3
	SBMLR	86.23	92.75	84.06	86.96	87.68	86.23	87.68	90.58	87.77	2.55	5
	IMoDEFS	93.18	88.65	91.33	95.73	96.27	86.79	86.96	93.45	91.95	2.89	4
	BDIE	92.03	97.10	92.03	92.03	91.30	86.96	95.20	93.48	**92.11**	3.49	**1**
Ionosphere	ReliefF	87.75	82.91	84.62	89.46	92.02	85.19	90.88	93.45	88.28	3.55	5
	mRMR	90.31	87.46	86.61	90.03	86.89	86.89	90.88	94.30	89.17	2.53	3
	SBMLR	88.89	87.46	87.75	90.03	91.17	89.46	89.46	90.88	89.39	1.25	2
	IMoDEFS	93.68	85.86	85.64	83.87	91.28	85.23	90.63	93.46	88.71	3.72	4
	BDIE	89.46	87.18	88.03	88.60	92.02	86.89	91.45	93.73	**89.67**	2.32	**1**
SPECTF	ReliefF	80.00	75.00	80.00	73.75	71.25	70.00	76.25	80.00	75.78	3.75	3
	mRMR	77.50	82.50	80.00	78.75	76.25	72.50	71.25	87.50	78.28	4.92	2
	SBMLR	82.50	83.75	78.75	73.75	76.25	77.50	83.75	85.00	80.16	3.87	**1**
	IMoDEFS	76.88	73.88	71.65	66.88	74.63	72.50	66.25	85.00	73.46	5.54	5
	BDIE	75.00	70.00	80.00	75.00	73.75	67.50	66.25	81.25	73.59	5.09	4
Sonar	ReliefF	67.79	76.44	79.33	82.21	77.88	72.60	75.48	80.29	76.50	4.32	5
	mRMR	70.19	79.33	80.77	86.06	79.33	71.63	81.25	84.13	79.09	5.20	3
	SBMLR	75.00	80.77	80.77	85.58	82.21	74.04	82.69	84.62	80.71	3.91	**1**
	IMoDEFS	68.77	76.76	79.33	75.05	81.83	72.60	73.99	84.18	76.56	4.73	4
	BDIE	72.60	78.37	81.73	85.10	80.29	75.00	80.77	85.58	79.93	4.23	2
Musk	ReliefF	69.96	71.85	71.43	75.84	72.27	70.17	79.62	80.67	73.98	3.95	3
	mRMR	62.82	61.97	64.08	77.73	65.34	65.34	72.48	80.46	68.78	6.69	5
	SBMLR	72.48	75.42	75.84	79.20	68.49	70.80	75.84	79.62	74.71	3.65	2
	IMoDEFS	71.65	74.02	70.72	76.09	72.33	71.22	70.70	80.65	73.42	3.23	4
	BDIE	73.74	78.36	80.25	84.45	75.63	71.64	72.90	82.98	**77.49**	4.47	**1**
GLRC	ReliefF	51.97	61.18	61.84	70.39	68.42	67.76	71.71	72.37	65.71	6.50	2
	mRMR	60.53	63.16	53.29	65.13	64.47	61.84	66.45	62.50	62.17	3.79	5
	SBMLR	57.89	65.13	64.47	51.97	61.84	60.53	63.82	65.79	61.43	4.34	4
	IMoDEFS	56.55	63.85	70.65	65.07	67.72	65.38	61.95	72.61	65.47	4.71	3
	BDIE	47.37	67.76	78.95	78.95	68.42	66.45	63.16	77.63	**68.59**	9.88	**1**
Colon	ReliefF	72.58	72.58	79.03	69.35	70.97	61.29	70.97	77.42	71.77	5.04	2
	mRMR	64.52	58.06	62.90	64.52	67.74	64.52	80.65	69.35	66.53	6.18	5
	SBMLR	62.90	69.35	72.58	66.13	62.90	66.13	72.58	66.13	67.34	3.58	4
	IMoDEFS	65.23	70.79	62.90	71.65	65.54	63.85	70.17	72.40	67.82	3.57	3
	BDIE	61.29	72.58	80.65	77.42	66.13	83.87	72.58	83.87	**74.80**	7.69	**1**
SRBCT	ReliefF	39.76	33.73	46.99	39.76	39.76	39.76	36.14	38.55	39.31	3.56	5
	mRMR	97.59	93.98	93.98	92.77	56.63	83.13	85.54	90.36	86.75	12.21	4
	SBMLR	93.98	97.59	90.36	98.80	61.45	81.93	91.57	97.59	89.16	11.65	2
	IMoDEFS	91.55	93.35	92.97	97.64	60.36	82.78	87.07	93.42	87.39	11.05	3
	BDIE	97.59	98.80	97.59	98.80	59.04	83.13	90.36	95.18	**90.06**	12.76	**1**
Leukemia-2	ReliefF	95.83	97.22	97.22	93.06	95.83	91.67	87.50	97.22	94.44	3.26	2
	mRMR	73.61	76.39	76.39	73.61	72.22	70.83	68.06	72.22	72.92	2.60	5
	SBMLR	91.67	76.39	88.89	79.17	90.28	87.50	81.94	86.11	85.24	5.15	3
	IMoDEFS	76.52	79.64	71.85	78.65	73.33	70.67	69.85	73.55	74.26	3.41	4
	BDIE	95.83	97.22	98.61	100.00	93.06	93.06	93.06	95.83	**95.83**	2.50	**1**
Leukemia-3	ReliefF	92.35	95.77	91.93	84.49	95.77	91.93	84.49	95.77	91.56	4.39	2
	mRMR	76.22	74.50	73.76	75.63	61.83	76.39	73.17	76.39	73.49	4.56	5
	SBMLR	94.44	93.06	95.83	90.28	81.94	77.78	80.56	94.44	88.54	6.80	3
	IMoDEFS	75.39	74.83	76.61	78.23	72.85	73.52	77.06	76.39	75.61	1.70	4
	BDIE	100.00	97.22	98.61	97.22	94.44	91.67	94.44	95.83	**96.18**	2.48	**1**

