A Steering-Matrix-Based Multiobjective Evolutionary Algorithm for High-Dimensional Feature Selection

Fan Cheng, Feixiang Chu, Yi Xu, and Lei Zhang, *Member, IEEE*

***Abstract*—In recent years, multiobjective evolutionary algorithms (MOEAs) have been demonstrated to show promising performance in feature selection (FS) tasks. However, designing an MOEA for high-dimensional FS is more challenging due to the curse of dimensionality. To address this problem, in this article, a steering-matrix-based multiobjective evolutionary algorithm, called SM-MOEA, is proposed. In SM-MOEA, a steering matrix is suggested and harnessed to guide the evolution of the population, which not only improves the search efficiency greatly but also obtains the feature subsets with high quality. Specifically, each element SM(i, j) in the steering matrix SM reflects the probability of the jth feature that is selected in the ith individual (feature subset), which is generated by considering the importance of both the feature j and the individual i. Based on the suggested steering matrix, two important operators referred to as dimensionality reduction and individual repairing operators are developed to effectively steer the population evolution in each generation. In addition, an effective initialization and update strategy for the steering matrix is also designed to further improve the performance of SM-MOEA. The experimental results on 12 high-dimensional datasets with the number of features ranging from 3000 to 13 000 demonstrate the superiority of the proposed algorithm over several state-of-the-art algorithms (including single-objective and MOEAs for high-dimensional FS) in terms of both the number and quality of the selected features.**

***Index Terms*—Evolutionary algorithm, feature selection (FS), high-dimensional data, multiobjective optimization.**

I. Introduction

FEATURE selection (FS) is an important task in data mining and machine learning, whose aim is to reduce the dimensionality of the dataset and increase the performance of a classification algorithm, such as the k-nearest neighbor (KNN) and support vector machine (SVM) [1]. Due to its importance, several FS algorithms have been proposed and these can generally be classified into two categories: 1) filter methods and 2) wrapper methods [2]. Filter FS methods use feature relevance criteria, such as mutual information or the Pearson correlation coefficient, to select the feature subset. Wrapper methods utilize a classification algorithm as a "black box" to evaluate the quality of the selected features. Although the filter methods are computationally less expensive than the wrapper ones, they ignore the performance of the selected features on a classification algorithm; thus, the feature subsets achieved by the filter methods are often worse than those achieved by the wrapper methods [2]. Therefore, in this study, we mainly focus on wrapper FS methods.

In the studies on wrapper FS methods, various search techniques, including the exhaustive search [3], greedy search [4], heuristic search [5], and random search [6], have been applied to obtain good feature subsets. Although these FS methods can select high-quality features, most of them often converge to local optima [2]. To this end, in recent years, the evolutionary computation (EC) technique has attracted significant attention from the FS community because of its well-known global searchability. In addition, compared with other search techniques, EC does not require domain knowledge or assumptions on the search space and is more suitable for the FS problem [7]. Based on the number of optimized objectives, existing EC-based wrapper FS methods can be divided into two groups: 1) single-objective and 2) multiobjective EC-based FS algorithms.

The single-objective EC-based FS algorithms adopt different EC paradigms, such as the genetic algorithm (GA) [8], particle swarm optimization (PSO) [9], and ant colony optimization (ACO) [10], to solve the FS problem, where the classification performance and the number of selected features are often aggregated into one optimized objective by using a tradeoff parameter. The performance of these single-objective FS algorithms is promising; however, setting a suitable value for the tradeoff parameter is a difficult problem, especially when prior knowledge is unavailable in real applications. To this end, multiobjective evolutionary algorithms (MOEAs) have been suggested to solve the FS problem, which have at least two merits. First, the classification performance and the number of selected features can be balanced without using the tradeoff parameter by utilizing the multiobjective

Manuscript received February 12, 2020; revised July 11, 2020 and December 30, 2020; accepted January 20, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62076001, Grant 61976001, Grant 61876184, and Grant U1804262; in part by the Natural Science Foundation of Anhui Province under Grant 2008085QF309 and Grant 1708085MF166; and in part by the Humanities and Social Sciences Project of Chinese Ministry of Education under Grant 18YJC870004. This article was recommended by Associate Editor Y. Jin. *(Corresponding author: Lei Zhang.)*

Fan Cheng, Feixiang Chu and Lei Zhang are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Computer Science and Technology, Anhui University, Hefei 230039, China (e-mail: chengfan@mail.ustc.edu.cn; cifeixiang@163.com; zl@ahu.edu.cn).

Yi Xu is with the School of Computer Science and Technology, Shanghai Jiaotong University, Shanghai 200139, China (e-mail: 644327005@qq.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2021.3053944.

Digital Object Identifier 10.1109/TCYB.2021.3053944

optimization technique. Second, multiobjective FS algorithms can provide multiple solutions in only one run, which is particularly suitable for satisfying the different needs of real applications.

In this regard, many multiobjective FS algorithms using different EC techniques have been proposed recently. For example, Xue *et al.* proposed a multiobjective PSO-based wrapper FS algorithm, called CMDPSO [11], and their experimental results showed that the proposed CMDPSO can obtain feature subsets with high quality. Since then, more multiobjective PSO-based FS algorithms have been developed [12]–[14]. Recently, Hancer *et al.* [15] suggested a Pareto front FS algorithm based on the artificial bee colony (ABC) optimization. Despite that these MOEA-based FS algorithms showing their competitive performance; however, they are poorly scalable with respect to high-dimensional datasets, thereby reducing their applicability to large-scale applications [16]–[18]. The main reason for the lack of scalability is that in these MOEAs, the length of the individual is equal to the number of features in the dataset; thus, the search space increases exponentially as the number of features increases.

To address the aforementioned problem, in this article, we propose a steering-matrix-based MOEA, namely, SM-MOEA, which is suitable for high-dimensional FS. The main contributions of this study can be summarized as follows.

1) A steering matrix SM is proposed and harnessed to improve the search efficiency significantly and obtain high-quality feature subsets. Specifically, each element SM(i, j) in the proposed steering matrix is generated by considering the importance of the jth feature in the dataset and that of the ith individual in the population. Thus, this steering matrix can approximate the importance of features in population individuals, which can be used to steer the evolution of the population toward a better direction.
2) Based on the proposed steering matrix, an MOEA, called SM-MOEA, is suggested for high-dimensional FS. SM-MOEA employs two important operators (a dimensionality reduction operator and an individual repairing operator) based on the steering matrix, and they are designed to guide the evolution of the population, which can recursively downsize the number of selected features in population individuals and maintain a low classification error rate. In addition, an effective initialization and update strategy is also designed for the steering matrix to further improve the performance of SM-MOEA.
3) The effectiveness of the proposed SM-MOEA is evaluated on 12 high-dimensional datasets, in which the feature sizes range from 3000 to 13 000. The experimental results demonstrate that the proposed SM-MOEA outperforms state-of-the-art EC-based single-objective and multiobjective FS algorithms in terms of both the error rate and the number of selected features.

The remainder of this article is organized as follows. Preliminary definitions for multiobjective optimization and recent studies on EC-based FS are described in Section II. The proposed SM-MOEA is detailed in Section III. The effectiveness of SM-MOEA is verified in Section IV, by comparing it with several state-of-the-art EC-based FS algorithms. Finally, we provide our conclusions and directions for future work in Section V.

II. Preliminaries and Related Work

In this section, preliminary definitions pertaining to multiobjective optimization are given and then recent research related to EC-based FS methods is reviewed in detail.

A. Multiobjective Optimization

In many machine-learning or data mining applications, such as FS [7], [15]; instance selection [19]; pattern mining [20]; P2P recommendation [21]; and community detection [22], [23], tasks can be formulated as multiobjective optimization problems (MOPs) characterized by multiple objectives that may conflict with each other. Formally, the MOP can be defined as

$$\text{Minimize} \quad \mathcal{F}(I) = (\mathcal{F}_1(I), \mathcal{F}_2(I), \ldots, \mathcal{F}_m(I))^T \tag{1}$$

where $I = (i_1, i_2, \ldots, i_n) \in \mathcal{R}$ (the n-dimensional decision space) is the decision vector and m is the number of objectives.

Given two decision vectors I_1 and I_2, $I_1 \succ I_2$ means that I_1 dominates I_2 or I_2 is dominated by I_1 if $\mathcal{F}_i(I_1) \leq \mathcal{F}_i(I_2)$ for all $i = 1, 2, \ldots, m$ and $\mathcal{F}(I_1) \neq \mathcal{F}(I_2)$. For decision vector $I \in \mathcal{R}$, if there is no $I^* \in \mathcal{R}$ satisfying $I^* \succ I$, then I is referred to as a *nondominated solution*. The set of all nondominated solutions is known as the *Pareto set*, which is defined as $\text{PS} = \{I \in \mathcal{R} | \nexists I^* \in \mathcal{R}, I^* \succ I\}$. The projection of the Pareto set into the objective space is known as the *Pareto front*, which is defined as $PF = \{\mathcal{F}(I) | I \in \text{PS}\}$. The aim of MOEAs is to find a set of nondominated solutions capable of approximating the true Pareto front as closely as possible [24].

B. Related Work

In recent years, many EC-based FS algorithms have been proposed. Thus, the following three lines of related work are reviewed: 1) single-objective EC-based FS algorithms; 2) multiobjective EC-based FS algorithms; and 3) evolutionary algorithms for high-dimensional FS.

1) Single-Objective EC-Based FS Algorithms: Many single-objective EC-based FS algorithms that adopt different EC paradigms have been proposed. For example, Siedlecki and Sklansky [25] proposed a GA-based algorithm to select features for designing automatic pattern classifiers. Experiments on both real-world and artificial datasets were conducted to demonstrate the superiority of their proposed algorithm over traditional non-EC FS (such as exhaustive search and greedy search) methods. Since then, more GA-based FS algorithms with different enhancement techniques have been suggested, and the interested readers can refer to [8], [26], and [27]. PSO, as a swarm intelligence optimization technique, has also been successfully applied to solve the problem of FS, and a series of representative PSO-based studies was conducted by Tran *et al.* [28]–[30], who designed different initialization strategies, fitness functions, and search mechanisms to obtain high-quality feature subsets.

In addition to the aforementioned GA-based and PSO-based single-objective FS algorithms, a few interesting FS methods that use other EC paradigms were also developed, for example, genetic programming (GP) [31]; ACO [10]; differential evolution (DP) [32], [33]; and memetic algorithms (MA) [34]. These single-objective EC-based FS algorithms exhibited promising performance; however, most of them combine the classification performance and the number of selected features into one objective by introducing a tradeoff parameter. Assigning a suitable value to this parameter is a difficult problem, especially when prior knowledge obtained in real applications is not available. To this end, MOEAs have been applied to solve the FS problem.

