



An interactive filter-wrapper multi-objective evolutionary algorithm for feature selection

Zhengyi Liu, Bo Chang, Fan Cheng*

Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Information Materials and Intelligent Sensing Laboratory of Anhui Province, School of Computer Science and Technology, Anhui University, Hefei 230601, China

ARTICLE INFO

Keywords:

Feature selection
Multi-objective evolutionary algorithm
Guiding strategy
Repairing strategy
Initialization strategy

ABSTRACT

As an important task in data mining, feature selection can improve the performance of classification by eliminating the redundant or irrelevant features in original data. It is mainly divided into filter method and wrapper method, and each one has its own advantages. To make full use of the advantages of two methods, in this paper, an interactive filter-wrapper multi-objective evolutionary algorithm, named GR-MOEA is proposed, where guiding and repairing strategies are used to select feature subsets with high quality. To be specific, a wrapper population and a filter population are evolved simultaneously in the proposed algorithm. To utilize the merits of two populations, an interactive scheme is designed, which includes a wrapper to filter guiding strategy and a filter to wrapper repairing strategy. The guide strategy is to use the good solutions in the wrapper population to steer the filter population towards a better direction. While in the repairing strategy, some features in the wrapper population are repaired by the useful information in filter population, which can avoid the trapping of local optimum in wrapper population. To further enhance the performance of GR-MOEA, two effective initialization strategies are also developed. Empirical studies are conducted by comparing the proposed algorithm with several state-of-the-art on different datasets, and the experimental results demonstrate the superiority of GR-MOEA over the comparison methods in obtaining the feature subsets with higher qualities.

1. Introduction

In the real-world, many problems (e.g. classification) involve a large number of features, however, not all the features are useful and some of them are irrelevant, redundant or even noisy, which often brings the problem solving with great challenges (such as overfitting, high computational complexity etc.). To tackle the issue, as a data pre-processing technique, feature selection is suggested, whose aim is to reduce the number of features by removing these irrelevant, redundant features [1]. By using feature selection, people can speed up the learning process, simplify the learned model, and increase the performance. Due to the merits above, feature selection has been successfully used in many data mining fields, such as cancer detection [2], text classification [3], face recognition [4], recommendation system [5] and so on.

A variety of methods with promising performance have been suggested, which include exhaustive search [6], greedy search [7], heuristic search [8] and random search [9]. Despite these feature selection methods can achieve the feature subsets with high quality, however, most of them suffer from the trapping in local optimum [10]. To solve the problem, Evolutionary Algorithm (EA) has been adopted in recent years due to its well-known global search ability, for example, Genetic

Algorithm (GA) [11–13], Particle Swarm Optimization (PSO) [14,15], Ant Colony Optimization (ACO) [16–18], Artificial Bee Colony (ABC) [19,20] etc. Empirical studies have shown the competitiveness of these EAs in solving the feature selection problem.

As common in feature selection, the existing Evolutionary Computation (EC) based feature selection algorithms are mainly categorized into three types, which are filter method, wrapper method, and hybrid (combining filter and wrapper) method. The filter methods [21] achieve the feature subset according to the characteristics of the features, and can obtain the result efficiently. The wrapper methods [22,23] select the features using a classifier evaluation. It means they are more time consuming, however, their performance is often better than the filter ones.

To combine the advantages of filter and wrapper methods, the hybrid methods have been suggested recently, where filter and wrapper methods are fused to achieve feature subsets with high quality [24–27]. Despite the experimental results have demonstrated the promising performance of the hybrid methods, most of them combine filter and wrapper methods in a simple way, which do not make full use of the advantages of both methods. For example, in the work of [26] and [28], the filter and wrapper methods are simply integrated in sequence. Similarly, the authors of [24,25,27] utilize one method (e.g. filter) as the

* Corresponding author.

E-mail addresses: liuzywen@ahu.edu.cn (Z. Liu), change9513@163.com (B. Chang), chengfan@mail.ustc.edu.cn (F. Cheng).

main algorithm and the other (e.g wrapper) is used as the aiding one. Different from these works, the motivation of our work is to fully utilize the merits of both two methods, which can obtain the feature subset with higher quality. In this paper, an interactive hybrid Guiding and Repairing based Multi-Objective Evolutionary Algorithm (GR-MOEA) is suggested, where an interactive procedure is introduced between two methods to fuse them effectively.

The main contributions of this paper can be summarized as follows:

- (1) An interactive hybrid method is suggested, where filter and wrapper method are fully integrated to effectively solve the problem of feature selection. To be specific, a wrapper population and a filter population are evolved simultaneously in the proposed method. During the evolution, a wrapper to filter guiding strategy is designed, which uses the good solutions in the wrapper population to steer the filter population towards a better direction. Meanwhile, a filter to wrapper repairing strategy is also designed, where some features in the wrapper population are repaired by the useful information in filter population, which can avoid the trapping of local optimum in wrapper population. Interactive method by guiding and repairing strategies promote the evolution of both populations.
- (2) Based on the suggested interactive hybrid method above, a guiding and repairing based multi-objective evolutionary algorithm, named GR-MOEA is proposed, where Non-dominated Sorting Genetic Algorithm-II (NSGA-II) is adopted as the framework of the wrapper population and filter population. In GR-MOEA, the designed guiding strategy and repairing strategy are utilized to make full use of the advantages of both populations, and obtain the high quality feature subsets. In addition, two initialization strategies are also developed to initialize the wrapper population and filter population, respectively, which can further improve the performance of GR-MOEA.
- (3) The effectiveness of the proposed GR-MOEA are verified on 12 public datasets by comparing with several state-of-the-art (including filter methods, wrapper methods and hybrid methods). The experimental results have demonstrated the superiority of the proposed GR-MOEA over the comparison methods in terms of both classification accuracy and the number of selected features.

The rest of this paper is organized as follows. The second part presents the related work. The third part gives our algorithm. The fourth part gives the experimental results and analysis. The fifth part summarizes the full text, and puts forward the future work.

2. Related work

In recent years, many EC based feature selection algorithms with promising performance have been proposed, which can be largely divided into three categories: the filter based feature selection approaches, the wrapper based feature selection approaches and the hybrid (combining filter and wrapper) based feature selection approaches [10]. In the following, we will review them, respectively.

2.1. Filter approaches

The EC based filter feature selection approaches obtain the feature subset by utilizing statistical characteristics of the data for evaluation, and adopt different EC techniques for the inner optimization [10]. For example, Xue et al. [29] proposed a filter-based feature selection algorithm named (CMDfsmI) using the ideas of Crowding, Mutation and Dominance, where the Mutual Information and entropy in the data are adopted as two optimized objectives. To solve the bi-objective optimization problem, an effective non-dominated sorting based Binary Particle Swarm Optimization (BPSO) is designed. Experimental results have shown the promising performance of the proposed CMDfsmI. Chen et al. [16] suggested an Ant Colony Optimization (ACO) based algorithm using Rough Set for Feature Selection, which is named RSFSACO. In RSFSACO, mutual information based feature significance is used as heuristic

information for ACO, which can search the optimal feature subset. Hancer et al. [30] designed a new fuzzy mutual information based measure to evaluate the relevance of feature subsets. Based on the design measure, a multi-objective ABC method for feature selection in classification is developed. The empirical results have justified the effectiveness of the developed method. Recently, Hancer et al. [31] proposed two Differential Evolution (DE) approaches for filter feature selection based on information theory. In their work, a new filter criterion that combines the concepts of mutual information, ReliefF [32] and Fisher Score [33] is firstly designed. Then based on the designed criterion, a single objective DE feature selection approach as well as a multi-objective DE feature selection approach are presented. The results on different datasets have indicated that two DE approaches can generally provide good performance in terms of the feature subset size and the classification accuracy. More other filter based feature selection algorithms can be found in [30,34,35].

Although these filter based feature selection algorithms can select the features efficiently, their performance is not very competitive, because they obtain feature subsets without evaluating them on the classifier. To this end, some feature selection algorithms use a classifier (e.g. K-Nearest Neighbor (KNN)) to evaluate the quality of the selected features, which are categorized into the wrapper based approaches [1].

2.2. Wrapper approaches

In the past decade, a variety of wrapper based feature selection algorithms by using EC techniques have been suggested. Hamdani et al. [36] proposed a NSGA-II based wrapper algorithm for feature selection, where 1-Nearest Neighbor (1NN) is used to evaluate the feature subset during the evolution. Experimental results on different datasets have shown that compared with filter based feature selection algorithms the proposed algorithm can obtain the feature subsets with higher quality.

In recognizing the superiority of this work, more wrapper based EC feature selection algorithms are suggested. For example, Xue et al. [23] designed a PSO based feature selection algorithm, which is termed as Crowding, Mutation and Dominance Particle Swarm Optimization (CMDPSO). In CMDPSO, the ideas of crowding, mutation and dominance are integrated into PSO, which can search the Pareto front solutions effectively. The empirical results have show that CMDPSO not only outperforms the traditional feature selection algorithms, such as Linear Forward Selection (LFS) [37] and Greedy Stepwise Backward Selection (GSBS) [38], but also has better performance than the wrapper based feature selection algorithms under the frameworks of NSGA-II, the Strength Pareto Evolutionary Algorithm-2 (SPEA-2) and the Pareto Archived Evolution Strategy (PAES). Other wrapper based PSO feature selection algorithms can be found in [39]. Hancer et al. [40] developed a Multi-Objective Artificial Bee Colony based wrapper feature selection algorithm (MOABC). In the developed algorithm, the original ABC is firstly modified by integrating the non-dominated sorting procedure and genetic operators. Then based on the modified ABC, two different implementations are provided, which can achieve the feature subsets with high quality. Zhang et al. [41] proposed a Multi-Objective Feature Selection algorithm using Binary Differential Evolution with self-learning (MOFS-BDE). In MOFS-BDE, the differential evolution is responsible for exploring the search space, while the self-learning strategy is utilized to effectively exploit these potential areas. In addition, three new operators (binary mutation operator, one-bit purifying search operator, non-dominated sorting operator) are also suggested, which can further improve the performance of MOFS-BDE. Wang et al. [42] proposed a Fast Multi-objective Feature Selection algorithm based on ABC (FMABC-FS), where an acceleration approach with variable sample size is embedded to obtain the feature subsets efficiently. Specifically, a selection strategy of representative samples and a ladder-like sample utilization strategy are developed to reduce the size of samples used in the evolutionary process. Based on two strategies, it can select the features with high qualities.

Besides the algorithms mentioned above, there also exist many other effective wrapper based algorithms with different EC techniques, such as ant colony optimization [43,44], whale optimization [45], butterfly optimization [46] and so on.

2.3. Hybrid filter and wrapper approaches

The filter based and wrapper based EC approaches have shown their promising performance in solving the problem of feature selection, and each method has its own advantages. For instance, the filter based feature selection methods often have less computational cost, while the wrapper based ones can achieve higher quality of solution [1]. To this end, recently, many efforts have been made to design the hybrid based approaches, which combine the advantages of filter and wrapper methods to obtain better feature subsets.

A representative work is [24] proposed by Butler et al.. In their work, the authors aim to explore combinations of filter and wrapper methods in PSO for feature selection with the goal of maintaining the classification performance of wrappers while approaching the efficiency of filters. To achieve the goal, two hybrid based PSO algorithms are developed. To be specific, in the first algorithm (named as FastPSO), the majority of the evaluations are based on filters and the wrapper evaluations are performed only when a better solution (in terms of the filter measure) is found. The second algorithm (named as RapidPSO) further reduces the times of wrapper evaluations by only using the wrapper method to evaluate particles whose current P_{best} is greater than the previous P_{best} with a threshold. Experimental results have demonstrated the superiority of two hybrid algorithms over the single filter or wrapper methods. Since then, more hybrid based feature selection approaches are suggested.

Ghosh et al. [25] proposed a wrapper-filter combination of ACO for feature selection, named Wrapper-Filter combination of ACO for Feature Selection (WFACOFS). In WFACOFS, the features are firstly selected by a filter method (based on feature similarity). Then the selected features are further evaluated by a wrapper method (using KNN and multi-layer perceptron classifier). In addition, a memory to keep the best ants and feature dimension-dependent pheromone update has also been designed in WFACOFS. The comparison of results have shown that the proposed algorithm outperformed most of the state-of-the-art algorithms used for feature selection.

