Cluster Computing (2025)28:285
https://doi.org/10.1007/s10586-024-04988-1

SFE-EANDS: a simple, fast, and efficient algorithm with external archive and normalized distance-based selection for high-dimensional feature selection

Rui Zhong[1] · Yang Cao[2] · Essam H. Houssein[3] · Jun Yu[4] · Masaharu Munetomo[1]

Received: 30 August 2024 / Revised: 11 November 2024 / Accepted: 8 December 2024

Abstract
SFE (i.e., Simple, Fast, and Efficient) is a novel evolutionary approach to deal with the high-dimensional feature selection challenge. The excellent performance and easy implementation of SFE have quickly attracted widespread attention. However, the original SFE is prone to stagnation in the later stages of optimization and lacks effectiveness in escaping local optima due to the single-agent search pattern. To handle these issues, we introduce two strategies (i.e., the External Archive and the Normalized Distance-based Selection mechanism) to improve the performance of SFE and propose SFE-EANDS. The external archive saves the optimal solution found so far to guide the direction of optimization, and the normalized distance-based selection offers a probability for an inferior solution to be accepted. To comprehensively evaluate the performance of SFE-EANDS, we implement experiments on 21 high-dimensional datasets, and ten advanced approaches are applied as competitor algorithms. The experimental results and statistical analyses confirm the effectiveness and efficiency of our proposed SFE-EANDS. Moreover, we have integrated four widely used classifiers into the SFE-EANDS framework to investigate its robustness and scalability. Based on numerical experiments, we recommend integrating the SFE-EANDS with the k-nearest neighbors (KNN) with $k = 1$ to solve the high-dimensional feature selection tasks.

Keywords High-dimensional feature selection · External archive · Normalized distance-based selection · Classification accuracy

1 Introduction

The rapid growth in the artificial intelligence (AI) field drives the explosive increment in the scale of data, and extracting practical knowledge from this vast amount of data has become an essential topic. Feature selection, also known as attribute selection or variable subset selection, is a typical task that involves choosing a representative subset of relevant features for model construction [1, 2]. Since redundant and irrelevant features are removed, feature selection can bring many benefits when solving complex high-dimensional problems, e.g., simplify the model complexity [3], avoid the curse of dimensionality [4, 5],

✉ Rui Zhong
zhongrui@iic.hokudai.ac.jp

Yang Cao
yang.cao.y4@elms.hokudai.ac.jp

Essam H. Houssein
essam.halim@mu.edu.eg

Jun Yu
yujun@ie.niigata-u.ac.jp

Masaharu Munetomo
munetomo@iic.hokudai.ac.jp

[1] Information Initiative Center, Hokkaido University, Sapporo, Japan

[2] Graduate School of Information Science and Technology, Hokkaido University, Sapporo, Japan

[3] Faculty of Computers and Information, Minia University, Minia, Egypt

[4] Institute of Science and Technology, Niigata University, Niigata, Japan

Published online: 28 April 2025

improve data compatibility with a learning model class [6], and encode inherent symmetries in the input space [7, 8]. As a popular research topic in the AI community, numerous techniques have been proposed to deal with various feature selection tasks, and they can be classified into three categories: the filter method, the wrapper method, and the embedded method. We will briefly introduce them in Sect. 2.1.

As one of the most well-studied methods, the wrapper method has attracted widespread attention due to its easy implementation and high accuracy. This approach usually regards the feature selection task as a binary optimization problem. Supposing a trial sample can contain a maximum of D features, then the number of all possible combinations is $2^D - 1$. Since the exhaustive search becomes an NP-hard problem, meta-heuristic algorithms are often employed as the search engineer for wrapper-based feature selection. Fortunately, with the unprecedented development of the evolutionary computation (EC) community, many evolutionary algorithms (EAs) have great potential to deal with various feature selection tasks. In the past two decades, many EAs for feature selection have sprung up like mushrooms: Leardi et al. [9] adopted the genetic algorithm (GA) for feature selection and found that the feature subset found by GA is more efficient than those obtained by classical methods. Khushaba et al. [10] proposed a differential evolution based technique for feature selection (DEFS), where the feature distribution factors associated roulette wheel weighting scheme are employed to map from continuous to discrete space. Liu et al. [11] introduced the mechanism of the survival of the fittest to particle swarm optimization (PSO) and proposed a multi-swarm PSO (MSPSO), a swarm request rule is adopted in MSPSO and transferred to the binary version. Emary et al. [12] proposed two different variants of binary grey wolf optimizer (GWO). In the first version, the first three best solutions are binarized, and then stochastic crossover is activated among three basic solutions to generate the new solution. In the second version, the sigmoid function is used in the original GWO to squash the continuous to discrete search space. Mafarja et al. [13] introduced two transfer functions (i.e., sigmoid and tanh) to the whale optimization algorithm (WOA) and proposed SWOA (sigmoid function based) and TWOA (tanh function based) for the high-dimensional feature selection task. Pourpanah et al. [14] applied the brainstorm optimization (BSO) to the feature selection task, where the Fuzzy ARTMAP (FAM) model is employed as an incremental learning neural network, and the BSO inspired by the human brainstorming process acts as the feature selection approach. Mohamed et al. [15] developed two novel search strategies for the slime mold algorithm (SMA): the two-phase mutation mechanism and the attacking-feeding strategy. The two-phase mutation further exploits better solutions around the best-so-far and the attacking-feeding strategy trades off exploration and exploitation based on the memory saving of each particle. Then, four kinds of binary SMA based on S-shaped and V-shaped transformation functions were proposed. Too et al. [16] proposed a novel dragonfly algorithm (DA) with a hyper-learning scheme that involves the concept of personal best and personal worst solutions in the offspring generation process, which has the potential to improve the food finding and enemy fleeing behaviors, and the embedded V-shape function transfer it to binary version. Thanks to its high accuracy, high-level flexibility, and excellent applicability, EA is one of the most popular techniques for such feature selection tasks.