the AVG. If the AVG values of the two algorithms are equal, then the algorithm with a smaller STD value is better. Besides, the bold values in the table represent the best classification accuracy.

Table 2 shows the comparison of the classification performance of the proposed algorithm with three filtering algorithms and an improved differential algorithm. It can be seen from Table 2 that the proposed algorithm achieves the best average classification accuracy on nine data sets. At the same time, the performance on the remaining three data sets is not the worst. The proposed algorithm has an average classification accuracy of more than 90% on the wine, WDBC, SRBCT, Leukemia-2 and Leukemia-3 data sets, and it is also significantly higher than other methods. The main purpose of Table 2 is to verify the robustness of the proposed algorithm on different classifiers. Obviously, we can see that BDIE algorithm can select feature subset with stable classification ability. In other words, the features selected by the proposed algorithm contain the key classification information. This is because the population-based heuristic search strategy is adopted in this paper, and it is different from other traditional filter-based algorithms such as ReliefF, mRMR and SBMLR. Each individual represents a subset of candidate features, and the individual is updated

TABLE 3. The classification (%) performance comparison between BDIE and population-based algorithms.

Dataset	Algo.	NB	LR	SVM	KNN	AB	DT	C4.5	RF	AVG	STD	RNAK
Wine	GA	96.07	93.82	95.51	96.07	89.89	86.52	91.01	94.94	92.98	3.26	2
	ACO	97.19	97.19	96.07	97.19	85.96	79.78	91.57	93.82	92.35	5.98	3
	PSO	94.38	92.13	93.82	88.20	87.64	85.39	89.33	94.94	90.73	3.34	5
	DE	94.94	96.07	95.51	93.26	88.20	85.76	89.26	95.31	92.29	3.71	4
	BDIE	95.51	96.07	96.63	95.51	89.89	88.76	94.38	97.75	**94.31**	3.03	**1**
Lymph	GA	82.43	77.03	78.38	77.70	66.89	79.05	79.05	81.08	77.70	4.40	2
	ACO	81.76	79.05	74.32	79.73	68.92	70.95	73.65	82.43	76.35	4.75	4
	PSO	75.00	72.97	72.30	75.00	65.54	78.38	75.68	76.35	73.90	3.62	5
	DE	82.43	77.03	78.38	77.70	66.89	79.05	77.03	81.08	77.45	4.38	3
	BDIE	81.76	79.73	84.46	82.43	66.22	72.30	75.68	83.78	**78.29**	6.00	**1**
WDBC	GA	92.03	94.93	93.48	91.30	92.75	86.96	88.41	91.30	91.39	2.44	2
	ACO	86.96	93.48	93.48	91.30	92.03	86.96	86.96	89.86	90.13	2.69	5
	PSO	91.30	96.38	95.65	89.86	92.03	83.33	90.58	92.03	91.39	3.74	3
	DE	91.30	93.48	92.03	90.58	92.03	90.58	86.96	92.03	91.12	1.80	4
	BDIE	92.03	97.10	92.03	92.03	91.30	86.96	95.20	93.48	**92.11**	3.49	**1**
Ionosphere	GA	91.17	87.46	88.03	88.03	87.18	87.75	92.31	95.16	89.64	2.72	3
	ACO	89.74	86.89	87.18	89.74	87.75	88.03	89.74	89.46	88.57	1.15	4
	PSO	89.46	81.48	82.34	90.03	83.76	88.89	88.32	92.02	87.04	3.68	5
	DE	90.31	87.75	88.03	91.74	91.45	87.46	90.88	93.16	**90.10**	1.98	**1**