2) Multiobjective EC-Based FS Algorithms: The multiobjective FS problem can be defined as follows [7], [15]:

$$\text{Minimize}\begin{cases}\mathcal{F}_1(I) = |I| \\ \mathcal{F}_2(I) = \frac{\text{FP+FN}}{\text{TP+TN+FP+FN}}\end{cases} \tag{2}$$

where I is a selected feature subset, $\mathcal{F}_1(I)$ is the size of this selected feature subset, and $\mathcal{F}_2(I)$ represents the error rate using these selected features on a classification algorithm, for example, KNN or SVM. TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively.

In the past decade, various multiobjective FS algorithms using EC techniques have been suggested. For example, Xue *et al.* [11] proposed a multiobjective PSO-based wrapper FS algorithm, called CMDPSO. In CMDPSO, standard PSO was first extended to address the MOPs. Then, the concepts of crowding distance, mutation, and nondominated sorting were introduced to PSO to search for the Pareto front solutions. Experimental results on different datasets have shown that CMDPSO not only outperformed the single-objective FS algorithms, such as LFS and GSBS, but also had better performance than the multiobjective FS algorithms under the frameworks of NSGA-II, SPEA2, and PAES. In recognizing the superiority of CMDPSO, more multiobjective PSO-based FS algorithms with promising performance have been developed [12]–[14].

Recently, Hancer *et al.* suggested a Pareto front FS algorithm, called MOABC, which is based on ABC optimization. The design of MOABC is based on a new multiobjective ABC framework that employs nondominated sorting and a genetically inspired search. Then, two versions were implemented based on two different representations: 1) Bin-MOABC (binary version) and 2) Num-MOABC (continuous version). The empirical results confirmed the efficiency and effectiveness of MOABC. In addition, the results indicated that Bin-MOABC achieved a better feature subset than Num-MOABC. Other MOEAs for FS were also proposed in [35]–[37].

3) Evolutionary Algorithms for High-Dimensional FS: EA (including single-objective and multiobjective)-based algorithms can obtain feature subsets with high quality and have been successfully applied to solve different FS tasks [7]. However, in certain real FS applications, such as biomedical data mining [17] and image recognition [38], the dimensionality of the dataset is very large, which brings a great challenge to the EA-based algorithms. The reason is that these algorithms usually adopt binary encoding, in which case, the length of an individual equals the number of features in the data, and the search space increases exponentially with the increment of the number of features. To solve the challenge, some preliminary explorations for high-dimensional FS were suggested.

For example, in the single-objective EC-based area, Gu *et al.* [39] proposed the use of a competitive swarm optimizer (CSO), which is a variant of PSO, to select features for high-dimensional classification. In the CSO, the swarm replaced the global or personal best position with randomly selected competitors. Specifically, during the process of evolution, the entire swarm was randomly divided into two groups and then a pairwise mechanism was implemented between two particles from these two groups to choose the winner and loser. Finally, the winner set directly survived into the next generation, whereas the loser set learned from the winner set. The experimental results demonstrated that compared with the state-of-the-art PSO-based FS methods, the classification performance of CSO was more accurate and the algorithm selected smaller feature subsets on high-dimensional datasets.

Tran *et al.* proposed a single-objective variable-length PSO (VLPSO) [28] for FS on high-dimensional classification, where particles with different lengths were utilized to downsize the search space and improve the performance of the PSO algorithm. Specifically, VLPSO started by ranking and rearranging features based on their relevance. Subsequently, it adopted a division mechanism in the first loop, which generated exemplars with different lengths for each particle. The second loop was the evolutionary process, during which a length-changing procedure was called, if *gbest* has not been improved for a certain time. The results on ten high-dimensional datasets revealed that VLPSO achieved significantly better performance than the fixed-length PSO-based and non-PSO-based FS methods.

Xue *et al.* [40] designed a self-adaptive PSO, called SaPSO, for high-dimensional FS. SaPSO maintained multiple candidate solution generation strategies (CSGSs) with different characteristics in a pool and used previous experiences of generating promising solutions to adaptively choose the suitable CSGSs to generate new solutions in subsequent generations. Empirical studies on various datasets revealed its effectiveness and efficiency. Recently, a novel PSO-based algorithm (SPS-PSO) was proposed [41] to solve high-dimensional FS problems. A CSGS self-adaptive mechanism and a parameter self-adaptive mechanism were developed to guide the evolution. The experimental results showed that SPS-PSO had good global and local search abilities when solving high-dimensional FS problems.

In the multiobjective EC-based area, to the best of our knowledge, only a few MOEAs for high-dimensional FS were reported. The most closely related work involved the proposal of SparseEA [42], which is an evolutionary algorithm for large-scale sparse MOPs and can be applied to solve FS problems. SparseEA is based on a new population initialization strategy and genetic operators in consideration of the sparse nature of the Pareto optimal solutions. The experimental results verified that most of the solutions obtained by SparseEA were sparse, and the algorithm showed its effectiveness and efficiency in solving high-dimensional FS. Another related algorithm is DMEA-FS, developed by Li *et al.* [43]. In DMEA-FS, FS was transformed into a many-objective optimization problem and solved by a suggested dividing-based evolutionary algorithm. Numerical experiments showed that DMEA-FS can effectively handle high-dimensional FS tasks. Recently, Nguyen *et al.* [44] proposed an MOEA based on multiple reference points for FS in classification. Two mechanisms (static and dynamic) were appropriately designed to enable the algorithm to overcome the challenges presented by a highly discontinuous Pareto front, imbalance preferences, and partially conflicting objectives in FS. The experimental results demonstrated that their proposed algorithm could effectively solve high-dimensional FS problems.

In contrast to the aforementioned studies, this article presents a steering-matrix-based MOEA, called SM-MOEA, for high-dimensional FS. SM-MOEA includes two important operators, called dimensionality reduction operator and individual repairing operator, based on a well-designed steering matrix. These operators are developed to evolve individuals in the population to recursively downsize the number of selected features in individuals and obtain feature subsets with high quality. In the next section, we describe the proposed SM-MOEA in detail.

III. Proposed SM-MOEA

In this section, we present details of the proposed SM-MOEA for FS on high-dimensional datasets. First, we present the main idea of SM-MOEA and then introduce the proposed steering matrix, which is the crucial component of SM-MOEA. Furthermore, we introduce the two evolutionary operators based on the steering matrix: 1) the dimensionality reduction operator and 2) the individual repairing operator, which can be used to guide the evolution of the population, with the aim to significantly improve the search efficiency and obtain feature subsets with high quality. In addition, a strategy to adaptively update the steering matrix is also suggested, which can further improve the performance of SM-MOEA. Finally, the general framework of SM-MOEA is presented. The overall flowchart of SM-MOEA is shown in Fig. 1 and the notations used in this article are summarized in Table I.

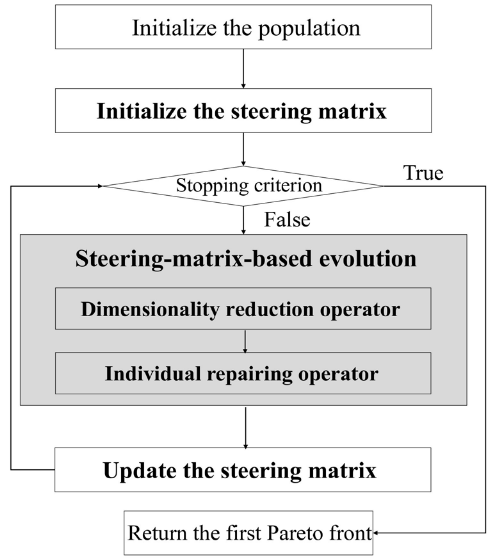


Fig. 1. Overall flowchart of the proposed SM-MOEA.

TABLE I
Notations Used in This Article

$\mathbf{SM}$	steering matrix, where $\mathbf{SM}(i,j)$ denotes the importance of the j-th feature to the i-th individual
$Data$	given dataset
F	set of all features in $Data$
P	population
$FI(F_j)$	importance of the j-th feature on $Data$
$II(P_i)$	importance of the i-th individual on P
$SU(F_j)$	symmetric uncertainty importance of feature F_j to class C on $Data$
$H(F_j)$	entropy of F_j
$H(F_j\|C)$	conditional entropy of F_j for class C
$DR(P_i)$	number of individual P_i dominating others in the current population
$\mathbf{FP}^t(i,j)$	probability of the j-th feature flipping to the i-th individual based on $\mathbf{SM}$ at the t-th generation
P^t	population of the t-th generation
P^{t+1}	population of the (t+1)-th generation
$P^{t'}$	reduced population of P^t
$Elites$	elite individuals of the population

A. Steering Matrix

1) Main Idea: In this study, the steering matrix (denoted as **SM**) was developed for the implementation in SM-MOEA to steer the evolution of individuals in the population, with the aim of solving the challenge of the curse of dimensionality in high-dimensional FS. Specifically, for each feature (suppose the jth feature) of each individual in a population (suppose the ith individual), a steering value [denoted as $\mathbf{SM}(i,j)$] between 0 and 1 is generated and used to determine whether the jth feature is included in the ith individual. Specifically, the larger the value of $\mathbf{SM}(i,j)$ is, the greater the importance of the jth feature to the ith individual. Thus, the jth feature will have a larger probability of being selected in the ith individual. Conversely, a smaller value of $\mathbf{SM}(i,j)$ means that the jth feature is of lower importance to the ith individual; in other words, the jth feature will have a lower probability of being selected in the ith individual.

Given a dataset *Data*, let F be the set of all features in *Data* and P be the population. The steering matrix for population P can be defined as follows.

Definition 1 (Steering Matrix): The steering matrix for P is denoted as **SM** with the size of $|P|$ rows and $|F|$ columns, where each element $\mathbf{SM}(i,j)$ represents the steering value ranging from 0 to 1 for the jth ($j = 1 \cdots |F|$) dimension (feature) of the ith ($i = 1 \cdots |P|$) individual in P. Specifically, each $\mathbf{SM}(i,j)$ in the steering matrix is generated by considering the following two facts.

1) The importance of the jth feature on the dataset, and it is denoted as $\mathrm{FI}(F_j)$.
2) The importance of the ith individual on the population, and it is denoted as $II(P_i)$.

TABLE II
EXAMPLE OF THE STEERING MATRIX

$\mathrm{SM}(i,j)$	F_1	F_2	F_3	F_4	F_5	F_6	F_7
P_1	0.2	0.4	0.3	0.1	0.5	0.9	0.8
P_2	0.3	0.1	0.8	0.6	0.3	0.6	0.7

Formally

$$\mathbf{SM}(i,j) = \mathrm{Norm}\big(\mathrm{FI}\big(F_j\big) \times II(P_i)\big) \tag{3}$$

where $\mathrm{Norm}(x)$ denotes the normalization of x.

