Dimensionality Reduction in Evolutionary Algorithms-Based Feature Selection for Motor Imagery Brain-Computer Interface

Ping Tan[a], Xin Wang[b], Yong Wang[b,*]

[a]*School of Computer Science and Information Technology, Hunan University of Technology and Business, Changsha 410205, China.*
[b]*School of Automation, Central South University, Changsha 410083, China.*

Abstract

For the classification of motor imagery brain-computer interface (BCI) based on electroencephalography (EEG), appropriate features are crucial to obtain a high classification accuracy. Considering the characteristics of the EEG signals, the time-frequency-space three-dimensional features are extracted. Due to a considerable number of the extracted features, the performance of a classifier will degrade. Therefore, it is necessary to implement feature selection. However, existing feature selection methods are easy to fall into a local optimum of a high-dimensional feature selection problem. In this paper, a dimensionality reduction mechanism (called DimReM) is proposed, which gradually reduces the dimension of the search space by removing some unimportant features. In principle, DimReM transforms a high-dimensional feature selection problem into a low-dimensional one. DimReM does not introduce any additional parameters and its implementation is simple. To verify its effectiveness, DimReM is combined with different evolutionary algorithms and different classifiers to select features on various kinds of datasets. Compared with evolutionary algorithms without dimensionality reduction, their augmented versions equipped with DimReM can find feature subsets with higher classification accuracies while smaller numbers of selected features.

Keywords: Brain-computer interface, evolutionary algorithms, motor imagery, dimensionality reduction, feature selection

*Corresponding author.
Email address: ywang@csu.edu.cn (Yong Wang)

1. Introduction

Brain-computer interface (BCI) is a communication system that allows its users to interact with external devices using the brain signal directly, and it does not depend on the peripheral nerves and muscles [1, 2]. BCI analyzes the brain signal data collected from specific tasks and converts the brain information into control commands that can be used to control computers or communication devices. It provides a new way of communication, which can help people who suffer from devastating neuromuscular injuries and neurodegenerative diseases to restore their communication ability to some extent, assist patients with epilepsy, stroke, and other diseases to biofeedback treatment, and so on. The BCI technology has attracted wide attention from many fields, such as neurology, rehabilitation engineering, psychology, computer science, engineering, and mathematics.

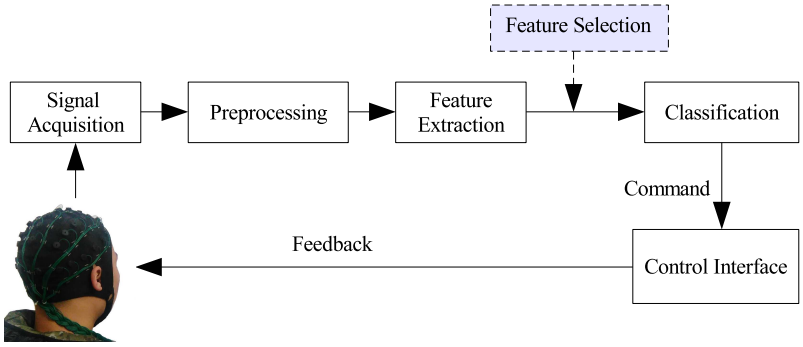


Figure 1: Framework of BCI.

The basic framework of BCI is shown in Fig. 1, which includes five modules: signal acquisition, preprocessing, feature extraction, classification, and control interface. The electroencephalography (EEG) signal is one of the most common bio-potential signals used in the signal acquisition module, because it is non-implantable, non-invasive, inexpensive, and easy to use. The high temporal resolution and multichannel of the EEG signals result in that a considerable number of features will be extracted and some of these features may be redundant, irrelevant, or trivial [3, 4].

Since the brain can be divided into various functional areas, some of the acquired EEG signals from adjacent EEG electrodes may come from the same functional area, and the features extracted from these signals may be redundant. For

example, in BCI Competition III dataset IVa (http://www.bbci.de/competition/iii/), the EEG signals x_{C1} and x_{C3} are acquired from 'C1' and 'C3', where 'C1' and 'C3' are the locations of EEG electrodes and they are close to the primary motor cortex. If five motor-related frequency features $F_{i_{C1}}$ $(i = 1, \ldots, 5)$ and $F_{i_{C3}}$ $(i = 1, \ldots, 5)$ are extracted from x_{C1} and x_{C3}, respectively, by the same method, then $F_{i_{C1}}$ will be similar to $F_{i_{C3}}$ $(i = 1, \ldots, 5)$, and thus they are redundant.

If some features have no relationship with the classification, then they are called the irrelevant features. For example, in motor imagery BCI, the EEG signals from motor-related areas (e.g., the primary motor area, the pre-motor area, and the supplementary motor area) are very important for classification. However, the EEG signals derived from other functional areas (e.g., the auditory area) may be independent of motor imagery. Under this condition, the extracted features from these EEG signals have no relationship with the classification, and thus cannot improve the classification performance.

In addition to the redundant and irrelevant features, it is necessary to note that there may exist some trivial features, which have very little effect on improving the classification accuracy, but will cause overfitting of a classifier.

Obviously, the redundant, irrelevant, and trivial features increase the computation burden of the training process of the classifier, degrade the generalization ability of the classifier, and decrease the classification accuracy. Therefore, feature selection should be employed before the classification, as shown in Fig. 1. The task of feature selection is to select some important features from all features, with the purpose of reducing the feature dimensionality, accelerating the training process, simplifying the classifier model, and improving the classification accuracy [5–7].

Feature selection is a challenging problem mainly due to the following two issues: the large search space and the interference of redundant, irrelevant, and trivial features [8]. Firstly, the search space grows exponentially with the increase of the number of features. For example, the total number of possible feature subsets is $(2^n - 1)$ for n features. Secondly, in order to select some important features, it is necessary to remove the redundant, irrelevant, and trivial features, since they have side effects on the classification performance. According to whether it is independent of the subsequent classifier, feature selection can be divided into two categories [7]: filter methods and wrapper methods.

The filter methods do not depend on any classifier. They generally use the statistical measures of the training data to evaluate a feature's importance, which include distance function [9], rough set [10], mutual information [11], fuzzy set [12], statistical correlation coefficient [13], etc. However, they cannot guarantee

the optimal feature subset.

The wrapper methods employ a classifier as a "black box" to evaluate the feature subsets based on the classification performance. Although they are computationally intensive and time-consuming, the wrapper methods have better performance than the filter methods. It is because the wrapper methods consider the performance of the selected features on a classifier, which is ignored by the filter methods [7]. So the wrapper methods have the potential to obtain a subset of features with higher classification performance. The wrapper methods are currently a hot topic in the field of feature selection. This paper focuses mainly on the application of the wrapper methods on the EEG signals.

In principle, the wrapper methods contain two important components: subset search, the aim of which is to generate a candidate feature subset from the original feature set, and subset evaluation, which makes use of a classifier to assess the goodness of this feature subset. In recent years, evolutionary algorithms (EAs) have become effective subset search methods. Moreover, various EAs, such as differential evolution (DE) [14, 15], particle swarm optimization (PSO) [16–20], and genetic algorithm (GA) [21–23], have achieved better performance than traditional subset search methods. Compared with traditional subset search methods, EAs do not require domain knowledge and do not need any assumptions about the search space, such as nonlinearity and separability. Another advantage is that EAs are population-based search algorithms and have powerful search ability.

For the classification of motor imagery BCI based on EEG, proper features are crucial to obtain a high classification accuracy. Considering the characteristics of the EEG signals, the time-frequency-space three-dimensional features are extracted, which forms a feature set with a considerable number of features. Under this condition, feature selection should be carried out in a high-dimensional search space. Note, however, that EAs are easily trapped into a local optimum due to high dimensionality.

To improve the performance of EAs on a high-dimensional feature selection problem of motor imagery BCI based on EEG, the main idea of this paper is to remove unimportant features (i.e., redundant, irrelevant, and trivial features) in the iterative process of EAs. By doing this, the dimension of the search space can be reduced and the important features can be maintained simultaneously. To this end, we propose a dimensionality reduction mechanism (called DimReM) in EAs-based feature selection.

The main contribution of this paper are summarized as follows:

- The current feature selection methods aim at directly selecting some im-

portant features. However, this paper proposes an opposite point of view. DimReM first determines whether a feature is unimportant by taking advantage of the information from evolution. Afterward, the unimportant features are deleted generation by generation. As a result, the important features are maintained.

- DimReM has a simple structure, and does not introduce any additional parameter and complicated operator.
- DimReM is readily embedded into different EAs. Moreover, we have successfully integrated DimReM with three EAs and three classifiers.
- Systematic experiments have been conducted on the EEG datasets and three datasets from other fields to verify the effectiveness of DimReM. The results verify that DimReM can find feature subsets with higher classification accuracies while smaller numbers of features.

The rest of this paper is organized as follows. Section 2 introduces the related work, including feature extraction and feature selection. In Section 3, three EAs are briefly introduced. Section 4 gives the details of the proposed DimReM. Section 5 presents the experimental results and discussions. Finally, Section 6 concludes this paper.

2. Related Work

2.1. Feature Extraction

Since the EEG signals are usually nonlinear and non-stationary, how to extract their features is very important in BCI. Appropriate features are helpful to improve the performance of BCI.

The physiological studies indicate that EEG power changes with imagined movements in the motor cortex. This phenomenon is called sensorimotor rhythms (SMRs). SMRs include event-related desynchronization (ERD) and event-related synchronization (ERS) [24, 25], which are the basis of motor imagery BCI. In motor imagery BCI, common spatial patterns (CSP) algorithm is a successful feature extraction method for detecting ERD and ERS [26–30]. The CSP algorithm computes spatial filter that maximizes the variance of one class signals, and meanwhile minimizes the variance of another class signals [28–30]. For this reason, it is easy to distinguish motor imagery activities.

Suppose that n trials are performed on the left and right hand movement imagery, respectively. Let an $N \times T$ matrix $E_{j,i}$ describe the raw EEG data of the ith trial, where $j \in \{L, R\}$ denotes the left or right hand movement imagery, N denotes the number of recording electrodes, and T denotes the number of samples in each electrode.

The process of the CSP algorithm is introduced as follows:

1) Compute the normalized covariance of the ith trial of the jth type of motor imagery signals:

$$C_{j,i} = \frac{E_{j,i}E_{j,i}^T}{trace(E_{j,i}E_{j,i}^T)} \tag{1}$$

where ^{T} means the transpose operator and $trace(E_{j,i}E_{j,i}^T)$ is the amount of the diagonal elements of $E_{j,i}E_{j,i}^T$. Then, the spatial covariance of the left or right hand motor imagery signals is

$$\overline{C}_j = \frac{1}{n}\sum_{i=1}^{n} C_{j,i} \tag{2}$$

So the composite spatial covariance is

$$C_c = \overline{C}_L + \overline{C}_R \tag{3}$$

2) Perform the eigen decomposition on C_c:

$$C_c = U_c A_c U_c^T \tag{4}$$

where U_c denotes the matrix of eigenvectors and A_c means the diagonal matrix of eigenvalues. Note that the eigenvalues are supposed to be sorted in descending order for convenience.

3) Execute the whitening transformation as follows:

$$P = A_c^{-\frac{1}{2}} U_c^T \tag{5}$$

$$S_j = P\overline{C}_j P^T \tag{6}$$

S_L and S_R share common eigenvectors, i.e.,

$$\begin{array}{ll} \text{if} & S_L = BA_LB^T, \\ \text{then} & S_R = BA_RB^T \text{ and } A_L + A_R = I. \end{array}$$

where I is the identity matrix, B denotes the matrix of eigenvectors, and A_j means the diagonal matrix of eigenvalues. The spatial filter after whitening is

$$W = (B^T P)^T \tag{7}$$

So $E_{j,i}$ after spatial filtering is:

$$Z_{j,i} = W E_{j,i} \tag{8}$$

Note that $Z_{j,i}$ is an $N \times T$ matrix.

4) Calculate the feature value as follows:

$$f_{j,i,p} = log\left(\frac{var(Z_{j,i,p})}{\sum_{t=1}^{2M} var(Z_{j,i,t})}\right) \tag{9}$$

where $Z_{j,i,p}$ is the pth row of $Z_{j,i}$, $p \in \{1, 2, \ldots, M, N-M+1, \ldots, N\}$, and $var(Z_{j,i,p})$ is the variance of $Z_{j,i,p}$. Finally, $f = [f_{j,i,1}, f_{j,i,2}, \ldots, f_{j,i,2M}]^T$ is the extracted feature vector. Note that the value of M can be changed according to the quality of the EEG signals and the requirement of the construction of a classifier, but $2M$ should be smaller than N.