	BDIE	89.46	87.18	88.03	88.60	92.02	86.89	91.45	93.73	89.67	2.32	2
SPECTF	GA	77.50	73.75	78.75	75.00	72.50	70.00	73.75	81.25	75.31	3.41	3
	ACO	75.00	78.75	73.75	77.50	75.00	77.50	75.00	81.25	76.72	2.33	2
	PSO	78.75	81.25	76.25	61.25	68.75	67.50	65.00	73.75	71.56	6.58	5
	DE	81.25	82.50	77.50	67.50	75.00	78.75	75.00	78.75	**77.03**	4.37	**1**
	BDIE	75.00	70.00	80.00	75.00	73.75	67.50	66.25	81.25	73.59	5.09	4
Sonar	GA	66.35	75.96	76.44	88.94	78.37	73.56	76.44	81.73	77.22	6.06	2
	ACO	69.71	75.00	74.52	73.08	73.08	73.08	75.00	76.44	73.74	1.89	3
	PSO	59.62	70.19	68.27	63.94	72.60	69.23	70.19	66.83	67.61	3.86	5
	DE	64.90	74.04	74.04	79.33	71.15	75.00	70.19	78.85	73.44	4.41	4
	BDIE	72.60	78.37	81.73	85.10	80.29	75.00	80.77	85.58	**79.93**	4.23	**1**
Musk	GA	70.59	78.99	73.53	79.20	75.63	72.69	81.09	87.61	77.42	5.13	3
	ACO	60.71	69.33	69.75	75.63	70.17	69.75	73.95	81.93	71.40	5.71	4
	PSO	61.55	59.24	55.04	77.73	59.45	66.60	78.99	84.03	67.83	10.21	5
	DE	66.18	73.74	76.47	84.24	76.68	76.89	78.36	87.18	77.47	5.96	2
	BDIE	73.74	78.36	80.25	84.45	75.63	71.64	72.90	82.98	**77.49**	4.47	**1**
GLRC	GA	43.42	79.61	73.03	71.05	68.42	66.45	61.18	73.03	67.02	10.25	5
	ACO	48.03	75.66	78.29	78.29	68.42	64.47	61.84	79.61	**69.33**	10.25	2
	PSO	46.05	78.95	78.95	79.61	68.42	65.13	60.53	79.61	69.65	11.36	**1**
	DE	45.39	76.97	77.63	78.29	68.42	65.79	59.87	76.32	68.59	10.77	4
	BDIE	47.37	67.76	78.95	78.95	68.42	66.45	63.16	77.63	68.59	9.88	3
Colon	GA	61.29	75.81	62.90	74.19	67.74	67.74	66.13	75.81	68.95	5.33	4
	ACO	61.29	69.35	80.65	75.42	64.52	83.87	72.58	80.65	73.54	7.59	2
	PSO	67.74	59.68	72.58	56.45	51.61	59.68	64.52	64.52	62.10	6.19	5
	DE	60.48	76.49	81.01	77.46	64.56	81.61	69.32	71.52	72.81	7.19	3
	BDIE	61.29	72.58	80.65	77.42	66.13	83.87	72.58	83.87	**74.80**	7.69	**1**
SRBCT	GA	93.98	90.12	92.85	90.25	65.22	81.75	83.91	89.77	85.98	8.76	2
	ACO	84.32	86.65	81.22	85.04	69.84	80.60	81.37	88.09	82.14	5.29	5
	PSO	82.65	83.55	88.45	83.67	68.88	86.58	84.33	89.63	83.47	5.99	4
	DE	90.33	89.85	91.55	88.63	60.32	81.33	84.74	89.37	84.52	9.67	3
	BDIE	97.59	98.80	97.59	98.80	59.04	83.13	90.36	95.18	**90.06**	12.76	**1**
Leukemia-2	GA	91.67	76.39	88.89	79.17	90.28	87.50	81.94	86.11	85.24	5.15	3
	ACO	86.60	81.52	87.27	88.39	91.22	80.99	85.63	81.33	85.37	3.52	2
	PSO	69.44	65.28	80.56	83.33	76.39	79.17	80.56	83.33	77.26	6.17	5
	DE	90.75	74.66	85.03	79.62	91.08	86.32	79.89	86.58	84.24	5.38	4
	BDIE	95.83	97.22	98.61	100.00	93.06	93.06	93.06	95.83	**95.83**	2.50	**1**
Leukemia-3	GA	93.06	92.37	95.83	86.11	94.44	77.78	80.56	83.33	87.93	6.45	3
	ACO	90.52	91.37	92.01	87.22	91.74	75.87	82.77	92.32	87.98	5.49	2
	PSO	93.06	91.62	93.54	84.22	86.02	79.66	81.33	91.82	87.66	5.19	4
	DE	92.11	90.53	91.33	89.56	81.02	75.24	79.67	92.32	86.47	6.30	5
	BDIE	100.00	97.22	98.61	97.22	94.44	91.67	94.44	95.83	**96.18**	2.48	**1**