However, as the number of features increases, the search space will grow exponentially [17], and the curse of dimensionality [18] is inevitable, which decreases the performance of EAs rapidly. Thus, implementing a wrapper-based feature selection technique in high-dimensional data poses a significant challenge. A newly proposed feature selection technique, SFE (i.e., the abbreviation of Simple, Fast, and Efficient) algorithm [19] provides a different potential direction to deal with high-dimensional feature selection tasks. Two simple but effective search operators consist of SFE: non-selection operator and selection operator, which correspond to the exploration and exploitation, respectively. In the exploration stage, the non-selection operator searches for irrelevant and redundant features and removes them from the solution representation. In the exploitation stage, the selection operator searches for the significant features and stores them in the solution representation. As a single-agent search technique, the superiority of SFE can be observed from the comparison experiments with six recently proposed feature selection algorithms [19]. However, we have observed a deficiency in SFE techniques where, during the later stages of optimization, convergence tends to get stuck in stagnation, and the optimization direction easily becomes trapped in a local optimum. In the original paper, the authors also notice this issue and suggest combining SFE with PSO (SFE-PSO) that in the early stage, SFE leads the optimization, and if the optimization enters the late stage and the fitness of the search agent does not change in during certain iterations, SFE-PSO will switch to PSO and search in the sub-domain found by SFE. This approach hybridizes the advantages of SFE and EC techniques, which is promising to deal with the high-dimensional feature selection task. Thus, it is necessary to enhance the exploration ability of SFE and endow the ability to escape local optima.

The main objective of this paper is to further improve the performance of the SFE algorithm and propose an

enhanced version named SFE-EANDS. We introduce two efficient strategies: external archive and normalized distance-based selection to address the poor exploration behavior of SFE in the late stage of optimization. The external archive saves the global optimum found so far, and the normalized distance-based selection allows a probability of deteriorating offspring to be accepted. In numerical experiments, we first implement the comparison experiments among SFE, SFE-EANDS, and other eight advanced meta-heuristic approaches on 21 high-dimensional datasets. The experimental results and statistical analyses prove the efficiency of our proposed SFE-EANDS. Secondly, to investigate the effect of different classifiers on the performance of the SFE-EANDS algorithm, four kinds of common models including the K-nearest neighbors (KNN, k={1, 3, 5}), the logistic classifier, the multi-layer perceptron classifier (MLP), and the support vector classifier (SVC) are embedded to the SFE-EANDS algorithm independently. Based on the experimental results, we recommend integrating the SFE-EANDS with the KNN (k = 1) to solve the high-dimensional feature selection tasks.

The remainder of this paper is organized as follows. Section 2 introduces the related works. Section 3 provides a detailed implementation of our proposed SFE-EANDS. Numerical experiments are designed and executed in Sect. 4. Section 5 discusses the performance of SFE-EANDS and provides some open topics. Finally, Sect. 6 concludes our work.

2 Related works

2.1 The classification of feature selection approaches

In this section, we roughly classify the feature selection techniques based on three evaluation criteria: the filter method, the wrapper method, and the embedded method.

2.1.1 Filter method

The filter method aims to find interesting feature subsets through independent measurements (e.g., coefficient correlation [20], mutual information [21], and Kullback-Liebler divergence [22]), and the benefit of this approach is computationally efficient since it does not involve the learning algorithm and model categories [23]. However, the filter method may ignore some features that are not important separately but can be critical when combined with other features [24]. A demonstration of the filter method is shown in Fig. 1.

Here, the Pearson correlation coefficient is employed as an example. Assuming the data and labels are defined in Eq. (1).

$$X = \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \boldsymbol{x}_3 \\ \vdots \\ \boldsymbol{x}_N \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ x_{31} & x_{32} & \cdots & x_{3D} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix} \quad (1)$$

$$Y = \{y_1, y_2, ..., y_N\}$$

And the Pearson correlation coefficient of the j^{th} feature is calculated using Eq. (2).

$$r_j = \frac{\sum_{i=1}^{N}(x_{ij} - \bar{x_j})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{N}(x_{ij} - \bar{x_j})^2 \sum_{i=1}^{N}(y_i - \bar{y})^2}}, \quad (2)$$

where $\bar{x_j}$ and $\bar{y}$ are the mean of the j^{th} feature and label, respectively. r_j in $[-1, 1]$ denotes the correlation coefficient. Subsequently, the filter method based on the Pearson correlation coefficient identifies top-k features based on the value of r_j as the selected features.

2.1.2 Wrapper method

The wrapper method concentrates on a specific problem for selecting the most suitable feature subset and usually regards the feature selection task as a combinatorial optimization problem. The objective function is designed based on the predictor performance, and the search space is all possible combinations of feature subsets. Figure 2 shows the demonstration of the wrapper method.

A typical framework of the wrapper method contains two major steps. In the first step, a subset combination of features is found by the search algorithm, and in the second step, the predictive model is constructed to evaluate this combination [26]. We take the mean squared error (MSE) as the example of the objective function, as defined in Eq. (3).

$$\min MSE = \min \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i}), \quad (3)$$

where $\hat{y_i}$ is the predicted label. The search and evaluation procedures are repeated until the optimal feature subset is found or the computational budget is exhausted [27], thus this approach is the most accurate among the filter, the wrapper, and the embedded methods. However, it is also the most computationally expensive since it involves model training and feature subset evaluation several times.

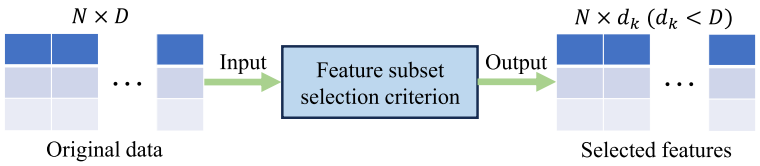


Fig. 1 A demonstration of the filter method [24]. N and D are the size and the dimension of the dataset, and d_k represents the dimension or the number of selected features of the optimum

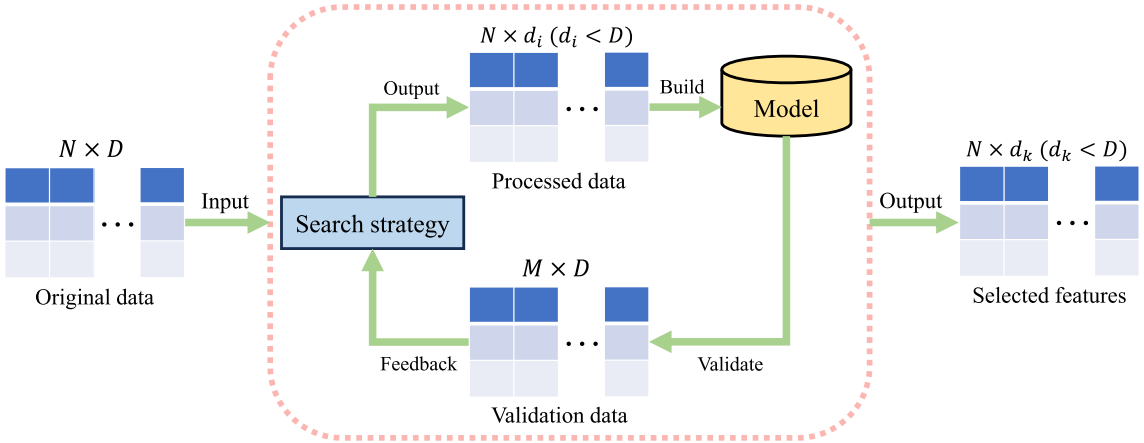


Fig. 2 A demonstration of the wrapper method [25]. N, D, and d_k have similar definitions, and d_i denotes the dimension of intermediate solutions

2.1.3 Embedded method

The embedded method integrates the feature selection algorithm as a part of the learning algorithm, where the feature selection process and model training are executed simultaneously [28]. The benefit of this approach is computationally cheap, considering the interaction with the model, and can avoid overfitting [29]. Figure 3 shows a demonstration of the embedded method.