Moradi et al. [27] suggested a Hybrid Particle Swarm Optimization Local Search algorithms (HPSO-LS) for feature subset selection by integrating a novel local search strategy. In HPSO-LS, the filter method (e.g. considering features' correlation information) is adopted to design the local search strategy. While the wrapper method (e.g. using KNN) is to evaluate the feature subset. The results demonstrated that HPSO-LS can outperform several well-known filter based and wrapper based algorithms with different EC techniques.

Huang et al. [26] developed a hybrid feature selection algorithm based on Binary State Transition Algorithm (BSTA) and ReliefF [32], which is called ReliefF-BSTA. The developed algorithm contains two phases: the filter phase and the wrapper phase. In the filter phase, ReliefF is used to efficiently reduce the large search space and provide important information on the features. Then in the following wrapper phase, a new nature-inspired algorithm BSTA (with KNN as the classifier) is adopted to obtain the optimal subsets. Similar the work of Huang et al. [26], Moslehi et al. [28] proposed a new Hybrid filter and wrapper method based on Genetic algorithm and Particle swarm optimization for Feature Selection(HGP-FS) in 2020, where Artificial Neural Network (ANN) is used as evaluation classifier. In the first filter phase, some features are selected according to their competency. Then in the second wrapper phase, a hybrid EC approach that combines GA and PSO is designed to get the final feature subsets. Empirical results have verified the effectiveness of the proposed HGP-FS.

Despite that the hybrid based feature selection algorithms have shown their superiority over the traditional filter and wrapper based al-

gorithms, most of them combine filter and wrapper methods in a simple way, which do not make full use of the advantages both methods. To this end, in this paper, an interactive filter-wrapper evolutionary approach termed GR-MOEA is suggested, where filter and wrapper methods interact fully using guiding and repairing strategies during the evolve and achieve the feature subsets with higher quality. In the following, we will describe the suggested GR-MOEA in detail.

3. Proposed method

In this part, we firstly present the framework and the objective functions of our GR-MOEA; then give two important strategies in GR-MOEA: (1) wrapper to filter guiding strategy (from wrapper population to filter population); (2) filter to wrapper repairing strategy (from filter population to wrapper population). With these two strategies, the two populations interact each other, and obtain good feature subsets. Lastly, we present the initial strategies for the wrapper and filter populations, which can further enhance the performance of GR-MOEA.

3.1. The framework of GR-MOEA and objective functions

3.1.1. The framework of GR-MOEA

GR-MOEA consists of wrapper population and filter population, and the whole algorithm includes three stages: initialization stage based on feature diversity and feature importance, evolution stage with guiding and repairing strategies, and the final merging stage. The main framework of GR-MOEA is shown in Fig. 1.

In the initialization stage, wrapper population and filter population are initialized based on feature diversity and feature importance, respectively. With these two initialization strategies, the proposed GR-MOEA can evolve from a good start point.

In the evolution stage, both populations adopt NSGA-II [47] framework to conduct the crossover, mutation and environmental selection. In order to make full use of the advantages of two populations, a wrapper to filter guiding strategy and a filter to wrapper repairing strategy are proposed, whose basic ideas are from the following two facts. Firstly, the wrapper population can obtain the better solutions than filter one due to the usage of the classifier, thus, the proposed guiding strategy is designed to guide the evolution of filter population by using the good quality solutions in wrapper population. Secondly, for the problem that the evolution of wrapper population may fall into local optimum [48], the proposed repairing strategy is developed to repair some local features in the wrapper population by the useful information in filter population. These two strategies make the two populations evolve better with mutual help.

In the merging stage, two populations are combined and evaluated by accuracy and reduction rate of features to take the non-dominated individuals as the final output. The whole procedure of GR-MOEA is shown in Algorithm 1.

3.1.2. Objective functions

The objective functions of the wrapper population adopt accuracy and reduction rate of features which are the popular objective functions used in many wrapper based feature selection methods [23]. Specifically, the two objectives are defined as:

$$WpopF_1(X_i) = Acc(X_i) \quad (1)$$

$$WpopF_2(X_i) = Rec(X_i) = 1 - \sum_{j=1}^n f_{ij}/n \quad (2)$$

where X_i denotes the i th individual (feature subset) in the population, $Acc(X_i)$ represents classification accuracy of the individual, $Rec(X_i)$ represents reduction rate of features in the individual and $f_{ij} = \{0, 1\}$. If the feature F_j is selected in the i th individual X_i , then $f_{ij} = 1$. Otherwise, $f_{ij} = 0$.

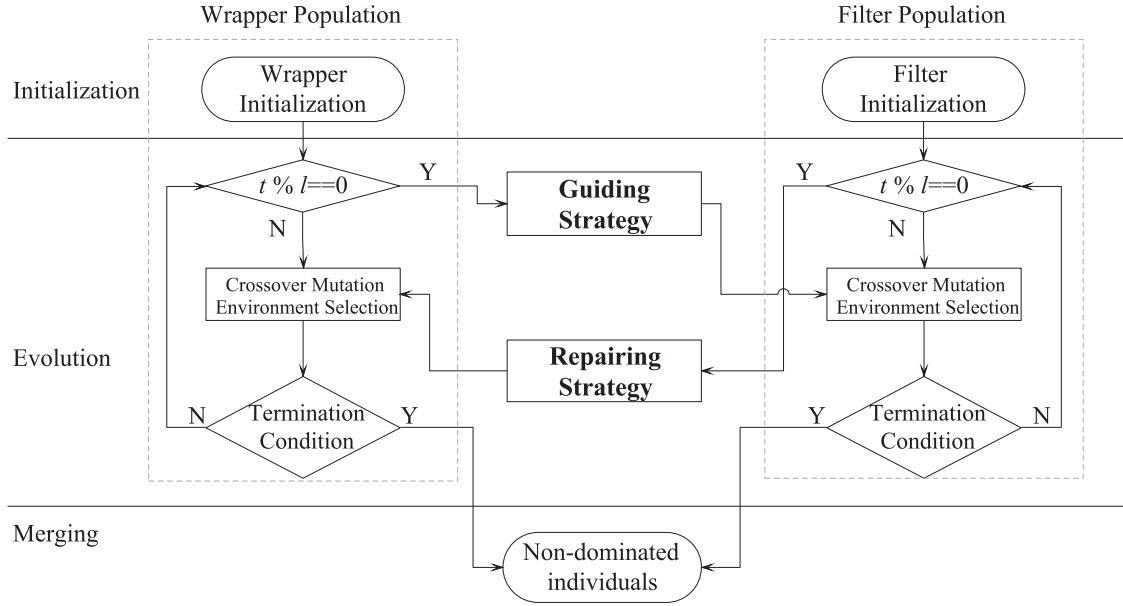


Fig. 1. The main framework of GR-MOEA.

Algorithm 1: GR-MOEA.

Input: D : training set; m : population size;
 $maxgen$: maximum number of generations;
 l : the number of generations between two interactions;
 pc : crossover probability; pm : mutation probability;

Output: $nonDomSet$: non-dominated solutions;

- 1: calculate relevance I and redundancy Red of features from D ;
- 2: set the initial weight $Weight^{(0)}$ of all features to 1;
- 3: /* Initialization stage */;
- 4: $WPop^{(0)} = WrapperInitialization(D, m/2)$;
- 5: $FPop^{(0)} = FilterInitialization(D, m/2)$;
- 6: /* Evolution stage */;
- 7: **for** $t = 0$; $t < maxgen$; $t++$ **do**
- 8: **if** $mod(t, l) == 0$ **then**
- 9: $[FPop^{(t)}, Weight^{(t)}] =$
 Guiding $(FPop^{(t)}, WPop^{(t-l)}, WPop^{(t)}, Weight^{(t-l)})$;
- 10: $[WPop^{(t)}] =$ **Repairing** $(WPop^{(t)}, FPop^{(t)})$;
- 11: **end if**
- 12: $FPop^{(t+1)} = Evolution(FPop^{(t)}, pc, pm)$;
- 13: $WPop^{(t+1)} = Evolution(WPop^{(t)}, pc, pm)$;
- 14: **end for**
- 15: /* Merging stage */;
- 16: $nonDomSet = Select(WPop^{(t)} \cup FPop^{(t)})$;
- 17: **return** $nonDomSet$.

The objective functions of the filter population adopt the relevance and redundancy, which are also widely used objectives in many filter based feature selection methods [29]. Specifically, they are defined as:

$$FpopF_1(X_i) = \sum_{j=1}^n f_{ij} * I(F_j, C); \quad \leftarrow \text{mutua information} \quad (3)$$

$$FpopF_2(X_i) = \sum_{j=1}^n f_{ij} * Red(F_j); \quad \leftarrow \text{redundancy} \quad (4)$$

where $I(F_j, C)$ represents relevance between feature F_j and class label C [31], which is expected to be maximized. $Red(F_j)$ represents average redundancy between feature F_j and other features [27], which is expected to be minimized.

3.2. Interactive evolutionary strategy

The wrapper and filter populations use traditional crossover and mutation operators in NSGA-II to generate the offsprings. In order to further utilize the advantages of both populations, interactive evolutionary strategy is proposed to improve the performance of the proposed method, which includes wrapper to filter guiding strategy and filter to wrapper repairing strategy.

3.2.1. Wrapper to filter guiding strategy

It is well known that compared with the filter method, the wrapper method evaluates the individual by combining a classifier, such as KNN [23,39–41], so the quality of obtained feature subset in wrapper method is often better than that of filter method. Therefore, in our GR-MOEA, we utilize the good solutions in wrapper population to guide the evolution of the filter population, which not only depends on the characteristics of the features itself, but also relies on the importance of the features in the wrapper population.

Specifically, the suggested guiding strategy is based on the following intuition. During the evolution, if the wrapper population is becoming better and the occurrence frequency of feature F_j in the whole wrapper population is increasing, or wrapper population is getting the worse and the occurrence frequency of feature F_j in the whole wrapper population is reducing, this implies that feature F_j has important contribution to current wrapper population. Therefore, the feature F_j should also has a large probability of be selected in the filter population, which can guide the evolution of filter population to a better direction, vice versa. Based on the intuition above, the useful feature information in wrapper population mainly includes two parts. One is the current performance of wrapper population, and the other is the occurrence frequency change of feature F_j in the current wrapper population.

To this end, a weight factor $Weight^{(t)}(F_j)$ is designed to evaluate the importance of feature F_j in the t^{th} generation of wrapper population, which is defined as:

$$Weight^{(t)}(F_j) = Weight^{(t-l)}(F_j) * WPop_{perf}^{(t)} * Feature_{occ}^{(t)}(F_j) \quad (5)$$

where $WPop_{perf}^{(t)}$ represents the performance of wrapper population from the $(t-l)^{th}$ to t^{th} generation, which is measured by the change of average accuracy and average reduction rate. The $Feature_{occ}^{(t)}(F_j)$ represents the occurrence frequency change of feature F_j from the $(t-l)^{th}$ to

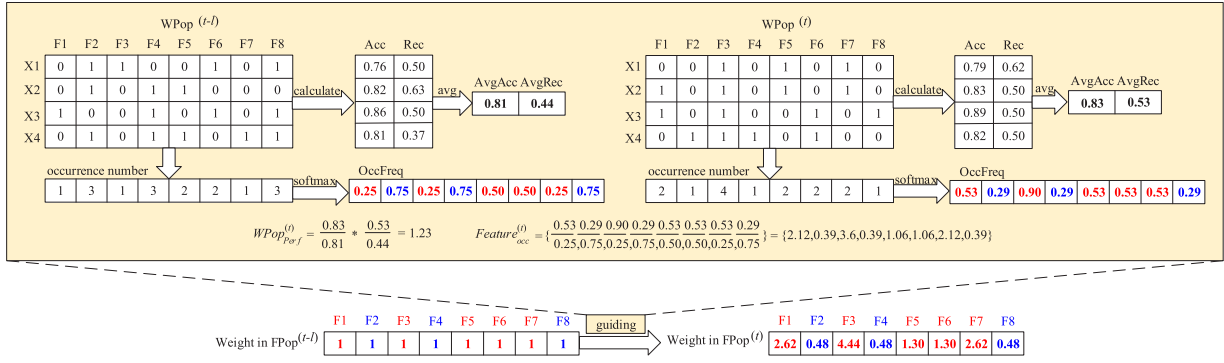


Fig. 2. An example of calculating the weight factor $Weight^{(t)}(F_j)$.

t^{th} generation in the whole wrapper population. They are defined as:

$$WPop_{perf}^{(t)} = \frac{AvgAcc^{(t)}}{AvgAcc^{(t-1)}} * \frac{AvgRec^{(t)}}{AvgRec^{(t-1)}} \quad (6)$$

$$Feature_{occ}^{(t)}(F_j) = \frac{OccFreq^{(t)}(F_j)}{OccFreq^{(t-1)}(F_j)}; \quad (7)$$

Fig. 2 shows an example of the calculation of the weight factor $Weight^{(t)}(F_j)$. In the figure, there are total eight features (F_1, \dots, F_8), and four individuals (X_1, \dots, X_4) in the wrapper population. The top left and top right give two generations of wrapper population $WPop^{(t-1)}$ and $WPop^{(t)}$, respectively. From each generation of wrapper population, we can calculate the accuracy Acc and the reduction rate Rec of each individuals in the wrapper population, and further average them to get the average accuracy $AvgAcc$ and average reduction rate $AvgRec$. In addition, we can also count the occurrence number of each feature in a generation, and further normalize them to get occurrence frequency $OccFreq$ by softmax function [49]. Next, based on formulas 6 and 7, we can get the performance of the wrapper population $WPop_{perf}^{(t)} = 1.23$ and the occurrence frequency changes of all eight features $Feature_{occ}^{(t)} = \{2.12, 0.39, 3.6, 0.39, 1.06, 1.06, 2.12, 0.39\}$.

