A Multi-objective Structure Variant Bacterial Heuristic Feature Selection Method in High-dimensional Data Classification

Hong Wang, Yikun Ou, and Yixin Wang(✉)

College of Management of Shenzhen University, Shenzhen, China
wangyixin2020@email.szu.edu.cn

Abstract. Feature selection (FS) has been studied as a multi-objective problem in recent years to improve the classification effect. In this paper, to expand the ability of Bacterial Foraging Optimization (BFO) in feature selection problems for classification, a Multi-objective Adapting Chemotaxis Bacterial Foraging Optimization (abbreviated as MOACBFO) is proposed. In MOACBFO, a structural variation strategic model is proposed to reduce the computational complexity for multi-objective problems and applied with a dynamically updated external matrix to record the performance of bacteria on two objectives. In addition, an adaptive chemotaxis step mechanism is designed to help the bacteria jump out of the local optimality. To further enhance the diversity of bacteria searching capability, a feature subset updating strategy is developed. The optimal feature subset and fitness value are stored in the external matrix continuously by comparing them with the historical value records. The performance of the proposed algorithm is demonstrated by comparing it with four other advanced swarm intelligent algorithms on 11 high-dimensional microarray datasets. The results indicate that the MOACBFO performs better in achieving a lower classification error rate using a small number of features.

Keywords: Bacterial foraging optimization · Feature selection · Structural variation · Adaptive chemotaxis step size

1 Introduction

Iterative updates in information technology increase the amount of data generated and retrieved, resulting in most of the data becoming high-dimensional. To deeply uncover the effective information hidden in the data, researchers are trying to design various advanced data mining methods. In this context, feature selection has become a widely studied method as it can reduce the dimensionality of data, using a small combination of features to express the most information of the primal high-dimensional data.

This paper is submitted to Special Session on Intelligent Data Mining: Techniques and Applications (Session Chair: Ben Niu, Hong Wang)

Y. Tan et al. (Eds.): DMBD 2021, CCIS 1454, pp. 342–357, 2021.
https://doi.org/10.1007/978-981-16-7502-7_34

Most of the features in the high-dimensional data are not always valid and necessary. It may contain a large number of redundant, and invalid information [1], which will significantly affect the effect of data analysis. For example, redundant features may increase the computation cost while invalid information may decrease the classification accuracy. Therefore, an appropriate method of feature selection can improve the reliability and validity of data analysis. At present, the FS for high-dimensional data has been applied in many fields, such as customer relationship management in enterprises [2], disease prediction [3] in modern medical diagnosis technologies, and customer recommendation [4] in electronic commerce.

FS is a preprocessing process, using certain evaluation criteria to select subsets from the original feature space [5]. The process of basic FS is followed in Fig. 1.

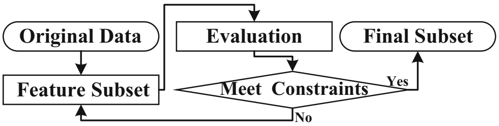


Fig. 1. General flow of basic FS.

In the process of FS, the feature subsets are generated according to some regulations or strategies [1]. Then, the fitness criteria are used to evaluate the effectiveness of the subsets, and the better performance features are stored in the final feature subset. The purpose of selecting feature subsets is to filter out invalid, incomplete, and redundant features as much as possible [2]. In other words, a good feature subset can use fewer features to present most of the attributes of the original data. Therefore, the information conveys by the original data can be better retained. From the processing, it can be seen that FS is also a typical optimization problem. In consequence, FS is now being studied extensively.

Aiming at reducing feature redundancy, J. Lee [3] designed a multi-variate feature ranking mechanism for FS in high-dimensional microarray data. This mechanism not only considers the relevance of features and targets but also considers the redundancy of repeating features. The ranking mechanism filters out the high-quality feature to generate new subsets. E. S. Hosseini and M. H. Moattar [4] also noticed the imbalance of the data features and instances. The candidate feature subsets were formed and evaluated by a multi-interactive information strategy to reduce the analysis bias caused by the imbalanced data.

In addition to data redundancy and imbalance, FS should also consider the computational complexity. In this regard, some studies used the differential evolution method to filter features, enhancing the ability of FS to select feature combination [5]. F. Aghaeipoor and M.M. Javidi [6] adopted a hybrid FS method of filter and wrapper methods, and selected feature subsets by mRMR and fuzzy rules to improve estimation accuracy and decrease the computing time.

Many researchers regard FS as a single objective task in the past. However, there are increasing amounts of researchers who beginning to pay attention to the interpretability and rationality of the feature subsets generated by FS. The inner connection of each feature in a subset needs to be considered. Especially, when classifying high-dimensional data, the algorithm may generate some large feature subsets without removing the redundant features, which is antithetical to the original intention of FS to reduce the dimension of data. Nevertheless, if the feature subsets are too small, it may filter out a lot of information. To make the subsets more concise with low dimensions, it is wise to consider FS as a multi-objective problem. This paper takes error rates of classification and size of feature subsets as the objective of FS.

