

Article

Classification of Gene Expression Data Using Multiobjective Differential Evolution

Shijing Ma, Xiangtao Li * and Yunhe Wang

School of Computer Science and Information Technology, Northeast Normal University, Changchun 130117, China; masj195@nenu.edu.cn (S.M.); wangyh082@nenu.edu.cn (Y.W.)

* Correspondence: lixt314@nenu.edu.cn; Tel.: +86-431-8453-6338

Academic Editor: Silvio Simani

Received: 6 September 2016; Accepted: 3 November 2016; Published: 15 December 2016

Abstract: Gene expression data are usually redundant, and only a subset of them presents distinct profiles for different classes of samples. Thus, selecting high discriminative genes from gene expression data has become increasingly interesting in bioinformatics. In this paper, a multiobjective binary differential evolution method (MOBDE) is proposed to select a small subset of informative genes relevant to the classification. In the proposed method, firstly, the Fisher-Markov selector is used to choose top features of gene expression data. Secondly, to make differential evolution suitable for the binary problem, a novel binary mutation method is proposed to balance the exploration and exploitation ability. Thirdly, the multiobjective binary differential evolution is proposed by integrating the summation of normalized objectives and diversity selection into the binary differential evolution algorithm. Finally, the MOBDE algorithm is used for feature selection, and support vector machine (SVM) is used as the classifier with the leave-one-out cross-validation method (LOOCV). In order to show the effectiveness and efficiency of the algorithm, the proposed method is tested on ten gene expression datasets. Experimental results demonstrate that the proposed method is very effective.

Keywords: multiobjective method; differential evolution algorithm; binary differential evolution; binary optimization

1. Introduction

Gene expression data are characterized by thousands of and even tens of thousands of measured genes on only a few tissue samples, which gives rise to difficulties for many classifiers [1,2]. Therefore, feature selection in the computational intelligence field [3,4] plays an important role in gene array-based cancer classification, because gene selection can help to remove the irrelevant and redundant features and choose a small subset of features to carry out the classification task in an optimal way. In general, feature selection can be categorized into wrappers and filters according to whether or not it is done independently of the learning algorithm [3,4]. By using the filter and wrapper techniques, many feature selection methods [5–8] have been proposed to optimize the efficiency of the search and selection process. For example, a novel correlation-based memetic framework (MA-C), which is a combination of genetic algorithm (GA) and local search (LS) using correlation-based filter ranking, was proposed [9]. The local filter method used here fine-tunes the population of GA solutions by adding or deleting features based on the symmetrical uncertainty (SU) measure. In order to take into account the experimental conditions and the time points simultaneously, Gutiérrez-Avilés D. et al. [10] presented the TriGen algorithm, a genetic algorithm that finds triclusters of gene expression. From the results, TriGen has proven to be capable of extracting groups of genes. In [11], Xue B. et al. propose three new initialization strategies and three new personal best and global best updating mechanisms in particle swarm optimization to develop novel feature selection approaches with the goals of maximizing the classification performance, minimizing the number of features and reducing the computational time.

The superior performance of this algorithm is due mainly to both the proposed initialization strategy, which aims to take advantage of both the forward selection and backward selection to decrease the number of features and the computational time, and the new updating mechanism, which can overcome the limitations of traditional updating mechanisms by taking the number of features into account, which reduces the number of features and the computational time. Based on the above analysis, the main purpose of the feature selection method is to maximize the model performance and to minimize the number of genes selected at the same time. That is to say, feature selection has two different objectives, including maximizing the classification performance and minimizing the number of genes selected. In some cases, these two objectives conflict. Based on the above situation, the feature selection may be more suitable to be designed for a multiobjective problem rather than a single-objective problem.

Recently, many multiobjective optimization approaches based on different evolutionary algorithms have been reported to solve feature selection [12,13]. For example, a hybrid multiobjective optimization method based on particle swarm optimization was proposed [5] to find a small set of non-redundant disease-related genes. Two objectives, including sensitivity and specificity, are simultaneously evaluated by the artificial neural network (ANN) classifier. Based on the real-life datasets of various types of cancers, the performance of multiobjective particle swarm optimization can perform better compared with sequential feature selection (SFS), the t-test and rank-sum. Xue B. et al. [14] proposed other multiobjective particle swarm optimizations, which are multiobjective binary particle swarm optimization (PSO) using the idea of non-dominated sorting (NSBPSO) and multiobjective binary PSO using the ideas of crowding, mutation and dominance (CMDBPSO). The proposed algorithms are examined and compared with a single-objective method on eight benchmark datasets. Experimental results show that the proposed multiobjective algorithms can evolve a set of solutions that use a smaller number of features and achieve better classification performance than using all features. Different from particle swarm optimization, the multiobjective genetic algorithm [15] was proposed to select the optimum subset and then the classification of gene expression data. Support vector machine with the radial basis function (RBF) kernel is used to measure the accuracy of the classification. This approach was tried on two benchmark gene expression datasets. It obtained encouraging results on those datasets as compared with an approach that used a single-objective strategy in a genetic algorithm. In [16], a different optimization algorithm based on an artificial immune system was used to solve feature selection in classification problems aiming at minimizing both the classification error and the cardinality of the subset of features. The algorithm is able to perform a multimodal search maintaining population diversity and controlling automatically the population size according to the problem. The experimental results show that parsimonious subsets of features and the classifiers produced a significant improvement in the accuracy. Another multiobjective artificial immune algorithm [17] was used to optimize the kernel and penalize the parameters of support vector machine (SVM). In the training stage of SVM, multiple solutions are found by using a multiobjective artificial immune algorithm, and then, these parameters are evaluated in the test stage. The proposed algorithm is applied to fault diagnosis of induction motors and anomaly detection problems, and successful results are obtained. Rubio-Escudero C. et al. [18] used EMO-CC (evolutionary multiobjective conceptual clustering) to obtain such gene product information, which retrieves meaningful substructures from network databases. The experiment results show that expectation maximization algorithm performs better than other algorithms for the analysis of microarray data. Romero-Zalaz R. et al. [19] proposed a multiobjective methodology to combine state-of-the-art algorithms into an aggregation scheme in order to obtain the optimal methods' aggregations. The results obtained by the multiobjective algorithm show a major improvement in sensitivity when our methodology is compared to the performance of individual methods for gene finding and gene expression problems. Based on the above discussion, many different multiobjective evolutionary algorithms have been used to handle the feature selection problem. However, these algorithms still have some drawbacks, such as low optimization efficiency, easily falling into local

optima and premature convergence. Moreover, this field of study is still in its early days; a large number of future research works is necessary in order to develop a multiobjective algorithm for feature selection.

Recently, the differential evolution algorithm was proposed as a powerful evolutionary algorithm [20–23], which has good global search and local search capabilities, and it can quickly search out all solutions from the solution space. Several variations of differential evolution (DE) have also been proposed to enhance the performance of the standard DE [24–31]. The algorithm was considered as an intelligent optimization method for heuristic random search in a continuous space. The algorithm consists of three different operators, including mutation, cross-over and selection operators. By these operators, the differential evolution algorithm can generate new individuals by combining the target vector and the trial vector. However, it should be noted that most of these algorithms work in continuous space rather than in discrete space. Therefore, in this paper, we propose a novel multiobjective binary differential evolution algorithm (MOBDE) to solve the binary problem in terms of the feature selection problem.

This paper uses a novel multiobjective differential evolution algorithm for the feature selection problem, and support vector machine (SVM) is used as the classifier with leave-one-out cross-validation (LOOCV). The Fisher-Markov selector is used to choose a fixed number of the top gene expression data features, and then, a multiobjective binary differential evolution algorithm based on the summation of normalized objectives and diversity selection is adopted to select the most important gene subsets. Finally, a classifier SVM is trained based on the gene subset and then used to predict the test sample. Numerical results of ten gene expression data are reported and compared with other algorithms. As is shown, the solutions obtained by the proposed approach are all superior to those best solutions obtained by other algorithms in the literature.

2. Computational Methods

In this part, we shall introduce a hybrid multiobjective binary differential evolution and support vector machine method (MOBDE) for feature selection. The flowchart of the proposed method is shown in Figure 1. As can be seen in this figure, there are mainly three important components, i.e., the Fisher-Markov selector component, the multiobjective binary differential evolution component and the support vector machine component.

In the first component, the Fisher-Markov selector method is used to select 180 top genes with the highest scores. These selected genes will then be utilized for the second component, multiobjective binary differential evolution component. In this component, at first, a randomly-generated initial solution will be represented by a binary (0/1) string. Then, a novel binary mutation method is proposed to balance the exploration and exploitation ability during the search process. After that, the multiobjective binary differential evolution is proposed by integrating the summation of normalized objectives and diversity selection into the algorithm.

By using MOBDE, the parameters of the support vector machine (SVM) in the third component and the features subset are dynamically optimized. Specifically, for feature selection, each gene is represented as a bit of binary encoded individual, where one denotes a gene selected and zero denotes a non-selected gene. For SVM, two important parameters of RBF kernels, i.e., c and γ , are taken into account. In this sense, the length of each individual is equal to $D + 2$, where D is the number of genes in the initial microarray dataset. Table 1 shows the solution representation of the algorithm.

Table 1. The solution representation.