In motor imagery BCI, the useful frequency of the EEG signals is in the range of 8-32 Hz. For different subjects, the reactive frequency bands are different. If the features are directly extracted by the CSP algorithm from the raw EEG signals, some unnecessary frequency signals may interfere with the feature extraction and degrade the classification performance. Thus, the frequency band filters should be employed before extracting the features by the CSP algorithm. Note that we choose the frequency range of 4-36Hz for frequency band filtering to cover the frequency range of 8-32Hz of the EEG signal in motor imagery BCI. In addition, the reactive time of each subject is different for the indicator signal, so the time domain features should be considered. Based on the above considerations, this paper extracts time-frequency-space three-dimensional features from the EEG signals.

The whole process of time-frequency-space feature extraction includes three steps:

- Divide the EEG signals into multiple frequency band signals using bandpass filters.
- Divide the frequency band signals of each trial into multiple time segments by short-time windows.

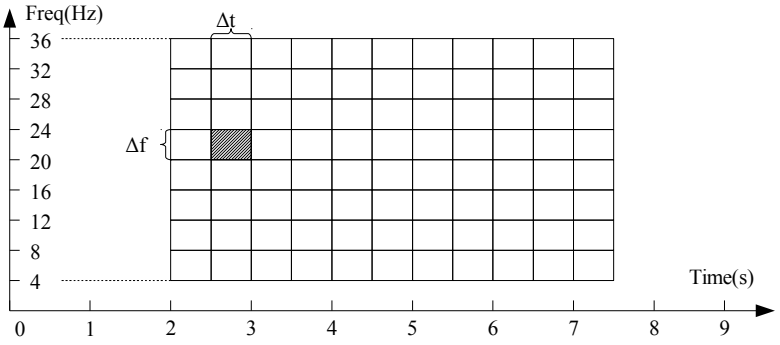


Figure 2: The time-frequency grids of time-frequency-space features.

- Finally, extract the features of the EEG signals by the CSP algorithm in each time-frequency grid.

The time-frequency grids are shown in Fig. 2. In each grid, several features can be extracted. In addition, the elliptic bandpass filter is used in the first step. The elliptical bandpass filter is superior to other types of bandpass filters because its transition band is steeper and its ripples in the passband and stopband are smaller. In the second step, the short-time sliding windows are employed, where the size of the windows is 1s and their overlap size is 0.5s.

2.2. Feature Selection

After extracting the time-frequency-space three-dimensional features, there are a large number of features which may lead to dimensionality curse. The dimensionality curse is that when the number of features exceeds a certain limit, the performance of a classifier will significantly degrade as the number of features increases. In addition, the higher the feature dimensionality, the greater the time cost in the training process. To address this problem, one of effective ways is feature selection, the aim of which is to find the optimal feature subset. Due to the powerful global search ability, EAs have been used in feature selection of motor imagery BCI based on EEG.

In [21] and [23], GA is used to search the space of features, and the fitness function is the weighted linear combination of the number of features and the accuracy of support vector machine (SVM). In [20], PSO-based rough set feature selection method is proposed to find the best subset of features, and the accuracy of a neighborhood classifier is used as the evaluation criterion for feature subset. Baig *et al.* [15] employed DE and Atyabi *et al.* [31] utilized PSO and GA to

discover the optimal feature subset, respectively. In [32], ant colony optimization, simulated annealing, GA, PSO, and DE are adopted to select features for EEG-based emotion recognition.

These EAs-based feature selection methods achieve promising performance. However, in these methods, the size of the search space is fixed and it is necessary to judge whether each feature should be selected or not in each iteration. Under this condition, some redundant, irrelevant, and trivial features will waste the computational resource. Besides, due to the large search space and the interference of redundant, irrelevant, and trivial features, EAs easily converge to a local optimum.

In this paper, we design EAs-based feature selection methods for motor imagery BCI via dimensionality reduction.

3. EAs

DE, GA, and PSO are three main branches of EAs. Many attempts have been made to improve their performance and expand their application fields. Since in essence, the feature selection problem is a binary optimization problem. In this section, we introduce three typical versions of DE, GA, and PSO to solve a binary optimization problem, respectively.

3.1. Novel Modified Binary DE (NMBDE)

DE [33, 34] has a simple structure and is easy to implement. However, it cannot solve the binary optimization problems directly since its crossover and mutation operators are executed in the continuous space. To address this problem, Wang *et al*. [35] proposed a new variant of DE, called NMBDE, which designs a probability estimation operator.

In NMBDE, the population contains *NP* individuals at generation t: $\mathcal{P}^t = \{\overrightarrow{x}_i^t = [x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t]^T, i = 1, 2, \ldots, NP\}$, where D is the number of variables. At generation $(t+1)$, for the jth $(j \in \{1, 2, \ldots, D\})$ variable of the ith individual, the probability estimation operator is defined as:

$$\begin{cases} P(x_{i,j}^{t+1}) = 1/(1 + e^{-2b*(MO-0.5)/(1+2F)}) \\ MO = x_{r1,j}^t + F * (x_{r2,j}^t - x_{r3,j}^t) \end{cases} \tag{10}$$

where F denotes the scaling factor, and $x_{r1,j}^t$, $x_{r2,j}^t$, and $x_{r3,j}^t$ are the jth variable of three mutual individuals randomly chosen from $\mathcal{P}^t$. In (10), b is a positive real constant, which is called the bandwidth factor to tune the range and shape of the probability distribution. An appropriate b value is beneficial to the search

efficiency and population diversity. In (10), *MO* represents the mutation operator of the standard DE, which is embedded into the probability estimation operator of NMBDE. After implementing the probability estimation operator in (10), we can obtain the probability of $x_{i,j}^{t+1}$ to be "1", i.e., $P(x_{i,j}^{t+1})$.

Then, the binary mutation operator is executed on $x_{i,j}^{t+1}$ as:

$$m_{i,j}^{t+1} = \begin{cases} 1, & \text{if } rand \leq P(x_{i,j}^{t+1}) \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

where $rand$ is a uniformly distributed random number between 0 and 1, $m_{i,j}^{t+1}$ denotes the mutant variable of $x_{i,j}^{t+1}$, and $\overrightarrow{m}_i^{t+1} = [m_{i,1}^{t+1}, m_{i,2}^{t+1}, \ldots, m_{i,D}^{t+1}]^T$ denotes the binary-coded mutant individual.

The crossover operator and the selection operator of NMBDE are the same with those of the standard DE.

3.2. GA

GA is derived from the Darwinian principle of "survival of the fittest" [36]. This principle implies that the fitter individuals can survive with a higher probability and are more likely to pass their good genetic features to the next generation [37].

In GA, a chromosome with D bits represents an individual, which is associated with a fitness value. Based on this characteristic, GA can be used for feature selection in a straightened way. For each bit of an individual, '1' and "0" indicate that the corresponding feature is selected or not selected, respectively. The fitness value can assess the quality of each individual and is the basis for genetic operations.

GA includes three genetic operators: selection, crossover, and mutation [38]. First, the roulette wheel selection is implemented, in which the probability of selecting an individual is directly proportional to its fitness value. Subsequently, the crossover operator splits up pair-wise individuals and recombines them. Specifically, some parts of two individuals are exchanged and merged to produce two new offspring. For example, the two-point crossover operator randomly generates two crossover points cp_1 and cp_2, where $cp_1 < cp_2$. If $rand < p_c$, then two individuals exchange the segments located between these two points, where $rand$ is a uniformly distributed random number between 0 and 1, and p_c is the crossover probability. Finally, the mutation operator randomly modifies some of bits with the mutation probability p_m, thus introducing new genetic structures.

3.3. Binary PSO (BPSO)

PSO, motivated by the collective behavior of organisms such as bird flocks, is a swarm intelligence method to solve continuous optimization problems [39]. PSO relies on the cooperation and information sharing among individuals in the group to find the optimal solution. It involves two important updating equations: the velocity updating equation and the position updating equation [40]. In order to apply the standard PSO to a binary space, BPSO is proposed in [41]. For a particle $\overrightarrow{x}_i^t = [x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t]^T$ $(i \in \{1, 2, \ldots, NP\})$ at generation t, BPSO first updates its velocity $\overrightarrow{v}_i^t = [v_{i,1}^t, v_{i,2}^t, \ldots, v_{i,D}^t]^T$ as follows:

$$v_{i,j}^{t+1} = \omega * v_{i,j}^t + c_1 * r_1 * (pbest_{i,j}^t - x_{i,j}^t) + c_2 * r_2 * (gbest_j^t - x_{i,j}^t) \tag{12}$$

where $j \in \{1, 2, \ldots, D\}$, $\overrightarrow{pbest}_i^t = [pbest_{i,1}^t,\ pbest_{i,2}^t,\ \ldots, pbest_{i,D}^t]^T$ is the previous best position of $\overrightarrow{x}_i^t$, $\overrightarrow{gbest}^t = [gbest_1^t, gbest_2^t, \ldots, gbest_D^t]^T$ is the historical best position of all particles, ω is the inertia weight, c_1 and c_2 are two acceleration factors, and r_1 and r_2 are two random numbers uniformly distributed in [0, 1]. At each generation, $v_{i,j}^{t+1}$ is limited to $[-v_{max}, v_{max}]$.

Afterward, BPSO updates its position through the sigmoid function conversion:

$$S(v_{i,j}^{t+1}) = 1/(1 + e^{-v_{i,j}^{t+1}}) \tag{13}$$

$$x_{i,j}^{t+1} = \begin{cases} 1, & \text{if } rand \leq S(v_{i,j}^{t+1}) \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

where $rand$ is a uniformly distributed random number between 0 and 1, and $S(\cdot)$ is the sigmoid function.

4. Proposed Method

4.1. Motivation

As introduced above, current EAs-based feature selection methods aim to select some important features from all features, which means that they focus on "selection". However, they will face the following two great challenges in motor imagery BCI based on EEG:

- The number of features extracted from the EEG signals is large, which leads to a high-dimensional search space. It is not an easy task to search for the important features in a high-dimensional search space.

- There exist interferences from some unimportant features (i.e., redundant, irrelevant, and trivial features), which causes the significant degradation of the search ability of EAs.

Recognizing these challenges, we deal with the feature selection problem from an opposite point of view. To be specific, we focus on "deletion" [1]. Although it is difficult to select some important features accurately, it is relatively easy to determine which features are unimportant. Along this line, we propose a dimensionality reduction mechanism, called DimReM, in EAs-based feature selection for motor imagery BCI based on EEG. DimReM first detects whether a feature is unimportant by utilizing the feedback information from evolution. If it has been detected, then we delete it. By deleting the unimportant features generation by generation, the important features are preserved in the end and the dimension of the search space reduces gradually during the search process. Note that the dimension of the search space in current EAs-based feature selection methods is fixed during the evolution.

4.2. DimReM

For each individual $\overrightarrow{x}_i^t = [x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t]^T$ ($i \in \{1, 2, \ldots, NP\}$) in $\mathcal{P}^t$, $x_{i,j}^t = 1$ ($j \in \{1, 2, \ldots, D\}$) represents that the jth feature is selected, and $x_{i,j}^t = 0$ represents that the jth feature is not selected. Suppose that $\overrightarrow{x}_{best}^t$ is the best individual in $\mathcal{P}^t$, f_{best}^t is the fitness value of $\overrightarrow{x}_{best}^t$, and S_j is the number of individuals which do not select the jth feature:

$$S_j = \sum_{i=1}^{NP} (x_{i,j}^t = 0),\ j = 1, 2, \ldots, D \tag{15}$$

Algorithm 1 gives the framework of DimReM. First, we find the maximum value of $\{S_1, \ldots, S_D\}$, denoted as S_{max} (Step 3). If only one element in $\{S_1, \ldots, S_D\}$ is equal to S_{max}, then the index of this element is denoted as k; else we randomly select one of the elements whose values are equal to S_{max}, and its index is denoted as k (Steps 4-8). If the kth bit of $\overrightarrow{x}_{best}^t$ is '0', then the kth bit of $\mathcal{P}^t$ is directly deleted and $\mathcal{P}_{DimReM}^t \leftarrow \mathcal{P}^t$ (Steps 9-11). Otherwise, the following attempt is made: 1) $\mathcal{Q}^t \leftarrow \mathcal{P}^t$ and delete the kth bit of $\mathcal{Q}^t$ (Steps 13-14); 2) evaluate $\mathcal{Q}^t$ and the best fitness value is denoted as $\hat{f}_{best}^t$ (Step 15); and 3) if $\hat{f}_{best}^t$ is better than f_{best}^t,

[1] As introduced in Section 1, there are certainly some redundant, irrelevant, and trivial features after feature extraction which should be deleted.