Table II presents an example of the steering matrix (**SM**) with seven features (F_1 and F_7) in dataset *Data* and two individuals (P_1 and P_2) in the population P. In this matrix, $\mathbf{SM}(1,6)$ equals 0.9, which is the largest value of the first individual. According to this value, the sixth feature in the first individual has the highest probability of being selected in P_1. In addition, $\mathbf{SM}(1,4)$ equals 0.1, which is the smallest value for the first individual. According to this value, the fourth feature in the first individual has the lowest probability of being selected in P_1. According to **SM**, many unimportant and irrelevant features can be reduced. Thus, the steering matrix can be exploited to guide the evolution of the population by recursively reducing the number of selected features as the population evolves, providing an effective solution to solve the curse of dimensionality.

2) Initialization of the Steering Matrix: The above analysis indicates that steering matrix **SM** is considerably important to overcome the challenge of high-dimensional FS. Thus, in our proposed algorithm, we should first initialize it properly. In what follows, we will give the details of how to design the two important functions: 1) FI(.) and 2) *II*(.) in (3).

The first important information that can be used to initialize the steering matrix is the importance of the feature on the dataset. In this study, the symmetric uncertainty (SU) [45] is adopted to measure the importance of feature F_j ($j = 1, \ldots, |F|$) to class C on the dataset, due to the fact that SU is a nonparametric measure and is frequently used for FS [46]. Specifically, SU is calculated by using the following:

$$\mathrm{FI}\big(F_j\big) = \mathrm{SU}\big(F_j\big) = \frac{H\big(F_j\big) - H\big(F_j|C\big)}{H\big(F_j\big) + H(C)} \tag{4}$$

where $H(F_j)$ is the entropy of F_j, and $H(F_j|C)$ is the conditional entropy of F_j for given C. The value of $\mathrm{SU}(F_j)$ ranges from 0 to 1. The higher the correlation of a feature with the class label, the more important it is. If it equals 1, then F_j is the most relevant feature to class C.

The second important information that can be used to initialize the steering matrix is the importance of the individual on the population. Specifically, the dominant relationship metric DR is used and defined as the number of each individual P_i ($i = 1, \ldots, |P|$) that is dominating others in the current population, and it is expressed as follows:

$$II(P_i) = \mathrm{DR}(P_i) = |\{P_k \in P | P_i \in P \wedge P_i \succ P_k\}| \tag{5}$$

where P_i and P_k are two individuals in population P. $P_i \succ P_k$ means that P_i dominates P_k. The higher the value of DR of an individual, the more important it is.

Algorithm 1: SteeringMatrixIni(Data, *P*)

Input: *Data*: given dataset; *P*: population;
Output: **SM**: steering matrix;
$F \leftarrow$ set of all features in *Data*;
for $j = 1$ to $|F|$ **do**
  Calculate $SU(F_j)$ using Eq. (4);
end for
for $i = 1$ to $|P|$ **do**
  Calculate $DR(P_i)$ using Eq. (5);
end for
for $i = 1$ to $|P|$ **do**
  for $j = 1$ to $|F|$ **do**
    Calculate $\mathbf{SM}(i,j)$ using Eq. (6);
  end for
end for

The steering matrix can be generated naturally by combining the vectors of SU and DR, which is formulated in (6). Note that, because the two suggested measures: 1) DR and 2) SU, have different scales, normalization operators are used. Specifically, DR is normalized by dividing the population size and SU is normalized by dividing the sum of SU values for all features

$$\mathbf{SM}(i,j) = \frac{\mathrm{DR}(P_i)}{|P|} \times \frac{\mathrm{SU}\big(F_j\big)}{\sum_{k=1}^{|F|} \mathrm{SU}(F_k)} \tag{6}$$

where $\mathbf{SM}(i,j)$ represents the steering value for the jth dimension (feature) of the ith individual, $|F|$ is the number of features, and $|P|$ is the population size.

Algorithm 1 provides the main procedure for the initialization of the steering matrix. Fig. 2 presents an example to illustrate the main steps of the initialization of the steering matrix. As shown in this example, there are eight features (F_1–F_8) in the dataset, where F_j denotes the jth feature and C represents the label or class of instance. According to (4), we can obtain vector $\mathrm{SU} = \langle 0.2, 0.6, 0.3, 0.4, 0.4, 0.7, 0.5, 0.1\rangle$ (step 1). We can also find that there are four individuals in the population marked from P_1 to P_4. According to (5), vector $\mathrm{DR} = \langle 3, 1, 0, 2\rangle$ (step 2). $\mathrm{DR}(P_1) = 3$ implies that the first individual P_1 dominates the other three individuals in the population. Based on the two derived vectors: 1) SU and 2) DR, this steering matrix can be obtained by using (6) (step 3). Taking the calculation of $\mathbf{SM}(1,2)$ as an example, according to (6), $\mathbf{SM}(1,2) = ([\mathrm{DR}(P_1) \times \mathrm{SU}(F_2)]/[4 \times \sum_{k=1}^{8} \mathrm{SU}(F_k)]) = (1.8/12.8) \approx 0.14$.

B. Steering-Matrix-Based Evolutionary Operators

In this study, we adopt binary encoding as the individual representation, which is popularly used in EC-based FS algorithms [7]. Specifically, if the jth feature of the ith individual is selected, the corresponding bit is set as 1, that is, $P(i,j) = 1$; otherwise, $P(i,j) = 0$. The key challenge in evolutionary algorithms for high-dimensional FS is to overcome the search inefficiency caused by the curse of dimensionality. To this end, based on the steering matrix derived earlier, two operators (the dimensionality reduction operator and individual repairing operator) are proposed to significantly improve the search efficiency and obtain high-quality feature subsets. The

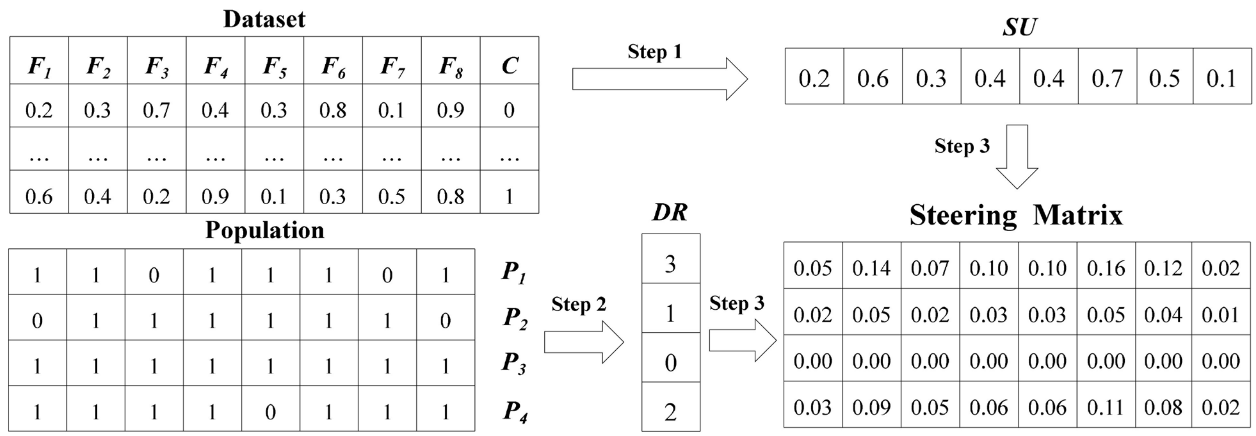


Fig. 2. Example to illustrate the initialization of the steering matrix.

two steering-matrix-based evolutionary operators are defined in the following section.

1) Steering-Matrix-Based Dimensionality Reduction Operator: As mentioned earlier, the search space of a high-dimensional FS problem becomes very large due to the curse of dimensionality, which reaches $2^{|F|}$ for a dataset with $|F|$ features [7]. To this end, we suggest a steering-matrix-based dimensionality reduction operator to improve the search efficiency considerably. The basic idea originates from the following intuition: in high-dimensional FS, population individuals at the beginning of evolution may have many irrelevant and redundant features; thus, unnecessary search space can be eliminated by focusing on certain important features. As the evolution proceeds, the search should gradually become more fine-grained.

Based on this intuition, we designed a steering-matrix-based dimensionality reduction operator to recursively downsize the number of selected features in population individuals as the evolution proceeds, where a dimensionality reduction flip probability for turning $P(i,j)$ from 1 to 0 is used for dimensionality reduction. Specifically, based on steering matrix $\mathbf{SM}$, flip probability $\mathbf{FP}^t(i,j)$ for the jth feature to the ith individual at the t-th generation is defined as follows:

$$\mathbf{FP}^t(i,j) = \frac{\exp^{-\gamma \cdot t}}{1 + \exp^{\left(\mathbf{SM}(i,j) - \overline{\mathbf{SM}(i)}\right)}} \tag{7}$$

where γ denotes the attenuation factor controlling the rate of zeroing for $P(i,j)$, t denotes the current number of generations, and $\overline{\mathbf{SM}}(i)$ denotes the average steering value of all features in the ith individual. The denominator of (7) indicates that if the value of the jth feature in the steering matrix is larger than the average value of all features in the ith individual, then the jth feature is more important than most of the features; thus, the probability of $P(i,j)$ flipping from 1 to 0 becomes smaller. Otherwise, if the value is smaller than the average value, then the probability of $P(i,j)$ flipping from 1 to 0 becomes larger. In addition, in the numerator of (7), parameter t adaptively controls the population reduction effect; that is, t promotes population reduction very quickly in the early stages of evolution, and its effect declines as the population evolves. Assume that $P(i,j) = 1$, if the value of $\mathbf{FP}^t(i,j)$ is close to 1, it implies that the jth feature of the ith individual at the t-th generation has a very high probability of flipping $P(i,j)$ from 1 to 0.

Algorithm 2: DimensionalityReduction($\mathbf{SM}$, P^t, γ)

Input: $\mathbf{SM}$: steering matrix; P^t: population of the t-th generation; γ: attenuation factor;

Output: $P^{t'}$: population after being processed by the reduction operator;

$P^{t'} \leftarrow P^t$;
Elites $\leftarrow$ non_dominated_sorting(P^t);
for $i = 1$ to $|P|$ **do**
  if P^t_i is not in *Elites* **then**
    for $j = 1$ to $|F|$ **do**
      Calculate $\mathbf{FP}^t(i,j)$ using Eq. (7);
      if $P^t(i,j) == 1$ and $\mathbf{FP}^t(i,j) \geq$ rand() **then**
        $P^{t'}(i,j) \leftarrow 0$;
      end if
    end for
  end if
end for

Algorithm 2 shows the dimensionality reduction procedure by using the steering matrix. The algorithm operates as follows. First, the elite individuals denoted as *Elites* from P^t (the population of the t-th generation) are obtained by using a nondominated sorting method [24]. Then, we only focus on processing the nonelite individuals in P^t by dimensionality reduction. For each nonelite individual, P^t_i $(1 \leq i \leq |P^t|)$, the probability of flipping $P^t(i,j)$ from 1 to 0 [i.e., $\mathbf{FP}^t(i,j)$] is computed according to (7) for the jth feature in the ith individual. If $P^t(i,j) == 1$ and $\mathbf{FP}^t(i,j) \geq$ rand() (a random number between 0 and 1), then $P^t(i,j)$ is set to 0, which means that the jth feature should be deleted from the ith individual.