1	2	3	4	...	$D + 2$
P_c	P_r	F_1	F_2	...	F_D

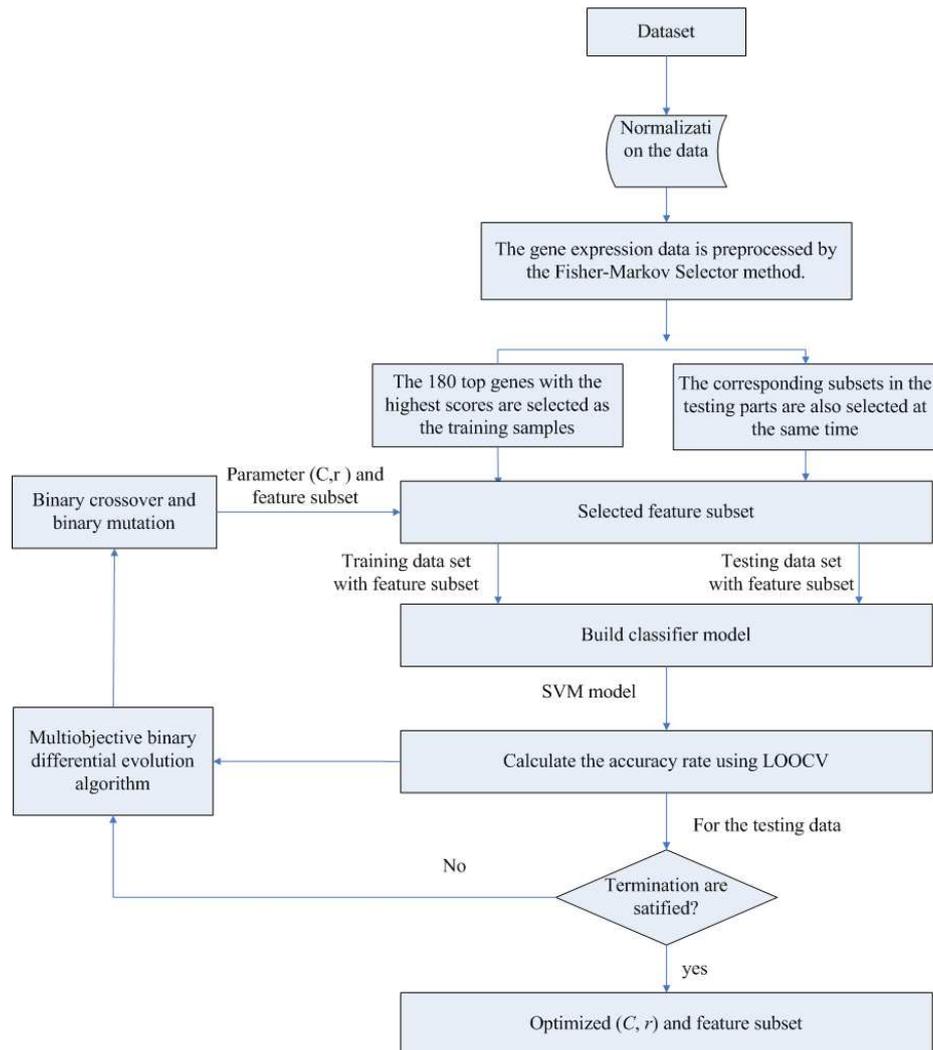


Figure 1. The framework of the multiobjective binary differential evolution method (MOBDE) with support vector machine (SVM).

From the above solution representation, P_c is the parameter C of the SVM, and P_γ denotes the parameter γ of SVM. In this paper, we use the evolutionary algorithm to optimize the parameters of SVM and the feature subset in each individual; the multiobjective function can be defined as below:

$$f_1 = SVM_{accuracy}; f_2 = \frac{D-R}{D}; \quad (1)$$

$$f = [f_1; f_2];$$

where $SVM_{accuracy}$ denotes the classification accuracy of SVM and R denotes the number of selected genes. Finally, The fitness values of each individual will be assessed by the accuracy of LOOCV.

2.1. Fisher-Markov Selector

In the field of machine learning, selecting suitable features is very important for classification. The Fisher-Markov selector is proposed by Cheng et al. [32] to identify the more useful features in describing essential differences among the possible groups. The authors present a way to represent essential discriminating characteristics together with sparsity as an optimization problem. In this paper, we use this method, and the detailed description can be seen in [32].

2.2. Multiobjective Differential Evolution Component

In this part, we shall introduce the proposed multiobjective binary differential evolution algorithm in detail.

As we know, differential evolution (DE) is a fairly novel population-based search heuristic, which is simple to implement and requires little parameter tuning compared with other search heuristics in continuous space.

The process of DE can be summarized into three major steps: mutation, cross-over and selection. In the mutation operator, the process of generating the mutation vectors $V_{i,G} = \{V_{1,i,G}, V_{2,i,G}, \dots, V_{D,i,G}\}$ is performed by target vector $X_{i,G} = \{X_{1,i,G}, X_{2,i,G}, \dots, X_{D,i,G}\}$ in the current population, where D denotes the dimension of the individual, i denotes the i -th of individual and G denotes the current iteration of the algorithm. In the DE algorithm, “DE/rand/1/bin” is the most common mutation strategies, as below:

$$V_{i,G} = X_{r1,G} + F \cdot (X_{r2,G} - X_{r3,G}); \quad (2)$$

where $r1, r2, r3 \in [1, \dots, NP]$, $r1 \neq r2 \neq r3 \neq i$ and F is the mutation factor of the differential evolution. NP is the size of the population.

In the cross-over operation, a recombination of the candidate solution $V_{i,G}$ and the parent $X_{i,G}$ produces an offspring solution $U_{i,G} = [U_{1,i,G}, U_{2,i,G}, \dots, U_{D,i,G}]$. Usually, the binomial cross-over is accepted, which is defined as follows:

$$U_{j,i,G} = \begin{cases} V_{j,i,G} & (\text{rand}_j[0,1] \leq CR) \text{ or } (j = j_{\text{rand}}) \\ X_{j,i,G} & \text{otherwise} \end{cases} \quad (3)$$

where $j \in [1, \dots, D]$; $\text{rand}_j \in [0, 1]$ is a random number between zero and one; $j_{\text{rand}} \in [1, \dots, D]$ is a randomly chosen index. CR is the cross-over rate.

A greedy selection is used to choose the next population (*i.e.*, $G = G + 1$) between the parent population and the offspring population. The selection operation is described as follows:

$$X_{i,G+1} = \begin{cases} U_{i,G} & f(U_{i,G}) \leq f(X_{i,G}) \\ X_{i,G} & \text{otherwise} \end{cases} \quad (4)$$

As we know, the original differential evolution algorithm is a continuous optimization algorithm, but the feature selection problem is a classic binary optimization problem. Therefore, the original continuous encoding scheme of DE cannot be used directly for gene selection problems. In order to make DE suitable to solve the gene selection problem, a binary differential evolution (BDE) algorithm is proposed first. In the proposed method, the initial population is represented as a vector in which each bit is a binary value of zero or one, where one denotes this gene is selected and zero denotes a non-selected gene. The objective function values are calculated, and then, new binary populations are transported into the mutation operators. The binary cross-over operations are used to generate the trial solution. Finally, greedy selection method is used to choose the better results for the next generation.

During the reconstruction of the mutation operation, the key idea is to use some appropriate operators in place of the arithmetic operators. In [33], He and Han used the XOR, AND and OR operations instead of the subtraction, multiplication and addition operations in the formula, which can be described as follows:

$$V_{j,i,G+1} = X_{j,r1,G} \odot F \otimes (X_{j,r2,G} \oplus X_{j,r3,G}) \quad (5)$$

where \oplus denotes the XOR operations, \otimes represents the AND operation and \odot denotes the OR operation. Note that in Formula (5), the use of OR operation will make the probability of a result be true. The probability of the binary “1” will be three times higher than the probability to be false (binary “0”). In other words, the binary “1” would be easily accumulated with the binary string $V_{j,i,G+1}$ of the trial solution after the OR operation. This would decrease the diversity of the algorithm. Accordingly,

in [34], another novel mutation operation is proposed by considering the distance of the $X_{r_1,G}$ and $X_{r_2,G}$ for each dimension:

$$V_{j,i,G+1} = \begin{cases} 1, & \text{if } (rand < F) \otimes (X_{j,r_1,G} \oplus X_{j,r_2,G}); \\ 0, & \text{otherwise} \end{cases}; \quad (6)$$

Compared with Formula (5), the new mutation strategy can enhance the diversity of the algorithm because it does not use the OR operation. However, in this formula, the value of the previous generation will be discarded. Therefore, it cannot inherit the advantage of the original individual from the previous population.