From above computation, we can find that the performance of the wrapper population $WPop_{perf}^{(t)}$ is greater than 1. It means the performance of the wrapper population is becoming the better. At the same time, if the occurrence frequency of feature F_j is increasing too, that is to say, the occurrence frequency change of the feature F_j is greater than 1, it means that the performance of the wrapper population is getting better with the occurrence frequency of feature F_j increasing, so the weight of this feature should be increased, vice versa. For example, the occurrence frequency change of the features F_1, F_3, F_5, F_6, F_7 is greater than 1. According to formula 5, their corresponding weight factors $Weight^{(t)}(F_j)$ are set as $\{1.23 * 2.12, 1.23 * 3.6, 1.23 * 1.06, 1.23 * 1.06, 1.23 * 2.12\}$ (by assuming $Weight^{(t-1)}(F_j) = 1$), which equals to $\{2.62, 4.44, 1.30, 1.30, 2.62\}$. In the bottom of Fig. 2, they are marked in red color. Similarly, the occurrence frequency change of the features F_2, F_4, F_8 is less than 1, then the weight factors of these features are reduced to $\{0.48, 0.48, 0.48\}$ based on the formula 5. It can be seen in the bottom of Fig. 2 which are marked in blue color.

The weight factor $Weight^{(t)}(F_j)$ achieved from wrapper population is further used as a cost coefficient to modify the two objectives of feature F_j in filter population. Due to the fact that the first objective $I(F_j, C)$ in filter population is to measure the relevance of feature F_j , the larger value of $I(F_j, C)$ is better. Thus the first objective is modified as:

$$I_w^{(t)}(F_j, C) = \begin{cases} I(F_j, C) * Weight^{(t)}(F_j), & WPop_{perf}^{(t)} \geq 1 \\ I(F_j, C) / Weight^{(t)}(F_j), & WPop_{perf}^{(t)} < 1 \end{cases} \quad (8)$$

when $WPop_{perf}^{(t)} < 1$, that is to say, the wrapper population is becoming the worse, then the influence should be contrary.

While for the second objective $Red(F_j)$ is suggested to evaluate the redundancy of feature F_j in filter population, and the smaller value is better. Thus, the second objective is modified as:

$$Red_w^{(t)}(F_j) = \begin{cases} Red(F_j) / Weight^{(t)}(F_j), & WPop_{perf}^{(t)} \geq 1 \\ Red(F_j) * Weight^{(t)}(F_j), & WPop_{perf}^{(t)} < 1 \end{cases} \quad (9)$$

By using the objective functions guided by the wrapper population, the filter population can be steered for a better direction, and obtain the solutions with higher quality. Algorithm 2 shows the whole procedure

Algorithm 2: Wrapper to filter Guiding Strategy.

Input: $FPop^{(t)}$: Filter population of t generation;

$WPop^{(t-1)}$: Wrapper population of $t-1$ generation;

$WPop^{(t)}$: Wrapper population of t generation;

$Weight^{(t-1)}$: Weight of $t-1$ generation;

Output: $[FPop^{(t)}, Weight^{(t)}]$;

1: **for** $j = 1; j \leq |D|; j++$ **do**

2: Calculate $Weight^{(t)}(F_j)$ using Eq. 5, Eq. 6 and Eq. 7;

3: **if** $WPop_{perf}^{(t)} \geq 1$ **then**

4: $I_w^{(t)}(F_j) = I(F_j, C) * Weight^{(t)}(F_j)$;

5: $Red_w^{(t)}(F_j) = Red(F_j) / Weight^{(t)}(F_j)$;

6: **else**

7: $I_w^{(t)}(F_j) = I(F_j, C) / Weight^{(t)}(F_j)$;

8: $Red_w^{(t)}(F_j) = Red(F_j) * Weight^{(t)}(F_j)$;

9: **end if**

10: **end for**

11: recalculate $FPopF_1$ and $FPopF_2$ in $FPop^{(t)}$ using $I_w^{(t)}$ and $Red_w^{(t)}$;

12: **return** $[FPop^{(t)}, Weight^{(t)}]$;

of the wrapper to filter guiding strategy.

3.2.2. Filter to wrapper repairing strategy

As mentioned before, the wrapper methods can often generate the better performance than the filter ones due to the individual evaluation based on the classifier. However, there may exist that some features are selected or not selected by all individuals during the wrapper population evolution [48], which makes the population trap into local optimum.

To tackle the issue, a filter to wrapper repairing strategy is suggested, where the all-one or all-zero features in the wrapper population are repaired by the corresponding features in the filter population. Therefore, the local optimum mentioned above can be avoided, and the performance of wrapper population is further enhanced.

To be specific, the repairing strategy consists of two stages. The first stage is to select the features in wrapper population, which will be repaired. The second stage is to utilize the filter population to perform the repairing process.

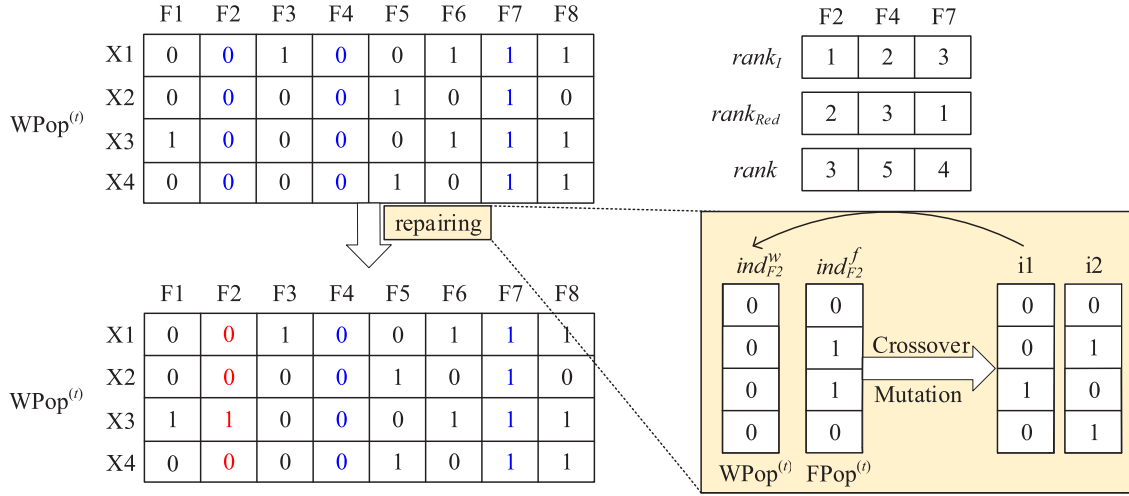


Fig. 3. An example of filter to wrapper repairing strategy.

We assume the set of all-one or all-zero features in the wrapper population denoted as $Wset_{01}$. In the first stage, we use filter population's objectives to evaluate the features in $Wset_{01}$. Specifically, for each feature F_i in $Wset_{01}$, we calculate its values of $I(F_i, C)$ and $Red(F_i, C)$, and sort all the elements in $Wset_{01}$ descend according to $I(F_i, C)$ and $Red(F_i, C)$, respectively. Then for each F_i , we get its rank number $rank_I(F_i)$ and $rank_{Red}(F_i)$. Thus, the total rank of feature F_i is $rank(F_i) = rank_I(F_i) + rank_{Red}(F_i)$. To make the repairing of filter population more accurate, only the features in $Wset_{01}$ with the top K values of $rank(F_i)$ are selected for future repairing, and $K = \lceil \frac{l}{maxgen} * |Wset_{01}| \rceil$, where $|Wset_{01}|$ is the number of features in set $Wset_{01}$, and $\lceil \cdot \rceil$ is an upper bound function.

After the top K features are selected, the suggested repairing strategy is then applied. To be specific, the feature F_i to be repaired in wrapper population and the corresponding one in filter population are viewed as two individuals ($ind_{F_i}^w$ and $ind_{F_i}^f$), whose length equals to the number of individuals in wrapper or filter population. Then the traditional crossover and mutation operators in GA are carried on $ind_{F_i}^w$ and $ind_{F_i}^f$ to generate two offsprings, one of which is randomly selected to replace the feature F_i in wrapper population. We note that to avoid the possible over-repairing, each feature in the wrapper population is repaired only once.

Fig. 3 shows an example of the proposed repairing strategy. From the figure, we can find that in the t th generation of the wrapper population $WPop^{(t)}$ three features F_2, F_4, F_7 are all-one or all-zero features. It means set $Wset_{01} = \{F_2, F_4, F_7\}$ and $|Wset_{01}| = 3$. In the first stage, we need to select the candidate features to be repaired in $Wset_{01}$. To be specific, these three features are ranked based on the relevance I and redundancy Red to generate rank score $rank_I(F_2) = 1, rank_I(F_4) = 2, rank_I(F_7) = 3$ and $rank_{Red}(F_2) = 2, rank_{Red}(F_4) = 3, rank_{Red}(F_7) = 1$, and then these two rank scores are summed to generate total rank score $rank(F_2) = 3, rank(F_4) = 5, rank(F_7) = 4$. Next, the top K features are selected to be repaired. In the example, let $l = 10$ and $maxgen = 100$, then $K = \lceil \frac{10}{100} * 3 \rceil = 1$, which indicates that feature F_2 is selected to be repaired. In the second stage, repairing strategy is applied between feature F_2 in wrapper population and filter population. Specifically, crossover and mutation operation are applied between $ind_{F_2}^w$ and $ind_{F_2}^f$ to generate two offsprings $i1$ and $i2$. Then the feature F_2 in the wrapper population is repaired by one of two offsprings $i1$ and $i2$ in the random manner, for example, feature F_2 is repaired by the offspring $i1$.

By using the suggested repairing strategy, the wrapper population can avoid the local optimum, and obtain the feature subsets with higher quality. Algorithm 3 shows the procedure of the filter to wrapper repairing strategy in detail.

Algorithm 3: Filter to wrapper Repairing Strategy.

Input: $WPop^{(t)}$: Wrapper population of t generation ;
 $FPop^{(t)}$: Filter population of t generation;
Output: $WPop^{(t)}$;

- 1: put the feature which should be repaired in $WPop^{(t)}$ into $Wset_{01}$;
- 2: select the top K features from $Wset_{01}$ based on $rank_I$ and $rank_{Red}$;
- 3: put the indexes of the top K features in $WPop^{(t)}$ into $idxs$;
- 4: **for** $i = 1; i \leq K; i++$ **do**
- 5: $idx = idxs(i)$ // idx is the idx^{th} column of feature in the wrapper population or the filter population;
- 6: $[o1, o2] = \text{Crossover}(ind_{F_{idx}}^w, ind_{F_{idx}}^f)$;
// $ind_{F_{idx}}^w$ and $ind_{F_{idx}}^f$ are the idx^{th} column of feature in $WPop^{(t)}$ and $FPop^{(t)}$;
- 7: $[i1, i2] = \text{Mutation}(o1, o2)$;
- 8: $x = \text{random}(i1, i2)$ // randomly select one from $i1$ and $i2$.
- 9: put x into the idx^{th} column of $WPop^{(t)}$;
- 10: **end for**
- 11: recalculate $WpopF_1$ and $WpopF_2$ in $WPop^{(t)}$;
- 12: **return** $WPop^{(t)}$;

3.3. Initialization strategy

In order to further improve the performance, we propose two initialization strategies for wrapper population and filter population in GR-MOEA, respectively.