2) Steering-Matrix-Based Individual Repairing Operator: Although the dimensionality reduction operator based on the steering matrix can significantly improve the search efficiency, certain important features may be incorrectly deleted from individuals in the population. To this end, we suggest an individual repairing operator to further improve the quality of the obtained feature subsets. The main idea is to use the steering matrix values of the elite individuals in the population to repair the nonelite individuals.

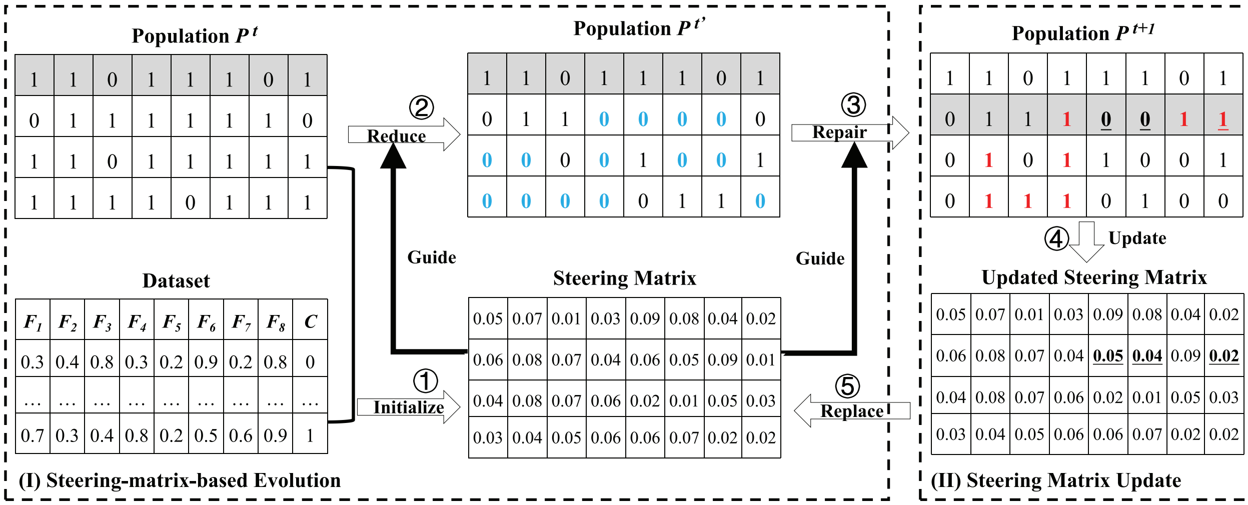


Fig. 3. Example to illustrate the steering-matrix-based evolution and the steering matrix update.

Algorithm 3: IndividualRepairing(**SM**, P^t, $P^{t'}$)

Input: **SM**: steering matrix;
P^t: population of the t-th generation;
$P^{t'}$: population reduced by Algorithm 2 based on P^t;
Output: P^{t+1}: population of the (t+1)-th generation;

$P^{t+1} \leftarrow P^{t'}$;
Elites $\leftarrow$ non_dominated_sorting(P^t);
for $i = 1$ to $|P^t|$ **do**
  if P_i^t is not in *Elites* **then**
    $P_e^t \leftarrow$ one elite individual randomly selected from *Elites*;
    for $j = 1$ to $|F|$ **do**
      if $\mathbf{SM}(i,j) >= \mathbf{SM}(e,j)$ **then**
        if $P^t(e,j) == 1$ **then**
          $P^{t+1}(i,j) \leftarrow P^t(i,j)$;
        end if
      else
        Calculate ρ using Eq. (8);
        if $\rho \geq$ rand() **then**
          $P^{t+1}(i,j) \leftarrow P^{t+1}(e,j)$;
        end if
      end if
    end for
  end if
end for

The pseudocode of the steering-matrix-based individual repairing operator is shown in Algorithm 3, the operation of which is described as follows. Let P^t and P^{t+1} be the population of the t-th generation and $(t+1)$th generation, respectively, and $P^{t'}$ be the reduced population by Algorithm 2 based on P^t. First, we initialize P^{t+1} with $P^{t'}$ and use the nondominated sorting method to choose the individuals that are in the first Pareto front of P^t as the elite individuals *Elites*. Then, we focus on the nonelite individuals in P^t to repair their corresponding ones in P^{t+1}. For each nonelite individual P_i^t, one elite individual P_e^t is randomly selected from *Elites* for repairing P_i^{t+1}. Specifically, we compare the steering values in **SM** for P_i^t and P_e^t for each feature F_j $(1 \leq j \leq |F|)$. There are two situations for the comparison between $\mathbf{SM}(i,j)$ and $\mathbf{SM}(e,j)$, which is depicted as follows.

1) If $\mathbf{SM}(i,j)$ is larger than or equal to $\mathbf{SM}(e,j)$, it means that feature F_j in individual P_i^t is not worse than P_e^t. Considering that if feature F_j is selected in P_e^t, then this feature in P_i^t should also have a high probability of being selected. If the scalar value of $P^t(i,j)$ is set from 1 to 0 in the dimensionality reduction stage, then it should revert to 1 in this stage. Thus, we use the scalar value of $P^t(i,j)$ as the value of $P^{t+1}(i,j)$.
2) If $\mathbf{SM}(i,j)$ is smaller than $\mathbf{SM}(e,j)$, which means feature j in individual P_i^t is worse than that of elite individual P_e^t. As a result, P_i^{t+1} should learn from P_e^{t+1} for feature F_j with probability ρ, as expressed by the following:

$$\rho = \frac{|\mathbf{SM}(e,j) - \mathbf{SM}(i,j)|}{\mathbf{SM}(e,j)}. \tag{8}$$

By using the two suboperators above, the possible errors caused by the fast dimensionality reduction operator can be repaired and the quality of the obtained feature subset may be further improved.

Fig. 3 (left panel) depicts an illustrative example of the steering-matrix-based evolution (dimensionality reduction and individual repairing), where the current population P^t of the t-th generation contains four individuals, and elite individuals are represented in gray. In P^t, the first individual is an elite individual and the remaining three individuals are nonelite. In the first stage of dimensionality reduction, reduced population $P^{t'}$ can be obtained on the basis of P^t by using Algorithm 2, where each element in $P^{t'}$ different from P^t is indicated in blue in boldface. For example, in the second individual in P^t, the values of the fourth, fifth, sixth, and seventh features are flipped from 1 to 0. In the second stage that entails individual repairs based on $P^{t'}$, the repaired population P^{t+1} can be obtained by applying Algorithm 3, where each element in P^{t+1} different from $P^{t'}$ is denoted in red and boldface. For example, the fourth feature of the second individual in $P^{t'}$ is flipped from 1 to 0 after dimensionality reduction and its steering value is 0.04, which is larger than that of the fourth feature of the elite individual (0.03). Moreover, due to the fact this feature is selected by the elite individual in $P^{t'}$, thus this feature in the second individual should revert to its original state, 1 [i.e.,

Algorithm 4: SteeringMatrixUpdating (**SM**, P^t, P^{t+1}, Data)

Input: **SM**: steering matrix;
P^t: parent population;
P^{t+1}: offspring population after applying Algorithm 2 and Algorithm 3;
Data: given dataset;
Output: **SM**: updated steering matrix;
Elites^t ← non_dominated_sorting(P^t);
Elites^{t+1} ← non_dominated_sorting(P^{t+1});
if the average error rate of *Elites*$^{t+1}$ is lower than that of *Elites*^{t} **then**
  for $i = 1$ to $|P^{t+1}|$ **do**
    if P_i^{t+1} is in *Elites*$^{t+1}$ **then**
      for $j = 1$ to $|F|$ **do**
        Update $\mathbf{SM}(i, j)$ using Eq. (9);
      end for
    end if
  end for
else
  SM=SteeringMatrixIni(*Data*, P^{t+1});
end if

Algorithm 5: Framework of SM-MOEA

Input: *Data*: given dataset;
$G_{\max}$: maximum number of generations;
γ: attenuation factor;
Output: *PF*: set of feature subsets in the Pareto front of the final population;
P^0 ← obtain the initial population by using the initialization method in [42];
SM ← SteeringMatrixIni(*Data*, P^0);
$t \leftarrow 1$;
while $t \leq G_{\max}$ **do**
  $P^t \leftarrow P^{t-1}$;
  $P^{t'}$ ← DimensionalityReduction (**SM**, P^t, γ);
  P^{t+1} ← IndividualRepairing (**SM**, P^t, $P^{t'}$);
  SM ← SteeringMatrixUpdating (**SM**, P^t, P^{t+1}, *Data*);
  $t{+}{+}$;
end while
PF ← *ParetoSolutions*(P^t);

$P^{t+1}(2, 4) = 1$]. The eighth feature of the second individual in $P^{t'}$, is still 0 after dimensionality reduction, whose steering value is 0.01, which is smaller than that of the elite individual (0.02). Thus, this feature in the second individual should learn from the elite individual with probability $\rho = 0.5$ calculated by using (8). In this example, if rand() is 0.3, which is smaller than ρ, then the eighth feature of the second individual should learn from the elite individual. Since $P^{t+1}(1, 8) = 1$, thus $P^{t+1}(2, 8) = 1$.

C. Steering Matrix Update

In order to further improve the effectiveness of the steering matrix, it is necessary to update the steering matrix at each generation. To this end, we propose a reward-and-punish-based strategy for adaptively updating the steering matrix **SM**. The pseudocode of the steering matrix updating strategy is provided in Algorithm 4. Let P^t be the parent population and P^{t+1} be the offspring population after the dimensionality reduction (Algorithm 2) and individual repairing (Algorithm 3). The algorithm to update the steering matrix operates as follows.

First, based on parent population P^t and offspring population P^{t+1}, the elite individuals *Elites*^{t} and *Elites*$^{t+1}$ can be obtained, respectively. Then, we can obtain the average error rate of *Elites*^{t} and *Elites*$^{t+1}$. If the average error rate of *Elites*$^{t+1}$ is not lower than that of *Elites*^{t}, indicating that the (t+1)th evolution by the steering matrix is not effective, then the steering matrix should be reinitialized by Algorithm 1 (punished). Otherwise, if the average error rate of *Elites*$^{t+1}$ is lower than that of *Elites*^{t}, implying that the guide of the population from P^t to P^{t+1} is effective, then the values in the steering matrix of elite individuals in P^{t+1} should be updated (rewarded). Specifically, for each elite individual P_i^{t+1} in *Elites*$^{t+1}$, if $P^t(i, j) == 0$ and $P^{t+1}(i, j) == 1$, then the value of $\mathbf{SM}(i, j)$ should be increased. Otherwise, if $P^t(i, j) == 1$ and $P^{t+1}(i, j) == 0$, then the value of $\mathbf{SM}(i, j)$ should be decreased. In this study, we use the update formulation in (9), which is a modification of a well-known parameter tuning technique, that is, the 1/5 successful rule [47]

$$\mathbf{SM}(i, j) = \mathbf{SM}(i, j) \cdot \exp^{\frac{1}{\sqrt{2}} \times \left[\Phi_{x!=y} \times \left(\Phi_{y>x} - \frac{1}{5}\right)\right]} \tag{9}$$

where $x = P^t(i, j)$, $y = P^{t+1}(i, j)$, and Φ_α is an indicator function that returns 1 if α is true and 0 otherwise. For example, if $P^t(i, j) == 0$ and $P^{t+1}(i, j) == 1$, then $\mathbf{SM}(i, j)$ is increased to $\mathbf{SM}(i, j) \times \exp^{(1/\sqrt{2}) \times (4/5)}$. Otherwise, if $P^t(i, j) == 1$ and $P^{t+1}(i, j) == 0$, then $\mathbf{SM}(i, j)$ is decreased to $\mathbf{SM}(i, j) \times \exp^{(1/\sqrt{2}) \times (-1/5)}$.