Algorithm 1 DimReM

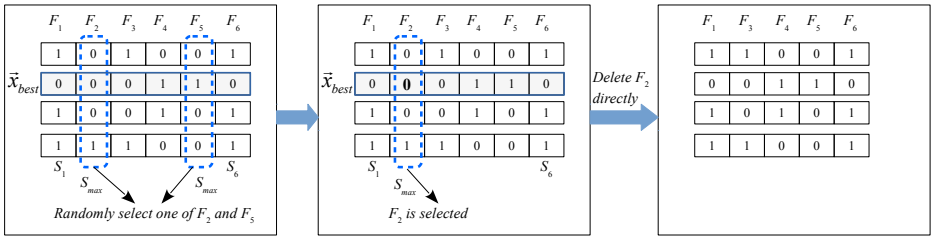
Input: $\mathcal{P}^t$, $\overrightarrow{x}^t_{best}$, and f^t_{best}, where $\overrightarrow{x}^t_{best}$ is the best individual in $\mathcal{P}^t$ and f^t_{best} is the fitness value of $\overrightarrow{x}^t_{best}$

1: $\mathcal{P}^t_{DimReM} = \emptyset$;
2: Calculate S_j $(j = 1, 2, \ldots, D)$ based on (15);
3: $S_{max} = max\{S_1, \ldots, S_D\}$;
4: **if** only one element in $\{S_1, \ldots, S_D\}$ is equal to S_{max} **then**
5: The index of this element is denoted as k;
6: **else**
7: Randomly select one of the elements whose values are equal to S_{max}, and its index is denoted as k;
8: **end if**
9: **if** $x^t_{best,k} = 0$ **then**
10: Delete the kth bit of $\mathcal{P}^t$;
11: $\mathcal{P}^t_{DimReM} \leftarrow \mathcal{P}^t$;
12: **else**
13: $\mathcal{Q}^t \leftarrow \mathcal{P}^t$;
14: Delete the kth bit of $\mathcal{Q}^t$;
15: Evaluate $\mathcal{Q}^t$ and the best fitness value is denoted as $\hat{f}^t_{best}$;
16: **if** $\hat{f}^t_{best}$ is better than f^t_{best} **then**
17: $\mathcal{P}^t_{DimReM} \leftarrow \mathcal{Q}^t$;
18: **else**
19: $\mathcal{P}^t_{DimReM} \leftarrow \mathcal{P}^t$;
20: **end if**
21: **end if**

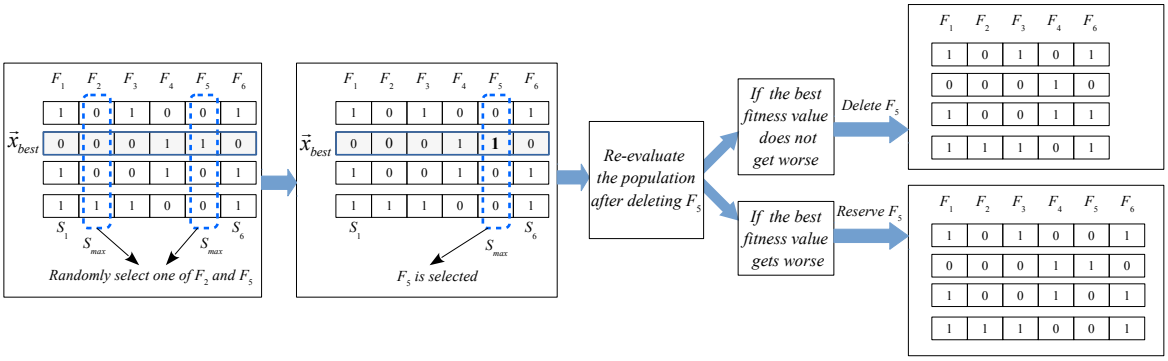
Output: $\mathcal{P}^t_{DimReM}$

then $\mathcal{P}^t_{DimReM} \leftarrow \mathcal{Q}^t$; else $\mathcal{P}^t_{DimReM} \leftarrow \mathcal{P}^t$ (Steps $16 - 20$). Finally, $\mathcal{P}^t_{DimReM}$ is the output of DimReM.

In **Algorithm 1**, we first focus on the feature which is not selected by the most individuals in the population, and consider that this feature is an unimportant feature with the highest probability among all features. Subsequently, if this feature is also not selected by the best individual in the population, which means that this feature is really an unimportant feature, then it is deleted. However, if this feature is selected by the best individual in the population, a further test will be carried out. We make an attempt to delete this feature from the population and, as a result, obtain a new population. If the best fitness value of the new population is better than or equal to that of the old population, which means that deleting this feature



(a) The feature to be deleted is not selected by the best individual



(b) The feature to be deleted is selected by the best individual

Figure 3: Illustration of DimReM.

has no side effect on the performance, then this feature is deleted since it is unimportant; otherwise, this feature is an important feature and should be reserved since deleting it will cause performance degradation.

From the above introduction, it is clear that in DimReM, the information of population is used to identify a potentially unimportant feature in each iteration. Moreover, the best individual is regarded as the feedback information from evolution to determine whether this feature should be deleted or not. As a result, DimReM gradually removes the unimportant features and finally achieves the dimensionality reduction of features.

The main characteristics of DimReM are summarized as follows:

- DimReM provides an effective way to identify and delete unimportant features by taking advantage of the information from evolution.
- DimReM does not introduce any additional parameter and complicated operator. Its implementation is simple.

- By deleting unimportant features, the search space becomes smaller, which is beneficial for EAs to enhance the search efficiency. Meanwhile, the number of dimensions of a feature subset also becomes smaller, which reduces the difficulty of the training task in a classifier.

- Due to the re-evaluation of population, DimReM ensures that the classification performance never gets worse when a feature is deleted in each time.

4.3. Principle Analysis

In this subsection, we analyze the principle of DimReM. An example of DimReM is shown in Fig. 3. Suppose that there are four individuals (i.e., $NP = 4$) in the population, each individual comprises of six features (i.e., $D = 6$): F_1, ..., F_6, and the second individual is the best individual $\overrightarrow{x}_{best}$.

The implementation of DimReM is explained as follows. Firstly, we calculate S_j ($j = 1, 2, \ldots, 6$) based on (15) and obtain the maximum value S_{max}. In this example, $S_{max} = max\{S_1, S_2, \ldots, S_6\}$ = $max\{1, 3, \ldots, 1\} = 3$. Obviously, $S_2 = S_5 = S_{max}$ as shown in Fig. 3. Therefore, it is necessary to randomly select one feature from F_2 and F_5 and to check whether it should be deleted or not. Fig. 3(a) and Fig. 3(b) depict what happens if F_2 or F_5 is selected, respectively:

- As shown in Fig. 3(a), F_2 is not selected by $\overrightarrow{x}_{best}$ since $x_{best,2} = 0$. Thus, F_2 is directly removed from the population. As a consequence, a six-dimensional feature subset is changed to a five-dimensional one.

- As shown in Fig. 3(b), F_5 is selected by $\overrightarrow{x}_{best}$ since $x_{best,5} = 1$. Therefore, it needs a further judgment by re-evaluating the population after F_5 has been removed. After re-evaluation, if the best fitness value does not get worse, then F_5 is deleted and the number of features is reduced from six to five. Otherwise, F_5 is reserved.

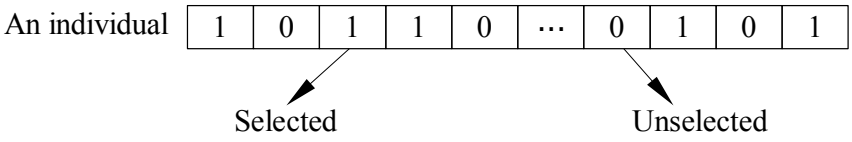


Figure 4: Encoding of an individual

Algorithm 2 Framework of DimReM-EAs

1: $t = 0$;
2: Initialize the population $\mathcal{P}^0$;
3: Evaluate $\mathcal{P}^0$;
4: **while** the stopping criterion is not met **do**
5: $\quad \mathcal{P}^t_{DimReM} \leftarrow$ DimReM($\mathcal{P}^t$) // see Algorithm 1;
6: $\quad$ Implement the evolutionary operators of EAs on $\mathcal{P}^t_{DimReM}$ to obtain $\mathcal{P}^{t+1}$;
7: $\quad t = t + 1$;
8: **end while**

4.4. EAs-Based Feature Selection with DimReM

In this paper, EAs are employed as the search engine to find the optimal feature subset. To test the effectiveness of DimReM, it is embedded into the three representative EAs introduced in Section 3 (i.e., NMBDE, GA, and BPSO) to solve the high-dimensional feature selection problem in motor imagery BCI based on EEG. For the sake of convenience, DimReM-embedded EAs are denoted as DimReM-EAs. In this paper, three DimReM-EAs are DimReM-NMBDE, DimReM-GA, and DimReM-BPSO. Next, we introduce how to embed DimReM into EAs.

4.4.1. Encoding

A feature selection problem can be regarded as a binary optimization problem, so an individual is represented by a binary-coded vector, which is shown in Fig. 4.

4.4.2. Fitness Function

The aim of the fitness function is to evaluate the quality of a feature subset. This paper makes use of the classification accuracy as the fitness function:

$$\textit{fitness} = \frac{T_P + T_N}{N_P + N_N} * 100\% \tag{16}$$

where T_P is the number of samples that are actually positive and are classified as positive by the classifier; T_N is the number of samples that are actually negative and are classified as negative by the classifier; N_P is the total number of positive samples; and N_N is the total number of negative samples.

4.4.3. DimReM-EAs

Firstly, the initial population $\mathcal{P}^0$ inducing *NP* binary-coded individuals is randomly generated. Then, DimReM is executed to obtain $\mathcal{P}^t_{DimReM}$. Afterward,

Table 1: Parameter Settings for Three DimReM-EAs and Their Original EAs

Algorithm	Parameter Setting
NMBDE and DimReM-NMBDE	$F = 0.8, CR = 0.2, b = 20$
GA and DimReM-GA	$p_c = 0.6, p_m = 0.03$
BPSO and DimReM-BPSO	$\omega = 1, c_1 = 2, c_2 = 1.5$

the evolutionary operators of EAs are implemented on $\mathcal{P}^t_{DimReM}$ to produce $\mathcal{P}^{t+1}$. Obviously, the dimension of $\mathcal{P}^{t+1}$ is smaller than or equal to that of $\mathcal{P}^t$. The above process repeats until the stopping criterion is met. The framework of DimReM-EAs for feature selection is shown in **Algorithm 2**.

It should be noted that DimReM does not increase any significant time complexity to the original EAs since DimReM does not affect the evolutionary operators of the original EAs.

5. Experimental Results and Analysis

To demonstrate the effectiveness of DimReM, the performance of three DimReM-EAs (i.e., DimReM-NMBDE, DimReM-GA, and DimReM-BPSO) was compared with that of their original EAs (i.e., NMBDE, GA, and BPSO) for feature selection, respectively. In this paper, the classification accuracy was used as the evaluation criterion and three different classifiers were employed to compute the classification accuracy, namely SVM, K-nearest neighbor (KNN) [42], and discriminant analysis (DA) [43]. Note that different EAs and different classifiers were combined in pairs. Our experiments were conducted on two types of datasets: 1) the EEG datasets of motor imagery BCI, and 2) other datasets of machine learning. For each pair which combined a EA with a classifier, 25 independent runs were executed on each dataset.