by the designed evaluation function to guide the population to search for the optimal solution direction. A great advantage of the algorithm is that it considers more feature combinations to get the global optimal solution. Therefore, the population-based BDIE can obtain better classification results.

Table 3 shows the classification results of the proposed algorithm and the four population-based approaches for feature selection. From Table 3, it is observed that BDIE achieves the best average classification accuracy on the Wine, Lymph, WDBC, Sonar, Musk, Colon, SRBCT, Leukemia-2 and Leukemia-3 data sets. For example, on the Leukemia-2 and Leukemia-3 data sets, the average classification accuracy of the proposed algorithm is 95.83% and 96.18%, respectively. For Leukemia-2 data set, the BDIE provides superior classification accuracy compared to GA, ACO, PSO and DE, and they are 10.59%, 10.46%, 18.58% and 11.59% higher than them respectively. Subsequently, the classification accuracy of the BDIE on the KNN classifier is as high as 100%. For the leukemia-3 data set, the average classification accuracy of the BDIE is 8.25%, 8.20%, 8.52% and 9.71% higher than the average classification accuracy of GA, ACO, PSO and DE algorithms, especially on the SVM classifier. The classification accuracy of the algorithm is 98.61%. The reason for this phenomenon is that the above four population-based

optimization algorithms only utilize the classification results to evaluate the feature subset, and they do not analyze the complex relationship between features deeply. While the distance between the sample within the class and the distance between the samples is considered in the paper, which helps to distinguish the differences of the samples. More importantly, more feature combinations are measured by the individual diversity indicators. It can expand the search range of the algorithm and increase the probability of finding the optimal combination of features.

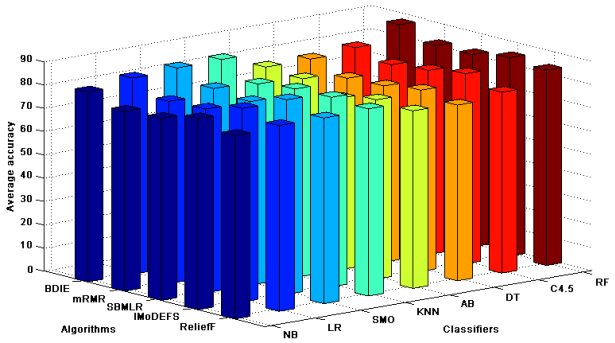


FIGURE 3. The average classification accuracy of BDIE and filter-based algorithms on 12 data sets.