The swarm intelligence (SI) algorithms are widely used in multi-objective FS due to their characteristics of simple structure and fewer mathematical rules. SI algorithms imitate the evolution processes of the biological population to construct systems with functions of search optimal solutions. It means to use the population to search a large number of solutions to a given problem and find out the optimal one. Common SI algorithms are artificial bee colony (ABC) [7], particle swarm optimization (PSO) [8], ant colony optimization (ACO) [8], and so on.

Focusing on the problems of multi-objective FS, researchers have made various improvements based on the original SI methods. In [9], the authors take the classification accuracy and the number of selected features as the objectives of FS. To reduce the computational cost of multi-objectives, it embeds the updating mechanism of PSO into the ABC to control the displacement of search individuals. Besides, a ladder-like sample combinations strategy is developed to increase search efficiency. Considering reducing the size of the subsets, Y. Zhou, et al. [10] chose to use the discrete method to compress the search space of the particle swarm, and set the size of particle population by a flexible cut-point method to better search optimal solutions.

Only reducing the size of the subsets can easily make the algorithms fall into the situation of local optimum. To escape the local optimum, W. Wei, et al. [11] design a vector-based elite learning rule and applied it to the immune algorithm to remove redundant features in the data being analyzed. However, elite learning often brings about the problem of premature convergence. To control the algorithms' speed of convergence, A. Santiago, et al. [12] propose a fuzzy selection of operators. All in all, SI methods are applied to multi-objective FS and they can be modified variously.

Except for the SI algorithms mentioned above, bacterial foraging optimization (BFO) is also a widely studied SI algorithm. BFO has been used to deal with a variety of complex optimization problems because of its outstanding searchability and versatility. Inspired by the foraging activities of E. coli, BFO simulates the life process of bacteria, Sect. 2 will give more details.

The common researches of BFO mainly focus on the improvement strategies of the algorithm, the problems of combinatorial optimization, and optimization scheduling. The improvement strategies mainly include parameters optimization, bacteria chemotaxis process optimization, and population structure optimization. For example, H. Chen et al. [13] designed a chaotic local search strategy to better control the convergence speed of BFO and the defect of the constant step length of bacteria chemotaxis, which increased the search diversity by dynamically changing the chemotaxis step length. In addition, the introduction of chaotic strategies has also expanded the search area of the bacteria population.

Except for the algorithm strategies' improvements, BFO also has multiple applications. B. Turanoğlu and G. Akkaya [14], aiming at the problem of dynamic facility layout, design a hybrid algorithm combining BFO and simulated annealing (SA) to solve such complex optimization problems. In addition to the problem of the dynamic facility, M. Kaur and S. Kadam [15], in the optimal scheduling problem, propose a multi-objective flora optimization algorithm to find the scheme closest to the optimal scheduling solution. However, the size of the population will significantly influence the search effect of BFO. For modulating the search population of the algorithm, H. Wang, et al. [16] design a multi-dimensional population mechanism of BFO based on the FS method. The algorithm can generate populations in different sizes, thereby generating different feature subsets in classification tasks to increase the variety of solutions. At present, there is very little research that focuses on BFO-based multi-objective FS, but it can be found from those existing studies that this area is very useful and meaningful because BFO performs well in multi-objective problems [17–19].

1.1 Goals

This research aims to reduce the dimensionality of high-dimensional data when doing classification tasks. So we design a modified BFO with the improved strategies of adaptive foraging process, structural variation, and update search bacteria population based on multi-objective FS. The specific contributions are as follows:

- Design a multi-objective BFO with structural variation to improve the problem of slow convergence of the original algorithm.
- Adaptively adjust the process of chemotaxis, replication, and elimination-dispersal of the improved BFO to increase the diversity of the search process.
- Design a feature subsets updating mechanism to identify features that can retain better classification effects and eliminate features that will weaken the classification accuracy.
- Combine the proposed algorithm with the KNN classifier to form a wrapper FS method to achieve efficient classification effects.

1.2 Organization

The rest of this article is structured as follows: Sect. 2 describes the principle of the basic BFO. The proposed algorithm and its improvement strategies are described in Sect. 3. Section 4 shows the experimental results and analysis, and the conclusion is placed in Sect. 5.

2 Basic Bacterial Foraging Optimization

The basic Bacterial Foraging Optimization (BFO) was proposed by K.M. Passino, which mimics the biological habits of E. coli. He designed chemotaxis, replication, and elimination-dispersion three main processes for the BFO algorithm. In addition, each bacterium involved activities of swimming, turning, attraction, and repulsion. It can be seen that BFO is a biological heuristic algorithm, and the researches show that this algorithm has strong searchability [20].