We also consider parent population P^t and offspring population P^{t+1} shown in Fig. 3 as an example. In this example, the second individual of P^{t+1}, represented in gray, is an elite one. According to the strategy for updating the steering matrix, only the values for the 5th, 6th, and 8th features in the 2nd individual of P^{t+1}, which are different from P^t, are updated by (9), and the adjusted values of the updated steering matrix are indicated in boldface and are underlined.

D. General Framework of SM-MOEA

The general framework of the proposed SM-MOEA is presented in Algorithm 5. Let *Data* be a given dataset, $G_{\max}$ be the maximum number of generations, and γ be the attenuation factor. SM-MOEA consists of the following three steps. In the first step, population P is initialized by the method used in [42], which can accelerate the training procedure of SM-MOEA. Then, a steering matrix **SM** is initialized by the proposed Algorithm 1. In the second step, the steering-matrix-based dimensionality reduction, shown in Algorithm 2, and individual repairing, depicted in Algorithm 3, are used to produce the next population. Then, the steering matrix is updated adaptively by using the proposed Algorithm 4. The process of dimensionality reduction, individual repairing, and steering matrix updating are repeated in the second step until the maximal generation, $G_{\max}$, is reached. In the last step, final population P^t is obtained. From P^t, the nondominated sorting method is adopted to obtain the optimal solutions (i.e., *PF*) in the first front.

IV. Experimental Results

This section presents a series of experiments that are conducted to empirically validate the performance of the proposed SM-MOEA on 12 benchmark high-dimensional datasets, whose feature sizes ranging from 3000 to 13 000. To this end, we compare the proposed SM-MOEA with five existing up-to-date evolutionary algorithms (two single-objective and three multiobjective EC-based algorithms) for FS. In the following, we first present the experimental settings including the comparison algorithms, benchmark datasets, and the adopted parameter settings and then report and analyze the experimental results in detail. Finally, we validate the effectiveness of the suggested strategies in SM-MOEA.

A. Experimental Settings

1) Comparison Algorithms: The performance of the SM-MOEA is compared with five representative FS evolutionary algorithms, namely: 1) CSO [39]; 2) VLPSO [28]; 3) SparseEA [42]; 4) Bin-MOABC [15]; and 5) MOEA/D-DYN [44], where CSO and VLPSO are single-objective EC-based methods, whereas the other three algorithms are multiobjective EC-based ones.

The two state-of-the-art single-objective evolutionary algorithms, CSO and VLPSO, are chosen for comparison because these two methods have the ability to process high-dimensional FS. Specifically, CSO is a variant of PSO [48] and adopts a competitive mechanism. In this method, all particles are divided into two groups where the pairwise comparison is applied and the more appropriate particle between the two is used as an exemplar for the other. VLPSO is a variant of comprehensive learning PSO [49], which enables particles to have different lengths in the evolution; therefore, it narrows the search space and can solve the high-dimensional FS effectively.

Three up-to-date MOEA-based FS algorithms, called Bin-MOABC, SparseEA, and MOEA/D-DYN, are considered as the baselines because these three algorithms can solve FS problems with high-dimensional decision variables. Bin-MOABC is a novel multiobjective ABC algorithm integrated with a nondominated sorting procedure and genetic operators where "Bin" means binary coding. SparseEA is a general algorithm that provides a new population initialization strategy and genetic operators by considering the sparse nature of the Pareto optimal solutions and thus can be successfully applied in solving high-dimensional FS. MOEA/D-DYN is a recently suggested MOEA for FS tasks. In that work, two mechanisms (static and dynamic) based on multiple reference points in the MOEA/D framework are first designed to address the difficulties (e.g., a highly discontinuous Pareto front, imbalance preferences, and partially conflicting objectives) in FS. Then, based on these two mechanisms, two algorithms" 1) MOEA/D-STAT and 2) MOEA/D-DYN, are proposed. The experimental results show that MOEA/D-DYN outperforms MOEA/D-STAT [44]. Therefore, in our experiments, we adopt MOEA/D-DYN as the comparision algorithm.

TABLE III
Statistics of the 12 Datasets

No.	Datasets	#Features	#Instances	#Classes
1	Lung	3,312	203	5
2	GLIOMA	4,434	50	4
3	Leukemia 1	5,327	72	3
4	DLBCL	5,469	77	2
5	9Tumor	5,726	60	9
6	Brain Tumor 1	5,920	90	5
7	Carcinom	9,182	174	11
8	Brain Tumor 2	10,367	50	4
9	Prostate	10,509	102	2
10	Leukemia 2	11,225	72	3
11	CLLSUB	11,340	111	3
12	11Tumor	12,533	174	11

TABLE IV
Parameter Values of the Baseline Algorithms Originally Recommended and Those of the Proposed SM-MOEA

Algorithm	Recommended parameter values
CSO [39]	$\phi = 0.1$, $Threshold\ for\ selected\ feature = 0.6$
VLPSO [28]	$\alpha = 7$, $\beta = 9$, $Divisions = 12$, $Threshold\ for\ selected\ feature = 0.6$
Bin-MOABC [15]	$Limit = 50$, $T = 10000$
SparseEA [42]	$Mutation = 1/Dimension$
MOEA/D-DYN [44]	$R = 100$, $T = \max\{R/10, 4\}$, $\sigma = 0.85$, $\alpha = 0.01$, $M = 0.4 * R$, $I = 4$ $Threshold\ for\ selected\ feature = 0.6$
SM-MOEA	$\gamma = 0.1$

2) Datasets: To examine the performance of the proposed SM-MOEA, extensive experiments are conducted on 12 high-dimensional datasets[1] that contain features and instances of different sizes. These datasets have been used in previous studies on high-dimensional FS [28], [30], [50]. Table III lists the detailed characteristics (i.e., the number of features, instances, and classes) of the adopted datasets, where the number of features is in ascending order.

For each dataset D, we first use ten-fold cross-validation to create the training and test sets. We consider one-fold as an example, in which 90% of the instances in D are chosen as training set D_{train}, and the remaining 10% are used as test set D_{test}. We utilize D_{train} to obtain feature subset F and adopt D_{test} to evaluate the quality of the obtained feature subset, F. During the evolution, 1NN with leave-one-out-cross-validation on set D_{train} is applied to calculate the classification error rate of individuals in the population; this implies that in the evolutionary process, each instance in D_{train} is used as the test set for 1NN, whereas the remaining instances in D_{train} are used to learn the 1NN classifier. By doing so, the FS bias can be avoided. To further reduce the variance, ten-fold cross-validation on each dataset is performed independently 30 times, which we refer to as 30 independent runs. We report the average results of the different algorithms on 30 independent runs, and the Wilcoxon rank-sum test with a significance level of 0.05 is adopted to perform statistical analysis.

3) Parameter Settings: For the five selected comparison algorithms, we adopt the originally recommended parameter values suggested in their original papers. Table IV lists the

[1]These datasets can be downloaded from "http://www.gems-system.org".

TABLE V
AVERAGE TEST RESULTS OF SM-MOEA VERSUS SINGLE-OBJECTIVE ALGORITHMS

Dataset (Size)	Algorithm	Mean Size ± Std	S_1	Mean Error ± Std	S_2
Lung (3,312)	CSO	1153.3 ± 242.3	-	**0.0557 ± 0.04**	+
	VLPSO	253.3 ± 172.3	-	0.1461 ± 0.12	-
	SM-MOEA	**36.1 ± 34.1**		0.1177 ± 0.06	
GLIOMA (4,434)	CSO	1273.9 ± 186.7	-	0.3248 ± 0.09	-
	VLPSO	217.9 ± 193.6	-	0.2833 ± 0.23	-
	SM-MOEA	**15.3 ± 12.2**		**0.1714 ± 0.17**	
Leukemia 1 (5,327)	CSO	1924.6 ± 240.6	-	0.3273 ± 0.06	-
	VLPSO	46.3 ± 31.3	-	**0.0349 ± 0.04**	≈
	SM-MOEA	**23.6 ± 36.2**		0.0557 ± 0.09	
DLBCL (5,469)	CSO	1820.6 ± 230.3	-	**0.0345 ± 0.05**	≈
	VLPSO	49.0 ± 32.9	-	0.1100 ± 0.13	-
	SM-MOEA	**11.2 ± 8.2**		0.0378 ± 0.06	
9Tumor (5,726)	CSO	2072.1 ± 355.3	-	0.6255 ± 0.07	-
	VLPSO	47.1 ± 18.7	-	**0.4833 ± 0.15**	+
	SM-MOEA	**19.8 ± 16.6**		0.5632 ± 0.23	
Brain Tumor 1 (5,920)	CSO	1977.4 ± 297.8	-	0.2985 ± 0.13	-
	VLPSO	25.8 ± 12.7	-	0.4042 ± 0.20	-
	SM-MOEA	**22.5 ± 18.7**		**0.0972 ± 0.14**	
Carcinom (9,182)	CSO	3521.6 ± 574.3	-	0.1893 ± 0.02	-
	VLPSO	125.5 ± 78.1	-	0.1418 ± 0.07	-
	SM-MOEA	**21.3 ± 16.9**		**0.0970 ± 0.12**	
Brain Tumor 2 (10,367)	CSO	3224.2 ± 666.3	-	0.4662 ± 0.09	-
	VLPSO	73.7 ± 56.4	-	0.3333 ± 0.19	-
	SM-MOEA	**12.5 ± 11.9**		**0.0667 ± 0.11**	
Prostate (10,509)	CSO	3866.9 ± 677.1	-	0.1382 ± 0.04	-
	VLPSO	17.5 ± 7.3	-	0.0750 ± 0.06	-
	SM-MOEA	**9.1 ± 8.1**		**0.0143 ± 0.03**	
Leukemia 2 (11,225)	CSO	3565.8 ± 568.7	-	0.1322 ± 0.19	-
	VLPSO	39.2 ± 11.4	-	0.0889 ± 0.07	≈
	SM-MOEA	**14.7 ± 11.7**		**0.0844 ± 0.08**	
CLLSUB (11,340)	CSO	4953.1 ± 737.5	-	0.5172 ± 0.11	-
	VLPSO	148.6 ± 164.3	-	0.2622 ± 0.14	-
	SM-MOEA	**16.5 ± 28.3**		**0.1866 ± 0.15**	
11Tumor (12,533)	CSO	4591.1 ± 570.9	-	0.2493 ± 0.07	≈
	VLPSO	216.1 ± 103.3	-	**0.2244 ± 0.08**	≈
	SM-MOEA	**25.2 ± 16.6**		0.2330 ± 0.09	

details of the parameters in the comparison algorithms and SM-MOEA. For a fair comparison, in all these evolutionary algorithms, the population size is set as 100, and the maximum number of generations is fixed as 100. In SM-MOEA, attenuation parameter γ, which controls the speed of downsizing the search space, is set to 0.1. The experiments are conducted on a server with a 2.2-GHz Intel Xeon CPU, 48-GB RAM, and the Windows 10 operating system.