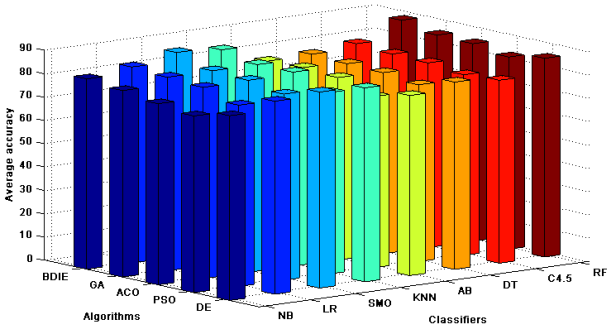


FIGURE 4. The average classification accuracy of BDIE and population-based algorithms on 12 data sets.

Figure 3 and Figure 4 show the average classification results of the algorithm over all 12 data sets. It can be seen that the average classification accuracy of BDIE on NB, LR, SVM, KNN, AB, DT, C4.5 and RF classifiers are 81.85%, 85.03%, 88.13%, 87.96%, 79.11%, 80.60%, 81.83% and 88.91%. From Figure 3, it can be detected that the SBMLR algorithm has a classification accuracy of 81.97% on the NB classifier, which is higher than the proposed algorithm. As shown in Figure 4, the GA algorithm has a classification accuracy of 79.18% on the AB classifier, which is slightly higher than the BDIE. However, for the remaining classifiers, the BDIE can achieve the best classification accuracy over all data sets. In addition, we can also find that all algorithms are better classified on the RF classifier than other classifiers. Besides, compared with the other eight algorithms, the proposed algorithm has the best classification result on the RF classifier. Therefore, the improvements are meaningful and the classification accuracy obtained by the BDIE is superior to those of other competitors.

4) DIMENSIONAL REDUCTION RATE OF BDIE

In terms of feature subset size, the feature reduction performance of the proposed algorithm is presented in the paper. The dimension reduction rate is calculated by $R = 1 - f/F$, where f is the number of the selected feature and F is the number of initial data set dimension. Figure 5 shows the reduction values of the BDIE over all 12 data sets. From Figure 5, it can be seen that the higher the dimension of the data set, the higher the reduction rate. For example, the reduction rates for Sonar, Musk, GLRC, Colon, SRBCT, Leukemia-2 and Leukemia-3 data sets are 81.67%, 89.76%, 96.42%, 99.05%, 99.65%, 99.70% and 99.72%, respectively. We can infer that the proposed algorithm can effectively remove irrelevant features and redundant features especially for high dimensional data sets, and it significantly reduces the size of the feature subset.

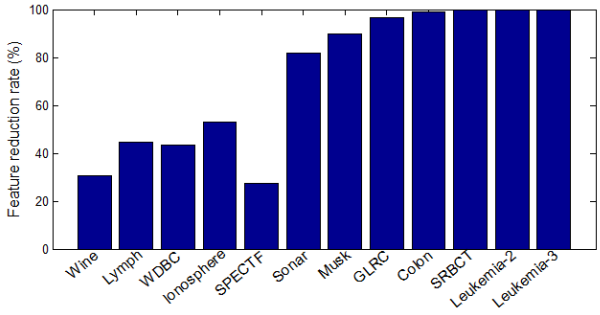


Fig. 5. The feature reduction rate of BDIE on 12 data sets.

5) COMPUTING TIME OF BDIE

For test the performance of the algorithm in terms of time cost, the average time cost of the competitors is calculated in this paper. As shown in Table 4, the filter-based algorithms usually need to choose the appropriate threshold (the feature importance value or the number of features to be selected) before determining the feature subset. In order to ensure the fairness of the comparison, we consider that the selected features number in filter-based algorithms is consistent with the number of features selected by the proposed algorithm. Therefore, the time cost on different data sets with the same feature subset size is analyzed. As can be seen from Table 4, the average time overhead of the ReliefF, mRMR, SBMLR, IMoDEFS and BDIE algorithms on all data sets are 4.76s, 1480.97s, 22.16s, 493.53s, and 193.27s, respectively. Based on the ranks, we can see that the RANK value of BDIE is 3. It means that the time expenses of the algorithm is not outstanding. This is mainly because the ReliefF and SBMLR algorithms only calculate the correlation between features and class labels without calculating the result between features, so they run faster. While the time usage of mRMR will increase rapidly with the size of the data set, because it needs to calculate the correlation between features. In addition, although IMoDEFS is an improved version of DE, its time cost is still not superior to that of BDIE.