Here, we take the L1 (LASSO) penalty [31] as an example. In the model training task, assuming the training data $X = \{X_1, X_2, ..., X_N\}$, where $X_i = \{x_1, x_2, ..., x_D\}$ is a vector with D features, training label $y = \{y_1, y_2, ..., y_N\}$, and y_i is a scalar. Based on the ordinary least-squares method [32], the training process can be described by Eq. (4).

$$\min L(y, f(X, W)) = \min \sum_{i=1}^{N} (y_i - f(X_i, W_i))^2 , \tag{4}$$

where $W = \{W_1, W_2, ..., W_D\}$ is the weight vector and can be obtained by minimizing the loss function in Eq. (4), and $f(X, W)$ is the prediction by the model. Then, the L1 method adds an extra constraint to the loss function and enforces some weights of irrelevant features close to 0.

$$\begin{aligned} \min L_{L1}(y, f(X, W)) = \min \sum_{i=1}^{N} (y_i - f(X_i, W_i))^2 \\ + \lambda \sum_{i=1}^{N} |W_i| \end{aligned} \tag{5}$$

Minimizing the loss function in Eq. (5) allows the predictive model to be optimized while the weight of some insignificant features is set to 0 and the feature selection

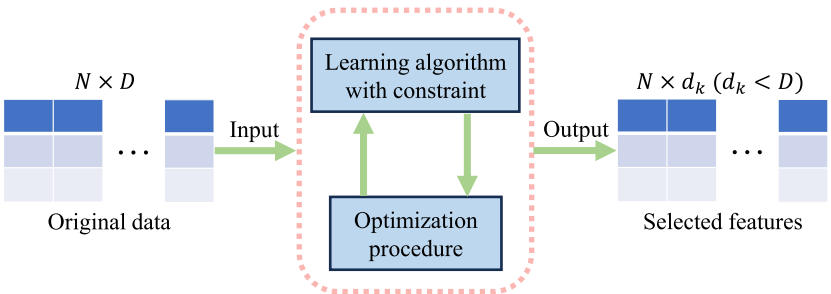


Fig. 3 A demonstration of the embedded method [30]

task is completed. Through the above explanation, the embedded method can achieve model training and feature selection simultaneously. In addition, L2 (ridge) penalty [33], elastic net [34] and random forest [35] are also popular techniques for embedded methods.

2.2 SFE algorithm

As a single-agent evolutionary approach, the SFE algorithm exhibits a simple structure, low computational cost, low memory consumption, and no complex operators. This section introduces four important components of the SFE algorithm: initialization, the non-selection operator, the selection operator, and the objective function design.

2.2.1 Initialization

Since the algorithm adopts a single-point search mode, the initial individual is generated randomly and a demonstration is given in Fig. 4.

SFE has the same binary encoding principle as most EAs for the feature selection task, where $x_i = 1$ indicates the i^{th} feature is selected, and $x_j = 0$ indicates the j^{th} feature is not selected. Equation (6) is applied to generate the initial search agent.

$$x_i = \mathbf{round}(\mathbf{rand}(0, 1)) \ , \tag{6}$$

where $\mathbf{rand}(0, 1)$ can generate a random number from 0 to 1 uniformly, and $\mathbf{round}(\cdot)$ is a rounding function.

2.2.2 The non-selection operator

The essence of the high-dimensional feature selection is the sparse optimization problem [36, 37] since there exist many unimportant and irrelevant features, and the objective of the non-selection operator is to eliminate the irrelevant features. Thus, in each iteration, the candidate size UN for reducing the selected features is calculated in Eq. (7).

$$\begin{aligned} UN &= \lceil UR \cdot nvar \rceil \\ UR &= (UR_{max} - UR_{min}) \cdot \frac{T_{max} - t}{T_{max}} + UR_{min} \end{aligned} \ , \tag{7}$$

where UR is an adaptive parameter to the iteration, $nvar$ is the total number of features, UR_{max} and UR_{min} are constants used to adjust UR and recommended settings are 0.3 and 0.001, T_{max} and t are maximum fitness evaluations and current fitness evaluations, respectively. Here, we provide a visualized example in Fig. 5 to explain how the non-selection operator works.

Supposing the feature indexes and the corresponding trial solution are shown in Fig. 5, and the indexes of selected features idx which equals to [1, 4, 8, 9, 12, 18, 19, 20]. In the first iteration, the $UR = 0.3$ and the trial solution contains 20 features, thus, $UN = 6$ can be computed by Eq. (7), and the non-selection operator randomly samples 6 indexes from idx. Note that the repeat indexes are allowable in the unselected feature indexes list K. Finally, the indexes of corresponding features are set to 0 in the new trial solution, and one iteration of the non-selection operator is finished.

Fig. 4 The encoding of a trial solution [19]

2.2.3 The selection operator

The selection operator highlights the exploitative search during optimization and will be activated only if all features are removed in the offspring solution and assist the algorithm in performing the non-selection mode again. Figure 6 shows the mechanism of the selection operator.

First, the indexes of unselected features idx are extracted based on the trial solution, and SN is a random integer determined from [1, $\mathbf{len}(idx)$]. In this example, the length of idx is 12 (i.e., $\mathbf{len}(idx)$=12), and we suppose $SN = 3$ while the elements in sample list $K = [7, 13, 17]$ are obtained by randomly sampled from idx with 3 times. Finally, in the new trial solution, the corresponding features are set to 1, the selection mode is finished, and the SFE algorithm is back to the non-selection mode.

2.2.4 Objective function

The classification accuracy is adopted to evaluate the trial solution of the feature selection task and is directly used as the fitness of the individual.

$$f(X) = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \cdot 100\% \ , \tag{8}$$

where T_P and T_N are the numbers of positive and negative instances that the classifier has correctly classified. Also, F_P and F_N are the numbers of positive and negative instances that the classifier has wrongly classified.

In summary, the pseudocode of SFE is summarized in Algorithm 1.