Therefore, in this paper, we propose a new mutation strategy, which can both increase the diversity of the algorithm and take advantage of the original population, as described in the following:

$$\begin{cases} (X_{j,r_1,G} \otimes X_{j,r_2,G}), & \text{if } (rand < F) \otimes (X_{j,r_1,G} = X_{j,r_2,G}); \\ (X_{j,r_1,G} \otimes X_{j,r_3,G}), & \text{if } (rand < F) \otimes (X_{j,r_1,G} \neq X_{j,r_2,G}); \\ X_{j,r_1,G}, & \text{otherwise} \end{cases}; \quad (7)$$

As can be seen in this formula, first, it does not use the OR operation, so this operation will not harm the diversity of the algorithm. Second, the values of the previous generation, e.g., $X_{j,r_1,G}$, $X_{j,r_2,G}$, $X_{j,r_3,G}$, will be kept with a probability. In this way, the algorithm can inherit the advantage of the original individual from the previous population. Following the binary mutation strategy, a binary cross-over operator is used to build a trial solution $U_{j,i,G+1}$ by combining the mutation vector and the target vector. The concept of the binary cross-over mechanism of BDE is similar to that of the original DE, though there is a difference in terms of the component data type. In BDE, the binary data is selected from the mutation vector if a random number is smaller than the cross-over rate; otherwise, the original solution is chosen to generate the trail solution. After the binary mutation and cross-over operator, the better solutions between the trial solution and the target solution will be retained to the next generation.

Based on the binary differential evolution algorithm, we will propose our multiobjective binary differential evolution algorithm (MOBDE). Specifically, in our method, two fitness objectives are taken into account for optimization. One is the accuracy of the classification, and the other is the number of selected genes. In order to tackle the feature selection problem, a non-dominated sorting process is often used to find the Pareto front. However, the non-dominated sorting process is always complex and time consuming. In order to solve this problem, Qu and Suganthan [35] used the summation of the normalized objective and diversity selection, and in this paper, we use a very similar method based on the summation of the normalized objective and diversity selection for the feature selection problem. For the summation of the normalized objective, first, we need to find the maximum and minimum value for every objective and calculate the different range of every objective; then, we need to sum all normalized objective values to obtain a single value. In this way, the multiobjective problem can be regarded as a single-objective optimization problem. However, this kind of transformation may cause the problem of lacking the diversity of the population. Therefore, the diversity selection method is used to maintain the diversity of the algorithm.

The preferential set and backup set are generated from the current population, and three rules are used to select the sets in the next process:

1. The preferential set can be selected in the next process firstly.
2. The backup set will be chosen based on the summation of the normalized objective and diversity selection if the preferential set is not sufficient for the solution.
3. While the individuals in the store exceed the maximum size, the required number of solutions will be randomly chosen from the preferential set.

Based on the above discussion, we can show the framework of our multiobjective binary differential evolution algorithm as follows in Algorithm 1.

Algorithm 1 Algorithm description of the MOBDE algorithm

Set the generation counter $G = 0$; and randomly initialize a population of NP individuals X_i .
 Initialize the parameters F, CR .
 Evaluate the fitness for each individual in P .
 Return the non-dominated solutions Ar_0 from the individual P .
while stopping criteria is not satisfied **do**
 for $i = 1$ to NP **do**
 select randomly $r_1 \neq r_2 \neq r_3 \neq i$
 for $j = 1$ to D **do**
 if $rand < CR \parallel j_{rand} = j$ **then**
 if $rand < F \ \& \ X_{r_1,j,G} = X_{r_2,j,G}$ **then**
 $U_{i,j} = \otimes(X_{r_1,j,G}, X_{r_2,j,G})$
 else if $rand < F \ \& \ X_{r_1,j,G} \neq X_{r_2,j,G}$ **then**
 $U_{i,j} = \otimes(X_{r_1,j,G}, X_{r_3,j,G})$
 else
 $U_{i,j} = X_{r_1,j,G}$
 end if
 else
 $U_{i,j} = X_{i,j,G}$
 end if
 end for
 end for
 Calculate the objective function for the new population
 Select the better individual based on the summation of normalized objectives and diversified selection
 Update the archive Ar_{G+1} based on the new individual
end while

2.3. Support Vector Machines

In our system, the support vector machine with the leave-one-out cross-validation serves as the evaluator of the multiobjective binary differential evolution algorithm. Let $x_i \in R^d, \forall i = 1, 2, \dots, n$, and $y_i \in -1, +1, \forall i = 1, 2, \dots, n$, be a set of training samples and the corresponding labels, respectively. Vapnik and Cortes [36] defined the SVM method as follows:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\omega\|^2 + C \cdot \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & y_i(\omega \cdot q_i + b) \geq 1 - \xi_i, i = 1, 2, \dots, n. \end{aligned} \quad (8)$$

where ω is a normal vector to the hyperplane and b is a constant, such that $\frac{b}{\|\omega\|}$ represents the Euclidean distance between the hyperplane and the original feature space. The ξ_i is the slack variables to control the training errors, and C is a penalty parameter of SVM. In this paper, the radial basis function (RBF) is used in SVM to obtain the optimal solution for classification. Considering two samples $q_i = [q_{i,1}, q_{i,2}, \dots, q_{i,d}]^T, q_j = [q_{j,1}, q_{j,2}, \dots, q_{j,d}]^T, i \neq j$ and i, j are the different samples, the RBF function is calculated by using $K(q_i, q_j) = \exp(-\gamma \|q_i - q_j\|^2)$, where $\gamma > 0$ is the width of the Gaussian. $K(q_i, q_j)$ is the kernel function.

For the RBF kernel function, C and γ are the very important parameters, and the performance of SVM depends on the choice of kernel function in terms of the parameters C and γ . If the value of C is large, the accuracy value of the training will perform better, but the test rate will perform worse.

Meanwhile, if the value of C is small, the accuracy rate will be unsatisfactory, though the test accuracy rate may be high. Sometimes, the parameter γ has a more effective effect on the test phase than the parameter C . In order to optimize the feature selection and parameter simultaneously, in the modified MOBDE, each individual is encoded to a string of binary bits associated with the number of genes, and the parameters C and γ of the SVM will be dynamically optimized by a real code differential evolution in Equations (3) and (4). Specifically, the constrained ranges of the value of C and γ are $[-5, 15]$ and $[-15, 5]$ respectively. In our method, the classification accuracy of the prediction models and the number of selected genes derived from all datasets will be measured by the LOOCV procedure discussed in Section 2.

2.4. Computational Complexity of the Multiobjective Binary Differential Evolution with Support Vector Machine

In this part, we will analyze the time complexity of the multiobjective binary differential evolution with support vector machine model. In the beginning of the algorithm, the Fisher-Markov selector is used to choose the suitable feature. Cheng Q. et al. [32] shows that the complexity of the Fisher-Markov selector is $O(n^2)$, where n is the size of the dataset. Then, for each iteration of MOBDE, the SVM needs to be called. Tsang et al. [37] shows that the data subroutines of standard SVM are $O(n^3)$, where the summation of the normalized objective values method is $O(M \cdot NP)$, where NP denotes the population size and M is the number of objectives. Therefore, for each iteration, the runtime complexity is $O(M \cdot NP \cdot n^3 + n^2)$. Suppose the total number of iterations is I ; the time complexity of the algorithm is then $O(I \cdot M \cdot NP \cdot n^3 + n^2)$. In this paper, M is two. Therefore, the time complexity is $O(2 \cdot I \cdot NP \cdot n^3 + n^2)$, i.e., $O(I \cdot NP \cdot n^3)$.

2.5. Why Use Each Finding in the Algorithm and the Strong Impact of the Finding

Firstly, the first problem is why we use the Fisher-Markov selector. The reason is that the Fisher-Markov selector selects the more suitable features to describe essential differences among the possible groups. This method uses the Markov random field optimization techniques to solve the formulated objective functions for simultaneous feature selection. The method is fast; in particular, it can be linear in the number of features and quadratic in the number of observations. The algorithm has been used to solve the high-dimensional microarray gene expression datasets better. Therefore, in this paper, we firstly use the Fisher-Markov selector to select the feature.

Secondly, the second problem is why we use the multiobjective binary differential evolution to solve this problem. As we know, the original differential evolution algorithm is a continuous optimization algorithm, but the feature selection problem is a classic binary optimization problem. Therefore, the original continuous encoding scheme of DE cannot be used directly for gene selection problems. In order to make DE applicable to the gene selection problem, a binary differential evolution (BDE) algorithm is proposed first. As shown in Section 2.2, previous work may either decrease the diversity of the algorithm or the new individual cannot inherit the advantage of the original individual from the previous population. Therefore, in this paper, we propose a new mutation strategy, which can both increase the diversity of the algorithm and take advantage of the original population.

3. Experimental Setup

To demonstrate the effectiveness of the MOBDE algorithm, the experiments are performed on 10 benchmark datasets. All of these characteristics of gene expression datasets are listed in Table 2. The gene expression datasets consist of 10 well-known datasets. These datasets have been widely used by researchers as a primary source of feature selection datasets. The library for support vector machines (LIBSVM) is proposed by Chang and Lin [38]. The datasets are classified by LIBSVM based on LOOCV. We compared our method with some binary differential evolution algorithms: binary DE [33], binary differential evolution (BDE) [34], binary differential evolution with artificial immune system (BDEAIS) [39], binary particle swarm optimization (BPSO) [40], binary genetic algorithm (BGA) [41]

and binary estimation distribution algorithm (BEDA) [42]. In this paper, we replace these methods with our binary method and then compare our method to show the effective of the algorithm. That is to say, all of these methods use the same multiobjective framework. Meanwhile, we compare our algorithm with the nondominated sorting genetic algorithm II (NSGAI) in order to show the difference of the summation of the normalized objective and diversified selection with the non-domination sorting process. As the same time, we also compare our method with some optimization methods, including SVM + grid search, improved binary particle swarm optimization (IBPSO), hybrid binary particle swarm optimization and tabu search (HPSOTS), PSO/GA [5–8] and some different versions of support vector machines. The parameters are as follows.