TABLE 4. The time cost comparison between BDIE and filter-based algorithms.

Dataset	ReliefF [30]		mRMR [31]		SBMLR [32]		IMoDEFS [33]		BDIE	
	T(s)	NF	T(s)	NF	T(s)	NF	T(s)	NF	T(s)	NF
Wine	0.55	9	0.89	9	1.03	9	89.04	9	21.43	9
Lymph	0.56	10	0.98	10	1.96	10	91.11	10	25.32	10
WDBC	0.84	17	0.98	17	1.04	17	98.65	17	67.43	17
Ionosphere	0.83	16	0.93	16	74.25	16	131.73	16	171.13	16
SPECTF	1.05	32	2.31	32	1.74	32	109.52	32	75.09	32
Sonar	1.22	11	1.11	11	0.95	11	133.81	11	165.61	11
Musk	2.03	17	2.13	17	0.99	17	192.33	17	218.03	17
GLRC	3.14	25	2671.62	25	2.33	25	374.29	25	145.44	25
Colon	4.36	19	3197.53	19	168.46	19	622.47	19	98.40	19
SRBCT	10.30	8	3202.31	8	6.25	8	1204.46	8	99.27	8
Leukemia-2	15.85	22	3958.76	22	3.10	22	1398.28	22	381.11	22
Leukemia-3	16.44	20	4732.08	20	3.83	20	1476.62	20	450.51	20
AVG	**4.76**	17.17	1480.97	17.17	22.16	17.17	493.53	17.17	159.90	17.17
RANK	**1**		5		2		4		3	

TABLE 5. The time cost comparison between BDIE and population-based algorithms.

Dataset	GA [34]		ACO [35]		PSO [36]		DE [17]		BDIE	
	T(s)	NF	T(s)	NF	T(s)	NF	T(s)	NF	T(s)	NF
Wine	126.73	6	1068.69	7	106.94	9	94.38	10	21.43	9
Lymph	109.52	8	1102.24	6	92.78	10	88.40	12	25.32	10
WDBC	73.77	12	1050.34	19	87.50	8	86.83	11	67.43	17
Ionosphere	472.62	13	1110.54	18	98.22	15	156.95	11	171.13	16
SPECTF	29.43	17	966.69	25	81.65	13	57.17	15	75.09	32
Sonar	175.79	24	1032.05	10	88.81	13	110.01	17	165.61	11
Musk	214.53	74	1259.47	21	104.82	18	467.15	54	218.03	17
GLRC	115.63	310	1132.15	103	92.72	160	198.20	245	145.44	25
Colon	90.57	992	2071.44	382	498.83	353	964.97	477	98.40	19
SRBCT	268.87	1259	2220.28	247	159.44	418	1317.51	563	99.27	8
Leukemia-2	605.22	3585	10040.49	1634	372.72	1504	2529.09	1889	381.11	22
Leukemia-3	698.69	3610	13701.81	1801	454.54	999	2608.65	1905	450.51	20
AVG	248.45	825.83	3063.02	356.08	186.58	293.33	723.28	434.08	**159.90**	**17.17**
RANK	3	5	5	3	2	2	4	4	**1**	**1**

In order to further verify the advantages of the algorithm in terms of time cost. The efficiency of population-based optimization algorithm is calculated. Then, the four algorithms including GA, ACO, PSO and DE algorithms are compared with the proposed algorithm. As shown in Table 5, the average time overhead of the BDIE algorithm is 159.90s, which is less than that of the other four algorithms. In terms of feature subset size, the average number of features obtained in this paper is 17.17, which is obviously superior to other algorithms. In other words, our algorithm can achieve better experimental results in terms of feature subset size and time cost.