Algorithm 1 SFE [19]

Require: Dimension: D, Max. iteration: T_{max}
Ensure: Trial solution: X
Initialize a search agent X by Eq. (6)
Evaluate the solution X by Eq. (8)
$t = 0$
while $t < T_{max}$ **do**
  $X_{new} \leftarrow \mathbf{copy}(X)$
  Calculate UR and UN by Eq. (7)
  Extract the indexes of selected features idx
  Generate the unselected feature indexes list K
  Set corresponding indexes in X_{new} to 0 % **non-selection operator**
  if All features are removed in X_{new} **then**
    $X_{new} \leftarrow \mathbf{copy}(X)$
    Extract the indexes of unselected features idx
    Generate the selected feature indexes list K
    Set corresponding indexes in X_{new} to 1 % **selection operator**
  end if
  if X_{new} has a better fitness value than X **then**
    $X \leftarrow X_{new}$
  end if
  $t \leftarrow t + 1$
end while
return X

3 Our proposal: SFE-EANDS

The efficiency and effectiveness of the original SFE algorithm can be observed from numerical experiments adequately, and this success is thanks to the non-selection operator design, the selection operator design, and the remarkable balance between exploitation and exploration behaviors. However, it is worth noticing that the convergence of optimization tends to get stuck in stagnation, and the optimization direction easily becomes trapped in a local

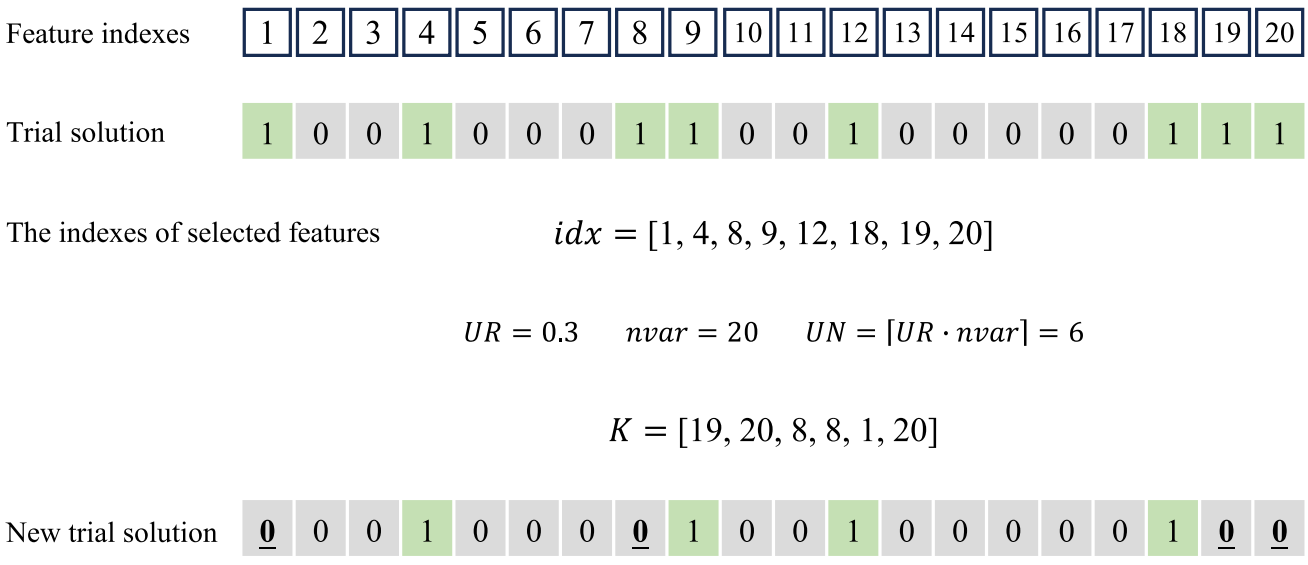


Fig. 5 A demonstration for the non-selection operator [19]. *idx* is the indexes of selected features, *UR* is a parameter to control the number of selected features, *nvar* is the number of the total features, *UN* can be calculated by *UR* and *nvar*, and *K* is an array to save the unselected features

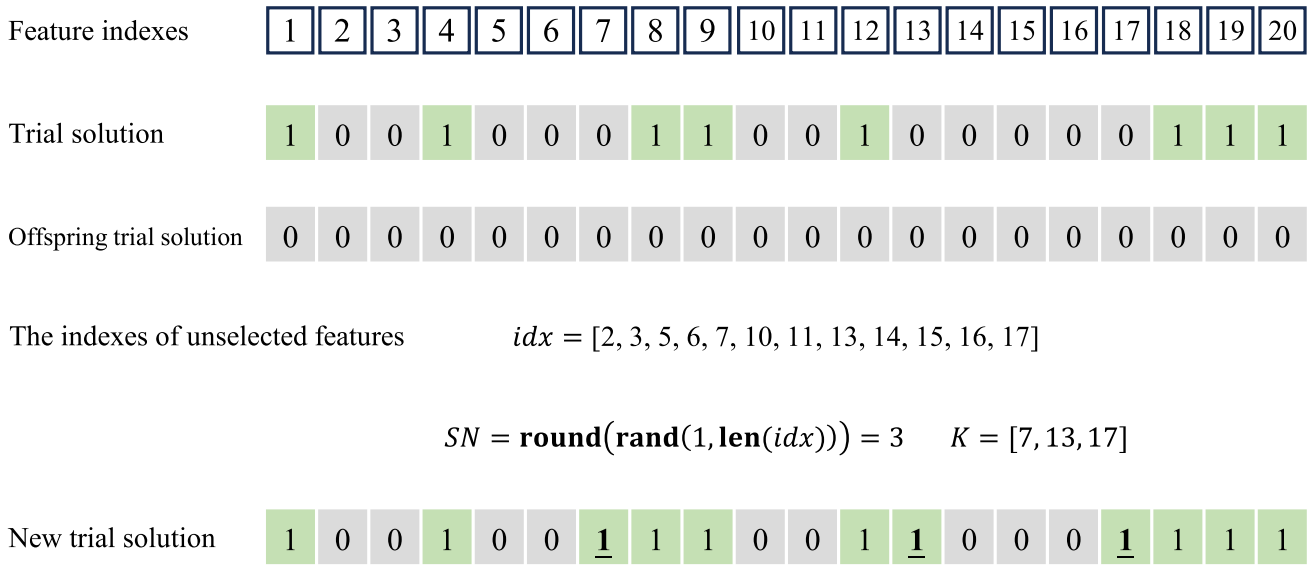


Fig. 6 A demonstration for the selection operator [19]. *idx* and *K* have similar definition in non-selection operator, and *SN* is a parameter to control the selection features

optimum in the late phase, we owe this issue to the basin of attractions [38] and the relatively weak exploration ability of SFE. Once the direction of the optimization falls into a local optimum, it is difficult to escape local optima. Therefore, we propose an improved version of the SFE algorithm named SFE-EANDS, which is embedded with two efficient strategies: the external archive and normalized distance-based selection. We first analyze the shortage of the SFE algorithm from the aspect of getting rid of the local optimum and introduce our proposed strategies.