Table 2. Format of gene expression classification data.

Dataset Number	Dataset Name	Number of Samples	Number of Classes	Number of Genes	Description
1	11_Tumors	60	9	5726	11 various human tumor types
2	9_Tumors	174	11	12,533	9 various human tumor types
3	Brain_Tumors1	90	5	5920	5 human brain tumor types
4	Brain_Tumors2	50	4	10,367	4 malignant glioma types
5	Leukemia1	72	3	5327	Acute myelogenous leukemia (AML), acute lymphoblastic leukemia (ALL) B-cell and ALL T-cell
6	Leukemia2	72	3	11,225	AML, ALL and mixed-lineage leukemia (MLL)
7	Lung_cancer	203	5	12,600	4 lung cancer types and normal tissues
8	SRBCT	83	4	2308	Small, round blue cell tumors (SRBCT) of childhood
9	Prostate_Tumor	102	2	10,509	Prostate tumor and normal tissues
10	DLBCL	77	2	5469	Diffuse large B-cell lymphomas (DLBCL) and follicular lymphomas

For all algorithms, the population size is 50; the maximum number of iterations is 100. For the different version of binary DE algorithms [33,34,39], the value of the F is 0.5, and the value of CR is 0.7. For the genetic algorithm, the cross-over rate is 0.7, and the mutation rate is 0.5. For the binary PSO, the values of $c1$ and $c2$ are both 2. For the binary estimation of the distribution algorithm, the probability of selection is 0.3. The parameters were selected (after some preliminary experiments) so as to result in roughly the best results generated by the algorithms used for comparison. However, with different strategies used by each algorithm, it is very difficult to ensure the best suitable parameters as reflected in the experiments.

3.1. Discussions and Analysis

As is discussed in the previous section, LOOCV is used in our algorithm. As the training set and test set are changing under the LOOCV strategy, the genes selected and the test accuracy are different each time. Tables 3 and 4 show the test accuracy and the number of genes selected in 10 runs on the ten datasets. As we can observe in Table 3, the results of the proposed methods are almost consistent on all datasets. Moreover, MOBDE can obtain 100% LOOCV accuracy with less than 10 selected genes for the Leukemia1, Leukemia2, small, round blue cell tumors (SRBCT) and diffuse large B-cell lymphomas (DLBCL) datasets. For another dataset, Brain_Tumor2, from the Table 3, we can find that MOBDE obtains 100% accuracy with smaller selected genes. For the average accuracy, the MOBDE algorithm can obtain 99% accuracy. Meanwhile, the average number of selected genes is 7.5. For the gene expression data 11_Tumors, the MOBDE algorithm can provide 97.19% accuracy with less than 40 selected genes. It is noted that the MOBDE can obtain more than 98% accuracy four times. For the dataset Lung Cancer in Table 4, MOBDE can provide 100% LOOCV accuracy two times. The average accuracy rate of MOBDE can provide 99.12 with less than 30 selected genes. Meanwhile, the MOBDE can obtain the average selected genes of 15.5. For the dataset Prostate_Tumor, the MOBDE algorithm

can provide 98.63% average LOOCV accuracy with 10.9 selected genes. For the dataset Brain_Tumor1, the MOBDE algorithm can also provide more than 97% classification accuracy. Among them, for Lung Cancer and Prostate Tumor, the algorithm can also find 100% classification accuracies for two and one times, respectively. From the point of view of the accuracy average in each independent run, the LOOCV accuracy and the number of selected genes obtained by MOBDE are shown in Figures 2 and 3.

Table 3. Experimental results for each run using multiobjective binary differential evolution (MOBDE) on 11_Tumors, 9_Tumors, Brain_Tumors1, Brain_Tumors2 and Leukemia1.

Run	11_Tumors		9_Tumors		Brain_Tumors1		Brain_Tumors2		Leukemia1	
	Acc	Selected Genes	Acc	Selected Genes	Acc	Selected Genes	Acc	Selected Genes	Acc	Selected Genes
1	97.13	34	93.33	20	98.89	15	100	6	100	6
2	94.83	23	96.67	25	98.89	14	100	9	100	5
3	94.83	31	91.67	17	97.78	11	100	6	100	4
4	97.71	31	91.67	20	98.89	7	100	5	100	5
5	98.28	22	86.67	17	96.67	11	100	7	100	6
6	98.28	40	93.33	17	96.67	12	96	10	100	8
7	96.55	24	91.67	17	98.89	11	100	9	100	6
8	98.28	23	93.33	14	95.56	10	96	9	100	8
9	98.85	26	93.33	38	96.67	12	98	8	100	6
10	97.13	21	95	22	97.78	12	100	6	100	5
Average	97.19	27.5	92.67	20.7	97.67	11.5	99	7.5	100	5.9
±S.D.	±1.42	±6.24	±2.62	±6.83	±1.22	±2.17	±1.7	±1.72	0	±1.29

Results for 10 runs are listed in this table. The best subset is shown in shaded cells. In this work, the accuracy is more important than the number of selected genes. Therefore, a solution with the best accuracy can be chosen from the final Pareto front. “Acc” denotes the accuracy of the classifications, and “Selected genes” represents the number of selected genes. The bolding denotes the best solutions.

Table 4. Experimental results for each run using MOBDE on Leukemia2, Lung_cancer, SRBCT, Prostate_Tumor and DLBCL.

Run	Leukemia2		Lung_Cancer		SRBCT		Prostate_Tumor		DLBCL	
	Acc	Selected Genes	Acc	Selected Genes	Acc	Selected Genes	Acc	Selected Genes	Acc	Selected Genes
1	100	5	98.52	12	100	8	98.04	6	100	7
2	100	8	100	27	100	6	99.02	8	100	8
3	100	4	98.03	15	100	6	99.02	22	100	4
4	100	8	98.52	15	100	4	100	10	100	4
5	100	7	99.51	16	100	5	99.02	8	100	5
6	100	7	100	14	100	5	97.06	14	100	3
7	100	5	99.02	10	100	6	98.04	11	100	6
8	100	5	99.02	14	100	4	99.02	13	100	8
9	100	5	99.02	15	100	5	99.02	10	100	4
10	100	6	99.51	17	100	5	98.04	7	100	7
Average	100	6	99.12	15.5	100	5.4	98.63	10.9	100	5.6
±S.D.	±0	±1.41	±0.65	±4.5	±0	±1.17	±0.83	±4.65	±0	±1.83

Results for 10 runs are listed in this table. The best subset is shown in shaded cells. In this work, the accuracy is more important than the number of selected genes. Therefore, a solution with the best accuracy can be chosen from the final Pareto front. “Acc” denotes the accuracy of the classifications, and “Selected genes” represents the number of selected genes. The bolding denotes the best solutions.

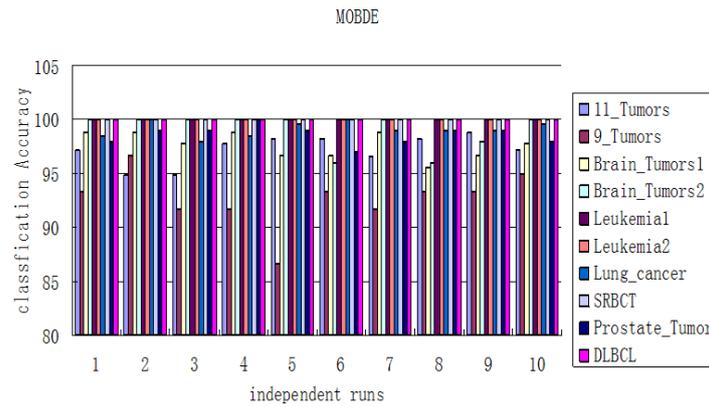


Figure 2. The accuracy obtained by MOBDE in each independent run.

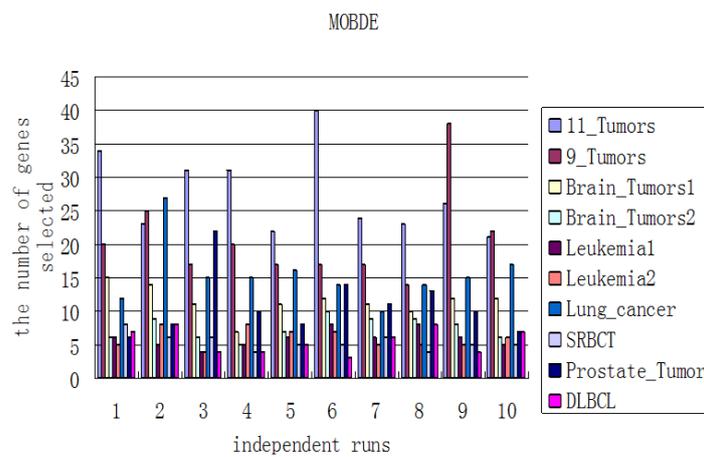


Figure 3. The number of selected gene obtained by MOBDE in each independent run.