The process of bacterial foraging is called chemotaxis. In the chemotaxis, bacteria swarm towards the place with a high concentration of nutrients. One chemotaxis contains two steps: tumbling and swimming. Bacteria select a random direction by tumbling to find a place with a high nutrient concentration. By swimming, bacteria continuously exploit the direction with high nutrient concentration. In addition, the bacteria colony also has a cell-to-cell signaling communication mechanism via special pheromone. Once a unit finds a good place, it will release an attraction pheromone to inform other units. On the contrary, if a bacterium is exposed to a lower nutrient concentration place or a noxious one, it will release a repulsive pheromone to warn other units to avoid approaching. In the BFO algorithm, this mechanism speeds up the convergence and finds the global optimum more efficiently. The position updating of bacterial foraging optimization during the chemotaxis stage is [20]:

$$\theta^i(j+1,k,l)=\theta^i(j,k,l)+C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{1}$$

The $\theta^i(j,k,l)$ presents the position of i_{th} bacterium in j_{th} chemotaxis, k_{th} and l_{th} stand for the process of reproduction elimination-dispersal, respectively. $C(i)$ means the step of chemotaxis and $\Delta(i)$ is a manually setting parameter that controls the actives of i_{th} bacterium. As there are attraction and repulsion actives between bacteria, the formula is followed:

$$J_{cc}\big(\theta^i(j,k,l),P(j,k,l)\big)=\sum\nolimits_j^S J_{cc}\big(\theta^i(j,k,l),\theta^j(j,k,l)\big) \tag{2}$$

Where $J_{cc}\big(\theta^i(j,k,l),P(j,k,l)\big)$ denotes the cell-to-cell signaling of i_{th} bacterium among other bacteria. $J_{cc}\big(\theta^i(j,k,l),\theta^j(j,k,l)\big)$ controls the communication level of attractant or repellant between i_{th} andj_{th} bacterium. Therefore, the i_{th} the bacterium will be updating with swarming effect not just its fitness $J(i,j,k,l)$:

$$J(i,j,k,l)=J(i,j,k,l)+J_{cc}\big(\theta^i(j,k,l),P(j,k,l)\big) \tag{3}$$

Reproduction simulates the most basic criteria in the natural world that is survival of the fittest. It improves the superiority of bacterial colonies and thus accelerates the convergence of BFO. Firstly, bacteria colonies are ranked according to the health level of each bacterium. Secondly, half of the bacteria with lower health levels are replaced by the half with higher. As a result, the overall health level in next-generation is improved. The health of bacteria are as follow:

$$J^i_{health}=\sum\nolimits_{j=1}^{N_c+1}J(i,j,k,l) \tag{4}$$

Where the J^i_{health} is the health level of i_{th} the bacterium, it is calculated by adding up each fitness value in N_c chemotaxis processes.

Elimination-dispersal emulates uncertainty in nature. In the life cycle of a bacteria colony, each bacterium could die abruptly or be transported to somewhere else randomly since the uncertain natural disasters. In BFO, the position of bacteria will be reset in a custom probability P_{ed}, which helps the optimization algorithm reduce premature convergence.

3 The Proposed Multi-objective Adapting Chemotaxis BFO Method

Although the basic BFO can realize solving optimization problems. However, the algorithm's computational time cost of BFO will be increased when faced with high-dimensional data. In addition, the original algorithm is more suitable for single-objective problems. In this paper, we regard FS as a multi-objective problem with two objectives, which are "minimize the size of feature subset" and "maximize classification effect (minimum error rate)".

This paper proposes an effective algorithm, multi-objective adapting chemotaxis bacteria foraging optimization (MOACBFO), the general flow of MOACBFO is shown in Fig. 2. '*h*' represents the running time of the program, '*iter*' means the iteration times of each running. '*m*' is the swimming number of times of each bacteria.

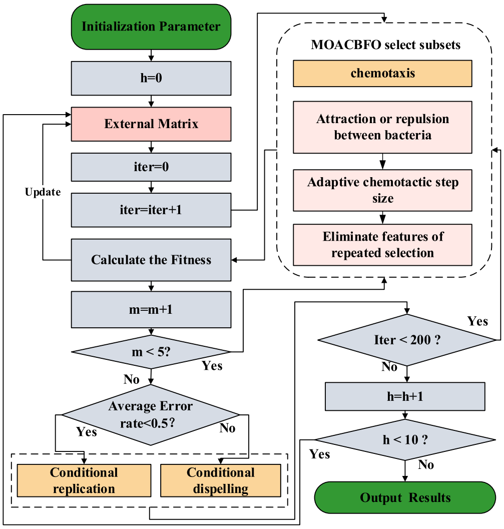


Fig. 2. General flow of MOACBFO.

Based on the original BFO, MOACBFO has four improved strategies, including *structural variation* (structure mutation of the basic BFO), *external matrix* mechanisms, *adaptive foraging* processes, and *feature subsets updating* mechanism.

3.1 Structure Variation

Traditional BFO is a multi-layer nested structure algorithm, when the bacterial population faces a high-dimensional search space, many calculation processes will be generated. However, as the scale of data analysis increases, many repetitive calculation results will be generated, which does not conform to the original intention of creating a concise and efficient algorithm. Therefore, this paper disassembled the nested structure of BFO, setting the three processes (chemotaxis, replication, and elimination-dispersal) to be carried out sequentially instead of being nested. We canceled the original nested loop structure of BFO and carried out serial and parallel transformations.