B. Results of the Comparison Between SM-MOEA and Baselines

1) SM-MOEA Versus Single-Objective Algorithms: Table V presents the average test error rate and feature subset size achieved by SM-MOEA and the two single-objective algorithms (i.e., CSO and VLPSO) on the 12 high-dimensional datasets. Note that CSO and VLPSO obtain only one feature subset in each run, whereas SM-MOEA obtains a set of nondominated feature subsets in one run. For a fair comparison, in each run, the objectives of nondominated feature subsets obtained by SM-MOEA are averaged.

As shown in this table, the third and fifth columns show the average feature subset sizes (with the standard deviation) and the average error rates (with the standard deviation) of all the solutions that are obtained in 30 independent runs. The smallest average size and the lowest average error rate obtained for each dataset are marked in bold. Columns S_1 and S_2, respectively, display the Wilcoxon significance test results of the average feature subset sizes and average error rate with a significance of 0.05, where symbols "+," "−," and "≈" denote that the result obtained by the baselines is significantly more accurate, significantly less accurate, and statistically similar to that obtained by the proposed method, SM-MOEA. In other words, the greater the number of "−," the more accurate the proposed method.

From Table V, it can be found that the average results of SM-MOEA are superior to those of CSO and VLPSO in almost all cases, especially on extremely high-dimensional datasets. Specifically, SM-MOEA obtains the smallest feature subsets in all cases, whereas it wins 17, ties 5, and loses 2 out of 24 comparisons in terms of the average error rate. The experimental results verify that our SM-MOEA is more effective than CSO and VLPSO on high-dimensional datasets by finding a set of superior tradeoff solutions with a few features.

2) SM-MOEA Versus Multiobjective Algorithms: In this section, we further validate the competitiveness of the proposed SM-MOEA by comparing it with three state-of-the-art MOEAs (i.e., Bin-MOABC, SparseEA, and MOEA/D-DYN) for high-dimensional FS. Table VI presents the comparison results among the four MOEAs on 12 high-dimensional datasets. Note that in Table VI, the same symbols are used as in Table V, where the additional column, hypervolume (HV), represents the HV value [51], which is a widely used metric to evaluate the diversity and convergence of different MOEAs. S_3 displays the Wilcoxon significance test results of HV with a significance of 0.05. The smallest average size, lowest average error rate, and highest average HV values obtained on each dataset are boldfaced.

As indicated by the results in Table VI, when measured by the number of selected features, the proposed SM-MOEA performs best on ten datasets and exhibits the second-best performance on the remaining two datasets. In addition, the comparison in terms of the error rate shows that SM-MOEA outperforms the other MOEAs on most datasets. Furthermore, the HV values of the different algorithms indicate that MOEA/D-DYN is the best method, whereas the proposed SM-MOEA is the second best. The highest HV of MOEA/D-DYN is attributed to its decomposition mechanisms, which can achieve both high convergence and good diversity. From the statistics in Table VI, we can conclude that SM-MOEA is a competitive multiobjective algorithm for solving high-dimensional FS problems.

The advantages of SM-MOEA are further illustrated in Fig. 4, in which the nondominated solutions obtained by Bin-MOABC, SparseEA, MOEA/D-DYN, and SM-MOEA in the objective space are plotted. From Fig. 4 we can find that the nondominated solutions obtained by SM-MOEA achieve superior performance over other up-to-date MOEAs on most datasets. In other words, SM-MOEA can use a smaller number of features to make the learned classification model more accurate.

TABLE VI
AVERAGE TEST RESULTS OF SM-MOEA VERSUS MULTIOBJECTIVE ALGORITHMS

Dataset (Size)	Algorithm	Mean Size ± Std	S_1	Mean Error ± Std	S_2	HV ± Std	S_3
Lung	Bin-MOABC	1064.6 ± 117.0	-	**0.0387 ± 0.03**	+	0.7026 ± 0.03	-
(3,312)	SparseEA	**11.9 ± 12.7**	+	0.0961 ± 0.06	+	0.9450 ± 0.03	+
	MOEA/D-DYN	113.3 ± 58.5	-	0.0428 ± 0.05	+	**0.9775 ± 0.02**	+
	SM-MOEA	36.1 ± 34.1		0.1177 ± 0.06		0.9057 ± 0.05	
GLIOMA	Bin-MOABC	1553.1 ± 100.1	-	0.2069 ± 0.17	-	0.5985 ± 0.11	-
(4,434)	SparseEA	16.3 ± 17.6	-	0.3622 ± 0.23	-	0.7873 ± 0.17	≈
	MOEA/D-DYN	82.6 ± 52.7	-	0.1894 ± 0.19	-	**0.9479 ± 0.05**	+
	SM-MOEA	**15.3 ± 12.1**		**0.1714 ± 0.17**		0.8238 ± 0.13	
Leukemia 1	Bin-MOABC	2017.0 ± 22.5	-	0.1179 ± 0.09	-	0.5800 ± 0.09	-
(5,327)	SparseEA	**16.8 ± 16.6**	+	0.0794 ± 0.11	-	0.9779 ± 0.05	≈
	MOEA/D-DYN	66.7 ± 26.4	-	0.0619 ± 0.09	≈	**0.9907 ± 0.01**	+
	SM-MOEA	23.6 ± 36.2		**0.0557 ± 0.09**		0.9765 ± 0.07	
DLBCL	Bin-MOABC	2019.6 ± 89.9	-	0.1077 ± 0.09	-	0.5988 ± 0.07	-
(5,469)	SparseEA	28.9 ± 32.4	-	0.1292 ± 0.11	-	0.9490 ± 0.08	≈
	MOEA/D-DYN	56.1 ± 24.0	-	0.0620 ± 0.11	-	**0.9819 ± 0.02**	+
	SM-MOEA	**11.2 ± 8.2**		**0.0378 ± 0.06**		0.9527 ± 0.07	
9Tumor	Bin-MOABC	2193.7 ± 22.4	-	0.5761 ± 0.16	≈	0.3336 ± 0.10	-
(5,726)	SparseEA	33.9 ± 31.8	-	0.6534 ± 0.17	-	0.4942 ± 0.12	≈
	MOEA/D-DYN	176.9 ± 81.7	-	**0.5518 ± 0.16**	≈	**0.6458 ± 0.06**	+
	SM-MOEA	**19.8 ± 16.6**		0.5632 ± 0.23		0.4694 ± 0.15	
Brain Tumor 1	Bin-MOABC	2292.0 ± 22.3	-	0.1816 ± 0.12	-	0.5306 ± 0.11	-
(5,920)	SparseEA	23.4 ± 22.3	-	0.2778 ± 0.10	-	0.7976 ± 0.07	≈
	MOEA/D-DYN	115.8 ± 70.9	-	0.1400 ± 0.09	-	**0.9280 ± 0.03**	+
	SM-MOEA	**22.5 ± 18.7**		**0.0972 ± 0.14**		0.8482 ± 0.12	
Carcinom	Bin-MOABC	3857.0 ± 36.0	-	0.1232 ± 0.05	-	0.5264 ± 0.05	-
(9,182)	SparseEA	47.2 ± 51.5	-	0.1870 ± 0.11	-	0.9201 ± 0.05	≈
	MOEA/D-DYN	296.7 ± 167.3	-	0.1020 ± 0.07	≈	**0.9357 ± 0.05**	+
	SM-MOEA	**21.3 ± 16.9**		**0.0970 ± 0.12**		0.8873 ± 0.11	
Brain Tumor 2	Bin-MOABC	4435.2 ± 29.9	-	0.3000 ± 0.21	-	0.4334 ± 0.12	-
(10,367)	SparseEA	105.8 ± 162.6	-	0.2623 ± 0.24	-	0.7422 ± 0.22	-
	MOEA/D-DYN	140.4 ± 79.3	-	0.1630 ± 0.14	-	**0.9247 ± 0.06**	≈
	SM-MOEA	**12.5 ± 11.9**		**0.0667 ± 0.11**		0.8422 ± 0.16	
Prostate	Bin-MOABC	4477.6 ± 126.9	-	0.1761 ± 0.08	-	0.5304 ± 0.07	-
(10,509)	SparseEA	88.0 ± 90.3	-	0.1306 ± 0.08	-	0.9234 ± 0.05	≈
	MOEA/D-DYN	142.4 ± 73.4	-	0.0743 ± 0.07	-	**0.9641 ± 0.03**	≈
	SM-MOEA	**9.1 ± 8.1**		**0.0143 ± 0.03**		0.9575 ± 0.06	
Leukemia 2	Bin-MOABC	4836.1 ± 31.9	-	0.1198 ± 0.10	-	0.5625 ± 0.05	-
(11,225)	SparseEA	136.2 ± 182.4	-	0.1970 ± 0.16	-	0.8786 ± 0.13	≈
	MOEA/D-DYN	87.5 ± 29.6	-	**0.0496 ± 0.07**	+	**0.9900 ± 0.01**	+
	SM-MOEA	**14.7 ± 11.7**		0.0844 ± 0.08		0.8916 ± 0.08	
CLLSUB	Bin-MOABC	4899.7 ± 33.4	-	0.4479 ± 0.14	-	0.3832 ± 0.08	-
(11,340)	SparseEA	97.8 ± 97.0	-	0.3233 ± 0.14	-	0.7982 ± 0.11	-
	MOEA/D-DYN	384.0 ± 301.7	-	0.3206 ± 0.13	-	0.8119 ± 0.06	-
	SM-MOEA	**16.5 ± 28.3**		**0.1866 ± 0.15**		**0.8456 ± 0.09**	
11Tumor	Bin-MOABC	5481.7 ± 34.2	-	0.1811 ± 0.09	+	0.4684 ± 0.06	-
(12,533)	SparseEA	92.9 ± 106.9	-	0.3388 ± 0.13	-	0.7785 ± 0.09	≈
	MOEA/D-DYN	307.1 ± 146.7	-	**0.1576 ± 0.08**	+	**0.9140 ± 0.04**	+
	SM-MOEA	**25.2 ± 16.6**		0.2330 ± 0.09		0.8143 ± 0.06	

From the above empirical results, we can conclude that the proposed SM-MOEA is a promising method in comparison with existing evolutionary algorithms for high-dimensional FS. The better performance of SM-MOEA is attributed to the proposed steering matrix, which utilizes both the importance of the feature on the dataset and that of the individual on the population, thus approximating the importance of features in different individuals. Subsequently, most of the irrelevant and redundant features of individuals in populations can be removed and repaired by the proposed steering-matrix-based dimensionality reduction and individual strategies. In addition, the initializing and updating strategies for the steering matrix can be used to further improve the overall performance of SM-MOEA. In the following section, we demonstrate the effectiveness of these proposed strategies in SM-MOEA.