5.1. Parameter Settings

The parameter settings of three DimReM-EAs and their original EAs are given in Table 1. For NMBDE and DimReM-NMBDE, the scaling factor F and the crossover control parameter *CR* were set to 0.8 and 0.2, respectively, and the bandwidth factor b was set to 20. For GA and DimReM-GA, the crossover probability p_c and mutation probability p_m were set to 0.6 and 0.03, respectively. For BPSO and DimReM-BPSO, the inertia weight ω was set to 1, and the acceleration factors c_1 and c_2 were set to 2 and 1.5, respectively. In this paper, for each

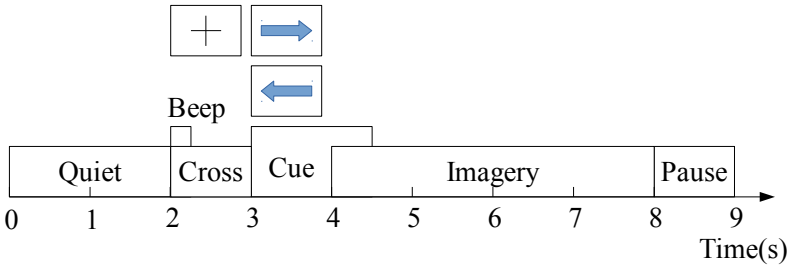


Figure 5: Timing scheme of the BCI paradigm.

algorithm, the population size *NP* was set to 100, and the maximum number of fitness evaluations was set to 20,000. In order to ensure a fair comparison, the three DimReM-EAs and their original EAs had the same stopping criterion, i.e., the maximum number of fitness evaluations. All experiments were run on a PC with Intel(R) Core(TM) i5-7500 CPU @ 3.40GHZ and 8.00GB RAM using MATLAB R2014a.

5.2. EEG Datasets

The EEG datasets can be divided into two groups: the first group and the second group come from BCI Competition III dataset IVa and BCI Competition IV dataset IIb, respectively. The EEG signal acquisition process of these datasets are shown in Fig. 5. The subject sits in front of a display screen, and keeps quiet between 0-2s. At 2s, a beep sound stimulus reminds the subject to concentrate, and a cross symbol appears on the display screen between 2-3s. At 3s, the cross symbol is replaced with a left or right arrow, and the subject imagines the movement according to the direction of the arrow. The entire trial process lasts about 9s. These two groups of datasets are slightly different, and are briefly introduced in the following.

5.2.1. The First Group of Datasets

The first group of datasets is recorded from five healthy subjects with 118 electrodes on the motor imagery of the right-hand task and the right-foot task. The sampling rate is 1000 Hz, and the signals are band-pass filtered between 0.05 and 200 Hz. For each subject, there are 280 trials in total and there are 140 trials per task. The datasets of five subjects are denoted as “aa”, “al”, “av”, “aw”, and “ay”. Moreover, each dataset is divided into the training trials and the test trials. In these datasets, 288 features are obtained after the time-frequency-space

Table 2: Classification Accuracy (%) of NMBDE and DimReM-NMBDE with Three Different Classifiers on the EEG Datasets. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	SVM		KNN		DA	
	NMBDE	DimReM-NMBDE	NMBDE	DimReM-NMBDE	NMBDE	DimReM-NMBDE
	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
aa	71.43±1.41 (125)	**79.46±0.63 (103)**	70.89±2.06 (142)	**76.25±1.85 (134)**	75.36±1.74 (142)	**77.68±2.53 (121)**
al	90.36±0.98 (142)	**95.00±1.49 (114)**	88.93±1.49 (145)	**93.57±1.60 (124)**	84.64±0.98 (141)	**94.29±2.33 (105)**
av	73.06±1.59 (137)	**78.67±0.84 (114)**	67.45±0.91 (143)	**73.16±1.82 (130)**	70.51±0.76 (142)	**73.06±1.82 (130)**
aw	84.20±0.51 (143)	**87.41±0.37 (115)**	71.79±0.66 (142)	**72.32±1.18 (130)**	76.25±0.37 (147)	**79.82±0.66 (122)**
ay	49.29±0.18 (124)	**54.13±2.49 (85)**	69.21±1.36 (139)	**75.95±1.14 (109)**	74.21±0.56 (141)	**79.05±0.43 (101)**
B0103T	92.00±0.68 (187)	**95.25±0.84 (141)**	91.75±0.52 (195)	**94.25±0.81 (164)**	91.00±1.63 (185)	**94.00±1.22 (178)**
B0203T	75.13±0.52 (188)	**79.00±0.56 (154)**	72.50±0.77 (191)	**77.38±1.73 (169)**	77.88±1.63 (187)	**82.75±1.57 (171)**
B0303T	66.13±0.68 (179)	**71.88±0.99 (140)**	67.88±0.56 (180)	**73.25±0.81 (166)**	71.00±0.71 (183)	**76.50±1.91 (163)**
B0403T	100.0±0.00 (194)	100.0±0.00 **(112)**	100.0±0.00 (198)	100.0±0.00 **(110)**	100.0±0.00 (204)	100.0±0.00 **(153)**
B0503T	98.13±0.00 (190)	**99.13±0.34 (130)**	98.13±0.00 (181)	**99.00±0.34 (137)**	97.50±0.44 (195)	**98.50±0.34 (176)**
B0603T	87.75±0.95 (181)	**92.25±0.71 (131)**	84.88±1.02 (194)	**87.75±1.14 (164)**	84.00±1.44 (191)	**89.13±3.32 (171)**
B0703T	94.38±0.00 (189)	**96.38±0.28 (129)**	94.00±0.34 (191)	**95.88±0.56 (162)**	97.00±0.52 (200)	**97.13±0.71 (175)**
B0803T	96.50±0.34 (182)	**98.63±0.52 (131)**	94.50±0.52 (190)	**96.25±0.00 (145)**	96.38±0.28 (201)	**98.00±0.68 (174)**
B0903T	94.88±0.28 (181)	**96.88±0.00 (124)**	94.75±0.56 (185)	**96.75±0.52 (155)**	95.00±1.17 (195)	**97.25±0.71 (172)**

three-dimensional feature extraction, and the performance is evaluated by the classification accuracy of the test trials.

5.2.2. The Second Group of Datasets

The second group of datasets is recorded from nine healthy subjects at C3, Cz, and C4 electrodes on the motor imagery of the right-hand task and the left-hand task. The sampling rate is 250 Hz, and the signals are band-pass filtered between 0.5 and 100 Hz. There are five session records for each subject (i.e., session 1, ..., session 5). In this paper, we only used the datasets of the third session as a representative, namely "B0103T", "B0203T", ..., "B0903T". For each subject, there are 160 trials in total and there are 80 trials per task. After the time-frequency-spatial three-dimensional feature extraction, 384 features are obtained. Due to the fact that this group of datasets is not divided into the training trials and the test trials, the average classification accuracy of 10×10-fold cross-validation is employed to evaluate the performance, which is different from the evaluation scheme of the first group of datasets.

5.3. Experiments on the EEG Datasets

Table 2 presents the results of NMBDE and DimReM-NMBDE with three different classifiers, in which "Mean CA" and "Std Dev" indicate the average and standard deviation of the classification accuracy over 25 runs, respectively, and the number in parentheses means the average size of the final feature subsets over

Table 3: Classification Accuracy (%) of GA and DimReM-GA with Three Different Classifiers on the EEG Datasets. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	SVM		KNN		DA	
	GA	DimReM-GA	GA	DimReM-GA	GA	DimReM-GA
	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
aa	76.96±2.92 (134)	**80.18±0.40 (107)**	80.18±3.54 (138)	**84.64±3.91 (130)**	77.32±2.41 (144)	**82.14±1.26 (127)**
al	93.93±0.98 (141)	**96.43±1.26 (90)**	95.36±0.98 (139)	**97.50±1.59 (123)**	90.71±1.49 (145)	**95.71±2.71 (95)**
av	77.76±1.38 (140)	**80.20±1.04 (123)**	73.88±1.11 (143)	**78.38±0.58 (104)**	76.02±2.70 (140)	**78.98±2.48 (125)**
aw	86.52±1.68 (144)	**89.73±1.45 (123)**	77.05±3.05 (141)	**82.59±1.05 (123)**	81.07±0.87 (149)	**82.77±1.43 (115)**
ay	55.87±1.03 (100)	**65.24±2.61 (57)**	76.43±2.15 (135)	**79.44±3.88 (115)**	78.65±0.76 (135)	**80.32±1.14 (111)**
B0103T	94.75±0.84 (182)	**97.63±0.81 (105)**	94.38±1.17 (187)	**95.38±0.56 (171)**	98.38±1.22 (178)	**100.0±0.00 (153)**
B0203T	77.75±1.37 (184)	**81.13±1.79 (121)**	78.63±1.68 (179)	**81.13±2.48 (176)**	95.38±2.36 (173)	**97.25±1.30 (156)**
B0303T	71.13±1.28 (178)	**74.38±0.99 (118)**	74.13±1.29 (187)	**76.25±2.65 (167)**	90.38±3.44 (167)	**98.00±1.49 (150)**
B0403T	100.0±0.00 (196)	100.0±0.00 **(99)**	100.0±0.00 (198)	100.0±0.00 **(100)**	100.0±0.00 (201)	100.0±0.00 **(155)**
B0503T	98.63±0.28 (181)	**99.25±0.28 (99)**	98.88±0.28 (184)	**99.13±0.56 (111)**	100.0±0.00 (184)	100.0±0.00 **(154)**
B0603T	91.25±1.25 (170)	**93.00±1.73 (113)**	89.00±1.63 (189)	**91.25±1.53 (169)**	97.25±2.34 (173)	**99.50±0.81 (155)**
B0703T	95.63±0.00 (185)	**97.00±0.28 (102)**	96.13±0.81 (193)	**96.88±0.77 (137)**	99.88±0.30 (182)	**100.0±0.00 (156)**
B0803T	97.88±0.34 (177)	**99.00±0.34 (99)**	96.63±0.34 (184)	**97.38±0.68 (123)**	99.75±0.34 (187)	**100.0±0.00 (158)**
B0903T	96.13±0.28 (179)	**97.13±0.34 (103)**	96.50±0.34 (196)	**97.63±0.52 (136)**	99.88±0.28 (190)	**100.0±0.00 (157)**

Table 4: Classification Accuracy (%) of BPSO and DimReM-BPSO with Three Different Classifiers on the EEG Datasets. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	SVM		KNN		DA	
	BPSO	DimReM-BPSO	BPSO	DimReM-BPSO	BPSO	DimReM-BPSO
	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
aa	77.86±0.75 (128)	**82.32±0.98 (97)**	76.61±2.04 (142)	**83.04±1.89 (108)**	77.50±1.60 (141)	**85.00±0.75 (117)**
al	94.28±0.80 (137)	**95.71±0.98 (122)**	96.07±0.80 (147)	**98.21±1.26 (116)**	94.64±2.82 (140)	**96.79±1.96 (117)**
av	76.94±1.11 (145)	**81.63±1.35 (106)**	74.49±0.95 (143)	**79.18±2.17 (113)**	76.12±0.67 (147)	**79.29±1.38 (116)**
aw	87.23±1.12 (134)	**90.09±0.86 (112)**	80.00±1.59 (147)	**85.36±1.43 (109)**	81.52±1.08 (147)	**84.46±0.86 (112)**
ay	57.30±1.55 (95)	**60.63±3.35 (70)**	71.59±1.33 (141)	**80.79±2.20 (103)**	77.62±0.35 (126)	**83.73±1.92 (102)**
B0103T	96.38±0.81 (173)	**97.50±0.44 (127)**	95.38±1.14 (192)	**96.00±1.05 (159)**	97.00±0.52 (186)	**98.63±0.52 (161)**
B0203T	79.38±0.99 (171)	**82.00±1.03 (135)**	78.13±1.40 (189)	**82.38±1.89 (155)**	86.75±1.56 (181)	**93.38±0.95 (162)**
B0303T	72.50±0.44 (164)	**76.38±1.73 (130)**	72.75±0.84 (186)	**78.63±2.48 (154)**	81.38±2.55 (179)	**89.13±2.85 (158)**
B0403T	100.0±0.00 (188)	100.0±0.00 **(148)**	100.0±0.00 (190)	100.0±0.00 **(146)**	100.0±0.00 (206)	100.0±0.00 **(158)**
B0503T	98.75±0.00 (186)	**98.88±0.28 (123)**	99.25±0.28 (189)	99.25±0.28 **(139)**	99.88±0.28 (190)	**100.0±0.00 (163)**
B0603T	91.88±0.77 (168)	**93.13±0.88 (135)**	88.50±1.30 (191)	**90.13±0.68 (166)**	92.50±1.59 (180)	**97.63±1.03 (164)**
B0703T	96.38±0.52 (183)	**96.50±0.34 (126)**	95.63±0.00 (191)	**96.88±0.76 (149)**	99.63±0.34 (191)	**100.0±0.00 (162)**
B0803T	98.00±0.28 (177)	**98.38±0.34 (131)**	96.50±0.34 (182)	**96.88±0.44 (145)**	99.63±0.34 (196)	**99.88±0.28 (162)**
B0903T	96.50±0.34 (180)	**96.88±0.00 (129)**	97.25±0.71 (185)	**97.38±0.52 (141)**	99.13±0.71 (191)	**100.0±0.00 (164)**

25 runs (i.e., the average number of the selected features). It can be seen from Table 2 that DimReM-NMBDE is able to find the feature subsets with higher average classification accuracies than NMBDE on all the datasets except dataset "B0403T". For dataset "B0403T", both NMBDE and DimReM-NMBDE achieve 100% average classification accuracy. On the other hand, the average numbers of features selected by DimReM-NMBDE are consistently smaller than those selected by NMBDE on all the datasets. For example, on dataset "al", the following

phenomena can be observed:

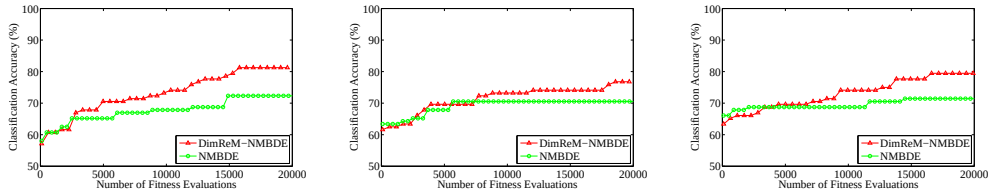
- In terms of DimReM-NMBDE, the average classification accuracies of the three classifiers are 95.00%, 93.57%, and 94.29%, respectively. In contrast, in terms of NMBDE, the average classification accuracies of the three classifiers are 90.36%, 88.93%, and 84.64%, respectively.