3.3.1. Wrapper population initialization based on feature diversity

The initialization strategy of the wrapper population focuses on improving the diversity of feature combinations, which can help the wrapper population search the optimal solution more effectively.

Suppose there are n features in the dataset and m individuals in the population. Then n features are divided into m groups. The number of the features in the first $m-1$ groups are all $\lfloor \frac{n}{m} \rfloor$, and the remained features are in the m th group. If $\lfloor \frac{n}{m} \rfloor \geq 2$, only one feature is selected randomly from each group for any individual in the wrapper population. If $\lfloor \frac{n}{m} \rfloor < 2$, the individual is initialized randomly.

Fig. 4 shows an example. Suppose there are 8 features in the dataset and 4 individuals in the wrapper population. Then these features are divided into 4 groups and only one feature in each group is selected and set as 1 for each individual.

The detail of initialization strategy in the wrapper population is shown in Algorithm 4.

	F1	F2	F3	F4	F5	F6	F7	F8
X1	0	1	1	0	0	1	0	1
X2	0	1	0	1	1	0	1	0
X3	1	0	0	1	0	1	1	0
X4	0	1	0	1	1	0	0	1
	Group1		Group2		Group3		Group4	

Initialization Matrix of Wrapper Population

Fig. 4. An example of wrapper population initialization.

Algorithm 4: Wrapper population initialization based on feature diversity.

Input: D : training set; m : population size;
Output: $WPop^{(0)}$;

```

1:  $n = |D|$ ; //the number of features.
2:  $WPop^{(0)} = \text{Zeros}(m, n)$ ;
3: for  $i = 1; i \leq m; i++$  do
4:   if  $\lfloor \frac{n}{m} \rfloor < 2$  then
5:     for  $j = 1; j \leq n; j++$  do
6:       if  $\text{Rand}(1) < 0.5$  then
7:          $WPop^{(0)}(i, j) = 1$ ;
8:       end if
9:     end for
10:  else
11:    divide individual  $X_i$  into  $m$  groups on average;
12:    select one feature from each group randomly;
13:    put the indexes of the selected features into  $idxs$ ;
14:    for  $j = 1; j \leq m; j++$  do
15:       $WPop^{(0)}(i, idxs(j)) = 1$ ;
16:    end for
17:  end if
18: end for
19: calculate  $WpopF_1$  and  $WpopF_2$  for  $WPop^{(0)}$ ;
20: return  $WPop^{(0)}$ ;
```

3.3.2. Filter population initialization based on feature importance

The initialization strategy of the filter population focuses on the importance of the features, which can help filter population choose the most important features from the beginning toward a good evolution direction. The more important the feature is, the higher the probability of being initialized to 1. We use the relevance and the redundancy to evaluate the importance of feature. To be specific, the larger the relevance is and the smaller the redundancy is, the more important the feature is. Therefore, for each feature its importance is calculated by following formula:

$$Pro(F_j) = \frac{I(F_j, C)}{Q * Red(F_j)} \quad (10)$$

where $I(F_j, C)$ represents relevance between feature F_j and class label C , and $Red(F_j)$ is the average redundancy between F_j and other features. To ensure the sparsity of the individual, a parameter $Q \geq 1$ is introduced. In the following Fig. 5, we will present an illustrative example to show how the individual in the filter population is initialized by using the formula 10.

Suppose there are 8 features in the dataset and 4 individuals in the filter population. Firstly, the probability vector $Pro = \{Pro(F_j) | j = 1, \dots, 8\}$ is computed by Eq. (10) (assuming $Q = 20$), which is shown in the last line of the top table of Fig. 5. Secondly, a random matrix $A \in \mathbb{R}^{4 \times 8}$ is generated, and each element $A_{ij} \in [0, 1]$, which is shown

in the left bottom table of Fig. 5. Thirdly, the initializing of individuals (X_1, \dots, X_4) in filter population can be described as following. For each bit f_{ij} in individual $X_i (i \in \{1, \dots, 4\})$ (corresponding to the feature $F_j (j \in \{1, \dots, 8\})$), if $Pro(F_j) > A_{ij}$, then $f_{ij} = 1$. Otherwise $f_{ij} = 0$. We take individual X_1 as an example, since $Pro(F_7) > A_{17}$, thus, $f_{17} = 1$. While for another feature F_k in X_1 , $Pro(F_k) < A_{1k}$, their corresponding $f_{1k} = 0$. Thus, the individual X_1 is initialized as (0,0,0,0,0,0,1,0). Likewise, other individuals are initialized, which are shown in the right bottom table of Fig. 5.

The detail of initialization strategy in the filter population is shown in Algorithm 5.

Algorithm 5: Filter population initialization based on feature importance.

Input: D : training set, m : population size;
Output: $FPop^{(0)}$;

```

1:  $n = |D|$ ; //the number of features
2: for  $j = 1; j \leq n; j++$  do
3:    $Pro(F_j) = \frac{I(F_j, C)}{Q * Red(F_j)}$ ;
4: end for
5:  $A = \text{Rand}(m, n)$ ;
6:  $FPop^{(0)} = \text{Zeros}(m, n)$ ;
7: for  $i = 1; i \leq m; i++$  do
8:   for  $k = 1; k \leq n; k++$  do
9:     if  $A(i, k) \leq Pro(F_k)$  then
10:       $FPop^{(0)}(i, k) = 1$ ;
11:    end if
12:   end for
13: end for
14: calculate  $FPopF_1$  and  $FPopF_2$  use  $I$  and  $Red$  for  $FPop^{(0)}$ ;
15: return  $FPop^{(0)}$ ;
```

4. Experimental results

In this part, we compare the proposed GR-MOEA with seven representative evolutionary algorithms (three hybrid methods, one filter method and three wrapper methods) for feature selection. To be specific, we firstly present the experimental setting including comparison algorithms, benchmark datasets and the adopted parameter settings, then report and analyze the experimental results in detail. Furthermore, we test the effectiveness of hybrid mode compared with wrapper or filter stand-alone mode, validate the effectiveness of the suggested strategies in GR-MOEA, study the influence of different classifier, discuss the sensitivity of the interaction parameter in the proposed algorithm and give running time analysis in the last part.

4.1. Experimental settings**4.1.1. Comparison algorithms**

The performance of our GR-MOEA is compared with seven representative feature selection evolutionary algorithms: FastPSO [24], RapidPSO [24], HPSO-LS [27], CMDfMI [29], Bin-MOABC [40], MOFS-BDE [41] and FMABC-FS [42]. The former three (FastPSO, RapidPSO and HPSO-LS) are all hybrid based algorithms that are same as our GR-MOEA. The FastPSO and RapidPSO are from [24], where a representative work of hybrid based evolutionary method for feature selection. In [24], the combination of filter and wrapper methods are explored in PSO with the aim of maintaining the classification performance of wrappers while approaching the efficiency of filters. The main difference between FastPSO and RapidPSO is that RapidPSO adopts less numbers of wrapper evaluations. HPSO-LS is another effective hybrid based algorithm, where the filter method is adopted to design the local search strategy, and the wrapper method is used to evaluate the feature subsets. To make the

	F1	F2	F3	F4	F5	F6	F7	F8
<i>I</i>	0.90	0.80	0.45	0.62	0.92	0.78	0.92	0.84
<i>Red</i>	0.15	0.13	0.85	0.66	0.11	0.46	0.10	0.19
<i>Pro</i>	0.30	0.31	0.03	0.05	0.42	0.08	0.46	0.22

	F1	F2	F3	F4	F5	F6	F7	F8
X1	0.42	0.53	0.27	0.38	0.44	0.31	0.45	0.72
X2	0.27	0.43	0.12	0.31	0.95	0.63	0.21	0.33
X3	0.63	0.31	0.47	0.91	0.28	0.07	0.31	0.35
X4	0.41	0.67	0.83	0.57	0.42	0.89	0.68	0.11

Random Matrix *A*

	F1	F2	F3	F4	F5	F6	F7	F8
	0	0	0	0	0	0	1	0
	1	0	0	0	0	0	1	0
	0	1	0	0	1	1	1	0
	0	0	0	0	1	0	0	1

Initialization Matrix of Filter Population

Fig. 5. An illustrative example for the initialization of the filter population.

Table 1
The detailed characteristics of 12 experimental datasets.

Data Set	#Features	#Classes	#Instances
Wine	13	3	178
Heart	13	2	270
Hepatitis	19	2	155
WDBC	30	2	569
Ionosphere	34	2	351
SpamBase	57	2	4601
Hillvalley	100	2	606
Musk1	166	2	476
LSVT	309	2	126
Isolet	617	26	1559
Colon	2000	2	62
Leukemia	7129	2	72

experiments more convincing, a filter based CMDfsMI and three wrapper based Bin-MOABC, MOFS-BDE and FMABC-FS are also adopted as the comparison algorithms. The CMDfsMI is a filter based feature selection algorithm with promising performance, where mutual information and entropy are adopted as two optimized objectives. The Bin-MOABC is a wrapper algorithm, where the original artificial bee colony is modified by integrating the non-dominated sorting procedure and genetic operators. MOFS-BDE is a binary differential evolution algorithm with a self-learning strategy, where three proposed operators are embedded to solve the problem of multi-objective feature selection. FMABC-FS is a multi-objective evolutionary feature selection algorithm, which embeds an improved artificial bee colony algorithm based on the particle update model.

4.1.2. Datasets

To examine the performance of different algorithms, experiments are conducted on 12 public datasets available from UCI machine learning repository [50], which are popularly used in feature selection task. The detailed characteristics of these datasets are depicted in Table 1, in which ‘‘#Features’’, ‘‘#Classes’’ and ‘‘#Instances’’ denote the number of features, instances and the classes, respectively.

For fair comparison, the instances in each dataset are randomly divided into 70% as the training set and 30% as the test set for all the comparison algorithms. All the hybrid algorithms and the wrapper algorithms adopt KNN as the evaluation classifier since that KNN is a common used classification tool and frequently used in different feature selection tasks [23,39–41]. To simplify the calculation, in the following experiments, K is set to 5, and during the training, all the algorithms which adopt wrapper method use 10-fold cross validation to avoid feature selection bias. To further reduce the variance, 30 independent runs

are executed on each dataset, and we report the experimental results averaged on 30 runs.

4.1.3. Parameter settings

All the comparison algorithms use the recommended parameter values suggested in their original papers, except for FastPSO [24], RapidPSO [24], whose original train/test partitions are 2/3 and 1/3, and close to our partitions (70% and 30%). Table 2 lists the details of the parameters in comparison algorithms and our GR-MOEA. For fair comparison, in all these evolutionary algorithms, the population size m is set as 40 and the maximum number of iterations $maxgen$ is fixed as 100. There are two populations (filter and wrapper) in our GR-MOEA. The size of each population is 20 and the number of iterations of each population is 100. The filter population and the wrapper population interact with each other every $l = 10$ generations. The crossover operation uses multi-point crossover, and the crossover probability is 0.9. The mutation operation uses single-point mutation, and the mutation probability is $1/featurenumber$.

All the experiments are conducted on a PC with an Intel Core i5-2300 CPU, 3.0GHz, 4GB memory, and the Windows 10 64 bit operating system. The source code of our GR-MOEA is available by requesting to the authors through their emails.

4.2. Performance evaluation

In this section, the proposed GR-MOEA is compared with seven state-of-the-art algorithms on 12 datasets.

4.2.1. GR-MOEA vs hybrid algorithms

Since the proposed algorithm is a hybrid method, thus, in the first part of experiments we report the comparison results between the proposed GR-MOEA and three hybrid feature selection algorithms (i.e. FastPSO, RapidPSO and HPSO-LS). It should be noted that these three hybrid algorithms are all single-objective methods, and GR-MOEA is a multi-objective method. Therefore, the solutions with the highest classification accuracy in their final populations are chosen to compare.

Table 3 shows the comparison results of four hybrid feature selection algorithms in terms of accuracy and the number of selected features. Note that in the table the highest average classification accuracy and the smallest average size of feature subsets obtained on each dataset are marked in bold.