From the results in Table 5, we can find that the average percentage of genes selected is 0.0016. For the Leukemia1, Leukemia2, SRBCT and DLBCL datasets, our algorithm provides 100% LOOCV accuracy, even though the percentage of genes selected for these datasets is reduce to 0.0011, 0.0005, 0.0023 and 0.0010 of the total available, respectively. Therefore, it can demonstrate that not all features are necessary for achieving better classification accuracy. Figure 4 shows the percentage of genes selected.

Table 5. The genes, selected genes and percentage of gene selected percentage.

Dataset Name	Genes	Genes Selected	Percentage of Genes Selected
11_Tumors	5726	27.5	0.0048
9_Tumors	12,533	20.7	0.0017
Brain_Tumors1	5920	11.5	0.0019
Brain_Tumors2	10,367	7.5	0.0007
Leukemia1	5327	5.9	0.0011
Leukemia2	11,225	6	0.0005
Lung_cancer	12,600	15.5	0.0012
SRBCT	2308	5.4	0.0023
Prostate_Tumor	10,509	10.9	0.001
DLBCL	5469	5.6	0.001
Average	-	-	0.0016

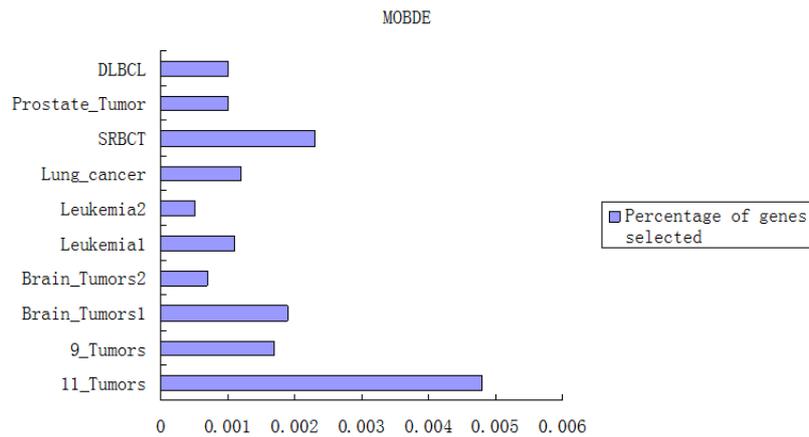


Figure 4. The percentage of genes selected.

In order to analyze each part of our algorithm, four different experiments are designed. The first one is to show the effectiveness of the Fisher-Markov selector. The second and third experiments are to show the effectiveness of the novel binary differential evolution algorithm compared with other meta-heuristic algorithms. The fourth experiment is to show the effectiveness of the MOBDE compared with the grid-search SVM without feature selection method. For the first experiment, we compare MOBDE with the Fisher-Markov selector and MOBDE without the Fisher-Markov selector. The results of the MOBDE with the Fisher-Markov selector and MOBDE without the Fisher-Markov selector are summarized in Table 6. The results represented in Table 6 show that both MOBDE with the Fisher-Markov selector and MOBDE without the Fisher-Markov selector provide 100% classification accuracy for Leukemia2 and DLBCL. MOBDE with the Fisher-Markov selector provides less genes selected for all datasets. For the 9_Tumors, Brain_Tumors1 and Brain_Tumors2 datasets, MOBDE can not only provide better classification accuracy, but also a lower number of genes selected. However, for 11_Tumors, Lung_cancer and Prostate_Tumor, MOBDE with the Fisher-Markov selector cannot obtain better classification accuracy than the latter, which demonstrates that the Fisher-Markov selector is not suitable for solving different problems. Overall, the Fisher-Markov selector is very effective in feature selection for the bioinformatics dataset.

For the second experiment, we compare our algorithm with three different versions of binary differential evolution: binary DE, BDE and BDEAIS. We replace our binary differential evolution in MOBDE by using the binary DE, BDE and BDEAIS [33,34,39]. That is to say, all of these methods use the same multiobjective framework to conduct a fair comparison. Therefore, the purpose of this experiment is to show the effectiveness of the new mutation strategy. Table 7 shows the results obtained by different binary differential evolution algorithms in terms of the mean and standard deviation (S. D.) of the classification accuracy and the number of genes selected. As can be seen in Table 7, for Leukemia2, SRBCT and DLBCL, all algorithms can obtain 100% LOOCV accuracy. However, MOBDE can obtain fewer genes selected. For the 11_Tumors dataset, the BDE can provide the best solution of 97.24% with the number of the genes selected being 48.4. The MOBDE can provide a similar accuracy of 97.19% and a lower number of genes selected of 27.5. For Leukemia1, three out of four algorithms can find the best accuracy with 100% LOOCV. For the rest of the datasets, the best classification and a lesser number of genes selected are provided by the MOBDE. Therefore, we can draw the conclusion that the MOBDE can obtain a better performance compared with other binary differential evolution algorithms.

Table 6. Comparative experimental results of the binary differential evolution algorithm with and without the Fisher-Markov selector.

Dataset Name	Evaluation	This Work (MOBDE)	MOBDE without the Fisher-Markov Selector
11_Tumors	Acc (%)	97.19	98.28
	Genes	27.5	236.67
9_Tumors	Acc (%)	92.67	91.67
	Genes	20.7	151.33
Brain_Tumors1	Acc (%)	97.67	97.04
	Genes	11.5	110
Brain_Tumors2	Acc (%)	99	98
	Genes	7.5	71.67
Leukemia1	Acc (%)	100	100
	Genes	5.9	75
Leukemia2	Acc (%)	100	100
	Genes	6	62.67
Lung_cancer	Acc (%)	99.12	99.26
	Genes	15.5	123.5
SRBCT	Acc (%)	100	100
	Genes	5.4	109.67
Prostate_Tumor	Acc (%)	98.63	99.35
	Genes	10.9	126.33
DLBCL	Acc (%)	100	100
	Genes	5.6	25.33

The bolding denotes the best solutions.

Table 7. Comparative experimental results of different binary differential evolution algorithms.

Dataset Name	Method	This Work (MOBDE)		Binary DE [34]		BDE [39]		BDEAIS [33]	
	Evaluation	Average	S.D.	Average	S.D.	Average	S.D.	Average	S.D.
11_Tumors	Acc (%)	97.19	1.42	93.97	0.41	97.24	0.63	95.29	0.48
	Genes	27.5	6.24	50	8.49	48.4	4.77	113.4	7.67
9_Tumors	Acc (%)	92.67	2.62	82	1.39	91.67	0	85.33	2.17
	Genes	20.7	6.83	17	6.85	44.8	9.91	107	11.34
Brain_Tumors1	Acc (%)	97.67	1.22	95.11	0.61	96	0.61	95.56	0
	Genes	11.5	2.17	17	11.25	56.8	6.53	97.4	8.17
Brain_Tumors2	Acc (%)	99	1.7	96.4	1.67	96	2.45	88.8	1.09
	Genes	7.5	1.72	11.6	7.89	38.2	9.93	86	6.67
Leukemia1	Acc (%)	100	0	100	0	100	0	98.33	0.62
	Genes	5.9	1.29	10.2	3.11	45.8	8.41	78.8	5.07
Leukemia2	Acc (%)	100	0	100	0	100	0	100	0
	Genes	6	1.41	11.4	6	42.2	1.92	77.2	2.49
Lung_cancer	Acc (%)	99.12	0.65	98.03	0.35	98.23	0.27	98.03	0.7
	Genes	15.5	4.5	18	8.15	51.8	7.89	49.5	33.23
SRBCT	Acc (%)	100	0	100	0	100	0	100	0
	Genes	5.4	1.17	11.2	4.38	40.6	5.64	74.8	1.64
Prostate_Tumor	Acc (%)	98.63	0.83	98.04	0	98.43	0.54	98.24	0.44
	Genes	10.9	4.65	15.4	10.95	49.8	13.23	103	9
DLBCL	Acc (%)	100	0	100	0	100	0	100	0
	Genes	5.6	1.83	8.8	1.79	44.4	3.13	92	18.25

The bolding denotes the best solutions.

In order to show the effectiveness of the binary differential evolution, we also compare our algorithm with other well-known metaheuristics, such as the genetic algorithm [40], particle swarm optimization [41] and the estimation of distribution algorithm [42]. In order to conduct a fair comparison, we replace our binary differential evolution in MOBDE by using these metaheuristics. All of these methods use the same multiobjective framework. Table 8 shows the results obtained by binary differential evolution, the genetic algorithm, particle swarm optimization and the estimation of distribution algorithm in terms of the mean and standard deviation (S. D.) of the classification accuracy and the number of genes selected. We can observe in this table that MOBDE clearly outperforms other algorithms in all of the datasets. Therefore, we can conclude that our proposed algorithm shows an efficient and better performance in comparison with these algorithms. In addition, we also list the time of these algorithms in Table 9.

Table 8. Comparative experimental results of the binary differential evolution algorithm with the binary genetic algorithm, binary particle swarm optimization and the binary estimation of binary estimation of distribution algorithm.