3.1 Shortage analysis of SFE

The original SFE algorithm adopts the better-acceptance criterion, which indicates that the offspring individual will be accepted only if it has better classification accuracy than the parent individual. Algorithm 1 line 16 shows this acceptance principle, and we owe the convergence stagnation in the late stage of optimization to this greedy selection mechanism. Once the optimization direction fails into a local optimum, the probability of jumping out of the local optimum is pretty low. A visual demonstration is shown in Fig. 7.

Here, we take a trial solution with a total of 20 features as an example, the global optimum and the local optimum are demonstrated in Fig. 7. Supposing the current trial solution cannot find a better offspring solution by the non-selection operator, thus the current trial solution is trapped into a local optimum, and the only way to obtain the global optimum is as follows: in step 1, the non-selection operator removes all features, and SFE algorithm enters the selection mode. In step 2, the selection operator is activated and the constructed solution has a similar structure to the global optimum. Then, in step 3, the non-selection operator continuously searches for better solutions and approaches the global optimum. Moreover, these continuous steps are strictly conditional:

- The non-selection operator that removes all features happens with a small probability.
- The global optimum must be the subset of the selection operator constructed solution (e.g., the global optimum is the subset of the solution in step 2) since the non-selection operator will only remove the redundant features.
- The selection operator constructed solution must have a higher classification accuracy than the original trial solution of the local optimum, otherwise, it will not be rejected by the SFE algorithm.
- The continuous search in step 3 cannot ensure the approach to the global optimum.

Moreover, being a single-agent evolutionary approach, the SFE algorithm exhibits sensitivity to both search positions and search directions. This characteristic is illustrated through a 3-D demonstration in Fig. 8. In this example, the red point denotes the initial solution, while the blue and green points represent the global optimum and local optima of this problem. Notably, if the initial solution moves with the directions indicated by the green arrows, it easily falls into a local optimum. This sensitivity is caused by the inherent greedy selection mechanism, which can be influenced by the basin of attractions. Additionally, the SFE algorithm lacks the occasional loci modification mechanism, such as the mutation mechanism, further contributing

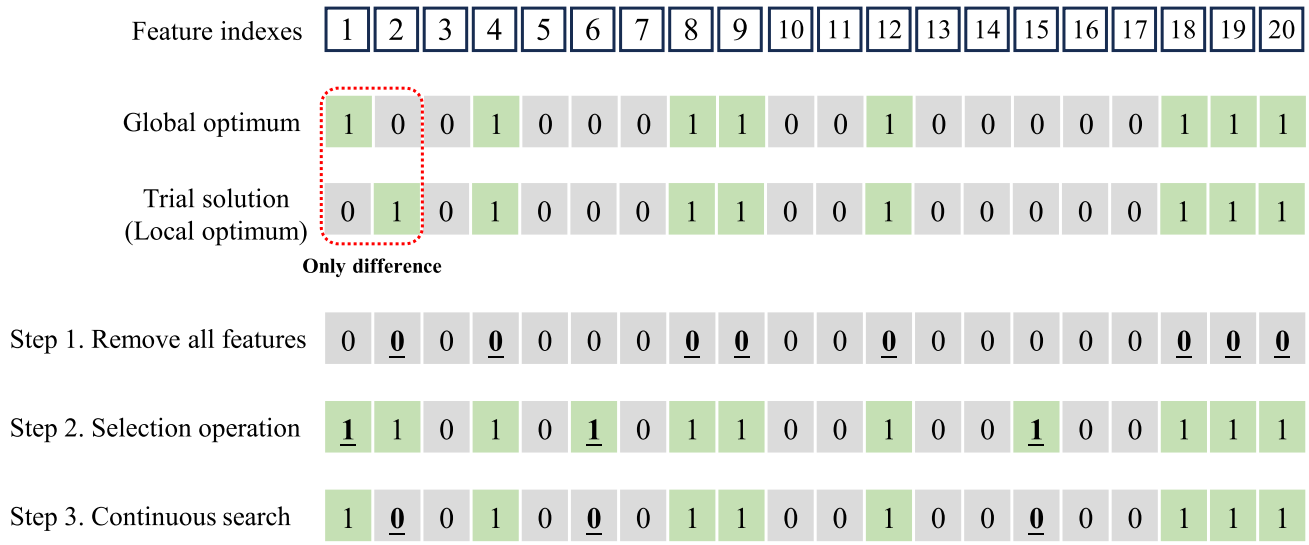


Fig. 7 A demonstration of SFE to escape local optima

to the challenge of escaping from local optima. Therefore, addressing this challenge and designing effective strategies to avoid the attraction from local optima is a severe challenge for the advancement of the SFE algorithm.

From the above explanation, although the difference between the trial solution of the local optimum and the global optimum in Fig. 7 only exists in the 1^{st} and 2^{nd} dimensions, it is almost impossible to jump out from the local optimum and further find the global optimum.

3.2 External archive and normalized distance-based selection

This paper attempts to modify the acceptance criterion to alleviate convergence stagnation and enhance the ability to escape local areas. The core scheme of the proposed selection mechanism is that even the inferior offspring solution still has a probability of being accepted. Besides, the introduction of the normalized distance-based selection mechanism is expected to enhance the search diversity during the optimization.

We introduce an external archive to save the historical best trial solution so far. Given that the SFE algorithm is a single-agent evolutionary approach and the deteriorating solution can be accepted in our proposed selection mechanism, thus, the current optimal solution in our proposal cannot ensure survival, and the non-selection and selection operator may destroy this superior solution construction. Therefore, we adopt a similar mechanism to the PSO, where each particle can memorize the historical best particle and historical best swarm position, and this historical experience can guide the direction of the optimization. Hence, if an offspring individual has a better fitness value than the individual saved in the external archive, the external archive will be updated. When the optimization is terminated, the solution saved in the external archive will be output as the optimal solution.