It can be observed from the table that GR-MOEA obtains the highest classification accuracy in 10 of 12 datasets and the smallest feature subsets in 11 of 12 datasets. In addition, by taking a closer look at Table 3 we can find that the proposed GR-MOEA shows better performance than the others especially on high-dimensional datasets, such as Isolet, Colon and Leukemia dataset. We take the Leukemia as an example,

Table 2
Parameter settings of different feature selection algorithms.

Algorithm	Parameter values
FastPSO[24],RapidPSO[24]	$vmax = 4, c1 = c2 = 2, w = 0.7298$.
HPSO-LS[27]	$vmax = 4, c1 = c2 = 2$.
CMDfsmi[29]	mutation rate = $1/featurenumber$, w is a random value in $[0.1, 0.5]$, $c1$ and $c2$ are random values in $[1.5, 2.0]$.
Bin-MOABC[40]	$Limit = 50, T = 10000$.
MOFS-BDE[41]	$scale\ factor F = 0.5 * random, CR = 0.3, T_{loc} = 5, \sigma = 0.01$.
FMABC-FS[42]	$T_1 = 50, L = 5$.
GR-MOEA	$l=10$, crossover probability=0.9, mutation rate= $1/featurenumber$, $Q=20$.

Table 3
Comparison results of different hybrid algorithms on the experimental sets.

Data Set	Accuracy(Std)				Feature number (Std)			
	FastPSO[24]	RapidPSO[24]	HPSO-LS[27]	GR-MOEA	FastPSO[24]	RapidPSO[24]	HPSO-LS[27]	GR-MOEA
Wine	97.45(1.54)	97.45(2.43)	96.17(1.39)	96.23(2.19)	7.42(1.51)	9.61(1.71)	5.43(0.51)	5.65(1.38)
Heart	83.62(2.85)	83.43(3.47)	81.46(1.28)	83.70(4.05)	7.13(1.60)	6.18(1.10)	4.00(0.82)	5.72(3.26)
Hepatitis	83.63(2.29)	82.59(4.35)	83.90(6.24)	84.43(3.63)	6.57(1.42)	7.26(1.48)	6.67(0.65)	3.70(1.08)
WDBC	97.37(1.04)	97.32(0.97)	96.65(1.70)	96.88(1.06)	18.43(2.12)	19.97(2.18)	12.13(0.51)	7.19(4.36)
Ionosphere	85.63(3.51)	82.60(3.54)	88.11(3.98)	90.60(2.21)	11.38(1.57)	17.05(3.51)	10.20(3.48)	3.61(1.97)
SpamBase	90.51(0.82)	88.93(1.48)	81.23(3.23)	91.45(0.72)	21.87(3.12)	23.98(3.86)	21.50(0.52)	15.15(2.62)
Hillvalley	51.35(2.28)	54.52(2.06)	57.73(3.27)	58.53(1.78)	65.34(8.18)	64.55(7.02)	10.27(3.59)	9.79(4.63)
Musk1	79.81(1.50)	80.06(2.36)	83.95(2.85)	84.69(2.47)	93.52(9.32)	95.28(13.35)	59.20(1.49)	16.68(5.47)
LSVT	83.92(1.82)	82.67(4.61)	78.19(3.52)	84.87(4.15)	96.61(17.18)	110.56(10.88)	106.40(2.06)	9.04(3.88)
Isolet	80.97(1.82)	77.09(1.25)	79.78(1.58)	82.76(2.11)	318.17(9.67)	330.67(16.51)	229.14(25.86)	39.37(5.12)
Colon	70.78(8.35)	71.26(10.47)	77.38(4.09)	79.32(8.05)	491.51(136.64)	519.52(125.18)	780.55(101.55)	21.63(9.29)
Leukemia	79.43(5.00)	75.89(6.79)	89.28(3.36)	90.34(5.69)	1772.21(376.61)	1741.46(214.01)	398.53(127.43)	101.47(32.36)
win/tie/loss	10/0/2	10/0/2	12/0/0		12/0/0	12/0/0	11/0/1	

Table 4
Comparison results between GR-MOEA and four non-hybrid algorithms on the experimental sets.

Data Set	Accuracy(Std)					Feature number(Std)				
	CMDfsmi[29]	Bin-MOABC[40]	MOFS-BDE[41]	FMABC-FS[42]	GR-MOEA	CMDfsmi[29]	Bin-MOABC[40]	MOFS-BDE[41]	FMABC-FS[42]	GR-MOEA
Wine	96.73(1.97)	93.67(2.35)	95.16(3.45)	94.65(3.87)	96.23(2.19)	5.31(1.33)	3.28(1.06)	5.07(1.08)	5.27(1.26)	5.65(1.38)
Heart	82.76(3.38)	82.64(3.81)	81.89(3.57)	82.76(4.47)	83.70(4.05)	5.76(2.09)	3.51(2.52)	4.83(1.34)	4.03(1.09)	5.72(3.26)
Hepatitis	82.16(3.84)	82.72(5.13)	81.09(4.61)	82.32(5.00)	84.43(3.63)	6.81(2.75)	3.78(1.42)	5.90(1.71)	4.67(1.49)	3.70(1.08)
WDBC	96.84(0.96)	94.01(1.41)	95.89(1.16)	95.93(1.66)	96.88(1.06)	13.73(4.63)	3.23(1.26)	9.20(3.01)	8.70(3.34)	7.19(4.36)
Ionosphere	87.17(3.03)	89.68(2.19)	89.08(3.38)	88.48(3.67)	90.60(2.21)	12.67(0.94)	3.75(0.86)	4.03(0.89)	4.13(1.31)	3.61(1.97)
SpamBase	89.26(0.77)	89.33(2.94)	91.16(0.70)	91.89(0.81)	91.45(0.72)	25.20(5.72)	15.61(3.38)	14.67(2.88)	13.83(2.28)	15.15(2.62)
Hillvalley	55.45(2.04)	56.16(2.82)	57.31(2.71)	56.36(3.11)	58.53(1.78)	38.42(9.18)	6.42(2.27)	12.73(5.57)	15.97(4.92)	9.79(4.63)
Musk1	82.48(1.44)	84.29(2.39)	84.26(3.54)	84.17(3.67)	84.69(2.47)	93.14(6.73)	16.79(4.05)	35.20(7.86)	38.37(6.99)	16.68(5.47)
LSVT	84.65(6.52)	76.38(6.28)	81.32(5.84)	79.47(6.53)	84.87(4.15)	129.78(11.18)	3.68(1.42)	61.30(16.62)	79.13(14.59)	9.04(3.88)
Isolet	81.35(1.88)	84.05(1.93)	82.10(1.60)	83.88(1.53)	82.76(2.11)	281.90(15.73)	93.85(18.01)	164.40(11.80)	200.07(12.82)	39.37(5.12)
Colon	76.84(6.67)	78.10(1.04)	77.54(9.16)	70.88(11.54)	79.32(8.05)	934.70(27.60)	616.81(48.46)	801.40(58.19)	819.06(43.61)	21.63(9.29)
Leukemia	84.70(8.05)	90.05(6.23)	83.84(6.47)	84.27(7.49)	90.34(5.69)					
win/tie/loss	11/0/1	11/0/1	11/0/1	10/0/2		3472.50(48.47) 12/0/0	3026.43(76.22) 7/0/5	3009.10(164.61) 9/0/3	3221.58(65.42) 9/0/3	101.47(32.36)

which has 7129 features. The GR-MOEA can achieve accuracy value of 0.9034 with only choosing about one hundred features. The statistics in Table 3 has demonstrated the superiority of GR-MOEA over the three existing hybrid feature selection algorithms.

4.2.2. GR-MOEA vs non-hybrid algorithms

In the second part of experiments, we compare our GR-MOEA with four non-hybrid MOEA feature selection algorithms, which are the filter based CMDfsmi, the wrapper based Bin-MOABC, MOFS-BDE and FMABC-FS. Table 4 presents the comparison results of five MOEAs in terms of accuracy and the number of selected features. From the table we can find that no matter measured by accuracy or the size of the selected feature subset, the proposed GR-MOEA outperforms other four algorithms on most of the datasets, which demonstrates the superiority of GR-MOEA. We note that the super performance of GR-MOEA is attributed to the fact that it can achieve better Pareto solutions. To illustrate the fact, Fig. 6 depicts the Pareto solutions of five MOEAs on 12 datasets, from which we can observe that compared with CMDfsmi, Bin-

MOABC, MOFS-BDE and FMABC-FS the proposed GR-MOEA can obtain the better non-dominated solutions. Moreover, it is well-known that the empirical attainment function (EAF) plot [51,52] is a graphical tool to give visual information on the pairwise comparison of two MOEAs. In order to make our comparison more adequate, we also plot the EAFs of different MOEAs, and the results show the competitiveness of GR-MOEA compared with other MOEAs. Due to the space limitation, we present the details in the Supplementary Material.

In addition, for all the MOEAs, we also give the hypervolume indicator (HV) [53] of different algorithms averaged on all the Pareto fronts, and paired T-test results of these MOEAs over 30 independent runs. Note that the degree of freedom is 29 and the significance level in T-test is 0.05. If a p-value less than 0.05 indicates that the proposed algorithm is significantly better than comparison algorithm, while a p-value larger than 0.05 indicates the proposed algorithm equals to the comparison one.

From Table 5, we can see that HV values of our GR-MOEA are higher than those of comparison algorithms. Meanwhile p-values in T-test are

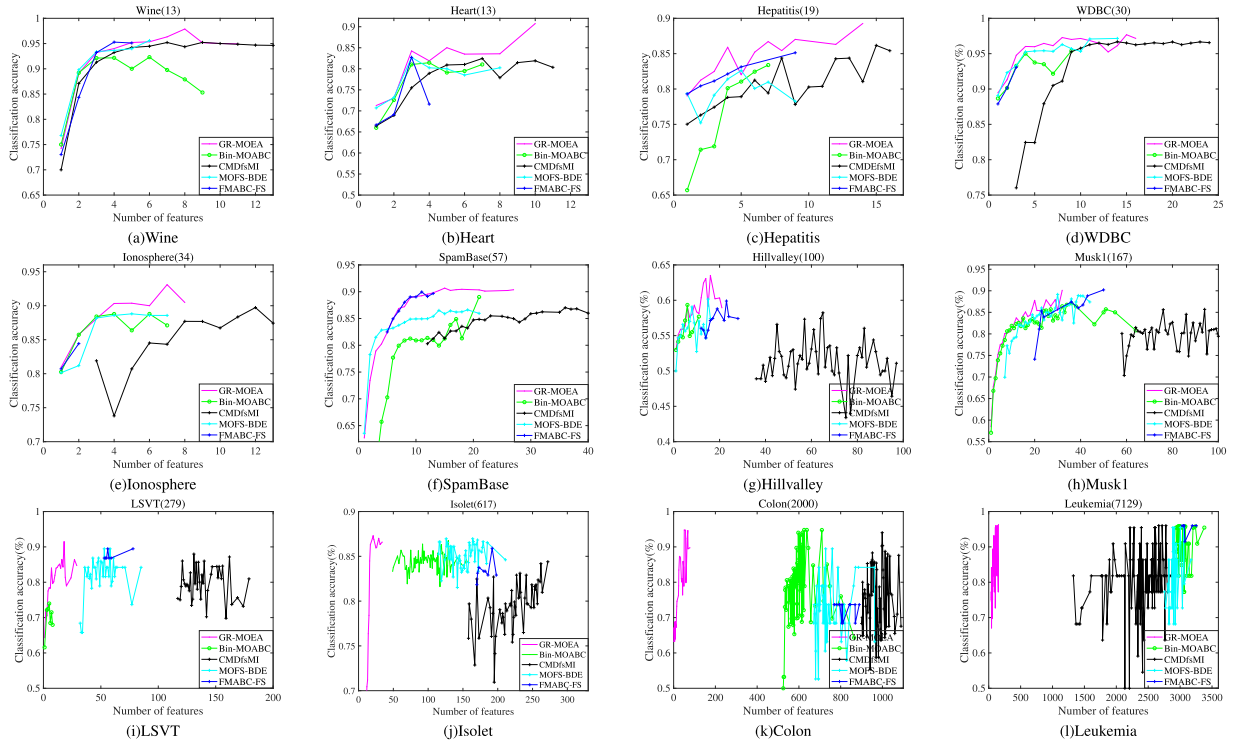


Fig. 6. Average results on all Pareto solutions obtained by CMDfsmI[29], Bin-MOABC[40], MOFS-BDE[41], FMABC-FS[42] and our GR-MOEA.