Dataset Name	Method Evaluation	This Work (MOBDE)		BGA [40]		BPSO [41]		BEDA [42]	
		Average	S.D.	Average	S.D.	Average	S.D.	Average	S.D.
11_Tumors	Acc (%)	97.19	1.42	97.13	1.52	94.44	0.33	93.3	0.88
	Genes	27.5	6.24	62.67	21.22	88.67	11.37	58.67	14.74
9_Tumors	Acc (%)	92.67	2.62	86.67	1.18	80.67	1.9	83	2.98
	Genes	20.7	6.83	35.2	5.45	87.4	10.21	31.4	4.45
Brain_Tumors1	Acc (%)	97.67	1.22	96	1.69	94.44	0	96.22	0.99
	Genes	11.5	2.17	34	13.64	89.2	5.5	22	8.75
Brain_Tumors2	Acc (%)	99	1.7	93.6	4.34	88	1.41	95.2	4.38
	Genes	7.5	1.72	25.6	14.88	80.6	9.53	12.2	4.6
Leukemia1	Acc (%)	100	0	98.33	1.16	98.33	0.62	100	0
	Genes	5.9	1.29	43.2	10.62	65.4	8.08	11.8	3.9
Leukemia2	Acc (%)	100	0	100	0	100	0	100	0
	Genes	6	1.41	44.8	3.42	62.2	1.48	9.6	1.82
Lung_cancer	Acc (%)	99.12	0.65	98.36	0.28	97.29	0.35	98.85	0.28
	Genes	15.5	4.5	31.33	9.24	83	21.21	31.67	10.5
SRBCT	Acc (%)	100	0	100	0	100	0	100	0
	Genes	5.4	1.17	47	2.45	57.4	3.3615	12.4	6.11
Prostate_Tumor	Acc (%)	98.63	0.83	97.65	1.12	97.25	0.44	97.84	0.44
	Genes	10.9	4.65	38.4	5.08	74.8	3.7	21.6	2.97
DLBCL	Acc (%)	100	0	99.74	0.58	100	0	100	0
	Genes	5.6	1.83	35.2	5.17	65	4.36	15.4	7.47

The bolding denotes the best solutions.

Table 9. Comparative experimental times of the differential evolution algorithm with different multiobjective optimization algorithms.

Dataset Name	MOBDE	BGA	BPSO	BEDA
11_Tumors	7945.6	5922.05	12,306.01	9082.42
9_Tumors	781.43	626.17	731.607	977.61
Brain_Tumors1	1191.58	1113.36	1552.9	1455.6
Brain_Tumors2	332.75	273.235	305.69	291.67
Leukemia1	495.13	614.4	734.03	524.458
Leukemia2	416.58	521.814	549.01	410.484
Lung_cancer	4695.81	6191.65	13,413.08	5092.3
SRBCT	770.32	1321.44	1001.7	877.149
Prostate_Tumor	1037.56	1151.62	1254.92	1293.71
DLBCL	364.5	441.242	668.58	384.981

The bolding denotes the best solutions.

In the test experiment, we compare our proposed method MOBDE with the grid-search SVM without feature selection method. The results are listed in Table 10. From the table, better results between the two algorithms are shown in shaded cells. It is easy to see that both the classification accuracy and the number of selected genes of MOBDE are superior to grid search SVM. This also demonstrates the effectiveness of MOBDE.

Table 10. Comparative experimental results of the binary differential evolution algorithm with grid search support vector machine (SVM).

Dataset Name	Evaluation	This Work (MOBDE)	SVM
11_Tumors	Acc (%)	97.19	89.08
	Genes	27.5	12533
9_Tumors	Acc (%)	92.67	51.67
	Genes	20.7	5726
Brain_Tumors1	Acc (%)	97.67	90
	Genes	11.5	5920
Brain_Tumors2	Acc (%)	99	90
	Genes	7.5	10367
Leukemia1	Acc (%)	100	97.22
	Genes	5.9	5327
Leukemia2	Acc (%)	100	94.44
	Genes	6	11225
Lung_cancer	Acc (%)	99.12	95.07
	Genes	15.5	12600
SRBCT	Acc (%)	100	98.8
	Genes	5.4	2308
Prostate_Tumor	Acc (%)	98.63	93.14
	Genes	10.9	10509
DLBCL	Acc (%)	100	96.1
	Genes	5.6	5469

The bolding denotes the best solutions.

3.2. Compared with Some Single-Objective Algorithms

In order to demonstrate the effectiveness of the proposed method, we also compared our work with some single-objective algorithms. It is worth mentioning that in the previous research, many single-objective algorithms only focused on the accuracy rate of the classification. Therefore, in this paper, we also use the accuracy rate as the compared criteria. Tables 11 and 12 show the results obtained by the MOBDE with other single-objective algorithms including IBPSO1 [8], IBPSO2 [6] and hybrid binary particle swarm optimization and tabu search (HPSOTS). As can be seen in Tables 11 and 12, we can find that the MOBDE algorithm can provide a higher LOOCV classification accuracy on all datasets compared with the other PSO algorithms [6–8] and the other SVM-based algorithms [43,44], except Leukemia1 data. For the Leukemia1 data, the algorithm MOBDE, IBPSO1 [8] and IBPSO2 [6] can all obtain a 100% accuracy rate, while IBPSO1 [8] can obtain a lesser number of genes compared with MOBDE. Based on the above analysis, we can conclude that when only considering the accuracy of classification, the MOBDE algorithm can also perform better than the other algorithms.

Table 11. Comparative experimental results of the binary differential evolution algorithm with some single-objective methods.

Dataset Name	Evaluation	This Work (MOBDE)	IBPSO1 [8]	IBPSO2 [6]	PSOTS [5]
11_Tumors	Acc (%)	97.19	95.06	93.1	-
	Genes	27.5	240.9	2948	
9_Tumors	Acc (%)	92.67	75.5	78.33	-
	Genes	20.7	240.6	1280	
Brain_Tumors1	Acc (%)	97.67	92.56	94.44	-
	Genes	11.5	11.2	754	
Brain_Tumors2	Acc (%)	99	92	94	-
	Genes	7.5	9.1	1197	
Leukemia1	Acc (%)	100	100	100	98.61
	Genes	5.9	3.5	1034	7
Leukemia2	Acc (%)	100	100	100	-
	Genes	6	6.7	1292	
Lung_cancer	Acc (%)	99.12	95.86	96.55	-
	Genes	15.5	14.9	1897	
SRBCT	Acc (%)	100	100	100	-
	Genes	5.4	17.5	431	
Prostate_Tumor	Acc (%)	98.63	97.94	92.16	-
	Genes	10.9	13.6	1294	
DLBCL	Acc (%)	100	100	100	-
	Genes	5.6	6	1042	

The bolding denotes the best solutions.

Table 12. Comparative experimental results of the binary differential evolution algorithm with MOBDE with The maximum margin criterion and support vector machine-based recursive feature elimination (MMC + SVM-RFE), support vector machine-based recursive feature elimination (SVM-RFE) and minimum-redundancy maximum-relevancy (MRMR).

Dataset Name	Evaluation	This Work (MOBDE)	MMC + SVM-RFE [43]	SVM-RFE with MRMR [44]
11_Tumors	Acc (%)	97.19	-	-
	Genes	27.5		
9_Tumors	Acc (%)	92.67	-	-
	Genes	20.7		
Brain_Tumors1	Acc (%)	97.67	67.8	-
	Genes	11.5	100	
Brain_Tumors2	Acc (%)	99	-	-
	Genes	7.5		
Leukemia1	Acc (%)	100	99.7	98.35
	Genes	5.9	100	37
Leukemia2	Acc (%)	100	96	-
	Genes	6	100	
Lung_cancer	Acc (%)	99.12	-	-
	Genes	15.5		
SRBCT	Acc (%)	100	98.7	-
	Genes	5.4	100	
Prostate_Tumor	Acc (%)	98.63	92.1	98.29
	Genes	10.9	30	10
DLBCL	Acc (%)	100	-	-
	Genes	5.6		

The bolding denotes the best solutions.

3.3. Compared with A Multiobjective Algorithm

In this section, we compare our algorithm with a well-known multiobjective optimization algorithm (NSGAI) [9]. The NSGAI algorithm is based on the non-dominated sorting and crowding distance method. Generally speaking, as is shown in Table 13, for all gene datasets, the MOBDE algorithm can provide better accuracy and smaller selected genes for most datasets compared with NSGAI. In Table 13, we also show the computation time comparison of these two algorithms. As can be seen in this table, in all instances, the computational time of our algorithm is less than that of NSGAI. The reason is that our algorithm is still efficient, and the time complexity is $O(INT^3)$, as discussed in Section 2. As for NSGAI, we can simply analyze its time complexity here. In NSGAI, for each iteration, the non-dominated sorting is $O(M(2N)^2)$, and the crowding-distance assignment is $O(M(2N)\log(2N))$, where N is the population size and M is the number of objectives; the data subroutines of standard SVM are $O(T^3)$; so the overall complexity of the iteration is $O(4MN^2T^3)$. Given the I iteration, the total time complexity of NSGAI is $O(4IMN^2T^3)$, i.e., $O(IN^2T^3)$. Obviously, our algorithm is more efficient than NSGAI based on the above analysis. There may be two reasons that our algorithm performs better than NSGAI with fewer selected genes. The first reason is that the new binary mutation strategy used in MOBDE tends to enhance the diversity of the population and share the previous good individuals with the next generation. The second reason is that there may be only very few genes that are necessary for achieving the better classification accuracy, and our method seems more efficient for selecting such genes.

Table 13. Comparative experimental results of the binary differential evolution algorithm with a different multiobjective optimization algorithm.