Subsequently, the normalized distance-based selection is employed in SFE-EANDS. The essence of the normalized

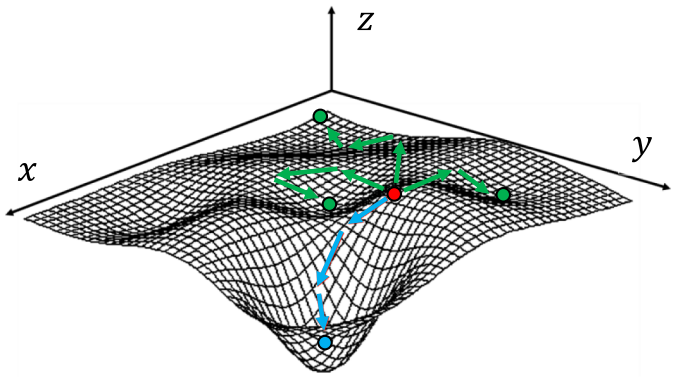


Fig. 8 A demonstration to illustrate the sensitivity of the SFE algorithm

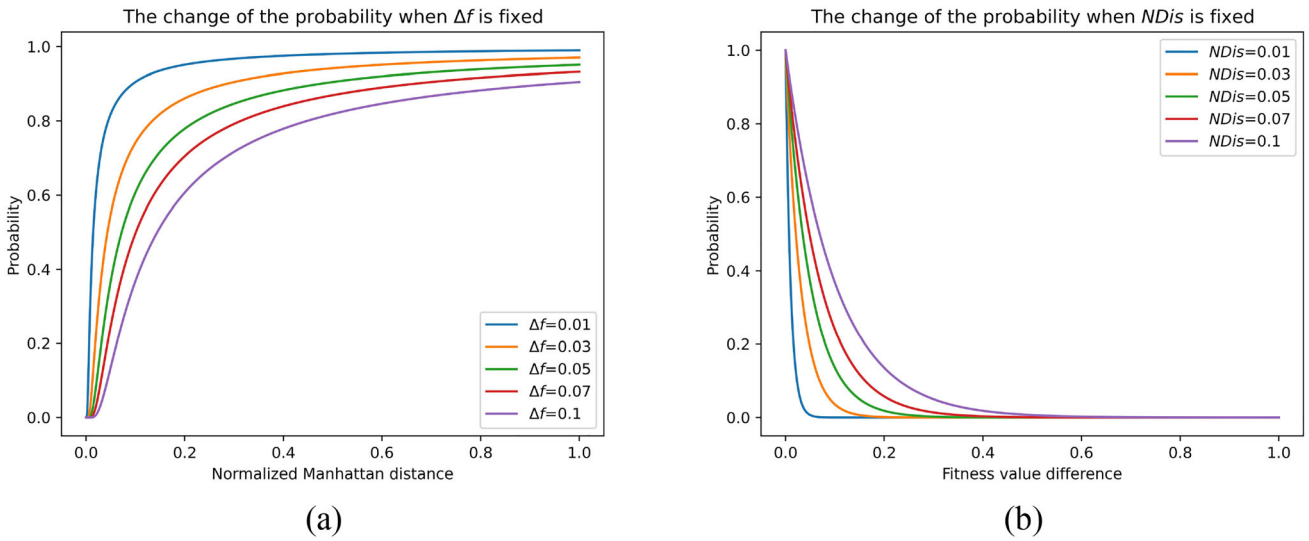


Fig. 9 The change of acceptance probability. **a** When Δf is fixed. **b** When *NDis* is fixed

distance-based selection is a kind of threshold-based selection mechanism, it allows the solution whose fitness value is not highly worse to have a probability of being accepted. As a kind of not-so-greedy selection scheme, the threshold-based selection mechanism has been extensively applied in optimization problems in noisy environments [39, 40] and feature selection tasks [41]. Here, we define the normalized distance-based selection in Eq. (9).

$$X = \begin{cases} X_{new}, if\ f(X_{new}) > f(X) \\ X_{new}, \qquad\qquad else\ if\ r \leq e^{-\frac{\Delta f}{NDis}} \\ X, \qquad\qquad otherwise \end{cases}, \tag{9}$$

where X and X_{new} are the parent the offspring individuals, respectively. $f(\cdot)$ is the objective function defined in Eq. (8), r is a random value in (0,1), $\Delta f = |f(X_{new}) - f(X)|$, and *NDis* denotes the normalized Manhattan distance between X_{new} and X, which can be calculated by Eq. (10).

$$NDis(X, X_{new}) = \frac{1}{D}\sum_{j=1}^{D} |X_j - X_{new,j}| \tag{10}$$

where D is the number of features, X_j and $X_{new,j}$ are the value in the j^{th} dimension of X and X_{new}. Since the objective function $f(\cdot)$ evaluates the classification accuracy of the trial solution and the range is from [0, 1], thus we normalize the calculation of the Manhattan distance in Eq. (10) to eliminate the influence from the dimension.

From our instinct, if the original trial solution X and the offspring trial solution X_{new} have a fixed absolute fitness difference (i.e., Δf is fixed), the acceptance probability will increase as the normalized Manhattan distance since this operation can enhance the diversity of the search agent and guide the direction of the optimization into unknown search sub-regions. On the contrary, if the original trial solution X and the offspring trial solution X_{new} have a fixed normalized Manhattan distance(i.e., *NDis* is fixed), the acceptance probability will decrease as the fitness difference increases since this degeneration of the classification accuracy is unaffordable. To visualize the above explanation, we fix the absolute fitness difference (Δf) to {0.01, 0.03, 0.05, 0.07, 1} and draw the change of the acceptance probability when normalized Manhattan distance (*NDis*) increases from [0, 1]. Similarly, we fix the *NDis* to {0.01, 0.03, 0.05, 0.07, 1} and draw the change of the acceptance probability when Δf increases from [0, 1]. Figure 9 shows the curves of these two cases.

The probability change tendency in Fig. 9a and b are consistent with our imagination such that, if the Δf is fixed, the acceptance probability will increase as the *NDis* becomes larger, and if the *NDis* is fixed, the acceptance probability will decrease as Δf the becomes larger. Therefore, the introduction of normalized distance-based selection to the SFE algorithm is reasonable. Finally, the pseudocode of SFE-EANDS is shown in Algorithm 2, and the difference between SFE and SFE-EANDS can be observed in Algorithm 2 line 3 and from line 22 to 24.