As Fig. 2 shows, in the search process of bacteria, the chemotaxis operation is performed first, and the classification effect of the intermediate process is used to decide whether the algorithm goes to replication operation or elimination-dispersal operation in the next stage of the algorithm. It is worth noting that replication or elimination-dispersal is not performed in every iteration, only if the conditions are met will one of the activities take place.

3.2 External Matrix

To make BFO more suitable for solving multi-objective problems, we introduce an external matrix (*EMatrix*) [21]. In each iteration, the matrix is used to store the location information of the population generated by the improved bacteria optimization algorithm, the classification error rate of the classifier, and the size of the feature subset generated each time. The classifier used in this paper is the K-nearest neighbor classifier, which is lightweight and efficient.

$$Fe_i = [x_{i1}, x_{i2}, \ldots, x_{in}], i = 1, 2, \ldots, z. \tag{5}$$

$$Pos_j = \left[Fe'_1, Fe'_2, \ldots, Fe'_k\right] \tag{6}$$

$$ErrR_j = \left[fit(Pos_1), fit(Pos_2), \ldots, fit(Pos_{iter})\right] \tag{7}$$

$$FitN_j = [Num(Pos_1), Num(Pos_2), \ldots, Num(Pos_{iter})] \tag{8}$$

$$EMatrix = \left[Pos_j, ErrR_j, FitN_j\right], j = 1, 2, \ldots, iter \tag{9}$$

Formula 5 represents the set of all features, ***'n'*** represents the number of features, and ***'z'*** represents the number of samples. ***'Pos'*** records the feature information of the j^{th} generation of bacterial colony search, and ***'k'*** represents the number of selected features, see *Formula* 6. ***'ErrR'*** represents the classification error rate obtained by training the feature subset generated by the j^{th} generation of flora into the classifier, see *Formula* 7 and ***'iter'*** represents the number of iterations. ***'FitN'*** represents the size of the feature subset generated by the j^{th} generation of flora, see *Formula* 8.

3.3 Adapting Foraging Process

The adaptive bacterial foraging process is mainly the adaptive change of the step length of bacterial chemotaxis. In addition, there are adaptive options for replication and elimination-dispersal. The basic step length of chemotaxis in BFO is invariable, but the movement of bacteria is not rigid in the real world. The fixed-step makes the bacteria easy to be caught in the same local place which is not beneficial for the diversified development of the population. In other words, the lack of diversity of the bacterium will lead to the results becoming incomplete. In this paper, a simple adaptive chemotaxis step-changing method is adopted [22]. In the MOACBFO, the initial step control mechanism for each bacterium is followed *formula* (10–11):

$$\gamma = \left|\left(1 - \frac{i}{Swarm}\right) * (Che_{start} - Che_{end}) + Che_{end}\right| \tag{10}$$

$$Che_{start} = \frac{|J_{new}(i,j)|}{|J_{new}(i,j)| + \gamma} = \frac{1}{\left(1 + \frac{\gamma}{|J_{new}(i,j)|}\right)} \tag{11}$$

'γ' is an adaptive parameter following *formula* (10), it controls the step length of chemotaxis ('Che$_{st}$'). $J_{new}(i,j)$ is the variation of *formula* (3). It removes ***'k'*** and ***'l'*** because the structure of BFO has been changed and the new strategy has followed the *formula* (14).

One bacterium may do reduplicative work if there is no communication between bacteria and the population. To reduce the computation cost, this paper introduces a learning strategy from PSO, see *formula* (12).

$$Che_{end} = Che_{start} + c_a R_a (PB_i - P_i) + c_b R_b (B - P_i) \tag{12}$$

'PB_i' stands for the best solution of i^{th} bacterium and 'B' is the best location searched by the entire bacteria groups. Table 1 shows the pseudo-code of MOACBFO.

Table 1. The pseudo-code of MOACBFO

```
Initialize the parameters (S=45, Nc =4,Ns=5,h=10)
    Swimming: input dynamic quantity characteristics
If the number of data features < 30
    Input feature number = 1:1: number of data features
Else
    Input feature number = 1:5:50
End (This step is to control the number of features)
For j=1:Nc
        i=1:S
        Do the chemotaxis (10-12); Record bacteria search information by (9); Get a fitness by (13)
    If new fitness < original fitness
        Record the best fitness, number of features, and position of bacteria
    End
        Replication:
    While m<Ns(tumbling times)
        Take tumble action in BFO, get fitness by (10-13), and updating the EMatrix.
        If fitness < threshold value
            Do Feature Subsets Updating Strategy, and then do replication.
            Elimination:
        Else (error rate doesn't meet expectations, no need to replication )
            Do elimination-dispersal by updating the position of bacteria. (see, TableII)
        End
    End
End
```

When the error rate is over 50%, the bacteria population will do the elimination-dispersal operation. A high error rate means the combination of features in the subset cannot reflect most information of the data. To save the computation resources, we set the MOACBFO only do elimination-dispersal instead of replication. On the contrary, if the error rate is small, the MOACBFO will take the replication activity.

3.4 Feature Subsets Updating Strategy

High dimensional data always bring thousands or more instances with many features. If the search population has been the same all the time, the diversity of feature subsets will lack change. This paper proposes a feature subsets updating strategy, see Table 2.