Dataset	Evaluation	This Work (MOBDE)	Time of MOBDE (s)	NSGA-II	Time of NSGAI (s)
11_Tumors	Acc (%)	97.19	7945.6	92.82	15015.56
	Genes	27.5		105	
9_Tumors	Acc (%)	92.67	781.43	80.33	1109.81
	Genes	20.7		66.2	
Brain_Tumors1	Acc (%)	97.67	1191.58	94	1264.95
	Genes	11.5		37	
Brain_Tumors2	Acc (%)	99	332.75	90	357.86
	Genes	7.5		15.6	
Leukemia1	Acc (%)	100	495.13	99.17	545.7
	Genes	5.9		12.6	
Leukemia2	Acc (%)	100	416.58	100	430.43
	Genes	6		25.2	
Lung_cancer	Acc (%)	99.12	4695.81	97.21	5278.53
	Genes	15.5		55.67	
SRBCT	Acc (%)	100	770.32	100	808.87
	Genes	5.4		23	
Prostate_Tumor	Acc (%)	98.63	1037.56	96.47	1426.49
	Genes	10.9		18.8	
DLBCL	Acc (%)	100	364.5	99.74	394.74
	Genes	5.6		11.2	

The bolding denotes the best solutions.

3.4. The Paired Wilcoxon's Signed Rank Test of Our Algorithm with Other Algorithms

In this part, the paired Wilcoxon's signed rank test is adopted to compare MOBDE with other algorithms to verify whether the experiment results of MOBDE are better than other algorithms [45]. The Wilcoxon's signed-rank test is a non-parametric statistical hypothesis test, which can be used as an alternative to the paired t-test when the results cannot be assumed to be normally distributed. In the paired Wilcoxon's signed rank test, the null hypothesis represents that there

is no significant improvement compared with other feature selection algorithms, and the alternative hypothesis represents that our algorithm is significantly different compared with other feature selection methods. As an example, we can compare our algorithm MOBDE with the well-known algorithm NSGAI. The null hypothesis and alternative hypothesis can be described respectively as follows: $H_0 : \varphi_{MOBDE} = \varphi_{NSGAI}$ and $H_1 : \varphi_{MOBDE} > \varphi_{NSGAI}$, where φ_{MOBDE} and φ_{NSGAI} denote the average accuracy and the number of features selected of MOBDE and NSGAI on all datasets, respectively. As can be seen in Table 14, we can find that the p -values obtained by the paired Wilcoxon's signed rank test between MOBDE and other algorithms are all less than the standard significance level 5%. Therefore, we can draw the conclusion that our algorithm MOBDE significantly outperforms the other algorithms.

Table 14. The Wilcoxon's signed rank test of our algorithm with other algorithms on accuracy and the number of features selected.

Accuracy		The Number of Feature Selected	
Paired Wilcoxon's Signed Rank Test	p -Value	Pair Wilcoxon's Signed Rank Test	p -Value
Binary DE [34]	0.01802	Binary DE [34]	0.004883
BDE [39]	0.02959	BDE [39]	0.0009766
BDEAIS [33]	0.01125	BDEAIS [33]	0.002945
BGA [40]	0.007074	BGA [40]	0.0009766
BPSO [41]	0.01125	BPSO [41]	0.0009766
EDA [42]	0.01802	EDA [42]	0.0009766
Grid search SVM	0.0009766	Grid search SVM	0.0009766
IBPSO [8]	0.01802	IBPSO [8]	0.04199
IBPSO [6]	0.01802	IBPSO [6]	0.0009766
NSGAI	0.007133	NSGAI	0.0009766

3.5. Independent Dataset

In order to check the effectiveness of the final model, an independent test set is needed to test the final model. We set aside approximately 20%–30% of the data for testing the final models by predicting the apoptosis protein locations. Firstly, the raw data constructed by Chen and Li [46] contained 317 apoptosis proteins. According to their subcellular locations, proteins were classified into six groups: 112 cytoplasmic proteins, 34 mitochondrial proteins, 52 nuclear proteins, 17 secreted proteins, 55 membrane proteins and 47 endoplasmic reticulum proteins. In addition, the 98 apoptosis proteins sourced from the paper [47] containing 43 cytoplasmic proteins, 30 plasma membrane-bound proteins, 13 mitochondrial proteins and 12 other proteins are used to test the algorithm. The following step should be used to prepare high quality datasets. We compared our method with other methods, such as PSORT [48] and GASVM [47]. For the leave-one-out cross-validation, each protein is singled out from the benchmark dataset as the test protein, and the remaining proteins would serve as the training dataset to train the predict model. Therefore, we used the leave-one-out cross-validation to evaluate the proposed method.

The support vector machine is used to measure the accuracy of the leave-one-out cross-validation on the feature subset produced by MOBDE. The datasets based on 317 apoptosis proteins are partitioned into one testing sample and $D-1$ training sample. Each individual will take turns being the testing dataset. The other $D-1$ individuals serve as the training dataset for determining the model prediction parameter. The swarm intelligence algorithm is used for selecting a near-optimal subset of informative features that is most relevant for the classification. The overall accuracy is 92.43%. The accuracy of the our method for cytoplasm proteins is 97.32%, for mitochondrial proteins is 88.24%, for nuclear proteins is 94.23%, for secreted proteins is 76.47%, for membrane proteins is 87.27% and for endoplasmic reticulum proteins is 93.62%. Then, we use the proposed method to predict the independent dataset including 98 proteins. The results are shown in Table 15. From Table 15, the accuracy is 95.92%. The accuracy of our method for cytoplasm proteins is 97.67%, for mitochondrial

proteins is 100%, for membrane proteins is 92.31% and for other proteins is 83.33%. From the compared results, the feature selection method can reduce the data dimensionality and find out an optimal amount of features that result in the better performance of the predict model. We hope that the promising results using the new feature selection method can improve the performance of protein subcellular location prediction.

Table 15. Prediction results with different models on the 317 apoptosis proteins dataset and 98 independent proteins dataset.

Model	Class						
317 Protein							
Training Dataset	Cyto	Mito	Memb	Nucl	Secr	Endo	Overall Accuracy (%)
PSORT	51.78	41.17	0	50	82.35	0.02	37.55
GASVM	89.28	91.17	92.72	86.53	88.23	91.48	89.91
This paper	97.32	88.24	87.27	94.23	76.47	93.62	92.43
98 Proteins							
Independent Dataset	Cyto	Mito	Memb	Other		Overall Accuracy (%)	
PSORT	58.13	30.76	0	25		28.47	
GASVM	90.7	92.31	86.67	91.7		90.34	
This paper	97.67	100	92.31	83.33		95.92	

4. Conclusions

The objective of this study is to provide a multiobjective optimization method for feature selection. Our proposed method called MOBDE embraces the strength of the binary differential evolution for classification method and find the smaller subsets. In the first stage, we use the Fisher-Markov selector method to rank the scores of the features and select the 180 top features as the input of the binary differential evolution. Then, a novel binary differential evolution is proposed to select the feature subset. Following that, we propose a multiobjective differential evolution method for the feature selection based on the summation of the normalized objective and diversity selection on ten gene expression datasets. According to the experiments, the following can be concluded.

1. The proposed method can find useful informative features in terms of classification accuracies
2. By using this feature selection method, there is no need to set the number of selected features since the proposed algorithm can automatically select the most useful features in terms of classification accuracies.
3. To show the effectiveness of the Fisher-Markov selector, the experiment of MOBDE with the Fisher-Markov selector and MOBDE without the Fisher-Markov selector is designed. The experimental results show that the Fisher-Markov selector is very effective in feature selection for the bioinformatics dataset.
4. To show the effectiveness of the proposed differential evolution, we compare our algorithm with three different versions of binary differential evolution: binary DE, BDE and BDEAIS. It is better than these different binary differential evolution algorithms in terms of classification accuracy and the number of selected features. Meanwhile, our algorithm also provides better solutions than other binary evolutionary algorithm, including BGA, BPSO and BEDA.
5. Compared with some single objective algorithms, our algorithm outperforms the best algorithm so far on these problems.

The proposed MOBDE algorithm is not only suitable for feature selection and classification in gene expression data, but also for other application domains, such as electricity load forecasting, face recognition and vehicle detection or any other high dimensional data classification.

Acknowledgments: The authors would like to thank the anonymous reviewers for their helpful comments. This research is supported by the National Natural Science Foundation of China under Grant No. 61603087 and also funded by the Natural Science Foundation of Jilin Province under Grant No. 20160101253JC.