Table 5

Experimental HV and T-test results of all non-hybrid method and GR-MOEA.

Data Set	HV(Std)					T-test p-value (GR-MOEA vs)			
	CMDfsmI[29]	Bin-MOABC[40]	MOFS-BDE[41]	FMABC-FS[42]	GR-MOEA	CMDfsmI[29]	Bin-MOABC[40]	MOFS-BDE[41]	FMABC-FS[42]
Wine	0.8762(0.0191)	0.8773(0.0232)	0.8621(0.0275)	0.8613(0.0242)	0.8837(0.0162)	0.1029(=)	3.87E-8(+)	0.0024(+)	3.46E-4(+)
Heart	0.7675(0.0322)	0.7643(0.0250)	0.7665(0.0273)	0.7605(0.0328)	0.7789(0.0337)	0.0279(+)	2.68E-6(+)	0.0104(+)	0.0101(+)
Hepatitis	0.7984(0.0405)	0.7604(0.0408)	0.7933(0.0389)	0.7978(0.0471)	0.8181(0.0349)	1.61E-4(+)	0.0010(+)	0.0071(+)	0.0237(+)
WDBC	0.9143(0.0231)	0.9191(0.0176)	0.9233(0.0077)	0.9259(0.0121)	0.9308(0.0255)	1.15E-13(+)	4.57E-7(+)	5.16E-5(+)	1.38E-4(+)
Ionosphere	0.6995(0.0235)	0.8713(0.0208)	0.8743(0.0227)	0.8661(0.0291)	0.8794(0.0209)	6.10E-26(+)	1.97E-4(+)	0.0025(+)	0.0370(+)
SpamBase	0.6833(0.0198)	0.8079(0.0338)	0.8701(0.0070)	0.8565(0.0210)	0.8711(0.0269)	3.43E-24(+)	4.83E-4(+)	0.2470(=)	0.0143(+)
Hillvalley	0.4328(0.0131)	0.5924(0.0301)	0.5789(0.0216)	0.5433(0.0319)	0.6161(0.0197)	4.07E-27(+)	0.0247(+)	9.62E-10(+)	2.72E-12(+)
Musk1	0.8599(0.0239)	0.8662(0.0209)	0.8255(0.0260)	0.7537(0.0382)	0.8688(0.0220)	9.89E-30(+)	0.0137(+)	4.55E-6(+)	6.41E-16(+)
LSVT	0.5503(0.0389)	0.7689(0.0543)	0.7456(0.0386)	0.6703(0.0539)	0.8433(0.0414)	1.88E-21(+)	5.11E-8(+)	7.38E-12(+)	5.17E-14(+)
Isolet	0.5087(0.0118)	0.7917(0.0249)	0.6829(0.0158)	0.6065(0.0184)	0.8264(0.0157)	2.24E-36(+)	0.0284(+)	1.10E-24(+)	3.70E-30(+)
Colon	0.5904(0.0358)	0.6344(0.0649)	0.7262(0.0536)	0.6538(0.0728)	0.8507(0.0604)	2.97E-22(+)	8.18E-8(+)	1.51E-10(+)	7.23E-13(+)
Leukemia	0.5844(0.0409)	0.7611(0.0351)	0.6684(0.0390)	0.6720(0.0443)	0.9312(0.0514)	6.37E-27(+)	2.31E-26(+)	1.73E-15(+)	1.70E-18(+)

all less than 0.05, except CMDfsmI on Wine dataset and MOFS-BDE on SpamBase dataset. It indicates that null hypothesis is accepted by 2 cases, and rejected by all the other cases. Based on the results of T-test, we can conclude that HV of our GR-MOEA is significantly better than those of compared MOEA based feature selection methods, and only equal to CMDfsmI on Wine dataset and MOFS-BDE on SpamBase dataset. The statistics in Table 5 have verified the better convergence and diversity of our GR-MOEA.

4.3. Hybrid effectiveness analysis

In order to show the hybrid effect of wrapper population and filter population in the proposed GR-MOEA, the comparison experiments among Wrapper, Filter and Hybrid are conducted, which denote wrapper population stand-alone mode, filter population stand-alone mode, and our hybrid GR-MOEA, respectively. From Table 6, we can see that our GR-MOEA is better on most datasets compared with stand-alone

Table 6
Accuracy and feature numbers comparisons of Wrapper, Filter and Hybrid mode.

Data Set	Accuracy(Std)			Feature number(std)		
	Wrapper	Filter	Hybrid	Wrapper	Filter	Hybrid
Wine	95.47(2.21)	95.51(3.19)	96.23(2.19)	5.40(0.5164)	7.91(1.35)	5.65(1.38)
Heart	81.75(6.16)	80.53(3.32)	83.70(4.05)	7.40(1.35)	7.73(1.62)	5.72(3.26)
Hepatitis	83.43(4.49)	82.03(4.60)	84.43(3.63)	3.80(2.39)	8.21(2.31)	3.70(1.08)
WDBC	96.67(1.35)	96.43(1.44)	96.88(1.06)	6.60(2.17)	9.25(2.72)	7.19(4.36)
Ionosphere	87.33(1.96)	85.17(3.19)	90.60(2.21)	4.80(1.97)	11.23(1.22)	3.61(1.97)
SpamBase	92.09(0.64)	88.86(0.81)	91.45(0.72)	18.30(2.67)	30.33(4.34)	15.15(2.62)
Hillvalley	56.95(2.31)	54.60(2.57)	58.53(1.78)	11.40(4.62)	59.20(4.51)	9.79(4.63)
Musk1	82.41(3.47)	77.85(3.78)	84.69(2.47)	14.30(6.16)	63.89(4.97)	16.68(5.47)
LSVT	81.84(8.55)	80.53(6.86)	84.87(4.15)	9.10(5.88)	56.47(8.02)	9.04(3.88)
Isotlet	81.98(1.89)	78.58(1.83)	82.76(2.11)	42.20(7.07)	111.41(6.90)	39.37(5.12)
Colon	74.63(8.84)	79.13(6.37)	79.32(8.05)	11.60(4.35)	88.87(8.11)	21.63(9.29)
Leukemia	81.82(9.82)	88.18(8.57)	90.34(5.69)	8.70(2.80)	171.13(28.02)	101.47(32.36)
win/tie/loss	11/0/1	12/0/0		7/0/5	12/0/0	

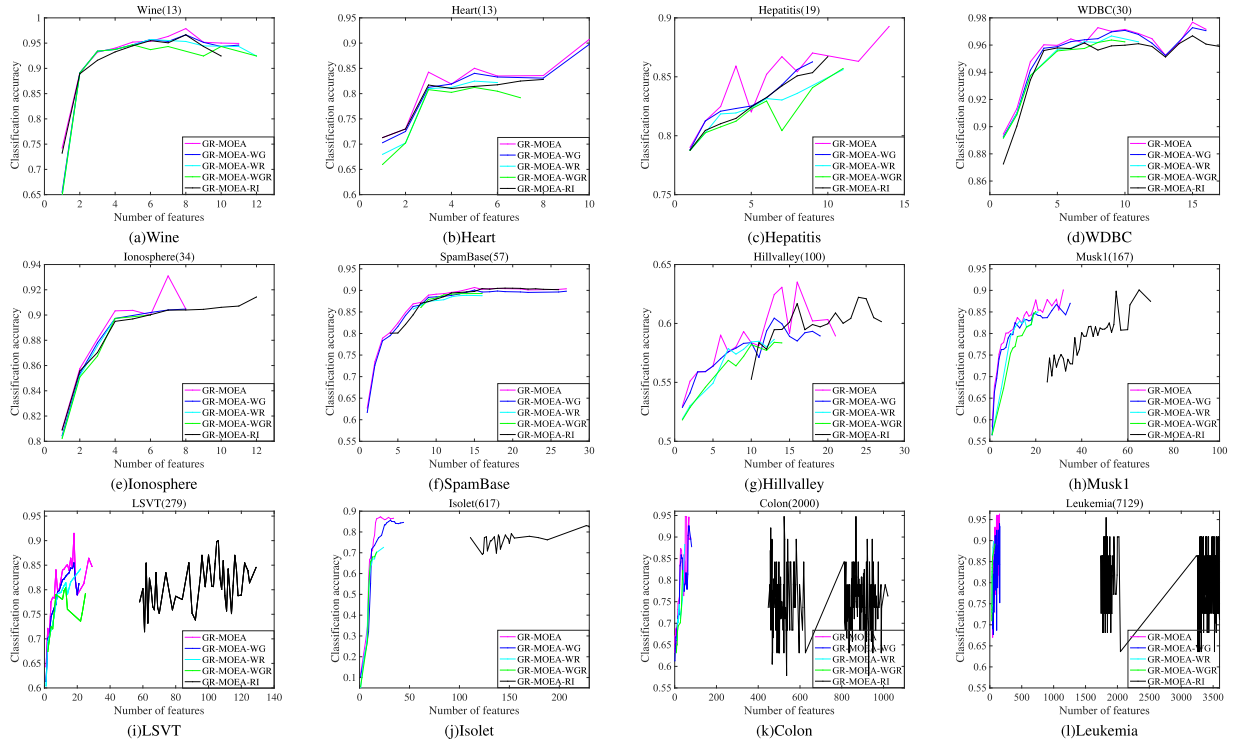


Fig. 7. Average results on all Pareto solutions obtained by GR-MOEA, GR-MOEA-WG, GR-MOEA-WR, GR-MOEA-WGR and GR-MOEA-RI.

mode according to the win/tie/loss of accuracy and feature number. Although the feature numbers of the wrapper stand-alone mode on Wine, WDBC, Musk1, Colon, Leukemia datasets are relatively small, their accuracies decrease obviously. To sum up, hybrid mode is superior to wrapper or filter stand-alone mode owing to the proposed interactive guiding and repairing strategies. In the following, we will verify these strategies in detail.

4.4. Strategy effectiveness analysis

In the proposed GR-MOEA, there are several strategies suggested, which include the wrapper to filter guiding strategy, the filter to wrapper repairing strategy and the initialized strategy. In this section, we detailed verify the effectiveness of these strategies. To be specific, we compare GR-MOEA with GR-MOEA-WG, GR-MOEA-WR, GR-MOEA-WGR and GR-MOEA-RI, which are all variants of original GR-MOEA. The former three variants are the same algorithm as GR-MOEA, except they abandon the guiding strategy, the repairing strategy and both strategies (guiding and repairing). The fourth variant GR-MOEA-RI replaces the

suggested initialized strategy with a random initialization. Fig. 7 shows the Pareto fronts of GR-MOEA and its variants.

From the figure we can find that compared with its variants (GR-MOEA-WG, GR-MOEA-WR, GR-MOEA-WGR and GR-MOEA-RI), the proposed algorithm has better Pareto solutions on almost all the experimental sets, which demonstrates the effectiveness of the suggested strategies. To be specific, GR-MOEA performs better than GR-MOEA-WG and GR-MOEA-WR, while GR-MOEA-WG and GR-MOEA-WR outperform GR-MOEA-WGR. This fact has indicated the effectiveness of the suggested guiding and repairing strategies. In addition, it can be also observed that GR-MOEA is better than GR-MOEA-RI, which has verified the effectiveness of the designed initializing strategy.