Algorithm 2 SFE-EANDS

Require: Dimension: D, Max. iteration: T_{max}
Ensure: Trial solution: X_{best}
Initialize a search agent X by Eq. (6)
Evaluate the solution X by Eq. (8)
$X_{best} \leftarrow \mathbf{copy}(X)$ % **external archive**
$t = 0$
while $t < T_{max}$ **do**
  $X_{new} \leftarrow \mathbf{copy}(X)$
  Calculate UR and UN by Eq. (7)
  Extract the indexes of selected features idx
  Generate the unselected feature indexes list K
  Set corresponding indexes in X_{new} to 0 % **non-selection operator**
  if All features are removed in X_{new} **then**
    $X_{new} \leftarrow \mathbf{copy}(X)$
    Extract the indexes of unselected features idx
    Generate the selected feature indexes list K
    Set corresponding indexes in X_{new} to 1 % **selection operator**
  end if
  if X_{new} has a better fitness value than X **then**
    if X_{new} has a better fitness value than X_{best} **then**
      $X_{best} \leftarrow \mathbf{copy}(X_{new})$ % **external archive update**
    end if
  else
    if $r \leq e^{-\frac{\Delta f}{NDis}}$ in Eq. (9) is satisfied **then**
      $X_{best} \leftarrow \mathbf{copy}(X_{new})$ % **normalized distance-based selection**
    end if
  end if
  $t \leftarrow t + 1$
end while
return X

4 Numerical experiments

This section introduces the numerical experiments in detail. Section 4.1 introduces the experiment settings: Experimental datasets and compared methods. Section 4.2 shows the experimental and statistical results.

4.1 Experiment settings

To evaluate the performance of our proposed SFE-EANDS, we implement comprehensive numerical experiments on 21 high-dimensional datasets. Table 1 summarizes the information of these datasets, which can be downloaded from.[1] We compare our proposed SFE-EANDS with SFE [19], BPSO [42], BDE [43], SWOA [13], TWOA [13], ISHADE [44], MPAKNN [45], BSSFS [46], BCCOA [47], and mRIME [48]. The population size of BPSO, BDE, SWOA, TWOA, ISHADE, and MPAKNN is set to 30, and the maximum fitness evaluations (*FEs*) are 6,000, which is consistent with the experimental setting in [19]. KNN (k=1) with 5-fold cross-validation is employed as the classifier. All compared algorithms use the recommended hyper-parameter setting in the original paper. In addition, we design a series of experiments to investigate the robustness and scalability of SFE-EANDS in various classifiers. Here, we adopt the KNN (k={1, 3, 5}), the Logistic classifier, the multi-layer perceptron classifier (MLP), and the support vector classifier (SVC) as classifiers, each algorithm in both is independently executed 30 times. All experiments in this paper are programmed by Python 3.11 and implemented in Lenovo Legion R9000P, which is equipped with Windows 11, AMD Ryzen 7 5800 H with Radeon Graphics 3.20 GHz, and 16GB RAM.

[1] http://csse.szu.edu.cn/staff/zhuzx/datasets.htmlhttps://jundongl.github.io/scikitfeature/datasets.htmlhttps://figshare.com/articles/dataset/Microarray_data_rar/7345880/2https://file.biolab.si/biolab/supp/bicancer/projections/info/gastricGSE2685.html

Table 1 The summary of 21 high-dimensional feature selection datasets [19]

Dataset	#Ins	#Fes	#Classes	Area
ALL_AML_4	72	7,129	2	Biological
ALLAML	72	7,129	4	Biological
CLL_SUB_111	111	11,340	3	Biological
CML treatment	28	12625	2	Biological
colon	62	2000	2	Biological
GLI_85	85	22283	2	Biological
GLIOMA	50	4434	4	Biological
leukemia	72	7070	2	Biological
Leukemia_1	72	5327	3	Biological
Leukemia_3	72	11225	3	Biological
lung	203	3312	5	Biological
lymphoma	96	4026	9	Biological
nci9	60	9712	9	Biological
ORL	400	1024	40	Face Image
orlraws10P	100	10304	10	Face Image
Prostate_GE	102	5966	2	Biological
SRBCT	83	2308	4	Biological
SMK_CAN_187	187	19993	2	Biological
TOX_171	171	5748	4	Biological
warpAR10P	130	2400	10	Face Image
Yale	165	1024	15	Face Image

4.2 Experimental results

4.2.1 Compare with other EC algorithms

We compare our proposed SFE-EANDS with other EC algorithms, and each algorithm is equipped with the KNN ($k = 1$) for a fair comparison. Table 2 summarizes the mean and standard deviation (std) classification accuracy on 21 high-dimensional datasets with 30 independent trial runs. Figures 10 and 11 visualize the convergence curves of SFE-EANDS versus other algorithms. In addition, Holm's multiple comparison test is applied to determine whether there is a significant difference between SFE-EANDS and compared algorithms. $+$, $\approx$, and $-$ are applied to represent that our proposed SFE-EANDS is significantly better, with no significance, and significantly worse with the compared method, and the best value is in bold.

Another metric of the feature selection task is the number of the selected features. Table 3 lists the mean scale of the optimal feature subset, while Fig. 12 provides the average classification accuracy and the average number of selected features of optimizers.

4.2.2 Investigate the impact of classifiers on performance

We equip SFE-EANDS with KNN, Logistic classifier, MLP, and SVC to investigate the robustness and scalability of SFE-EANDS when equipped with different classifiers. In addition, the classification task with all features is also implemented to demonstrate the efficiency of SFE-EANDS. Table 4 concludes the experimental results of two categories: SFE-EANDS with the corresponding classifier and no feature selection technique with the corresponding classifier. Figure 13 and 14 provide the convergence curves. Similarly, Holm's multiple comparison test is also applied to determine the significance. Table 5 lists the mean scale of optimal solutions, and the average classification accuracy and the average number of selected features on 21 datasets are visualized in Fig. 15.

5 Discussion

5.1 Computational complexity analysis

To analyze the computational complexity of the proposed SFE-EANDS, we define the parameters used in our proposal as follows: the number of features is D, the maximum iteration is T, and the computational cost of the fitness evaluation is C. Simply, the computational complexity of the non-selection and selection operators are both $O(D)$. To analyze the computational complexity of the normalized distance-based selection mechanism, three cases are involved:

(1). If the offspring individual has a better fitness value, then it will be directly accepted, and the computational complexity is $O(1)$.

(2). Else if $r \leq e^{-\frac{\Delta f}{NDis}}$ is satisfied, in this case, the computational complexity of Δf and $NDis$ are $O(1)$ and $O(D)$, respectively.

(3). Else, the offspring individual will be rejected, and the computational complexity is $O(1)$.