- The average numbers of features selected by DimReM-NMBDE with the three classifiers are 114, 124, and 105, respectively. However, when NMBDE is combined with the three classifiers, the average numbers of the selected features are 142, 145, and 141, respectively.

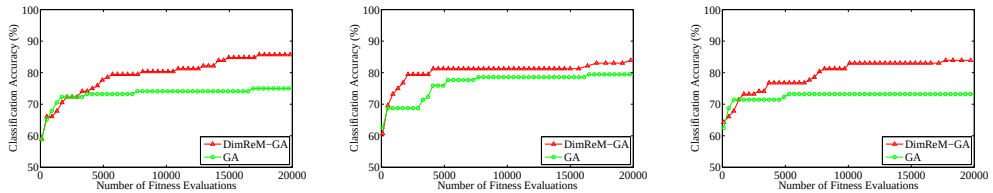
- Overall, after embedding DimReM, the average classification accuracies of the three classifiers are increased by 4.64%, 4.64%, and 9.65%, respectively, and the average numbers of features are reduced by 28, 21, and 36 at the same time.

Tables 3 and 4 present the results of GA and DimReM-GA and the results of BPSO and DimReM-BPSO, respectively. Similar to 2, it can also be observed from Tables 3 and 4 that DimReM-GA and DimReM-BPSO have the capability to find the feature subsets with higher average classification accuracies than GA and BPSO on all the datasets except dataset "B0403T" and dataset "B0503T". Furthermore, the average numbers of features selected by DimReM-GA and DimReM-BPSO are consistently smaller than those selected by GA and BPSO, respectively.

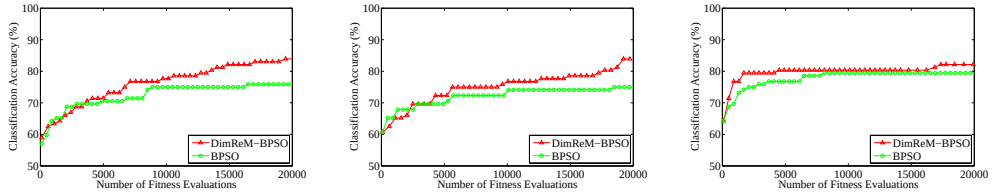
As shown in Tables 2, 3, and 4, different classifiers produce different average classification accuracies on different datasets. For example, for dataset "ay" in Table 2, according to the average classification accuracy, the best, median, and worst classifiers with NMBDE are DA, KNN, and SVM, respectively. However, for dataset "B0603T" in Table 2, SVM, KNN, and DA combined with NMBDE rank the first, second, and third, respectively, according to the average classification accuracy. So it is necessary to employ multiple classifiers to verify the effectiveness of DimReM. From Tables 2, 3, and 4, one can conclude that no matter which classifier is used, DimReM-EAs is capable of finding feature subsets with higher or equal average classification accuracies against their original EAs. The above discussion demonstrates that DimReM-EAs are insensitive to classifiers. The reasons are twofold: 1) in each iteration, DimReM-EAs can ensure that the classification accuracy of a classifier will not decrease after deleting a feature, and 2) the evolutionary operators of EAs can guide the selected features toward a higher classification accuracy.



(a) NMBDE & DimReM-NMBDE with SVM (b) NMBDE & DimReM-NMBDE with KNN (c) NMBDE & DimReM-NMBDE with DA



(d) GA & DimReM-GA with SVM (e) GA & DimReM-GA with KNN (f) GA & DimReM-GA with DA

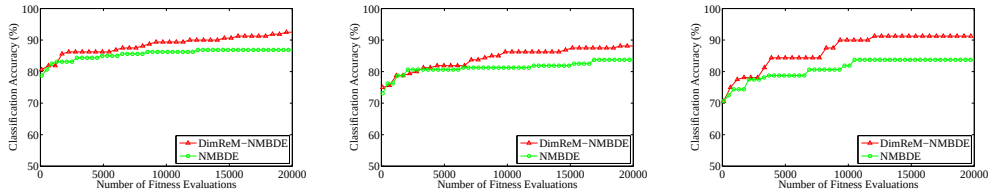


(g) BPSO & DimReM-BPSO with SVM (h) BPSO & DimReM-BPSO with KNN (i) BPSO & DimReM-BPSO with DA

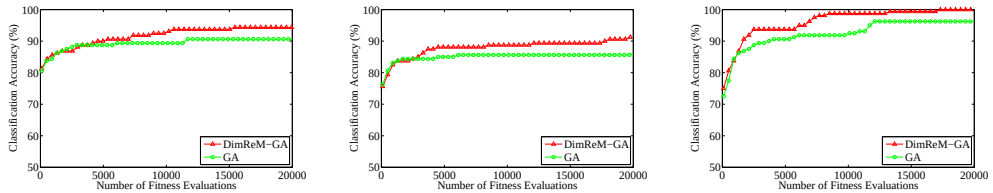
Figure 6: Evolution of the classification accuracy for dataset 'aa'.

It is also necessary to emphasize that in some scenarios, despite the average classification accuracies are not improved, the average numbers of features selected by DimReM-EAs are smaller than those selected by their original EAs. In the context of the same classification accuracy, the smaller number of features can improve the generalization and robustness of a classifier.

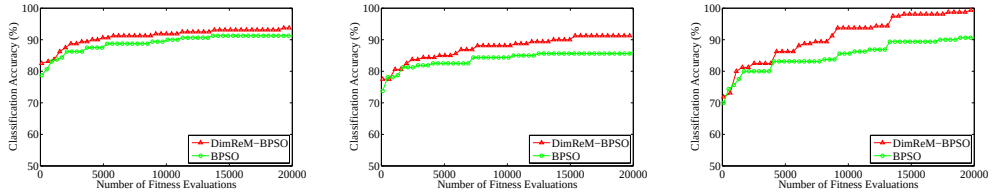
Figs. 6 and 7 provide the evolution of the classification accuracy of each classifier on dataset "aa" and dataset "B0603T" in a typical run, respectively. As shown in these two figures, DimReM-EAs consistently keep higher classification accuracies than their original EAs after some generations for each classifier.



(a) NMBDE & DimReM-NMBDE with SVM (b) NMBDE & DimReM-NMBDE with KNN (c) NMBDE & DimReM-NMBDE with DA



(d) GA & DimReM-GA with SVM (e) GA & DimReM-GA with KNN (f) GA & DimReM-GA with DA



(g) BPSO & DimReM-BPSO with SVM (h) BPSO & DimReM-BPSO with KNN (i) BPSO & DimReM-BPSO with DA

Figure 7: Evolution of the classification accuracy for dataset 'B0603T'.

5.4. Further Experiments on Three Other Datasets

To further test the effectiveness of the proposed DimReM, we carried out comparative experiments on three datasets in other fields, including Madelon, Musk (Version 1), and hERG.

5.4.1. Description of Three Other Datasets

Both the Madelon and Musk (Version 1) datasets are from the UCI dataset repository. The Madelon dataset is a machine-learning dataset which is a two-class classification problem with continuous input variables. The difficulty of this problem is multivariate and highly nonlinear. This dataset has 500 features, and its provider divides the dataset into a training set consisting of 2000 samples and a verification set consisting of 600 samples. For this dataset, the performance is

Table 5: Classification Accuracy (%) of NMBDE and DimReM-NMBDE with Three Different Classifiers on Three Other Datasets. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	SVM		KNN		DA	
	NMBDE	DimReM-NMBDE	NMBDE	DimReM-NMBDE	NMBDE	DimReM-NMBDE
	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
Madelon	69.41±0.57 (253)	**70.03±1.50 (238)**	65.73±0.28 (239)	**67.47±0.56 (224)**	65.60±0.37 (247)	**68.63±0.66 (219)**
Musk (V1)	97.10±1.81 (73)	**98.23±0.17 (53)**	93.95±0.28 (81)	**95.59±0.58 (64)**	88.44±0.46 (86)	**89.34±0.40 (72)**
hERG	87.01±0.40 (36)	**88.84±0.46 (27)**	83.88±0.38 (56)	**86.53±0.61 (39)**	86.02±0.33 (53)	**87.56±0.50 (41)**

Table 6: Classification Accuracy (%) of GA and DimReM-GA with Three Different Classifiers on Three Other Datasets. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	SVM		KNN		DA	
	GA	DimReM-GA	GA	DimReM-GA	GA	DimReM-GA
	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
Madelon	73.37±1.92 (248)	**74.83±1.22 (227)**	69.37±1.66 (240)	**71.20±1.77 (216)**	69.00±0.68 (251)	**70.57±0.84 (223)**
Musk (V1)	97.85±0.34 (70)	**98.15±0.23 (43)**	95.25±0.51 (87)	**95.88±0.46 (64)**	90.71±0.58 (88)	**91.01±0.57 (56)**
hERG	87.62±0.91 (38)	**88.42±0.89 (31)**	85.72±1.19 (54)	**86.22±0.54 (39)**	87.55±0.74 (55)	**88.06±0.38 (36)**

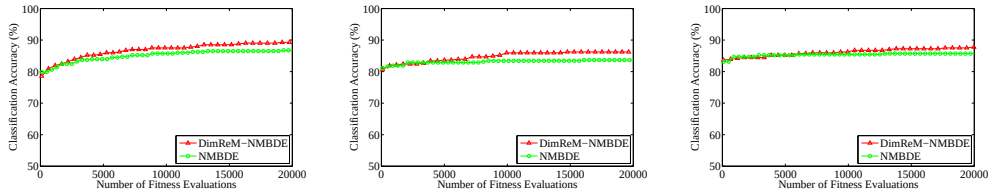
Table 7: Classification Accuracy (%) of BPSO and DimReM-BPSO with Three Different Classifiers on Three Other Datasets. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	SVM		KNN		DA	
	BPSO	DimReM-BPSO	BPSO	DimReM-BPSO	BPSO	DimReM-BPSO
	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
Madelon	72.39±0.26 (247)	**74.45±1.89 (221)**	68.83±0.79 (241)	**71.33±0.55 (205)**	69.53±0.57 (243)	**71.97±0.62 (219)**
Musk (V1)	97.98±0.17 (68)	**98.53±0.39 (47)**	95.42±0.64 (79)	**96.52±0.48 (65)**	90.05±0.19 (83)	**91.48±0.51 (63)**
hERG	88.08±0.75 (36)	**89.14±0.23 (29)**	85.87±0.43 (50)	**86.42±0.42 (41)**	87.34±0.63 (54)	**88.42±0.29 (35)**

evaluated by the classification accuracy of the verification set.

The Musk (Version 1) dataset is used to judge whether the new molecule is musk or non-musk. This dataset has 166 features and 476 conformations (or samples) from 92 molecules. Due to the fact that a molecule has multiple conformations, the relationship between feature vectors and molecules is a many-to-one relationship. If any conformation of the molecule is judged as a musk, the molecule should be classified as "musk". If none conformation of the molecule is judged as a musk, the molecule is classified as "non-musk". For this dataset, the average classification accuracy of 10×10-fold cross-validation is used to evaluate the performance. For details, please refer to http://archive.ics.uci.edu/ml/index.php.