4.5. Different classifiers influence analysis

In order to analyse the influence of different classifiers on the performance of GR-MOEA, we attempt to use Heterogeneous oblique Random Forest (HobRaF) [54], Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB) and Random Forest (RF) as the evaluation

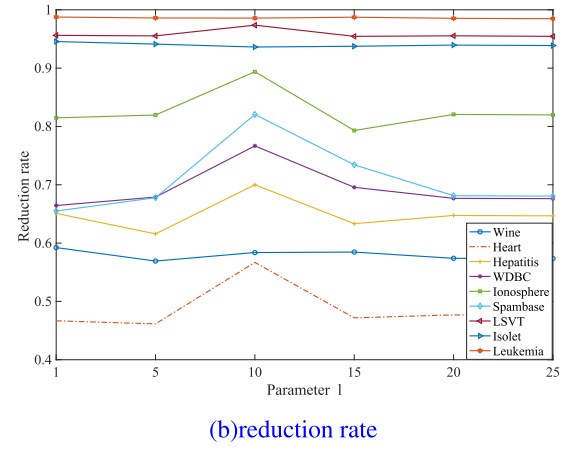
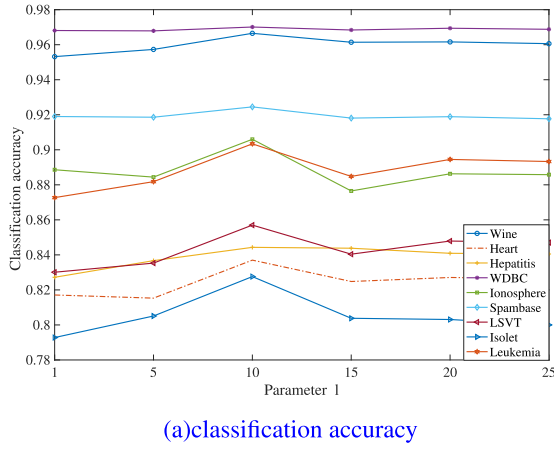


Fig. 8. The classification accuracy and reduction ratio of GR-MOEA with the different l .

classifier. In the experiment, SVM, DT, NB, RF, KNN adopt the default parameters setting from Matlab Toolbit, and the number of decision tree is set as 10 in RF and HobRaF [54]. Tables 7 and 8 give the comparison results with different classifiers in terms of accuracy, feature numbers and the running time. From two tables, we can find that our GR-MOEA with KNN classifier obtains the better accuracy with the fastest speed. Considering the balance between the classification accuracy and time cost, KNN is more suitable to be used as the evaluation classifier in our GR-MOEA. Furthermore, in order to show the effectiveness of feature selection, we also give the classification accuracy with all the features by KNN classifier, which is denoted as “KNN-All” in Table 7. Comparing the classification accuracy of “KNN” and “KNN-All”, we can observe that because our GR-MOEA removes the possibly redundant features, its performance is significantly better than the method using all the features.

4.6. Sensitivity analysis of parameter in GR-MOEA

As mentioned in Algorithm 1, there is an important parameter l in the proposed GR-MOEA, which is used to control the frequency of interactions between the filter population and the wrapper population. In this section, we will empirically investigate the influence of parameter l on GR-MOEA. Fig. 8 plots the experimental results of GR-MOEA with $l \in \{1, 5, 10, 15, 20, 25\}$. Note that since feature sizes of different datasets vary greatly, the number of selected features on these datasets are quite different. To make the figure more clear, in Fig. 8(b), we adopt the reduction rate (the percents of the not selected features in the total features) as the Y-axis.

As can be seen from the figure that if the value of l is set too large (such as $l = 25$), there is few interactions between two populations, which can not take the advantages of both populations to enhance the performance of GR-MOEA. On the contrary, if the value of l is set very small (such as $l = 1$), there are too many interactions, which means that some useless information in each population may be interchanged, and deteriorates the performance of GR-MOEA. Finally, the comparison results also show that when $l = 10$, the proposed method achieves both better accuracy and higher reduction rate. Therefore, in this paper, we set the value of l as 10.

4.7. Time analysis

To further analyze the performance of different feature selection algorithms, we compare GR-MOEA with other 7 state-of-the-arts in terms of running time, and the comparison results is shown in Table 9. As shown in the table, when compared with the hybrid methods, our GR-MOEA has the advantages on 8 of 12 datasets. When compared with non-hybrid method, the fastest method is filter based CMDfMI, and the speed of our GR-MOEA is the second fastest.

In addition, we also analyze the influence of the number of instances and the number of features on the running time of our GR-MOEA. From Table 9 we can find when the instances in the datasets are less and features are more, such as, Colon (62 instances and 2000 features) and Leukemia (72 instances and 7129 features), the speed of our GR-MOEA is very fast compared with other hybrid methods and wrapper methods. The reason is attributed to the suggested strategies in GR-MOEA (e.g. the initialization strategy, the hybrid framework, and the interaction between wrapper and filter populations), which ensures the proposed algorithm achieves better performance with less running time.

When the instances are more and features are less, for example, on the SpamBase dataset (4601 instances and 57 features), our observation is that the speed of our GR-MOEA is a little bit slow. The reason mainly lies in the fact that with the increasing number of instances the evaluation time of the wrapper population in our algorithm becomes longer, which results in the total running time of GR-MOEA is a bit long. However, the comparison results in Tables 3 and 4 have shown that GR-MOEA can obtain feature subset with higher quality than most of the comparison algorithms.

5. Conclusions and future work

In this paper, an interactive hybrid guiding and repairing based MOEA, named GR-MOEA is proposed to solve the problem of feature selection. The main contributions of the proposed GR-MOEA includes the following aspects. (1) Two populations (filter population and wrapper population) simultaneously evolve in GR-MOEA, and during the evolution they interact with each other to obtain the final feature subset with higher quality. (2) Two well-designed interactive strategies: the guiding strategy (from wrapper to filter) and the repairing strategy (from filter to wrapper) are suggested, which can make full use of the merits of two populations to select the features with high quality. (3) To further improve the performance of GR-MOEA, two effectively initialization strategies are also developed. The comparison results on different datasets have demonstrated the superiority of GR-MOEA over the existing state-of-the-art in terms of both accuracy and the number of selected features.

The experimental results have shown that it is a promising way to solve the the problem of feature selection by designing the hybrid based MOEA, and in this paper we only consider a simple framework of NSGA-II. How to develop more effective MOEA with other frameworks (such as MOEA/D [55] and AR-MOEA [56]) is an interesting future work to be further investigated. In addition, in the proposed GR-MOEA the binary encoding scheme is adopted to represent an individual. Although this encoding scheme is easy, it is not suitable for the dataset with high dimensions, since it may cause the curse of dimensionality. Thus, another

Table 7
Accuracy and feature numbers comparisons of different classifiers.

Data Set	Accuracy(Std)							Feature number(Std)						
	HobRaF [54]	SVM	DT	NB	RF	KNN	KNN-All	HobRaF [54]	SVM	DT	NB	RF	KNN	KNN-All
Wine	98.11(3.16)	97.23(1.90)	93.89(2.52)	96.99(2.70)	95.15(2.26)	96.23(2.19)	95.35(2.35)	5.40(1.14)	7.17(1.21)	5.10(1.19)	6.91(1.20)	7.10(2.23)	5.65(1.38)	13
Heart	84.22(5.13)	83.83(3.56)	80.73(7.13)	85.56(3.45)	82.12(3.97)	83.70(4.05)	80.45(4.53)	5.14(0.89)	7.01(1.78)	4.50(1.08)	7.40(1.51)	2.90(0.99)	5.72(3.26)	13
Hepatitis	85.17(4.18)	84.65(6.27)	79.53(5.25)	79.34(2.64)	80.74(3.25)	84.43(3.63)	83.12(4.34)	7.20(2.39)	7.41(2.82)	5.40(1.35)	5.12(1.79)	5.12(1.79)	3.70(1.08)	19
WDBC	97.10(1.17)	95.94(1.36)	93.63(2.19)	93.33(2.68)	95.62(2.68)	96.88(1.06)	96.65(1.38)	8.20(2.17)	10.33(4.19)	6.40(2.68)	8.00(1.41)	8.30(3.31)	7.19(4.36)	30
Ionosphere	92.38(1.78)	90.78(3.15)	86.87(3.04)	85.69(3.27)	91.05(1.75)	90.60(2.21)	84.19(3.71)	10.40(4.56)	11.47(2.49)	4.80(2.04)	8.10(1.85)	9.10(4.85)	3.61(1.97)	34
SpamBase	93.18(0.74)	90.45(1.43)	90.69(1.21)	89.33(1.84)	93.33(1.00)	91.45(0.72)	89.52(0.64)	10.49(2.49)	16.30(1.74)	15.80(3.61)	18.39(2.61)	19.60(1.51)	15.15(2.62)	57
Hillvalley	78.11(0.66)	53.78(2.26)	58.02(2.25)	50.74(2.82)	50.01(1.73)	58.53(1.78)	54.01(2.42)	19.20(3.56)	13.40(8.19)	7.70(2.31)	5.60(3.56)	10.30(2.91)	9.79(4.63)	100
Musk1	86.65(3.48)	84.71(2.25)	83.27(4.47)	81.12(7.14)	81.98(2.73)	84.69(2.47)	81.79(3.30)	18.20(5.36)	20.49(3.72)	12.10(3.51)	18.43(5.54)	15.30(3.50)	16.68(5.47)	166
LSVT	78.95(4.32)	88.95(2.20)	80.39(4.68)	78.42(4.96)	82.74(6.20)	84.87(4.15)	81.23(6.05)	14.80(5.26)	24.80(3.11)	13.10(2.42)	12.83(4.39)	8.64(2.76)	9.04(3.88)	309
Isolet	82.95(6.88)	85.79(1.29)	80.06(1.64)	78.95(2.88)	79.59(2.00)	82.76(2.11)	78.85(1.40)	42.39(6.47)	40.69(4.38)	39.50(5.99)	38.47(6.21)	39.60(7.38)	39.37(5.12)	617
Colon	81.21(6.15)	77.15(5.26)	76.21(9.10)	78.95(4.18)	75.79(9.67)	79.32(8.05)	74.74(9.71)				42.90(5.28)	21.63(9.29)		2000
Leukemia	90.93(5.93)	88.95(2.20)	86.18(9.59)	82.70(7.91)	85.73(8.45)	90.34(5.69)	78.64(7.56)	52.43(10.71)	40.81(20.26)	40.80(10.56)	28.20(10.05)			7129
								219.45(9.30)	174.20(11.73)	166.70(29.38)	88.10(38.45)	149.70(35.87)	101.47(32.36)	

Table 8
Running time (s) comparison of different classifiers.

Data Set	Time(s)					
	HobRaF [54]	SVM	DT	NB	RF	KNN
Wine	1090.57	301.05	59.05	82.08	341.19	64.98
Heart	1174.73	169.96	66.17	76.31	337.19	62.86
Hepatitis	577.93	183.86	59.23	35.46	350.50	67.22
WDBC	1450.75	204.24	75.56	87.93	442.07	73.80
Ionosphere	1528.53	202.74	73.31	88.56	411.39	71.02
SpamBase	26800.16	1850.62	210.23	138.58	1351.41	151.84
Hillvalley	7215.64	379.00	191.25	109.03	784.82	81.42
Musk1	2700.41	225.43	87.32	120.49	511.78	72.37
LSVT	616.68	197.96	66.28	106.14	404.75	68.84
Isolet	36125.84	9811.82	541.03	795.51	1376.21	139.64
Colon	354.39	197.05	66.59	195.08	399.46	70.74
Leukemia	332.41	250.01	89.91	646.20	469.72	88.57
win/tie/loss	12/0/0	12/0/0	8/0/4	10/0/2	12/0/0	

Table 9
Running time (s) of FastPSO, RapidPSO, HPSO-LS, CMDfsMI, Bin-MOABC, MOFS-BDE, FMABC-FS and GR-MOEA on the experimental sets.

Data Set	Hybrid vs GR-MOEA				Non-Hybrid vs GR-MOEA				
	FastPSO[24]	RapidPSO[24]	HPSO-LS[27]	GR-MOEA	CMDfsMI[29]	Bin-MOABC[40]	MOFS-BDE[41]	FMABC-FS[42]	GR-MOEA
Wine (178 instances and 13 features)	81.25	58.43	62.41	64.98	0.72	145.20	270.15	260.21	64.98
Heart (270 instances and 13 features)	85.49	73.51	70.12	62.86	0.66	138.38	267.78	276.59	62.86
Hepatitis (155 instances and 19 features)	96.33	85.05	60.94	67.22	0.65	131.64	260.75	273.39	67.22
WDBC (569 instances and 30 features)	128.52	113.13	132.67	73.80	0.71	144.63	282.86	240.03	73.80
Ionosphere (351 instances and 34 features)	119.23	109.98	110.93	71.02	0.68	135.20	304.79	261.97	71.02
SpamBase (4601 instances and 57 features)	138.72	122.40	2615.52	151.84	1.96	918.38	640.11	341.99	151.84
Hillvalley (606 instances and 100 features)	43.96	25.87	405.99	81.42	0.93	152.48	383.56	371.42	81.42
Musk1 (476 instances and 166 features)	131.45	117.92	312.38	72.37	0.86	141.63	404.24	332.99	72.37
LSVT (126 instances and 309 features)	125.18	105.74	156.89	68.84	0.89	128.32	334.64	339.76	68.84
Isolet (1559 instances and 617 features)	582.13	455.68	5833.62	139.64	4.29	1466.93	1022.23	601.67	139.64
Colon(62 instances and 2000 features)	155.87	132.78	418.86	70.74	2.78	386.67	290.67	299.28	70.74
Leukemia(72 instances and 7129 features)	245.61	205.17	593.75	88.57	12.46	592.63	540.81	450.71	88.57
win/tie/loss	11/0/1	9/0/3	10/0/2		0/0/12	12/0/0	12/0/0	12/0/0	

interesting future work is to extend the proposed method to the feature selection tasks with large scale.