Before updating the features in a subset, the position of each bacterium *'P'* needs to be found. The basic position updating formula is followed:

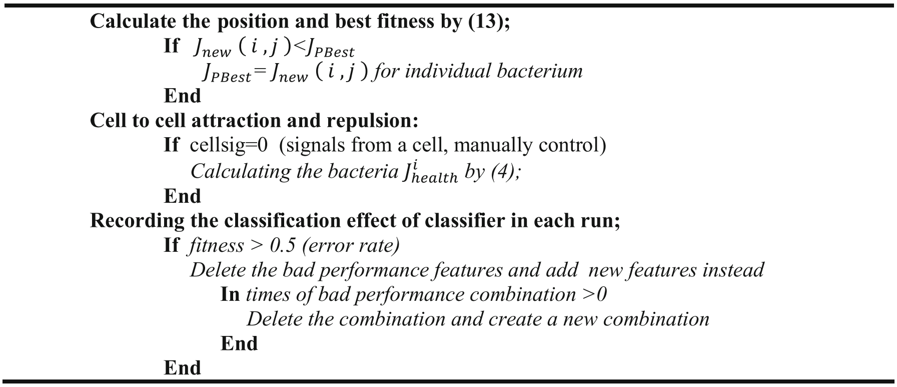
$$P(i, j+1) = Pos(i, j) + C_{en}(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{13}$$

After recording the information of position into *EMatrix*, the fitness (error rate) must be calculated and saved into *EMatrix*. The fitness value $J_{new}(i, j)$ is acquired by *formula* (14). It uses the K-nearest neighbor to be the classifier, which takes the *'P'* as the input value.

$$J_{new}(i, j) = Classifier_{KNN}(P(i, j+1)) \tag{14}$$

The pseudo-code of feature subsets updating strategy is below.

Table 2. The pseudo-code of feature subsets updating



```
Calculate the position and best fitness by (13);
    If $J_{new}(i,j) < J_{PBest}$
        $J_{PBest} = J_{new}(i,j)$ for individual bacterium
    End
Cell to cell attraction and repulsion:
    If cellsig=0 (signals from a cell, manually control)
        Calculating the bacteria $J^i_{health}$ by (4);
    End
Recording the classification effect of classifier in each run;
    If fitness > 0.5 (error rate)
        Delete the bad performance features and add new features instead
            In times of bad performance combination >0
                Delete the combination and create a new combination
            End
    End
```

4 Experiments and Analysis

In this section, the proposed algorithm is compared with four derived versions of traditional swarm intelligent algorithms. The experiment is carried on 11 microarray data.

4.1 Experiment Design and Parameter Initialization

The comparing algorithms are BPSO [23], BFO [20], BFOLIW [24] and BWFS [21]. Table 3 records the characteristics of the comparing algorithms. To compare the MOACBFO to the multi-objective algorithm, the comparing algorithms in this paper are all added to the external matrix mechanisms used in this paper. As for this, the final comparison algorithms are BPSO, BFO, BFOLIW, and BWFS in a multi-objective version.

Table 3. Algorithms for comparison

Basic algorithms	Characteristic
BPSO	FS can be treated as a 0/1 problem. For the features, 0 stands for unselected and 1 stands for selected. The binary particle swarm algorithm is a classical algorithm and is suitable for solving 0/1 problems
BFO	As the basis of the improved method (MOACBFO) in this paper, it is necessary to show the capacity of BFO in classification targets based on FS
BFOLIW	BFOLIW integrates the liner weight strategy to the basic BFO
BWFS	BWFS imports the roulette wheel mechanism into BFO to increase bacterial diversity

To make the comparison experiment more fair and reliable, the parameters set was referred to most of the literature. As the popular size S is usually 30 to 50, this paper takes 45 as the popular size and dimension. The maximum iterations are 200. The running time of the program is 10.

4.2 Data Sources

The testing datasets are 11 microarray data from the UCI machine learning repository [24]. The microarray datasets are high-dimension data about cancer, which have two or more categories and more than 5,000 features. The experiment compares the error rate and the size of the subset of the algorithm.

Table 4. Microarray datasets

Datasets	Features	Instance	Class
9_Tumors	5726	60	9
11_Tumors	12,533	174	11
14_Tumors	15,009	308	26
Brain_Tumor_1	5920	90	5
Brain_Tumor_2	10367	50	40
Leukemia_1	5328	72	3
Leukemia_2	11225	73	3
Lung_Cancer_I	12600	203	5
SRBCT	2309	83	4
DLBCL	5470	77	2
Prostate_Tumor	10509	102	2

4.3 Experiment Results and Analysis

Figure 3 and Fig. 4 show the experiment results of the comparisons. The abscissa axis represents the size of the subsets generated by the algorithms, and the axis of ordinates records the error rate of classification. From the results, MOACBFO always gets a minimum error rate. It reflects that the MOACBFO has a good ability in searching the optimization result.