Author Contributions: Shijing Ma and Xiangtao Li designed the algorithm and experiments. Shijing Ma performed the experiments. Yunhe Wang analyzed the experimental results. Shijing Ma and Xiangtao Li wrote the algorithm and the experimental sections of the paper. Xiangtao Li organized the structure of the paper. All authors have read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, X.; Yin, M. Multiobjective binary biogeography based optimization for feature selection using gene expression data. *IEEE Trans. Nanobiosci.* **2013**, *12*, 343–353.
2. Liu, B.; Tian, M.; Zhang, C.; Li, X. Discrete Biogeography Based Optimization for Feature Selection in Molecular Signatures. *Mol. Inform.* **2015**, *34*, 197–215.
3. Mitra, S.; Kundu, P.P.; Pedrycz, W. Feature selection using structural similarity. *Inform. Sci.* **2012**, *198*, 48–61.
4. Xue, B.; Zhang, M.J.; Browne, W.N. Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach. *IEEE Trans. Cybern.* **2013**, *43*, 1656–1671.
5. Mukhopadhyay, A.; Mandal, M. A Hybrid Multiobjective Particle Swarm Optimization Approach for Non-redundant Gene Marker Selection. In Proceedings of the International Conference on Bio-Inspired Computing: Theories and Applications, Huangshan, China, 12–14 July 2013; pp. 205–216.
6. Chuang, L.Y.; Chang, H.W.; Tu, C.J.; Yang, C.H. Improved binary PSO for feature selection using gene expression data. *Comput. Biol. Chem.* **2008**, *32*, 29–38.
7. Li, S.; Wu, X.; Tan, M. Gene selection using hybrid particle swarm optimization and genetic algorithm. *Soft Comput.* **2008**, *12*, 1039–1048.
8. Mohamad, M.S.; Omatu, S.; Deris, S.; Yoshioka, M. A modified binary particle swarm optimization for selecting the small subset of informative genes from gene expression data. *IEEE Trans. Inf. Technol. Biomed.* **2011**, *15*, 813–822.
9. Kannan, S.S.; Ramaraj, N. A novel hybrid feature selection via Symmetrical Uncertainty ranking based local memetic search algorithm. *Knowl.-Based Syst.* **2010**, *23*, 580–585.
10. Gutiérrez-Avilés, D.; Rubio-Escudero, C.; Martínez-Álvarez, F.; Riquelme, J.C. TriGen: A genetic algorithm to mine triclusters in temporal gene expression data. *Neurocomputing* **2014**, *132*, 42–53.
11. Xue, B.; Zhang, M.; Browne, W.N. Particle swarm optimisation for feature selection in classification: Novel initialization and updating mechanisms. *Appl. Soft Comput.* **2014**, *18*, 261–276.
12. Huang, B.; Buckley, B.; Kechadi, T.M. Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications. *Expert Syst. Appl.* **2010**, *37*, 3638–3646.
13. Zhou, A.; Qu, B.Y.; Li, H.; Zhao, S.Z.; Suganthan, P.N.; Zhang, Q. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm Evolut. Comput.* **2011**, *1*, 32–49.
14. Xue, B.; Cervante, L.; Shang, L.; Browne, W.N.; Zhang, M. A multiobjective particle swarm optimisation for filter-based feature selection in classification problems. *Connect. Sci.* **2012**, *24*, 91–116.
15. Mohamad, M.S.; Omatu, S.; Deris, S.; Misman, M.F.; Yoshioka, M. A multiobjective strategy in genetic algorithms for gene selection of gene expression data. *Artif. Life Robot.* **2009**, *13*, 410–413.
16. Castro, P.A.; Von Zuben, F.J. Multi-objective feature selection using a Bayesian artificial immune system. *Int. J. Intell. Comput. Cybern.* **2010**, *3*, 235–256.
17. Aydin, I.; Karakose, M.; Akin, E. A multiobjective artificial immune algorithm for parameter optimization in support vector machine. *Appl. Soft Comput.* **2011**, *11*, 120–129.
18. Rubio-Escudero, C.; Martínez-Álvarez, F.; Romero-Zaliz, R.; Zwir, I. Classification of gene expression profiles: Comparison of K-means and expectation maximization algorithms. In Proceedings of the IEEE Eighth International Conference on Hybrid Intelligent Systems (HIS'08), Barcelona, Spain, 10–12 September 2008; pp. 831–836.
19. Romero-Zaliz, R.; Rubio-Escudero, C.; Zwir, I.; del Val, C. Optimization of multi-classifiers for computational biology: Application to gene finding and expression. *Theor. Chem. Acc.* **2010**, *125*, 599–611.
20. Storn, R.; Price, K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359.

21. Das, S.; Suganthan, P.N. Differential evolution: A survey of the state-of-the-art. *IEEE Trans. Evolut. Comput.* **2011**, *15*, 4–31.
22. Wang, Y.; Li, H.X.; Yen, G.G.; Song, W. MOMMOP: Multiobjective optimization for locating multiple optimal solutions of multimodal optimization problems. *IEEE Trans. Cybern.* **2015**, *45*, 830–843.
23. Wang, Y.; Li, H.X.; Huang, T.; Li, L. Differential evolution based on covariance matrix learning and bimodal distribution parameter setting. *Appl. Soft Comput.* **2014**, *18*, 232–247.
24. Wang, Y.; Cai, Z.; Zhang, Q. Enhancing the search ability of differential evolution through orthogonal cross-over. *Inf. Sci.* **2012**, *185*, 153–177.
25. Liu, H.; Cai, Z.; Wang, Y. Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Appl. Soft Comput.* **2010**, *10*, 629–640.
26. Zheng, Y.J.; Xu, X.L.; Ling, H.F.; Chen, S.Y. A hybrid fireworks optimization method with differential evolution operators. *Neurocomputing* **2015**, *148*, 75–82.
27. Guo, S.M.; Yang, C.C. Enhancing differential evolution utilizing eigenvector-based cross-over operator. *IEEE Trans. Evolut. Comput.* **2015**, *19*, 31–49.
28. Wang, Y.; Liu, Z.Z.; Li, J.; Li, H.X.; Yen, G.G. Utilizing cumulative population distribution information in differential evolution. *Appl. Soft Comput.* **2016**, *48*, 329–346.
29. Li, X.; Yin, M. Modified differential evolution with self-adaptive parameters method. *J. Comb. Optim.* **2016**, *31*, 546–576.
30. Wang, Y.; Cai, Z.; Zhang, Q. Differential evolution with composite trial vector generation strategies and control parameters. *IEEE Trans. Evolut. Comput.* **2011**, *15*, 55–66.
31. Zhang, J.; Sanderson, A.C. JADE: Adaptive differential evolution with optional external archive. *IEEE Trans. Evolut. Comput.* **2009**, *13*, 945–958.
32. Cheng, Q.; Zhou, H.; Cheng, J. The Fisher-Markov selector: Fast selecting maximally separable feature subset for multiclass classification with applications to high-dimensional data. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1217–1233.
33. He, X.; Han, L. A novel binary differential evolution algorithm based on artificial immune system. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 2267–2272.
34. Gong, T.; Tuson, A.L. Differential evolution for binary encoding. In *Soft Computing in Industrial Applications*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 251–262.
35. Qu, B.Y.; Suganthan, P.N. Multi-objective evolutionary algorithms based on the summation of normalized objectives and diversified selection. *Inf. Sci.* **2010**, *180*, 3170–3181.
36. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297.
37. Tsang, I.W.; Kwok, J.T.; Cheung, P.M. Core vector machines: Fast SVM training on very large data sets. *J. Mach. Learn. Res.* **2005**, *6*, 363–392.
38. Chang, C.C.; Lin, C.J. LIBSVM—A Library for Support Vector Machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27.
39. Kanlikilicer, A.E.; Keles, A.; Uyar, A.S. Experimental analysis of binary differential evolution in dynamic environments. In Proceedings of the 9th Annual Conference Companion on Genetic and Evolutionary Computation, London, UK, 07–11 July 2007; pp. 2509–2514.
40. Lin, S.W.; Ying, K.C.; Chen, S.C.; Lee, Z.J. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Syst. Appl.* **2008**, *35*, 1817–1824.
41. Oreski, S.; Oreski, G. Genetic algorithm-based heuristic for feature selection in credit risk assessment. *Expert Syst. Appl.* **2014**, *41*, 2052–2064.
42. Saeys, Y.; Degroove, S.; Aeyels, D.; Van de Peer, Y.; Rouzé, P. Fast feature selection using a simple estimation of distribution algorithm: A case study on splice site prediction. *Bioinformatics* **2003**, *19*, 179–188.
43. Nijijima, S.; Kuhara, S. Recursive gene selection based on maximum margin criterion: A comparison with SVM-RFE. *BMC Bioinform.* **2006**, *7*, 543–561.
44. Mundra, P.A.; Rajapakse, J.C. SVM-RFE with MRMR filter for gene selection. *IEEE Trans. Nanobiosci.* **2010**, *9*, 31–37.
45. Chiclana, F.; García, J.T.; del Moral, M.J.; Herrera-Viedma, E. A statistical comparative study of different similarity measures of consensus in group decision making. *Inf. Sci.* **2013**, *221*, 110–123.

46. Chen, Y.L.; Li, Q.Z. Prediction of the subcellular location of apoptosis proteins. *J. Theor. Biol.* **2007**, *245*, 775–783.
47. Kandaswamy, K.K.; Pugalenthi, G.; Moller, S.; Hartmann, E.; Kalies, K.U.; Suganthan, P.N.; Martinetz, T. Prediction of apoptosis protein locations with genetic algorithms and support vector machines through a new mode of pseudo amino acid composition. *Protein Pept. Lett.* **2010**, *17*, 1473–1479.
48. Horton, P.; Park, K.J.; Obayashi, T.; Fujita, N.; Harada, H.; Adams-Collier, C.J.; Nakai, K. WoLF PSORT: Protein localization predictor. *Nucleic Acids Res.* **2007**, *35*, 585–587.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).