The hERG dataset is from molecule pharmacy. A voltage-gated potassium channel, which is encoded by the human ether-$\grave{a}$-go-go-related gene (hERG or



(a) NMBDE & DimReM-NMBDE with SVM (b) NMBDE & DimReM-NMBDE with KNN (c) NMBDE & DimReM-NMBDE with DA

(d) GA & DimReM-GA with SVM (e) GA & DimReM-GA with KNN (f) GA & DimReM-GA with DA

(g) BPSO & DimReM-BPSO with SVM (h) BPSO & DimReM-BPSO with KNN (i) BPSO & DimReM-BPSO with DA

Figure 8: Evolution of the classification accuracy for dataset 'hERG'.

Kv11.1), plays a significant role in regulating the exchange of cardiac action potential and resting potential during cardiac depolarization and repolarization [44]. So assessing hERG-associated cardiotoxicity is an essential stage during the drug design or discovery process. The hERG dataset consists of 392 molecules (or samples) and 131 descriptors (or features). For this dataset, we used the average classification accuracy of 10×10-fold cross-validation to evaluate the performance.

5.4.2. Experimental Result on Three Other Datasets

The results of three DimReM-EAs and their original EAs integrated with three different classifiers are presented in Tables 5, 6, and 7, which again indicate that DimReM-EAs have better overall performance than EAs in terms of the average

Table 8: Classification Accuracy (%) of DimReM-NMBDE and NA-DimReM-NMBDE on the EEG Datasets. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	SVM		KNN		DA	
	NA-DimReM-NMBDE	DimReM-NMBDE	NA-DimReM-NMBDE	DimReM-NMBDE	NA-DimReM-NMBDE	DimReM-NMBDE
	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
aa	**85.89±1.16 (87)**	79.46±0.63 (103)	**84.11±1.94 (88)**	76.25±1.85 (134)	**84.28±1.35 (88)**	77.68±2.53 (121)
al	**97.14±2.04 (81)**	95.00±1.49 (114)	**97.14±0.98 (90)**	93.57±1.60 (124)	**97.14±2.04 (82)**	94.29±2.33 (105)
av	78.47±0.43 (92)	**78.67±0.84** (114)	72.02±1.30 (92)	**73.16±1.82** (130)	**74.39±0.66 (103)**	73.06±1.82 (130)
aw	84.46±0.97 (83)	**87.41±0.37** (115)	71.91±1.67 (94)	**72.32±1.18** (130)	**81.07±0.59 (97)**	79.82±0.66 (122)
ay	49.92±0.86 (70)	**54.13±2.49** (85)	**76.27±0.81 (88)**	75.95±1.14 (109)	78.57±0.74 (87)	**79.05±0.43** (101)
B0103T	**95.25±0.84 (127)**	95.25±0.84 (141)	**95.38±0.56 (131)**	94.25±0.81 (164)	**94.13±1.69 (154)**	94.00±1.22 (178)
B0203T	**80.38±0.71 (136)**	79.00±0.56 (154)	**80.88±1.05 (142)**	77.38±1.73 (169)	81.63±0.71 (165)	**82.75±1.57** (171)
B0303T	**74.00±0.71 (131)**	71.88±0.99 (140)	72.00±1.44 (135)	**73.25±0.81 (126)**	75.75±2.27 (155)	**76.50±1.91 (163)**
B0403T	100.0±0.00 (114)	100.0±0.00 (112)	100.0±0.00 (115)	100.0±0.00 (110)	100.0±0.00 (153)	100.0±0.00 (153)
B0503T	99.00±0.34 (121)	**99.13±0.34 (120)**	98.38±0.00 (135)	**99.00±0.34 (134)**	98.38±0.34 (164)	**98.50±0.34** (176)
B0603T	92.25±0.71 (131)	**92.35±0.71** (131)	87.13±0.56 (137)	**87.75±1.14 (164)**	88.25±2.18 (167)	**89.13±3.32** (171)
B0703T	**97.00±0.28** (131)	96.38±0.28 (129)	95.00±0.34 (127)	**95.88±0.56** (162)	**97.63±0.68 (167)**	97.13±0.71 (175)
B0803T	**98.75±0.44 (127)**	98.63±0.52 (131)	**97.00±0.52 (135)**	96.25±0.00 (145)	98.00±1.12 (167)	98.00±0.68 (174)
B0903T	96.88±0.00 (122)	96.88±0.00 (124)	**97.00±0.52 (128)**	96.75±0.52 (155)	95.75±0.68 (161)	**97.25±0.71** (172)

classification accuracy and the average number of the selected features. Taking the Madelon dataset in Table 5 as an example, after embedding DimReM, the average classification accuracies of the three classifiers are increased by 0.62%, 1.74%, and 3.03%, respectively, and the average numbers of features are reduced by 15, 15, and 28 at the same time.

Fig. 8 plots the evolution of the classification accuracy of each classifier on the hERG dataset in a typical run. Similar to Figs. 6 and 7, DimReM-EAs consistently maintain higher classification accuracies after some iterations than EAs for each classifier.

5.5. Experiments of DimReM with and without a Further Attempt

As introduced in Section 4.2, when a feature is not selected by the most individuals but is selected by the best individual, a further attempt will be carried out: we judge whether this feature should be deleted or not. The aim of this subsection is to verify the effectiveness of this further attempt.

We conducted the experiments of DimReM with and without this further attempt on the EEG datasets. For the sake of convenience, the prefix "NA-" denotes that DimReM directly deletes a feature without this further attempt. Tables 8, 9, and 10 present the results of NA-DimReM-EAs and DimReM-EAs.

As can be seen from Table 8, when NMBDE is considered as the search algorithm, NA-DimReM-NMBDE is better than and worse than DimReM-NMBDE on 20 cases and 17 cases, respectively, which means that NA-DimReM-NMBDE

Table 9: Classification Accuracy (%) of DimReM-GA and NA-DimReM-GA on the EEG Datasets. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	SVM		KNN		DA	
	NA-DimReM-GA	DimReM-GA	NA-DimReM-GA	DimReM-GA	NA-DimReM-GA	DimReM-GA
	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
aa	**86.43±1.47 (88)**	80.18±0.40 (107)	**85.89±2.13 (110)**	84.64±3.91 (130)	**84.11±2.92 (102)**	82.14±1.26 (127)
al	95.71±3.24 (60)	**96.43±1.26** (90)	96.78±2.93 (103)	**97.50±1.59** (123)	95.35±2.71 (81)	**95.71±2.71** (95)
av	78.77±1.06 (106)	**80.20±1.04** (123)	**78.98±2.61** (121)	78.38±0.58 (104)	79.49±1.32 (113)	**78.98±2.48** (125)
aw	82.68±2.43 (93)	**89.73±1.45** (123)	80.00±2.57 (126)	**82.59±1.05 (123)**	**83.66±1.03 (105)**	82.77±1.43 (115)
ay	53.73±3.94 (55)	**65.24±2.61** (57)	78.65±2.96 (111)	**79.44±3.88** (115)	78.81±1.33 (99)	**80.32±1.14** (111)
B0103T	97.13±0.84 (113)	**97.63±0.81 (105)**	**96.13±0.93 (154)**	95.38±0.56 (171)	100.0±0.00 (154)	100.0±0.00 (153)
B0203T	**81.63±1.22** (124)	81.13±1.79 (121)	81.13±1.68 (168)	81.13±2.48 (176)	**98.13±2.99** (155)	97.25±1.30 (156)
B0303T	**75.13±0.81** (126)	74.38±0.99 (118)	75.50±1.03 (174)	**76.25±2.65 (167)**	97.88±1.04 (151)	**98.00±1.49 (150)**
B0403T	100.0±0.00 (102)	100.0±0.00 (99)	100.0±0.00 (98)	100.0±0.00 (100)	100.0±0.00 (159)	100.0±0.00 (155)
B0503T	99.13±0.34 (97)	**99.25±0.28** (99)	**99.38±0.00** (112)	99.13±0.56 (111)	100.0±0.00 (156)	100.0±0.00 (154)
B0603T	92.50±0.94 (108)	**93.00±1.73** (113)	89.75±1.14 (170)	**91.25±1.53 (169)**	**99.63±0.56 (154)**	99.50±0.81 (155)
B0703T	**97.63±0.28** (104)	97.00±0.28 (102)	**97.63±0.28 (104)**	96.88±0.77 (137)	100.0±0.00 (157)	100.0±0.00 (156)
B0803T	98.50±0.56 (100)	**99.00±0.34 (99)**	**97.63±1.12 (114)**	97.38±0.68 (123)	100.0±0.00 (159)	100.0±0.00 (158)
B0903T	97.00±0.53 (103)	**97.13±0.34** (103)	**98.00±0.52 (123)**	97.63±0.52 (136)	100.0±0.00 (155)	100.0±0.00 (157)

Table 10: Classification Accuracy (%) of DimReM-BPSO and NA-DimReM-BPSO on the EEG Datasets. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	SVM		KNN		DA	
	NA-DimReM-BPSO	DimReM-BPSO	NA-DimReM-BPSO	DimReM-BPSO	NA-DimReM-BPSO	DimReM-BPSO
	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
aa	**86.07±1.85 (95)**	82.32±0.98 (97)	**85.18±4.07 (107)**	83.04±1.89 (108)	**88.21±2.85 (92)**	85.00±0.75 (117)
al	95.71±2.04 (88)	95.71±0.98 (122)	97.86±1.49 (104)	**98.21±1.26** (116)	96.43±1.13 (97)	**96.79±1.96** (117)
av	80.71±0.84 (106)	**81.63±1.35** (106)	**80.92±2.29 (111)**	79.18±2.17 (113)	78.47±1.38 (111)	**79.29±1.38** (116)
aw	85.89±1.32 (92)	**90.09±0.86** (112)	82.50±1.02 (106)	**85.36±1.43** (109)	83.57±1.59 (105)	**84.46±0.86** (112)
ay	48.65±0.35 (84)	**60.63±3.35 (70)**	80.32±2.13 (103)	**80.79±2.20** (103)	80.24±1.17 (97)	**83.73±1.92** (102)
B0103T	95.88±0.56 (126)	**97.50±0.44** (127)	**96.38±1.20 (145)**	96.00±1.05 (159)	99.50±0.52 (161)	98.63±0.52 (161)
B0203T	81.50±0.56 (138)	**82.00±1.03 (135)**	**82.75± 1.51 (149)**	82.38±1.89 (155)	**93.50±2.28 (161)**	93.38±0.95 (162)
B0303T	75.25±0.56 (131)	**76.38±1.73 (130)**	77.13±2.92 (154)	**78.63±2.48** (154)	87.88±4.06 (158)	**89.13±2.85** (158)
B0403T	100.0±0.00 (148)	100.0±0.00 (148)	100.0±0.00 (115)	100.0±0.00 (149)	100.0±0.00 (156)	100.0±0.00 (158)
B0503T	98.50±0.34 (125)	**98.88±0.28** (123)	99.25±0.28 (139)	99.25±0.28 (139)	100.0±0.00 (160)	100.0±0.00 (163)
B0603T	93.00±0.36 (129)	**93.13±0.88** (135)	**90.38±2.19 (154)**	90.13±0.68 (166)	97.25±0.71 (160)	**97.63±1.03** (164)
B0703T	**96.63±0.71** (132)	96.50±0.34 (126)	**97.25±0.71** (152)	96.88±0.76 (149)	100.0±0.00 (162)	100.0±0.00 (162)
B0803T	98.13±0.44 (123)	**98.38±0.34** (131)	96.75±0.52 (145)	**96.88±0.44** (145)	**100.0±0.00 (159)**	99.88±0.28 (162)
B0903T	96.50±0.34 (124)	**96.88±0.00** (129)	97.38±0.28 (143)	97.38±0.52 (141)	100.0±0.00 (164)	100.0±0.00 (164)

is slightly superior to DimReM-NMBDE. However, from Tables 9 and 10, DimReM-EAs outperform NA-DimReM-EAs. Specifically, DimReM-GA and DimReM-BPSO surpass NA-DimReM-GA and NA-DimReM-BPSO on 18 cases and 21 cases, respectively, and lose on 15 cases and 11 cases, respectively.

From the above discussion, one can conclude that, overall, the performance of DimReM with this further attempt is better than the performance of DimReM without this further attempt.