Besides, MOACBFO can get a small number of features in a subset with a lower error rate compared with other methods. The results in Fig. 3 and Fig. 4 prove that MOACBFO has a better comprehensive performance. Especially in the datasets DLBCL, Lung-Cancer, Prostate-Tumor, Brain-Tumor2, Leukemia1, and SRBCT, MOACBFO gets better diversity of results.

To compare the effect of the algorithms in more detail, Table 5 shows the numerical results of average error rates and subsets' size of all methods. As shown in the Table 5, the MOACBFO can generally reach an error rate of less than 0.005. Although the number of features in each dataset is not the smallest one, the error rate of MOACBFO is low. Since the results need to fit the two objectives (minimize error rates and feature subsets).

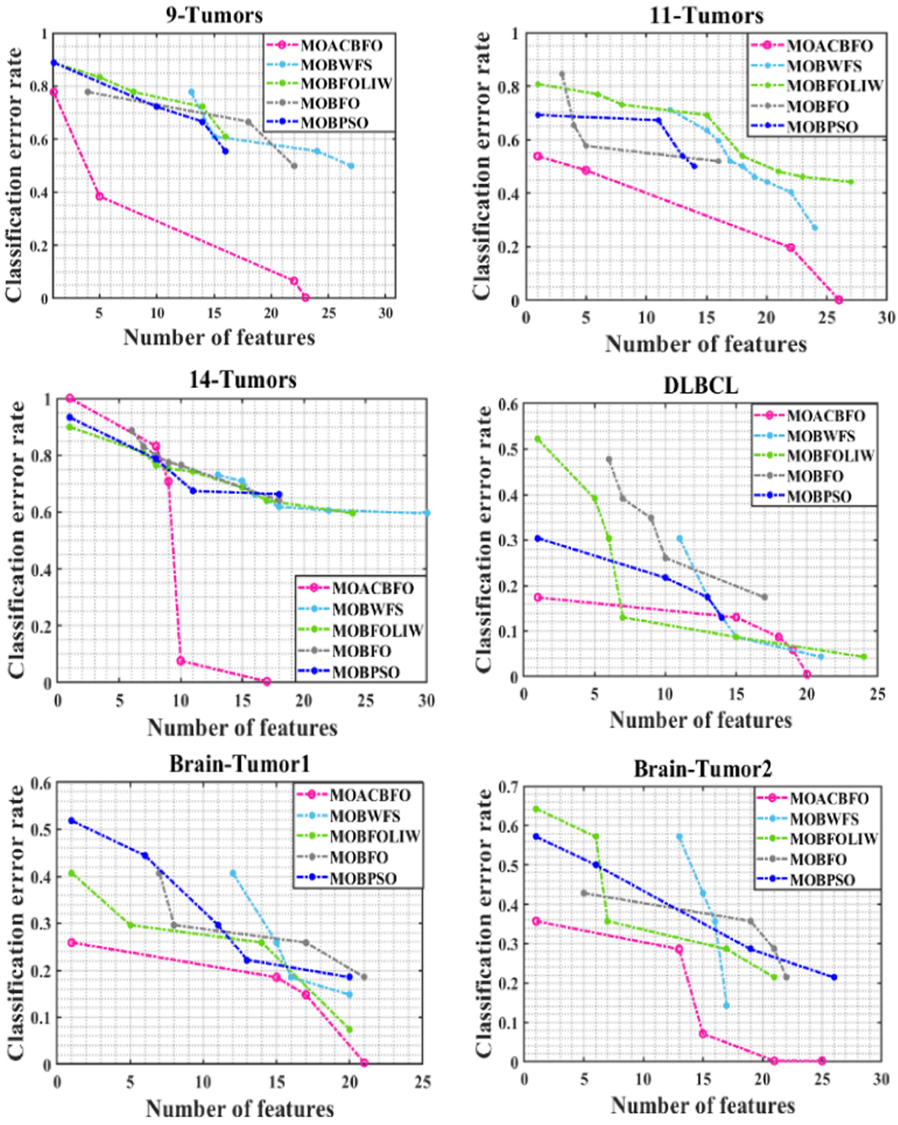


Fig. 3. Experiments results of all algorithms in comparison on microarray datasets.