C. *Effectiveness of the Proposed Strategies in SM-MOEA*

In this section, we verify the effectiveness of the proposed strategies used in SM-MOEA, namely, the steering matrix's initialization strategy, steering-matrix-based individual repairing strategy, and the steering-matrix's update strategy. Fig. 5 shows the comparison results of the proposed SM-MOEA and its three variants: 1) SM-MOEA-RI; 2) SM-MOEA-WR; and 3) SM-MOEA-WU, on 12 datasets, where all the components remained the same as those in SM-MOEA except for the steering matrix initialization, which was replaced by a random initialization (RI) strategy, dimensionality reduction without individual repairing (WR) strategy, and steering matrix without adaptive updating (WU) strategy.

As observed in Fig. 5, the proposed SM-MOEA outperforms the other three variants in terms of both the number of

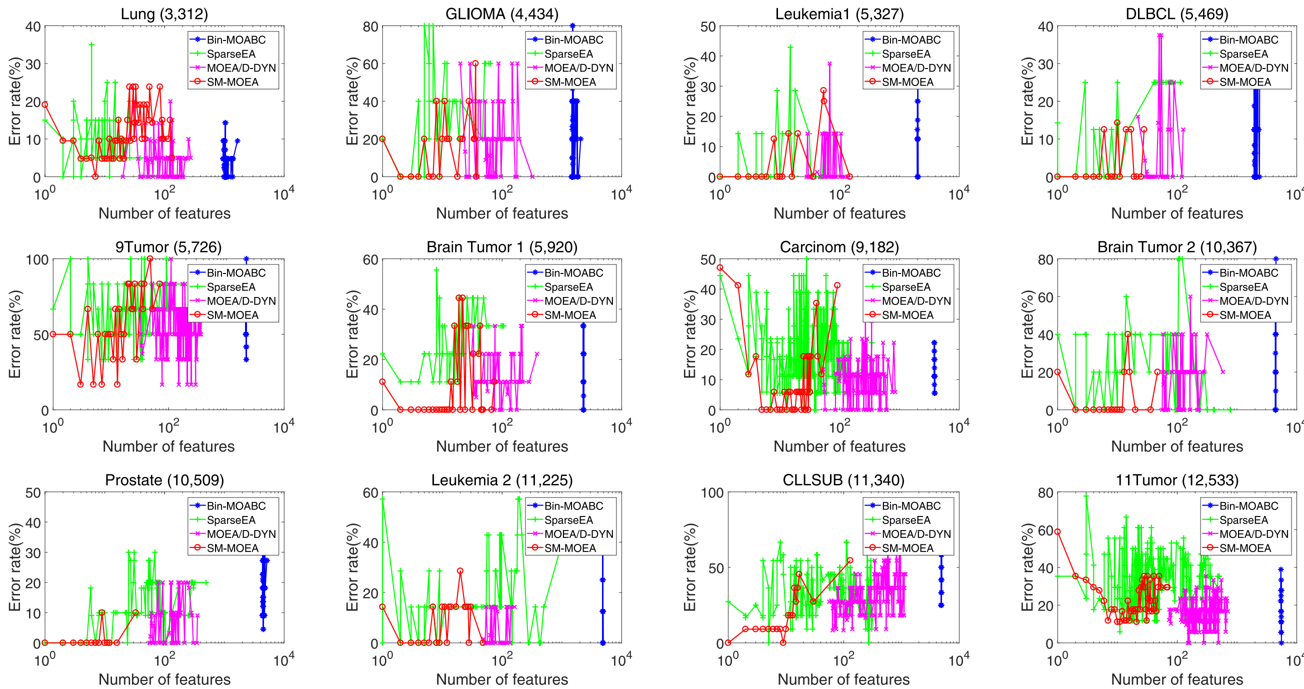


Fig. 4. Average results on all nondominated solutions obtained by the four MOEAs (i.e., Bin-MOABC, SparseEA, MOEA/D-DYN, and SM-MOEA) in the objective space on the 12 datasets.

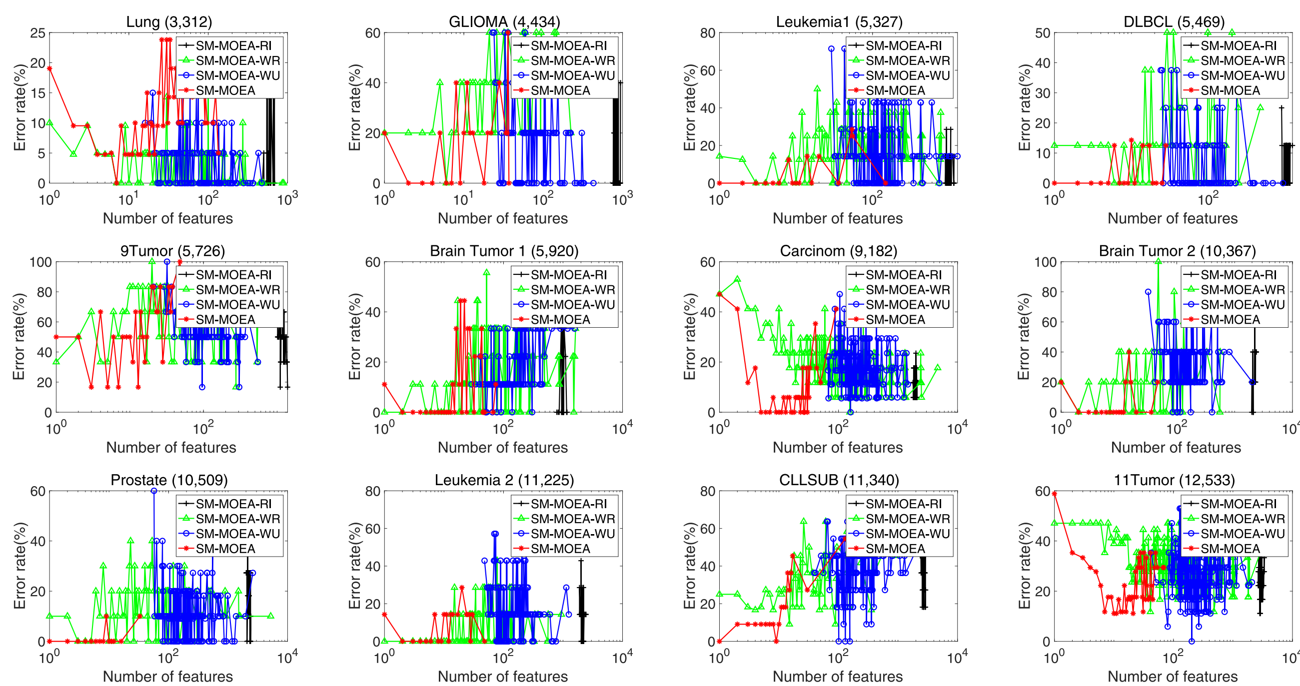


Fig. 5. Average results on all nondominated solutions obtained by SM-MOEA and its three variants in the objective space on the 12 datasets.

features and the error rate on most of the 12 high-dimensional datasets. To be specific, due to the abandonment of the individual repairing strategy, SM-MOEA-WR (green line) has a higher average error rate than the SM-MOEA, especially on the datasets, such as Brain Tumor 2, Prostate, and CLLSUB. Meanwhile, by taking a close look at Fig. 5, we can find that SM-MOEA-WR also selects more features than the original SM-MOEA. The reason is mainly attributed to the following fact. In the suggested repairing strategy, there is a common case, in which nonelite individuals should learn from the elite individuals with probability ρ. In this case, if the ith feature in elite individuals is not selected (which often occurs in high-dimensional FS), then after performing repairing operator, this feature may be also deleted in the nonelite individuals, thereby leading SM-MOEA to select fewer features than SM-MOEA-WR. As for the other two variants: 1) SM-MOEA-RI (black line) and 2) SM-MOEA-WU (blue line), since they cause the steering matrix to lose the "steering" ability, which makes the population easily fall into the local optimum and degrades their performance. Thus, the effectiveness of the proposed strategies used in SM-MOEA is verified.

V. Conclusion

In this article, we proposed a steering-matrix-based MOEA, called SM-MOEA, for high-dimensional FS. In SM-MOEA, a steering matrix was first designed by considering both the importance of the feature on the dataset and that of the individual on the population; thus, can more closely approximate the importance of features in population individuals. Then, during the evolution, two steering-matrix-based dimensionality reduction and individual repairing operators were proposed, which can remove most of irrelevant and redundant features in population individuals without sacrificing the classification accuracy. In addition, the steering matrix initialization and updating strategies were also suggested to further improve the performance of SM-MOEA. Based on these components, SM-MOEA can significantly improve the search efficiency and obtain high-quality feature subsets. The experimental results on 12 real-world high-dimensional datasets demonstrated the superiority of the proposed algorithm over several state-of-the-art single-objective and MOEAs in terms of both the number of selected features and classification accuracy.

The performance of the proposed SM-MOEA confirmed the effectiveness of the steering matrix for high-dimensional FS, which is also a sparse combinatorial optimization problem. Thus, it would be interesting to extend the concept of the steering matrix to other large-scale sparse combinatorial optimization problems, such as itemset mining [52] and network structural vulnerability analysis [53]. Moreover, it is worth noting that as a wrapper method, the cost of evaluating an individual (feature subset) in the proposed method for high-dimensional FS is still expensive. In the future, we would like to exploit surrogate-assisted methods [54] to reduce the calculation time of evaluation functions, thus further improving the efficiency of SM-MOEA.

References

[1] J. Gui, Z. Sun, S. Ji, D. Tao, and T. Tan, "Feature selection based on structured sparsity: A comprehensive study," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 7, pp. 1490–1507, Jul. 2017.

[2] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.

[3] M. S. Raza and U. Qamar, "A hybrid feature selection approach based on heuristic and exhaustive algorithms using rough set theory," in *Proc. Int. Conf. Internet Things Cloud Comput.*, 2016, pp. 1–7.

[4] I. Tsamardinos, G. Borboudakis, P. Katsogridakis, P. Pratikakis, and V. Christophides, "A greedy feature selection algorithm for big data of high dimensionality," *Mach. Learn.*, vol. 108, no. 2, pp. 149–202, 2019.