Table 11: Classification Accuracy (%) of the Seven Methods with SVM on the EEG Datasets. "Acc" and "Num" Denotes the Classification Accuracy and the Number of Features Obtained by PCA, ICA, and NCA, respectively. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	PCA	ICA	NCA	VLPSO	DimReM-NMBDE	DimReM-GA	DimReM-BPSO
	Acc (Num)	Acc (Num)	Acc (Num)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
aa	64.28 (41)	65.18 (45)	61.60 (11)	69.64±1.38 (154)	79.46±0.63 (103)	80.18±0.40 (107)	**82.32±0.98** (97)
al	71.42 (37)	75.00 (103)	67.86 (17)	80.36±1.18 (145)	95.00±1.49 (114)	**96.43±1.26** (90)	95.71±0.98 (122)
av	59.69 (33)	61.22 (49)	57.14 (12)	63.27±1.98 (149)	78.67±0.84 (114)	80.20±1.04 (123)	**81.63±1.35** (106)
aw	65.18 (15)	68.30 (38)	63.39 (29)	66.97±0.96 (142)	87.41±0.37 (115)	89.73±1.45 (123)	**90.09±0.86** (112)
ay	54.36 (27)	56.75 (103)	60.32 (28)	67.06±3.64 (131)	54.13±2.49 (85)	**65.24±2.61** (57)	60.63±3.35 (70)
B0103T	87.50 (16)	87.50 (89)	90.62 (50)	88.13±0.75 (186)	95.25±0.84 (141)	**97.63±0.81** (105)	97.50±0.44 (127)
B0203T	61.25 (21)	65.00 (109)	71.88 (16)	67.50±1.31 (167)	79.00±0.56 (154)	81.13±1.79 (121)	**82.00±1.03** (135)
B0303T	60.00 (28)	60.62 (17)	61.88 (13)	61.88±1.83 (181)	71.88±0.99 (140)	74.38±0.99 (118)	**76.38±1.73** (130)
B0403T	98.75 (10)	99.37 (20)	100.00 (76)	100.0±0.00 (182)	100.0±0.00 (112)	100.0±0.00 (99)	100.0±0.00 (148)
B0503T	94.37 (10)	95.00 (17)	98.75 (44)	97.50±0.33 (188)	99.13±0.34 (130)	**99.25±0.28** (99)	98.88±0.28 (123)
B0603T	79.37 (13)	81.87 (204)	89.37 (12)	88.13±1.24 (194)	92.25±0.71 (131)	93.00±1.73 (113)	**93.13±0.88** (135)
B0703T	90.00 (14)	93.12 (17)	91.25 (12)	91.88±0.57 (187)	96.38±0.28 (129)	**97.00±0.28** (102)	96.50±0.34 (126)
B0803T	93.75 (18)	93.12 (54)	96.88 (27)	96.25±0.76 (199)	98.63±0.52 (131)	**99.00±0.34** (99)	98.38±0.34 (131)
B0903T	90.00 (10)	90.62 (70)	93.75 (22)	94.38±0.12 (186)	96.88±0.00 (124)	**97.13±0.34** (103)	96.88±0.00 (129)

5.6. Comparison among PCA, ICA, NCA, VLPSO, and DimReM-EAs

To further demonstrate the performance of DimReM-EAs, we compared DimReM-EAs with two traditional methods, i.e., principal component analysis (PCA) [45] and independent component analysis (ICA) [46], and two state-of-the-art methods, i.e., neighborhood component analysis (NCA) [47] and variable-length PSO (VLPSO) [48]. It is worth noting that PCA and ICA implement feature selection by dimensionality reduction. Therefore, by comparing with PCA and ICA, we can ascertain the effectiveness of our dimensionality reduction in DimReM-EAs.

Tables 11, 12, and 13 provide the results of PCA, ICA, NCA, VLPSO, DimReM-NMBDE, DimReM-GA, and DimReM-BPSO on the EEG datasets with the three classifiers, respectively. Note that since PCA, ICA and NCA are deterministic methods, "Acc" denotes the classification accuracy obtained by PCA, ICA, and NCA, and "Num" denotes the corresponding number of features.

From Tables 11, 12, and 13, although the numbers of features selected by PCA and NCA are smaller than those derived from DimReM-EAs, their classification accuracies are consistently lower than DimReM-EAs. Similar to PCA and NCA, the numbers of features selected by ICA are smaller than those provided by DimReM-EAs on all cases except datasets 'ay', 'ay', and 'B0603T'; however, the classification accuracies of ICA are consistently lower than DimReM-EAs. In addition, VLPSO is inferior to DimReM-EAs in terms of both the classification accuracy and the number of features. We can also observe from Tables 11, 12

Table 12: Classification Accuracy (%) of the Seven Methods with KNN on the EEG Datasets. "Acc" and "Num" Denotes the Classification Accuracy and the Number of Features Obtained by PCA, ICA, and NCA, Respectively. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	PCA	ICA	NCA	VLPSO	DimReM-NMBDE	DimReM-GA	DimReM-BPSO
	Acc (Num)	Acc (Num)	Acc (Num)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
aa	59.82 (27)	59.82 (107)	58.04 (105)	66.97±1.97 (145)	76.25±1.85 (134)	**84.64±3.91** (130)	83.04±1.89 (108)
al	69.64 (36)	69.64 (56)	73.21 (19)	78.57±1.34 (153)	93.57±1.60 (124)	97.50±1.59 (123)	**98.21±1.26** (116)
av	59.18 (43)	59.18 (23)	59.69 (78)	64.29±2.54 (152)	73.16±1.82 (130)	78.38±0.58 (104)	**79.18±2.17** (113)
aw	60.27 (42)	62.50 (256)	61.16 (21)	67.41±1.72 (149)	72.32±1.18 (130)	82.59±1.05 (123)	**85.36±1.43** (109)
ay	54.37 (22)	57.54 (140)	57.14 (208)	65.87±2.43 (153)	75.95±1.14 (109)	79.44±3.88 (115)	**80.79±2.20** (103)
B0103T	88.75 (13)	86.87 (14)	91.25 (61)	89.38±1.62 (192)	94.25±0.81 (164)	95.38±0.56 (171)	**96.00±1.05** (159)
B0203T	58.75 (64)	63.12 (2)	73.12 (137)	68.16±2.03 (211)	77.38±1.73 (169)	81.13±2.48 (176)	**82.38±1.89** (155)
B0303T	55.62 (36)	58.12 (161)	65.00 (109)	64.38±2.77 (201)	73.25±0.81 (166)	76.25±2.65 (167)	**78.63±2.48** (154)
B0403T	100.00 (13)	100.00 (13)	100.00 (12)	100.0±0.00 (197)	100.0±0.00 (110)	100.0±0.00 (100)	100.0±0.00 (146)
B0503T	95.00 (10)	95.62 (67)	96.87 (32)	96.88±0.67 (193)	99.00±0.34 (137)	99.13±0.56 (111)	**99.25±0.28** (139)
B0603T	73.12 (81)	76.25 (39)	82.50 (13)	80.00±1.04 (204)	87.75±1.14 (164)	**91.25±1.53** (169)	90.13±0.68 (166)
B0703T	90.00 (12)	92.50 (24)	92.50 (43)	91.25±1.31 (195)	95.88±0.56 (162)	96.88±0.77 (137)	96.88±0.76 (149)
B0803T	89.37 (16)	90.62 (163)	91.87 (15)	92.50±0.76 (186)	96.25±0.00 (145)	**97.38±0.68** (123)	96.88±0.44 (145)
B0903T	90.62 (12)	91.25 (49)	92.50 (12)	93.13±1.02 (203)	96.75±0.52 (155)	**97.63±0.52** (136)	97.38±0.52 (141)

Table 13: Classification Accuracy (%) of the Seven Methods with DA on the EEG Datasets. "Acc" and "Num" Denotes the Classification Accuracy and the Number of Features Obtained by PCA, ICA, and NCA, Respectively. "Mean CA" and "Std Dev" Indicate the Average and Standard Deviation of the Classification Accuracy over 25 Runs, Respectively, and "AN" in Parentheses Means the Average Number of the Selected Features over 25 Runs.

Dataset	PCA	ICA	NCA	VLPSO	DimReM-NMBDE	DimReM-GA	DimReM-BPSO
	Acc (Num)	Acc (Num)	Acc (Num)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)	Mean CA±Std Dev (AN)
aa	59.82 (37)	63.39 (10)	56.25 (13)	67.86±0.99 (134)	77.68±2.53 (121)	82.14±1.26 (127)	**85.00±0.75** (117)
al	75.00 (93)	75.00 (47)	67.86 (15)	80.38±2.38 (158)	94.29±2.33 (105	95.71±2.71 (95)	**96.79±1.96** (117)
av	60.20 (17)	61.22 (18)	61.22 (174)	64.29±1.76 (156)	73.06±1.82 (130)	78.98±2.48 (125)	**79.29±1.38** (116)
aw	63.83 (35)	63.39 (15)	60.71 (39)	66.52±1.12 (134)	79.82±0.66 (122)	82.77±1.43 (115)	**84.46±0.86** (112)
ay	55.56 (27)	62.70 (80)	60.31 (37)	67.46±2.07 (155)	79.05±0.43 (101)	80.32±1.14 (111)	**83.73±1.92** (102)
B0103T	86.87 (13)	86.87 (13)	86.25 (15)	83.75±1.45 (192)	94.00±1.22 (178)	**100.0±0.00** (153)	98.63±0.52 (161)
B0203T	59.37 (60)	66.25 (58)	70.00 (18)	70.00±1.17 (195)	82.75±1.57 (171)	**97.25±1.30** (156)	93.38±0.95 (162)
B0303T	60.00 (142)	61.25 (100)	59.37 (11)	63.13±3.31 (184)	76.50±1.91 (163)	**98.00±1.49** (150)	89.13±2.85 (158)
B0403T	100.0 (19)	99.37 (12)	99.37 (14)	100.0±0.00 (211)	100.0±0.00 (153)	100.0±0.00 (155)	100.0±0.00 (158)
B0503T	94.37 (16)	94.37 (20)	95.00 (10)	90.63±0.45 (181)	98.50±0.34 (176)	100.0±0.00 (154)	100.0±0.00 (163)
B0603T	81.87 (69)	79.37 (21)	85.62 (15)	77.50±1.47 (199)	89.13±3.32 (171)	**99.50±0.81** (155)	97.63±1.03 (164)
B0703T	90.62 (14)	91.50 (12)	91.25 (12)	90.00±0.73 (208)	97.13±0.71 (175)	100.0±0.00 (156)	100.0±0.00 (162)
B0803T	92.50 (22)	92.50 (20)	94.37 (12)	90.62±0.56 (196)	98.00±0.68 (174)	**100.0±0.00** (158)	99.88±0.28 (162)
B0903T	90.00 (10)	89.37 (12)	90.00 (17)	88.13±0.69 (204)	97.25±0.71 (172)	100.0±0.00 (157)	100.0±0.00 (164)

and 13 that DimReM-GA and DimReM-BPSO achieve the highest classification accuracy on 16 cases and 19 cases, respectively.

Therefore, DimReM-EAs are not only better than the traditional dimensionality reduction methods, but also outperform the state-of-the-art methods.

Remark 1: Based on the experiments on the EEG datasets and three oth-

er datasets, the performance improvement of DimReM-EAs against EAs can be achieved on different types of datasets and is insensitive to the classifier. Thus, DimReM is an effective and generic dimensionality reduction mechanism for EAs-based feature selection in a high-dimensional search space. The performance superiority of DimReM-EAs can be attributed the following fact: by removing unimportant features, the dimension of the search space has been reduced and the interference of unimportant features has been alleviated.

Remark 2: For a practical BCI system, the classification accuracy of a user's intention is the most important indicator. The purpose of feature selection is to improve the classification accuracy in the BCI system. If the classification accuracy and the number of features are considered as two objectives, we should make a tradeoff between them, which means that the classification accuracy will become lower with the decrease of the number of features. So, we consider the feature selection of the BCI system as a single-objective optimization problem to improve the classification accuracy.

6. Conclusion

In this paper, we introduced DimReM for EAs-based feature selection in motor imagery BCI based on EEG. DimReM takes advantage of the feedback information of population to identify and delete unimportant features gradually, thus transforming a high-dimensional feature selection problem into a low-dimensional one. DimReM does not add any significant burden, and does not require any additional control parameter. Its implementation is simple and it is easy to be embedded into EAs.

In the experiments, DimReM was combined with three different EAs and three different classifiers to solve the feature selection problems on the EEG datasets and three other datasets. The results suggest that DimReM is an effective way to assist EAs in finding a feature subset with a higher classification accuracy and smaller number of features simultaneously.

In the future, we plan to apply DimReM to deal with the feature selection problems in medical big data.