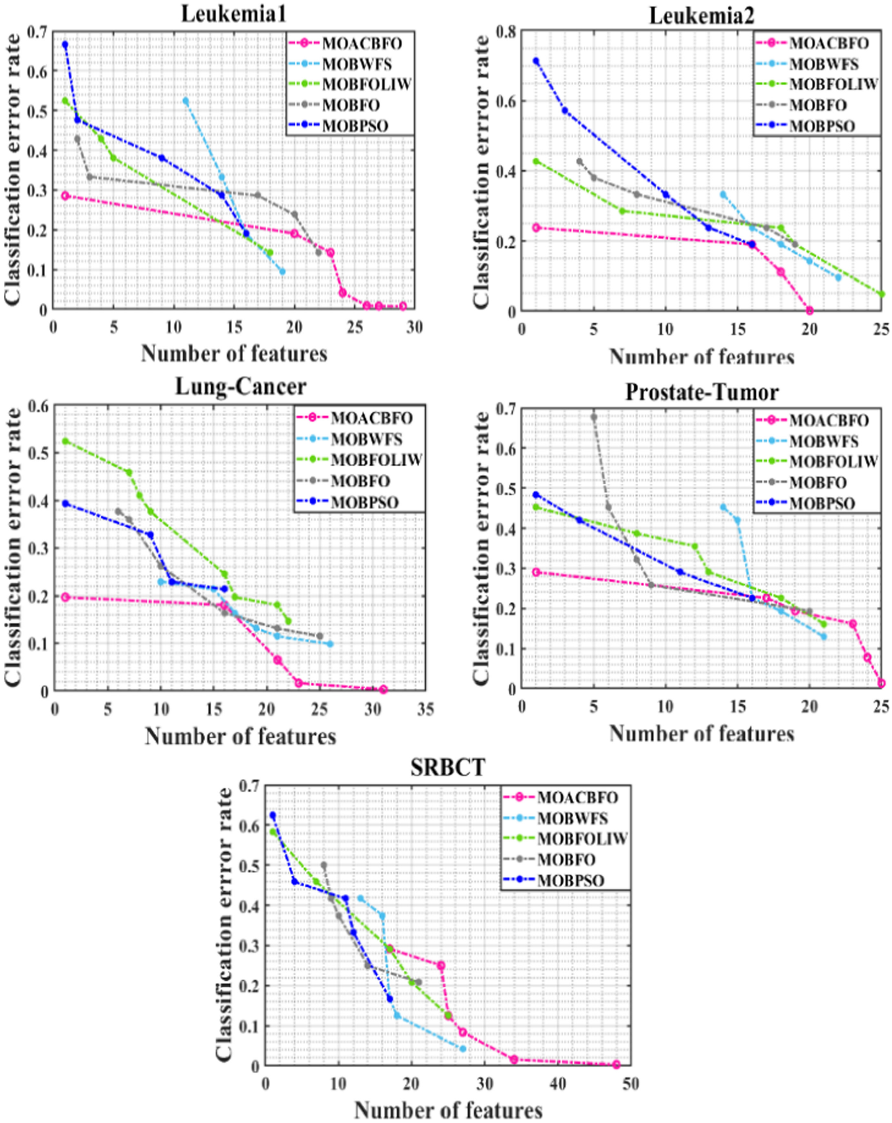


Fig. 4. Experiments results of all algorithms in comparison on microarray datasets.

Table 5. Average error rate and subset size of all algorithms

Datasets	Average	Algorithms				
		MO-B-PSO	MO-BFO	MO-LIW-BFO	MO-BWFS	MO-AC-BFO
		(Binary)	(Original)	(Linearly weight)	(Roulette wheels)	(Improved method in this paper)
9_Tumors	A_Size	**16**	22	**16**	27	23
	A_Error	0.556	0.500	0.611	0.500	**0.003**
11_Tumors	A_Size	**14**	16	27	24	26
	A_Error	0.500	0.519	0.442	0.269	**0.002**
14_Tumors	A_Size	18	18	24	30	**17**
	A_Error	0.663	0.640	0.596	0.596	**0.004**
Brain_Tumor_1	A_Size	20	21	20	20	21
	A_Error	0.185	0.185	0.074	0.148	**0.003**
Brain_Tumor_2	A_Size	26	22	21	**17**	25
	A_Error	0.214	0.214	0.214	0.143	**0.002**
DLBCL	A_Size	**14**	17	24	21	20
	A_Error	0.130	0.174	0.043	0.043	**0.005**
Leukemia_1	A_Size	**16**	22	18	19	29
	A_Error	0.190	0.143	0.143	0.095	**0.008**
Leukemia_2	A_Size	**16**	19	25	22	20
	A_Error	0.190	0.190	0.048	0.095	**0.002**
SRBCT	A_Size	**17**	21	25	27	48
	A_Error	0.167	0.208	0.125	0.417	**0.003**
Lung_Cancer_I	A_Size	**16**	25	22	26	31
	A_Error	0.213	0.115	0.148	0.098	**0.003**
Prostate_Tumor	A_Size	**16**	20	21	21	25
	A_Error	0.226	0.194	0.161	0.129	**0.013**

5 Conclusions

In general, this paper proposes a modified FS method MOACBFO and receives good performance in comparing with four swarm intelligence algorithms on 11 high-dimensional microarray datasets. MOACBFO effectively improves the classification results of the modified BFO method in a multi-objective optimization perspective. In addition, it achieves the four goals proposed in this paper. MOACBFO has a good convergence effect, better diversity of searching results, higher classification accuracy, and better effect during the multi-objective FS process in the target of classification. However, MOACBFO needs more comparison to become more stable. In the future, we should pay more attention to the application of MOACBFO and its modified version on practical problems.

Acknowledgment. This work is partially supported by The National Natural Science Foundation of China (Grants Nos. 71901152), Natural Science Foundation of Guangdong Province (2020A1515010752), and Natural Science Foundation of Shenzhen University (85303/00000155).