VI. CONCLUSION

In this paper, a novel binary differential evolution based on individual entropy for feature subset optimization is proposed. The individual entropy method is constructed to evaluate the diversity of the individual. And it can measure the complex relationship between features. Then, the influence of individual entropy on feature subset selection is considered in the design of objective function, which helps to select better combination of features. More importantly, two-stage mutation operator is designed according to different search stages, and a new adaptive crossover factor is presented. It is not only beneficial to discover the features with stronger correlation, but also to speed up the operation of the algorithm. Finally, experiments show that the proposed algorithm is superior to other algorithms. It can effectively improve the classification accuracy, reduce the size of feature subset and shorten the running time.

However, the proposed algorithm in this paper has limitations. For example, how to measure the stability of feature selection based on evolutionary algorithm and whether there is correlation between classifier and optimization objective. In the future, we will continue to explore feature selection algorithms based on evolutionary computing, especially the efficient search strategies and the reasonable evaluation criteria. At the same time, we will study the factors that affect the stability of feature selection.

REFERENCES

[1] J. Izetta, P. F. Verdes, and P. M. Granitto, "Improved multiclass feature selection via list combination," *Expert Syst. Appl.*, vol. 88, pp. 205–216, Dec. 2017.

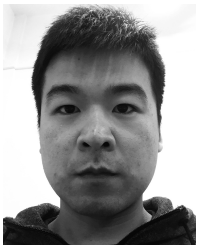
[2] A. Datta, S. Ghosh, and A. Ghosh, "Self-adaptive differential evolution for feature selection in hyperspectral image data," *Appl. Soft. Comput.*, vol. 13, no. 4, pp. 1969–1977, 2013.

[3] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A. M. Al-Zoubi, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," *Expert Syst. Appl.*, vol. 117, pp. 267–286, Mar. 2019.

[4] A. A. Yahya, A. Osman, A. R. Ramli, and A. Balola, "Feature selection for high dimensional data: An evolutionary filter approach," *J. Comput. Sci. Technol.*, vol. 7, no. 5, pp. 800–820, 2011.

[5] S. Oreski and G. Oreski, "Genetic algorithm-based heuristic for feature selection in credit risk assessment," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 2052–2064, 2014.