Declaration of Competing Interest

We declare that we have no financial and personal relationships with other people or organizations that can inappropriately influence our work, there is no professional or other personal interest of any nature

or kind in any product, service or company that could be construed as the review of the manuscript.

CRediT authorship contribution statement

Zhengyi Liu: Methodology, Writing - original draft. **Bo Chang:** Data curation, Formal analysis. **Fan Cheng:** Supervision, Investigation, Writing - review & editing.

Acknowledgments

This work is supported by Natural Science Foundation of Anhui Province (1908085MF182) and Key Program of Natural Science Project of Educational Commission of Anhui Province (KJ2019A0034), Humanities and Social Sciences Project of Chinese Ministry of Education (18YJC870004), University Synergy Innovation Program of Anhui Province (GXXT-2020-050), National Natural Science Foundation of China under Grant (62076001).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at [10.1016/j.swevo.2021.100925](https://doi.org/10.1016/j.swevo.2021.100925).

References

- [1] B. Xue, M. Zhang, W.N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, *IEEE Trans. Evol. Comput.* 20 (4) (2015) 606–626.
- [2] A. Zibakhsh, M.S. Abadeh, Gene selection for cancer tumor detection using a novel memetic algorithm with a multi-view fitness function, *Eng. Appl. Artif. Intell.* 26 (4) (2013) 1274–1281.
- [3] H. Uguz, A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm, *Knowl. Based Syst.* 24 (7) (2011) 1024–1032.
- [4] K. Mistry, L. Zhang, S.C. Neoh, C.P. Lim, B. Fielding, A micro-ga embedded PSO feature selection approach to intelligent facial emotion recognition, *IEEE Trans. Cybern.* 47 (6) (2016) 1496–1509.
- [5] N. Mirzadeh, F. Ricci, M. Bansal, Feature selection methods for conversational recommender systems, in: *Proceedings of the IEEE International Conference on e-Technology, e-Commerce and e-Service, IEEE, 2005*, pp. 772–777.
- [6] M.S. Raza, U. Qamar, A hybrid feature selection approach based on heuristic and exhaustive algorithms using rough set theory, in: *Proceedings of the International Conference on Internet of Things and Cloud Computing, 2016*, pp. 1–7.
- [7] I. Tsamardinos, G. Borboudakis, P. Katsogridakis, P. Pratikakis, V. Christophides, A greedy feature selection algorithm for big data of high dimensionality, *Mach. Learn.* 108 (2) (2019) 149–202.
- [8] S. Oreski, G. Oreski, Genetic algorithm-based heuristic for feature selection in credit risk assessment, *Expert Syst. Appl.* 41 (4) (2014) 2052–2064.
- [9] S. Kang, D. Kim, S. Cho, Efficient feature selection-based on random forward search for virtual metrology modeling, *IEEE Trans. Semicond. Manuf.* 29 (4) (2016) 391–398.
- [10] G. Chandrashekar, F. Sahin, A survey on feature selection methods, *Comput. Electr. Eng.* 40 (1) (2014) 16–28.
- [11] E. Aličković, A. Subasi, Breast cancer diagnosis using GA feature selection and rotation forest, *Neural Comput. Appl.* 28 (4) (2017) 753–763.
- [12] C. Khammassi, S. Krichen, A GA-LR wrapper approach for feature selection in network intrusion detection, *Comput. Secur.* 70 (2017) 255–277.
- [13] Z. Tao, L. Huiling, W. Wenwen, Y. Xia, GA-SVM based feature selection and parameter optimization in hospitalization expense modeling, *Appl. Soft Comput.* 75 (2019) 323–332.
- [14] B. Tran, B. Xue, M. Zhang, S. Nguyen, Investigation on particle swarm optimisation for feature selection on high-dimensional data: local search and selection bias, *Connect. Sci.* 28 (3) (2016) 270–294.
- [15] B. Tran, B. Xue, M. Zhang, A new representation in PSO for discretization-based feature selection, *IEEE Trans. Cybern.* 48 (6) (2017) 1733–1746.
- [16] Y. Chen, D. Miao, R. Wang, A rough set approach to feature selection based on ant colony optimization, *Pattern Recognit. Lett.* 31 (3) (2010) 226–233.
- [17] S. Kashef, H. Nezamabadi-pour, An advanced ACO algorithm for feature subset selection, *Neurocomputing* 147 (2015) 271–279.
- [18] P. Moradi, M. Rostami, Integration of graph clustering with ant colony optimization for feature selection, *Knowl. Based Syst.* 84 (2015) 144–161.
- [19] S. Palanisamy, S. Kanmani, Artificial bee colony approach for optimizing feature selection, *Int. J. Comput. Sci. Issues* 9 (3) (2012) 432–438.
- [20] E. Zorarpaci, S.A. Özel, A hybrid approach of differential evolution and artificial bee colony for feature selection, *Expert Syst. Appl.* 62 (2016) 91–103.
- [21] Z. Wang, M. Li, J. Li, A multi-objective evolutionary algorithm for feature selection based on mutual information with a new redundancy measure, *Inf. Sci.* 307 (2015) 73–88.
- [22] A. Unler, A. Murat, R.B. Chinnam, mr2PSO: a maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification, *Inf. Sci.* 181 (20) (2011) 4625–4641.
- [23] B. Xue, M. Zhang, W.N. Browne, Particle swarm optimization for feature selection in classification: a multi-objective approach, *IEEE Trans. Cybern.* 43 (6) (2012) 1656–1671.
- [24] T. Butler-Yeoman, B. Xue, M. Zhang, Particle swarm optimisation for feature selection: a hybrid filter-wrapper approach, in: *Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, 2015*, pp. 2428–2435.
- [25] M. Ghosh, R. Guha, R. Sarkar, A. Abraham, A wrapper-filter feature selection technique based on ant colony optimization, *Neural Comput. Appl.* 32 (12) (2019) 1–19.
- [26] Z. Huang, C. Yang, X. Zhou, T. Huang, A hybrid feature selection method based on binary state transition algorithm and relief, *IEEE J. Biomed. Health Inf.* 23 (5) (2018) 1888–1898.
- [27] P. Moradi, M. Gholampour, A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy, *Appl. Soft Comput.* 43 (2016) 117–130.
- [28] F. Moslehi, A. Haeri, A novel hybrid wrapper-filter approach based on genetic algorithm, particle swarm optimization for feature subset selection, *J. Ambient Intell. Humanized Comput.* 11 (3) (2020) 1105–1127.
- [29] B. Xue, L. Cervante, L. Shang, W.N. Browne, M. Zhang, A multi-objective particle swarm optimisation for filter-based feature selection in classification problems, *Connect. Sci.* 24 (2-3) (2012) 91–116.
- [30] E. Hancer, B. Xue, M. Zhang, D. Karaboga, B. Akay, A multi-objective artificial bee colony approach to feature selection using fuzzy mutual information, in: *Proceedings of the IEEE Congress on Evolutionary Computation, IEEE, 2015*, pp. 2420–2427.
- [31] E. Hancer, B. Xue, M. Zhang, Differential evolution for filter feature selection based on information theory and feature ranking, *Knowl. Based Syst.* 140 (2018) 103–119.
- [32] M. Robnik-Šikonja, I. Kononenko, Theoretical and empirical analysis of relief and rrelief, *Mach. Learn.* 53 (1-2) (2003) 233–269.
- [33] C.M. Bishop, et al., *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.
- [34] S.K. Nayak, P.K. Rout, A.K. Jagadev, T. Swarnkar, Elitism based multi-objective differential evolution for feature selection: a filter approach with an efficient redundancy measure, *J. King Saud Univ. Comput. Inf. Sci.* 32 (2) (2020) 174–187.
- [35] L.E.A.d.S. Santana, A.M. de Paula Canuto, Filter-based optimization techniques for selection of feature subsets in ensemble systems, *Expert Syst. Appl.* 41 (4) (2014) 1622–1631.
- [36] T.M. Hamdani, J.-M. Won, A.M. Alimi, F. Karray, Multi-objective feature selection with NSGA II, in: *Proceedings of the International Conference on Adaptive and Natural Computing Algorithms, Springer, 2007*, pp. 240–247.
- [37] M. Gutlein, E. Frank, M. Hall, A. Karwath, Large-scale attribute selection using wrappers, in: *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining, IEEE, 2009*, pp. 332–339.
- [38] R. Caruana, D. Freitag, Greedy attribute selection, in: *Machine Learning Proceedings 1994, Elsevier, 1994*, pp. 28–36.
- [39] M. Amoozegar, B. Minaei-Bidgol, Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism, *Expert Syst. Appl.* 113 (2018) 499–514.
- [40] E. Hancer, B. Xue, M. Zhang, D. Karaboga, B. Akay, Pareto front feature selection based on artificial bee colony optimization, *Inf. Sci.* 422 (2018) 462–479.
- [41] Y. Zhang, D.-w. Gong, X.-z. Gao, T. Tian, X.-y. Sun, Binary differential evolution with self-learning for multi-objective feature selection, *Inf. Sci.* 507 (2020) 67–85.
- [42] X. Wang, Y. Zhang, X. Sun, Y. Wang, C.-h. Du, Multi-objective feature selection based on artificial bee colony: an acceleration approach with variable sample size, *Appl. Soft Comput.* 88 (2020) 106–114.
- [43] R.J. Manoj, M.A. Praveena, K. Vijayakumar, An ACO-ANN based feature selection algorithm for big data, *Cluster Comput.* 22 (2) (2019) 3953–3960.
- [44] P. Shunmugapriya, S. Kanmani, A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC hybrid), *Swarm Evol. Comput.* 36 (2017) 27–36.
- [45] M. Mafarja, S. Mirjalili, Whale optimization approaches for wrapper feature selection, *Appl. Soft Comput.* 62 (2018) 441–453.
- [46] S. Arora, P. Anand, Binary butterfly optimization approaches for feature selection, *Expert Syst. Appl.* 116 (2019) 147–160.
- [47] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2) (2002) 182–197.
- [48] Z. Yong, G. Dun-wei, Z. Wan-qiu, Feature selection of unreliable data using an improved multi-objective PSO algorithm, *Neurocomputing* 171 (2016) 1281–1290.
- [49] P. Moradi, M. Rostami, A graph theoretic approach for unsupervised feature selection, *Eng. Appl. Artif. Intell.* 44 (2015) 33–45.
- [50] D. Dua, C. Graff, UCI machine learning repository, 2017, <http://archive.ics.uci.edu/ml>.
- [51] V.G. Da Fonseca, C.M. Fonseca, A.O. Hall, Inferential performance assessment of stochastic optimisers and the attainment function, in: *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization, Springer, 2001*, pp. 213–225.
- [52] M. López-Ibáñez, L. Paquete, T. Stützle, Exploratory analysis of stochastic local search algorithms in biobjective optimization, in: *Experimental Methods for the Analysis of Optimization Algorithms, Springer, 2010*, pp. 209–222.
- [53] R. Lacour, K. Klamroth, C.M. Fonseca, A box decomposition algorithm to compute the hypervolume indicator, *Comput. Oper. Res.* 79 (2017) 347–360.
- [54] R. Katuwal, P.N. Suganthan, L. Zhang, Heterogeneous oblique random forest, *Pattern Recognit.* 99 (2020) 107078.
- [55] B.H. Nguyen, B. Xue, P. Andreae, H. Ishibuchi, M. Zhang, Multiple reference points-based decomposition for multiobjective feature selection in classification: static and dynamic mechanisms, *IEEE Trans. Evol. Comput.* 24 (1) (2020) 170–184.
- [56] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator-based multiobjective evolutionary algorithm with reference point adaptation for better versatility, *IEEE Trans. Evol. Comput.* 22 (4) (2018) 609–622.