References

1. Guan, B., Zhao, Y., Yin, Y., Li, Y.: A differential evolution based feature combination selection algorithm for high-dimensional data. Inf. Sci. (Ny) **547**, 870–886 (2021)
2. Gangavarapu, T., Patil, N.: A novel filter–wrapper hybrid greedy ensemble approach optimized using the genetic algorithm to reduce the dimensionality of high-dimensional biomedical datasets. Appl. Soft Comput. J. **81**, 105538 (2019)
3. Joloudari, J.H., Saadatfar, H., Dehzangi, A., Shamshirband, S.: Computer-aided decision-making for predicting liver disease using PSO-based optimized SVM with feature selection. Inf. Med. Unlocked **17**, 100255 (2019)
4. Wang, H., Niu, B., Tan, L.: Bacterial colony algorithm with adaptive attribute learning strategy for feature selection in classification of customers for personalized recommendation. Neurocomputing **452**, 747–755 (2021)
5. Jain, D., Singh, V.: Feature selection and classification systems for chronic disease prediction: a review. Egypt. Informat. J. **19**(3), 179–189 (2018)
6. Kalimuthan, C., Arokia Renjit, J.: Review on intrusion detection using feature selection with machine learning techniques. Mater. Today Proc. **33**, 3794–3802 (2020)
7. Khaire, U.M., Dhanalakshmi, R.: Stability of feature selection algorithm: a review. J. King Saud Univ. Comput. Inform. Sci. (2019). https://doi.org/10.1016/j.jksuci.2019.06.012
8. Lee, J., Choi, I.Y., Jun, C.H.: An efficient multivariate feature ranking method for gene selection in high-dimensional microarray data. Expert Syst. Appl. **166**, 113971 (2021)
9. Hosseini, E.S., Moattar, M.H.: Evolutionary feature subsets selection based on interaction information for high dimensional imbalanced data classification. Appl. Soft Comput. J. **82**, 105581 (2019)
10. Aghaeipoor, F., Javidi, M.M.: A hybrid fuzzy feature selection algorithm for high-dimensional regression problems: an mRMR-based framework. Expert Syst. Appl. **162**, 113859 (2020)
11. Karaboga, D., Basturk, B.: On the performance of artificial bee colony (ABC) algorithm. Appl. Soft Comput. J. **8**(1), 687–697 (2008)
12. Okwu, M.O., Tartibu, L.K.: Particle swarm optimisation. In: Metaheuristic Optimization: Nature-Inspired Algorithms Swarm and Computational Intelligence, Theory and Applications. SCI, vol. 927, pp. 5–13. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-61111-8_2
13. Wang, X., Zhang, Y., Sun, X., Wang, Y., Du, C.: Multi-objective feature selection based on artificial bee colony: an acceleration approach with variable sample size. Appl. Soft Comput. J. **88**, 106041 (2020)
14. Zhou, Y., Kang, J., Sam Kwong, X., Wang, Q.Z.: An evolutionary multi-objective optimization framework of discretization-based feature selection for classification. Swarm Evol. Comput. **60**, 100770 (2021)
15. Wei, W., Chen, S., Lin, Q., Ji, J., Chen, J.: A multi-objective immune algorithm for intrusion feature selection. Appl. Soft Comput. J. **95**, 106522 (2020)

16. Santiago, A., Dorronsoro, B., Fraire, H.J., Ruiz, P.: Micro-genetic algorithm with fuzzy selection of operators for multi-Objective optimization: μFAME. Swarm Evol. Comput. **61**, 100818 (2021)
17. Chen, H., Zhang, Q., Luo, J., Xu, Y., Zhang, X.: An enhanced bacterial foraging optimization and its application for training kernel extreme learning machine. Appl. Soft Comput. J. **86**, 105884 (2020)
18. Turanoğlu, B., Akkaya, G.: A new hybrid heuristic algorithm based on bacterial foraging optimization for the dynamic facility layout problem. Expert Syst. Appl. **98**, 93–104 (2018)
19. Kaur, M., Kadam, S.: A novel multi-objective bacteria foraging optimization algorithm (MOBFOA) for multi-objective scheduling. Appl. Soft Comput. J. **66**, 183–195 (2018)
20. Wang, H., Tan, L., Niu, B.: Feature selection for classification of microarray gene expression cancers using bacterial colony optimization with multi-dimensional population. Swarm Evol. Comput. **48**, 172–181 (2019)
21. Passino, K.M.: Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Syst. **22**(3), 52–67 (2002)
22. Niu, B., Yi, W., Tan, L., Geng, S., Wang, H.: A multi-objective feature selection method based on bacterial foraging optimization. Nat. Comput. **20**(1), 63–76 (2019)
23. Majhi, R., Panda, G., Majhi, B., Sahoo, G.: Efficient prediction of stock market indices using adaptive bacterial foraging optimization (ABFO) and BFO based techniques. Expert Syst. Appl. **36**(6), 10097–10104 (2009)
24. Too, J., Abdullah, A.R., Saad, N.M.: A new co-evolution binary particle swarm optimization with multiple inertia weight strategy for feature selection. Informatics **6**(2), 21 (2019)