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

[6] S. Kang, D. Kim, and S. Cho, "Efficient feature selection-based on random forward search for virtual metrology modeling," *IEEE Trans. Semicond. Manuf.*, vol. 29, no. 4, pp. 391–398, Nov. 2016.

[7] X. Bing, M. Zhang, W. Browne, and Y. Xin, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[8] E. Aličković and A. Subasi, "Breast cancer diagnosis using GA feature selection and rotation forest," *Neural Comput. Appl.*, vol. 28, no. 4, pp. 753–763, 2017.

[9] A. Unler and A. Murat, "A discrete particle swarm optimization method for feature selection in binary classification problems," *Eur. J. Oper. Res.*, vol. 206, no. 3, pp. 528–539, 2010.

[10] R. J. Manoj, M. D. A. Praveena, and K. Vijayakumar, "An ACO–ANN based feature selection algorithm for big data," *Clust. Comput.*, vol. 22, no. 2, pp. 3953–3960, 2019.

[11] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1656–1671, Dec. 2013.

[12] M. Amoozegar and B. Minaei-Bidgoli, "Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism," *Expert Syst. Appl.*, vol. 113, pp. 499–514, Dec. 2018.

[13] H. B. Nguyen, B. Xue, I. Liu, P. Andreae, and M. Zhang, "New mechanism for archive maintenance in PSO-based multi-objective feature selection," *Soft Comput.*, vol. 20, no. 10, pp. 3927–3946, 2016.

[14] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 64–75, Jan./Feb. 2017.

[15] E. Hancer, B. Xue, M. Zhang, D. Karaboga, and B. Akay, "Pareto front feature selection based on artificial bee colony optimization," *Inf. Sci.*, vol. 422, pp. 462–479, Jan. 2018.

[16] B. Hu *et al.*, "Feature selection for optimized high-dimensional biomedical data using an improved shuffled frog leaping algorithm," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 15, no. 6, pp. 1765–1773, Nov./Dec. 2018.

[17] M. K. Ebrahimpour, H. Nezamabadi-Pour, and M. Eftekhari, "CCFS: A cooperating coevolution technique for large scale feature selection on microarray datasets," *Comput. Biol. Chem.*, vol. 73, pp. 171–178, Apr. 2018.

[18] C. Yan, J. Ma, H. Luo, G. Zhang, and J. Luo, "A novel feature selection method for high-dimensional biomedical data based on an improved binary clonal flower pollination algorithm," *Hum. Hered.*, vol. 84, no. 1, pp. 34–46, 2019.

[19] F. Cheng, J. Chen, J. Qiu, and L. Zhang, "A subregion division based multi-objective evolutionary algorithm for SVM training set selection," *Neurocomputing*, vol. 394, pp. 70–83, Jun. 2020.

[20] X. Zhang, F. Duan, L. Zhang, F. Cheng, Y. Jin, and K. Tang, "Pattern recommendation in task-oriented applications: A multi-objective perspective," *IEEE Comput. Intell. Mag.*, vol. 12, no. 3, pp. 43–53, Aug. 2017.

[21] L. Zhang, X. Wu, H. Zhao, F. Cheng, and Q. Liu, "Personalized recommendation in P2P lending based on risk-return management: A multi-objective perspective," *IEEE Trans. Big Data*, early access, May 8, 2020, doi: 10.1109/TBDATA.2020.2993446.

[22] L. Zhang, H. Pan, Y. Su, X. Zhang, and Y. Niu, "A mixed representation-based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2703–2716, Sep. 2017.

[23] X. Zhang, K. Zhou, H. Pan, L. Zhang, X. Zeng, and Y. Jin, "A network reduction-based multiobjective evolutionary algorithm for community detection in large-scale complex networks," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 703–716, Feb. 2020.

[24] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[25] W. Siedlecki and J. Sklansky, "A note on genetic algorithms for large-scale feature selection," *Pattern Recognit. Lett.*, vol. 10, no. 5, pp. 335–347, 1989.

[26] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Comput. Security*, vol. 70, pp. 255–277, Sep. 2017.

[27] T. Zhou, H. Lu, W. Wang, and Y. Xia, "GA-SVM based feature selection and parameter optimization in hospitalization expense modeling," *Appl. Soft Comput.*, vol. 75, pp. 323–332, Feb. 2019.

[28] B. Tran, B. Xue, and M. Zhang, "Variable-length particle swarm optimization for feature selection on high-dimensional classification," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 473–487, Jun. 2019.

[29] B. Tran, B. Xue, M. Zhang, and S. Nguyen, "Investigation on particle swarm optimisation for feature selection on high-dimensional data: Local search and selection bias," *Connection Sci.*, vol. 28, no. 3, pp. 270–294, 2016.

[30] B. Tran, B. Xue, and M. Zhang, "A new representation in PSO for discretization-based feature selection," *IEEE Trans. Cybern.*, vol. 48, no. 6, pp. 1733–1746, Jun. 2018.

[31] G. Dick, "Sensitivity-like analysis for feature selection in genetic programming," in *Proc. Genet. Evol. Comput. Conf.*, 2017, pp. 401–408.

[32] U. Mlakar, I. Fister, J. Brest, and B. Potočnik, "Multi-objective differential evolution for feature selection in facial expression recognition systems," *Expert Syst. Appl.*, vol. 89, pp. 129–137, Dec. 2017.

[33] E. Hancer, B. Xue, and M. Zhang, "Differential evolution for filter feature selection based on information theory and feature ranking," *Knowl. Based Syst.*, vol. 140, pp. 103–119, Jan. 2018.

[34] M. Zhang, J. Ma, M. Gong, H. Li, and J. Liu, "Memetic algorithm based feature selection for hyperspectral images classification," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, San Sebastian, Spain, 2017, pp. 495–502.

[35] K. Nag and N. R. Pal, "A multiobjective genetic programming-based ensemble for simultaneous feature selection and classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 499–510, Feb. 2016.

[36] B. Niu, W. Yi, L. Tan, S. Geng, and H. Wang, "A multi-objective feature selection method based on bacterial foraging optimization," *Nat. Comput.*, pp. 1–14, Jul. 2019. doi: 10.1007/s11047-019-09754-6.

[37] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020.

[38] K. Mistry, L. Zhang, S. C. Neoh, C. P. Lim, and B. Fielding, "A Micro-GA embedded PSO feature selection approach to intelligent facial emotion recognition," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1496–1509, Jun. 2017.

[39] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Comput.*, vol. 22, no. 3, pp. 811–822, 2018.

[40] Y. Xue, B. Xue, and M. Zhang, "Self-adaptive particle swarm optimization for large-scale feature selection in classification," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 5, pp. 1–27, 2019.

[41] Y. Xue, T. Tang, W. Pan, and A. X. Liu, "Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers," *Appl. Soft Comput.*, vol. 88, Mar. 2020, Art. no. 106031.

[42] Y. Tian, X. Zhang, C. Wang, and Y. Jin, "An evolutionary algorithm for large-scale sparse multiobjective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 380–393, Apr. 2020.

[43] H. Li, F. He, Y. Liang, and Q. Quan, "A dividing-based many-objective evolutionary algorithm for large-scale feature selection," *Soft Comput.*, vol. 24, pp. 6851–6870, Sep. 2019.

[44] B. H. Nguyen, B. Xue, P. Andreae, H. Ishibuchi, and M. Zhang, "Multiple reference points-based decomposition for multiobjective feature selection in classification: Static and dynamic mechanisms," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 170–184, Feb. 2020.

[45] W. Press, S. Teukolsky, W. Vetterling, and F. Brian, *Numerical Recipes in C*, 2nd ed. New York, NY, USA: Cambridge Univ. Press, 1992.

[46] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 856–863.

[47] P. Yang, K. Tang, and X. Yao, "Turning high-dimensional optimization into computationally expensive optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 143–156, Feb. 2018.

[48] F. Marini and B. Walczak, "Particle swarm optimization (PSO). A tutorial," *Chemometr. Intell. Lab. Syst.*, vol. 149, pp. 153–165, Dec. 2015.

[49] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[50] J. Apolloni, G. Leguizamón, and E. Alba, "Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments," *Appl. Soft Comput.*, vol. 38, pp. 922–932, Jan. 2016.

[51] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, 1998, pp. 292–301.

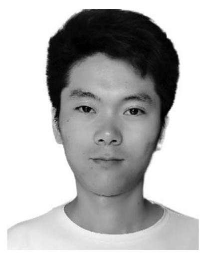
[52] L. Zhang, G. Fu, F. Cheng, J. Qiu, and Y. Su, "A multi-objective evolutionary approach for mining frequent and high utility itemsets," *Appl. Soft Comput.*, vol. 62, pp. 974–986, Jan. 2018.

[53] L. Zhang, J. Xia, F. Cheng, J. Qiu, and X. Zhang, "Multi-objective optimization of critical node detection based on cascade model in complex networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 3, pp. 2052–2066, Jul.–Sep. 2020.

[54] Q. Zhou, J. Wu, T. Xue, and P. Jin, "A two-stage adaptive multi-fidelity surrogate model-assisted multi-objective genetic algorithm for computationally expensive problems," *Eng. Comput.*, vol. 37, pp. 623–639, Aug. 2019.

Fan Cheng received the B.Sc. and M.Sc. degrees from the Hefei University of Technology, Hefei, China, in 2000 and 2003, respectively, and the Ph.D. degree from the University of Science and Technology of China, Hefei, in 2012.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University, Hefei. He has presented and published more than 40 papers at refereed conferences and in journals such as IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON BIG DATA, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, and *IEEE Computational Intelligence Magazine*. His main research interests include machine learning, imbalanced classification, multiobjective optimization, and complex networks.

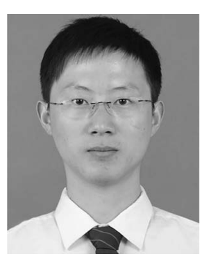


Feixiang Chu received the B.Sc. degree from Anhui University, Hefei, China, in 2018, where he is currently pursuing the master's degree with the School of Computer Science and Technology.

His research interests include multiobjective optimization methods, instance selection, and feature selection.

Yi Xu received the B.Eng. degree from Anhui University, Hefei, China, in 2020. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Shanghai Jiaotong University, Shanghai, China.

His research interests include evolutionary computation and its applications to multiobjective optimization.



Lei Zhang (Member, IEEE) received the B.Sc. degree from Anhui Agriculture University, Hefei, China, in 2007, and the Ph.D. degree from the University of Science and Technology of China, Hefei, in 2014.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University, Hefei. He has presented and published nearly 60 papers of refereed conferences and in journals, such as ACM SIGKDD, ACM CIKM, IEEE International Conference on Data Mining, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON BIG DATA, IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, *ACM Transactions on Knowledge Discovery From Data*, and *IEEE Computational Intelligence Magazine*. His current research interests include multiobjective optimization and applications, data mining, social network analysis, and pattern recommendation.

Dr. Zhang was a recipient of the ACM CIKM'12 Best Student Paper Award.