Acknowledgment

This work was supported in part by the Innovation-Driven Plan in Central South University under Grant 2018CX010, in part by the National Natural Science Foundation of China under Grant 61673397 and Grant 61976225, in part

by the National Social Science Foundation of China under Grant 19BGL111, in part by the Scientific Research Project of the Education Department of Hunan Province under Grant 18B338 and Grant 18A304, in part by the Open Fund of Key Laboratory of Hunan Province under Grant 2017TP1026, and in part by the Beijing Advanced Innovation Center for Intelligent Robots and Systems under Grant 2018IRS06.

References

[1] J. R. Wolpaw, N. Birbaumer, W. J. Heetderks, D. J. McFarland, P. H. Peckham, G. Schalk, E. Donchin, L. A. Quatrano, C. J. Robinson, T. M. Vaughan, Brain-computer interface technology: a review of the first international meeting, IEEE Transactions on Rehabilitation Engineering 8 (2) (2000) 164–173. doi:10.1109/TRE.2000.847807.

[2] L. F. Nicolas-Alonso, J. Gomez-Gil, Brain computer interfaces, a review, Sensors 12 (2) (2012) 1211–1279. doi:10.3390/s120201211.

[3] G. H. John, R. Kohavi, K. Pfleger, Irrelevant Features and the Subset Selection Problem, in: Machine Learning Proceedings 1994, Elsevier, 1994, pp. 121–129. doi:10.1016/B978-1-55860-335-6.50023-4.

[4] L. Yu, H. Liu, Efficient feature selection via analysis of relevance and redundancy, Journal of Machine Learning Research 5 (Oct) (2004) 1205–1224.

[5] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 491–502. doi:10.1109/TKDE.2005.66.

[6] C. K. Chow, C. N. Liu, Approximating discrete probability distributions with dependence trees, IEEE Transactions on Information Theory IT-14 (3) (1968) 462–467.

[7] B. Xue, M. Zhang, W. N. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Transactions on Evolutionary Computation 20 (4) (2016) 606–626. doi:10.1109/TEVC.2015.2504420.

[8] A. Jakulin, I. Bratko, Analyzing attribute dependencies, in: Knowledge Discovery in Databases: PKDD 2003, Vol. 2838, Springer Berlin

Heidelberg, Berlin, Heidelberg, 2003, pp. 229–240. doi:10.1007/978-3-540-39804-2_22.

[9] S. Ahmed, M. Zhang, L. Peng, Enhanced feature selection for biomarker discovery in LC-MS data using GP, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 584–591. doi:10.1109/CEC.2013.6557621.

[10] B. Xue, L. Cervante, L. Shang, W. N. Browne, M. Zhang, Binary PSO and rough set theory for feature selection: A multi-objective filter based approach, International Journal of Computational Intelligence and Applications 13 (02) (2014) 1450009. arXiv:https://doi.org/10.1142/S1469026814500096, doi:10.1142/S1469026814500096.

[11] H. B. Nguyen, B. Xue, I. Liu, M. Zhang, Filter based backward elimination in wrapper based PSO for feature selection in classification, in: 2014 IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 3111–3118. doi:10.1109/CEC.2014.6900657.

[12] B. Chakraborty, G. Chakraborty, Fuzzy consistency measure with particle swarm optimization for feature selection, in: 2013 IEEE International Conference on Systems, Man, and Cybernetics, 2013, pp. 4311–4315. doi:10.1109/SMC.2013.735.

[13] A. J. Tallón-Ballesteros, J. C. Riquelme, Tackling ant colony optimization meta-heuristic as search method in feature subset selection based on correlation or consistency measures, in: E. Corchado, J. A. Lozano, H. Quintián, H. Yin (Eds.), Intelligent Data Engineering and Automated Learning – IDEAL 2014, Springer International Publishing, Cham, 2014, pp. 386–393.

[14] S. Chattopadhyay, S. Mishra, S. Goswami, Feature selection using differential evolution with binary mutation scheme, in: 2016 International Conference on Microelectronics, Computing and Communications (MicroCom), 2016, pp. 1–6. doi:10.1109/MicroCom.2016.7522533.

[15] M. Z. Baig, N. Aslam, H. P. Shum, L. Zhang, Differential evolution algorithm as a tool for optimal feature subset selection in motor imagery EEG, Expert Systems with Applications 90 (2017) 184–195. doi:https://doi.org/10.1016/j.eswa.2017.07.033.

[16] A. Unler, A. Murat, A discrete particle swarm optimization method for feature selection in binary classification problems, European Journal of Operational Research 206 (3) (2010) 528–539. doi:https://doi.org/10.1016/j.ejor.2010.02.032.

[17] S. S. S. Ahmad, Feature and instances selection for nearest neighbor classification via cooperative PSO, in: 2014 4th World Congress on Information and Communication Technologies (WICT 2014), 2014, pp. 45–50. doi:10.1109/WICT.2014.7077300.

[18] M. C. Lane, B. Xue, I. Liu, M. Zhang, Gaussian based particle swarm optimisation and statistical clustering for feature selection, in: C. Blum, G. Ochoa (Eds.), Evolutionary Computation in Combinatorial Optimisation, Springer Berlin Heidelberg, Berlin, Heidelberg, 2014, pp. 133–144.

[19] J. Lv, M. Liu, Common spatial pattern and particle swarm optimization for channel selection in BCI, in: 2008 3rd International Conference on Innovative Computing Information and Control, 2008, pp. 457–457. doi:10.1109/ICICIC.2008.196.

[20] S. Udhaya Kumar, H. Hannah Inbarani, PSO-based feature selection and neighborhood rough set-based classification for BCI multiclass motor imagery task, Neural Computing and Applications 28 (11) (2017) 3239–3258. doi:10.1007/s00521-016-2236-5.

[21] L. Wang, G. Xu, J. Wang, S. Yang, L. Guo, W. Yan, GA-SVM based feature selection and parameters optimization for BCIahmed2013enhanced research, in: 2011 Seventh International Conference on Natural Computation, Vol. 1, 2011, pp. 580–583. doi:10.1109/ICNC.2011.6022083.

[22] C.-F. Tsai, W. Eberle, C.-Y. Chu, Genetic algorithms in feature and instance selection, Knowledge-Based Systems 39 (2013) 240–247. doi:https://doi.org/10.1016/j.knosys.2012.11.005.

[23] D. Garrett, D. A. Peterson, C. W. Anderson, M. H. Thaut, Comparison of linear, nonlinear, and feature selection methods for EEG signal classification, IEEE Transactions on Neural Systems and Rehabilitation Engineering 11 (2) (2003) 141–144. doi:10.1109/TNSRE.2003.814441.

[24] G. Pfurtscheller, Event-related synchronization (ERS): an electrophysiological correlate of cortical areas at rest, Electroencephalography and Clinical

Neurophysiology 83 (1) (1992) 62–69. doi:https://doi.org/10.1016/0013-4694(92)90133-3.

[25] J. Li, J. Liang, Q. Zhao, J. Li, K. Hong, L. Zhang, Design of assistive wheelchair system directly steered by human thoughts, International Journal of Neural Systems 23 (03) (2013) 1350013. arXiv:https://doi.org/10.1142/S0129065713500135, doi:10.1142/S0129065713500135.

[26] B. Blankertz, G. Dornhege, M. Krauledat, K.-R. Muller, G. Curio, The non-invasive berlin brain-computer interface: Fast acquisition of effective performance in untrained subjects, NeuroImage 37 (2) (2007) 539 – 550. doi:https://doi.org/10.1016/j.neuroimage.2007.01.051.

[27] H. Ramoser, J. Muller-Gerking, G. Pfurtscheller, Optimal spatial filtering of single trial EEG during imagined hand movement, IEEE Transactions on Rehabilitation Engineering 8 (4) (2000) 441–446. doi:10.1109/86.895946.

[28] K. K. Ang, Z. Y. Chin, H. Zhang, C. Guan, Filter bank common spatial pattern (FBCSP) in brain-computer interface, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), 2008, pp. 2390–2397. doi:10.1109/IJCNN.2008.4634130.

[29] Y. Zhang, G. Zhou, J. Jin, X. Wang, A. Cichocki, Optimizing spatial patterns with sparse filter bands for motor-imagery based brain-computer interface, Journal of Neuroscience Methods 255 (2015) 85–91. doi:https://doi.org/10.1016/j.jneumeth.2015.08.004.

[30] B. Blankertz, R. Tomioka, S. Lemm, M. Kawanabe, K. Muller, Optimizing spatial filters for robust EEG single-trial analysis, IEEE Signal Processing Magazine 25 (1) (2008) 41–56. doi:10.1109/MSP.2008.4408441.

[31] A. Atyabi, M. Luerssen, S. Fitzgibbon, D. M. W. Powers, Evolutionary feature selection and electrode reduction for EEG classification, in: 2012 IEEE Congress on Evolutionary Computation, 2012, pp. 1–8. doi:10.1109/CEC.2012.6256130.

[32] B. Nakisa, M. N. Rastgoo, D. Tjondronegoro, V. Chandran, Evolutionary computation algorithms for feature selection of EEG-based emotion recognition using mobile sensors, Expert Systems with Applications 93 (2018) 143–155.

[33] Y. Wang, Z.-Z. Liu, J. Li, H.-X. Li, J. Wang, On the selection of solutions for mutation in differential evolution, Frontiers of Computer Science 12 (2) (2018) 297–315.

[34] Y. Wang, H. Liu, H. Long, Z. Zhang, S. Yang, Differential evolution with a new encoding mechanism for optimizing wind farm layout, IEEE Transactions on Industrial Informatics 14 (3) (2018) 1040–1054. doi:10.1109/TII.2017.2743761.

[35] L. Wang, X. Fu, Y. Mao, M. I. Menhas, M. Fei, A novel modified binary differential evolution algorithm and its applications, Neurocomputing 98 (2012) 55–75. doi:https://doi.org/10.1016/j.neucom.2011.11.033.

[36] J. J. Grefenstette, Optimization of control parameters for genetic algorithms, IEEE Transactions on Systems, Man, and Cybernetics 16 (1) (1986) 122–128. doi:10.1109/TSMC.1986.289288.

[37] N. M. Razali, J. Geraghty, Genetic algorithm performance with different selection strategies in solving TSP, in: 2011 World Congress on Engineering, 2011, pp. 1–6.

[38] D. Whitley, A genetic algorithm tutorial, Statistics and Computing 4 (2) (1994) 65–85. doi:10.1007/BF00175354.

[39] R. Eberhart, J. Kennedy, A new optimizer using particle swarm theory, in: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, 1995, pp. 39–43. doi:10.1109/MHS.1995.494215.

[40] Z. Liu, Y. Wang, S. Yang, K. Tang, An adaptive framework to tune the coordinate systems in nature-inspired optimization algorithms, IEEE Transactions on Cybernetics 49 (4) (2019) 1403–1416. doi:10.1109/TCYB.2018.2802912.

[41] J. Kennedy, R. C. Eberhart, A discrete binary version of the particle swarm algorithm, in: 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Vol. 5, 1997, pp. 4104–4108 vol.5. doi:10.1109/ICSMC.1997.637339.

[42] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1) (1967) 21–27. doi:10.1109/TIT.1967.1053964.

[43] E. I. Altman, Financial ratios, discriminant analysis and the prediction of corporate bankruptcy, The Journal of Finance 23 (4) (1968) 589–609. doi:10.1109/TIT.1967.1053964.

[44] Z.-Z. Liu, J. Huang, Y. Wang, D. Cao, ECoFFeS: A software using evolutionary computation for feature selection in drug discovery, IEEE Access 6 (2018) 20950–20963. doi:10.1109/ACCESS.2018.2821441.

[45] I. T. Jolliffe, Principal Component Analysis, Springer Series in Statistics, Springer-Verlag, New York, 2002. doi:10.1007/b98835.

[46] Q. V. Le, A. Karpenko, J. Ngiam, A. Y. Ng, ICA with reconstruction cost for efficient overcomplete feature learning, Granada, Spain, 2011.

[47] S. Raghu, N. Sriraam, Classification of focal and non-focal EEG signals using neighborhood component analysis and machine learning algorithms, Expert Systems with Applications 113 (2018) 18–32. doi:10.1016/j.eswa.2018.06.031.

[48] B. Tran, B. Xue, M. Zhang, Variable-length particle swarm optimization for feature selection on high-dimensional classification, IEEE Transactions on Evolutionary Computation 23 (3) (2019) 473–487. doi:10.1109/TEVC.2018.2869405.