[6] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Appl. Soft Comput.*, vol. 18, pp. 261–276, May 2014.
[7] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Appl. Soft. Comput.*, vol. 43, pp. 117–130, Jun. 2016.
[8] B. Z. Dadaneh, H. Y. Markid, and A. Zakerolhosseini, "Unsupervised probabilistic feature selection using ant colony optimization," *Expert Syst. Appl.*, vol. 53, pp. 27–42, Jul. 2016.
[9] P. Shunmugapriya and S. Kanmani, "A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid)," *Swarm Evol. Comput.*, vol. 36, pp. 27–36, Oct. 2017.
[10] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[11] H. Peng, C. Ying, S. Tan, B. Hu, and Z. Sun, "An improved feature selection algorithm based on ant colony optimization," *IEEE Access*, vol. 6, pp. 69203–69209, 2018.
[12] W. Ke, C. Wu, Y. Wu, and N. N. Xiong, "A new filter feature selection based on criteria fusion for gene microarray data," *IEEE Access*, vol. 6, pp. 61065–61076, 2018.
[13] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
[14] G. Pampara, A. P. Engelbrecht, and N. Franken, "Binary differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2006, pp. 1873–1879.
[15] E. Zorarpacı and S. A. Özel, "A hybrid approach of differential evolution and artificial bee colony for feature selection," in *Expert Systems With Applications*. New York, NY, USA: Pergamon, 2016.
[16] F. Neri and V. Tirronen, "Recent advances in differential evolution: A survey and experimental analysis," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 61–106, 2010.
[17] L. Tang, Y. Dong, and J. Liu, "Differential evolution with an individual-dependent mechanism," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 560–574, Aug. 2015.
[18] K. Zielinski, P. Weitkemper, R. Laur, and K.-D. Kammeyer, "Parameter study for differential evolution using a power allocation problem including interference cancellation," in *Proc. IEEE Congr. Evol. Comput.*, 2006, pp. 1857–1864.
[19] J. Ronkkonen, S. Kukkonen, and K. V. Price, "Real-parameter optimization with differential evolution," in *Proc. IEEE Congr. Evol. Comput.*, Sep. 2005, pp. 506–513.
[20] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.
[21] S. M. Elsayed, R. A. Sarker, and T. Ray, "Differential evolution with automatic parameter configuration for solving the CEC2013 competition on real-parameter optimization," in *Proc. Evol. Comput.*, 2013, pp. 1932–1937.
[22] Q. Fan and X. Yan, "Self-adaptive differential evolution algorithm with discrete mutation control parameters," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1551–1572, 2015.
[23] X.-J. Bi and J. Xiao, "Classification-based self-adaptive differential evolution with fast and reliable convergence performance," *Soft Comput.*, vol. 15, no. 8, pp. 1581–1599, 2011.
[24] A. Viktorin, R. Senkerik, M. Pluhacek, T. Kadavy, and A. Zamuda, "Distance based parameter adaptation for differential evolution," in *Proc. IEEE Comput. Intell.*, Nov./Dec. 2018, pp. 1–7.
[25] S.-H. Wang, Y.-Z. Li, Y.-H. Yu, and H. Liu, "Self-adaptive differential evolution algorithm with improved mutation strategy," *Soft Comput.*, vol. 22, no. 10, pp. 3433–3447, 2018.
[26] D. Zaharie, "Critical values for the control parameters of differential evolution algorithms," in *Proc. MENDEL*, 2002, pp. 62–67.
[27] W. Qian, J. Chai, Z. Xu, and Z. Zhang, "Differential evolution algorithm with multiple mutation strategies based on roulette wheel selection," *Appl. Intell.*, vol. 48, no. 10, pp. 3612–3629, 2018.
[28] N. H. Awad, M. Z. Ali, and R. M. Duwairi, "Multi-objective differential evolution based on normalization and improved mutation strategy," *Natural Comput.*, vol. 16, no. 4, pp. 661–675, 2016.
[29] S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 64–79, Feb. 2008.
[30] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, Oct. 2003.
[31] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1226–1238, 2005.
[32] G. C. Cawley, N. L. Talbot, and M. Girolami, "Sparse multinomial logistic regression via Bayesian L_1 regularisation," in *Proc. Int. Conf. Neural Inf. Process.*, 2007, pp. 209–216.
[33] T. Bhadra and S. Bandyopadhyay, "Unsupervised feature selection using an improved version of differential evolution," *Expert Syst. Appl.*, vol. 42, no. 8, pp. 4042–4053, 2015.
[34] H. Dong, T. Li, R. Ding, and J. Sun, "A novel hybrid genetic algorithm with granular information for feature selection and optimization," *Appl. Soft. Comput.*, vol. 65, pp. 33–46, Apr. 2018.
[35] S. Kashef and H. Nezamabadi-Pour, "An advanced ACO algorithm for feature subset selection," *Neurocomputing*, vol. 147, pp. 271–279, Jan. 2015.
[36] B. Tran, B. Xue, and M. Zhang, "A new representation in pso for discretization-based feature selection," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1733–1746, Jun. 2017.
[37] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.
[38] J. He and G. Lin, "Average convergence rate of evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 316–321, Apr. 2016.



TAO LI is currently pursuing the Ph.D. degree in software engineering with the College of Computer Science and Technology, Harbin Engineering University, China. His research interests include intelligent information processing, data mining, and machine learning, especially in evolutionary computation and feature selection.

HONGBIN DONG received the B.S. and M.S. degrees in computer science and technology from the Harbin Ship Engineering College, China, in 1986 and 1995, respectively, and the Ph.D. degree in computer science and information technology from Beijing Jiaotong University, Beijing, China. Since 2005, he has been a Professor with the Computer Science and Technology Department, Harbin Engineering University, China. His research interests include natural computation, multi-agent systems, machine learning, and data mining. He is a member of the Committee on Artificial Intelligence and Pattern Recognition of the Chinese Computer Society and the Committee on Granular Computing and Knowledge Discovery of the Chinese Academy of Artificial Intelligence, and a Senior Member of the Computer Society.

JING SUN is currently pursuing the Ph.D. degree in software engineering with the College of Computer Science and Technology, Harbin Engineering University, China. Her research interests include machine learning, optimization algorithms, and feature selection.