In summary, the computational complexity of the normalized distance-based selection mechanism is max $\{O(1), O(D), O(1)\} = O(D)$, and the whole computational complexity of our proposed SFE-EANDS is $O(T \cdot (D + D + C)) := O(T \cdot (D + C))$. Although the theoretical analysis of the computational complexity between SFE and SFE-EANDS are identical, SFE-EANDS will consume more CPU time in practice since the computational complexity of the selection mechanism in SFE and SFE-EANDS are $O(1)$ and $O(D)$, respectively. Some EC approaches such as BPSO and binary butterfly optimization

Table 2 The mean and standard deviation (std) of optimal classification accuracy on 21 high-dimensional datasets with 30 independent trial runs

Dataset		BPSO	BDE	SWOA	TWOA	ISHADE	MPAKNN	BSSFS	BCCOA	mRIME	SFE	SFE-EANDS
ALL_AML_4	Mean	76.86+	80.57+	84.11+	85.60+	79.71+	77.82+	77.16+	79.73+	79.14+	95.31≈	**96.02**
	Std	1.46	0.57	2.26	2.06	0.70	1.15	1.07	2.33	1.07	2.90	1.82
ALLAML	Mean	84.29+	87.14+	92.28+	92.65+	86.86+	85.00+	84.30+	88.29+	85.96+	98.85+	**99.91**
	Std	0.57	1.07	3.13	2.01	0.70	1.16	1.60	3.63	1.26	1.30	0.34
CLL_SUB_111	Mean	67.38+	71.73+	73.00+	73.25+	69.93+	68.52+	67.42+	68.51+	73.16+	87.61≈	**87.84**
	Std	0.67	1.32	2.26	1.45	0.41	1.30	0.68	1.47	0.38	2.71	2.80
CML treatment	Mean	58.53+	59.33+	84.11+	82.09+	59.33+	59.33+	58.53+	72.53+	59.33+	88.22+	**97.20**
	Std	1.60	0.00	6.87	7.18	0.00	0.00	1.60	9.14	0.00	4.68	2.51
colon	Mean	85.26+	86.95+	92.03+	92.46+	86.59+	85.59+	85.59+	86.92+	85.90+	97.64+	**98.65**
	Std	1.05	0.05	2.75	2.01	0.67	0.79	0.67	1.02	0.79	1.57	1.33
GLI_85	Mean	91.76+	93.18+	94.63+	95.10+	93.88+	92.08+	92.00+	92.94+	92.00+	96.75+	**98.51**
	Std	0.00	0.47	1.24	0.86	0.47	0.52	0.47	1.29	0.47	1.79	1.00
GLIOMA	Mean	80.00+	85.20+	91.80+	91.07+	84.40+	80.73+	80.00+	87.60+	81.60+	96.47≈	**97.07**
	Std	0.00	0.98	2.15	2.24	1.50	1.50	1.26	1.50	1.50	2.29	1.91
leukemia	Mean	63.75+	69.64+	79.74+	79.21+	68.55+	65.50+	65.98+	70.78+	66.88+	92.41+	**94.23**
	Std	1.03	1.39	3.22	3.72	1.39	1.35	2.81	7.08	0.69	2.50	2.67
Leukemia_1	Mean	86.93+	92.32+	94.01+	93.93+	90.34+	88.50+	89.24+	88.91+	89.47+	99.03+	**99.82**
	Std	1.63	1.14	2.27	1.31	1.05	1.61	1.50	1.87	1.11	1.34	0.47
Leukemia_3	Mean	86.25+	89.60+	95.62+	95.80+	89.35+	87.12+	87.64+	89.90+	88.17+	98.50+	**99.96**
	Std	1.62	1.11	1.71	1.87	1.11	1.25	2.22	1.10	1.43	1.89	0.24
lung	Mean	97.14+	97.83+	97.57+	97.80+	97.83+	97.26+	97.05+	97.24+	97.54+	97.70+	**98.69**
	Std	0.37	0.24	0.31	0.35	0.24	0.27	0.31	0.24	0.00	0.73	0.39
lymphoma	Mean	93.97+	96.64+	96.73+	96.74+	96.84+	95.30+	94.57+	95.61+	95.59+	95.93+	**98.62**
	Std	0.39	0.40	0.83	0.82	0.00	0.78	1.00	0.41	1.02	1.30	1.01
nci9	Mean	50.33+	53.67+	58.78+	59.39+	53.67+	50.89+	49.00+	56.00+	51.33+	**71.89** ≈	71.06
	Std	1.25	1.25	2.75	2.53	1.25	1.41	2.26	3.43	1.25	4.90	4.62
ORL	Mean	94.60≈	95.55 −	94.98≈	94.70≈	**95.60** −	95.02≈	94.60≈	94.80≈	95.50 −	94.52≈	94.74
	Std	0.20	0.10	0.51	0.69	0.12	0.32	0.34	0.43	0.27	0.66	0.55
orlraws10P	Mean	96.00+	96.40+	99.53≈	99.90≈	96.00+	96.00+	96.00+	97.80+	96.00+	97.77+	**99.93**
	Std	0.00	0.49	0.62	0.30	0.00	0.00	0.00	1.33	0.00	1.80	0.25
Prostate_GE	Mean	86.34+	89.10+	88.81+	89.50+	87.92+	86.56+	85.40+	87.32+	87.33+	96.07≈	**96.63**
	Std	0.88	0.36	1.12	1.17	0.44	0.76	0.62	1.23	0.63	1.32	1.46
SRBCT	Mean	90.36+	94.36+	95.56+	97.56+	92.55+	90.79+	88.91+	92.22+	91.45+	99.76+	**100.00**
	Std	0.73	0.00	2.74	1.45	0.00	1.10	1.15	1.83	0.89	0.62	0.00
SMK_CAN_187	Mean	61.19+	62.79+	69.26+	69.24+	62.37+	61.76+	61.41+	62.81+	62.57+	75.99+	**77.79**
	Std	0.54	0.64	2.26	2.22	0.26	0.49	0.85	1.61	0.67	2.11	1.94
TOX_171	Mean	61.44+	65.09+	66.34+	67.16+	63.56+	63.32+	63.66+	62.38+	65.32+	81.30+	**83.16**
	Std	0.53	0.43	1.72	1.37	0.59	1.36	1.12	0.60	1.71	3.76	2.26
warpAR10P	Mean	59.08+	62.31+	74.51+	73.72+	62.00+	60.03+	59.23+	67.23+	61.08+	86.18+	**90.18**
	Std	0.58	0.84	2.33	3.37	0.62	0.83	0.49	3.97	0.78	3.48	3.92
Yale	Mean	76.73+	**79.64** ≈	77.19+	76.99+	79.03≈	77.60+	76.24+	76.73+	79.52≈	76.12+	79.49
	Std	0.82	0.48	0.81	0.86	0.62	0.91	0.80	0.73	0.24	1.75	1.67
+/≈/− summary		20/1/0	19/1/1	19/2/0	19/2/0	19/1/1	20/1/0	20/1/0	20/1/0	19/1/1	15/6/0	–

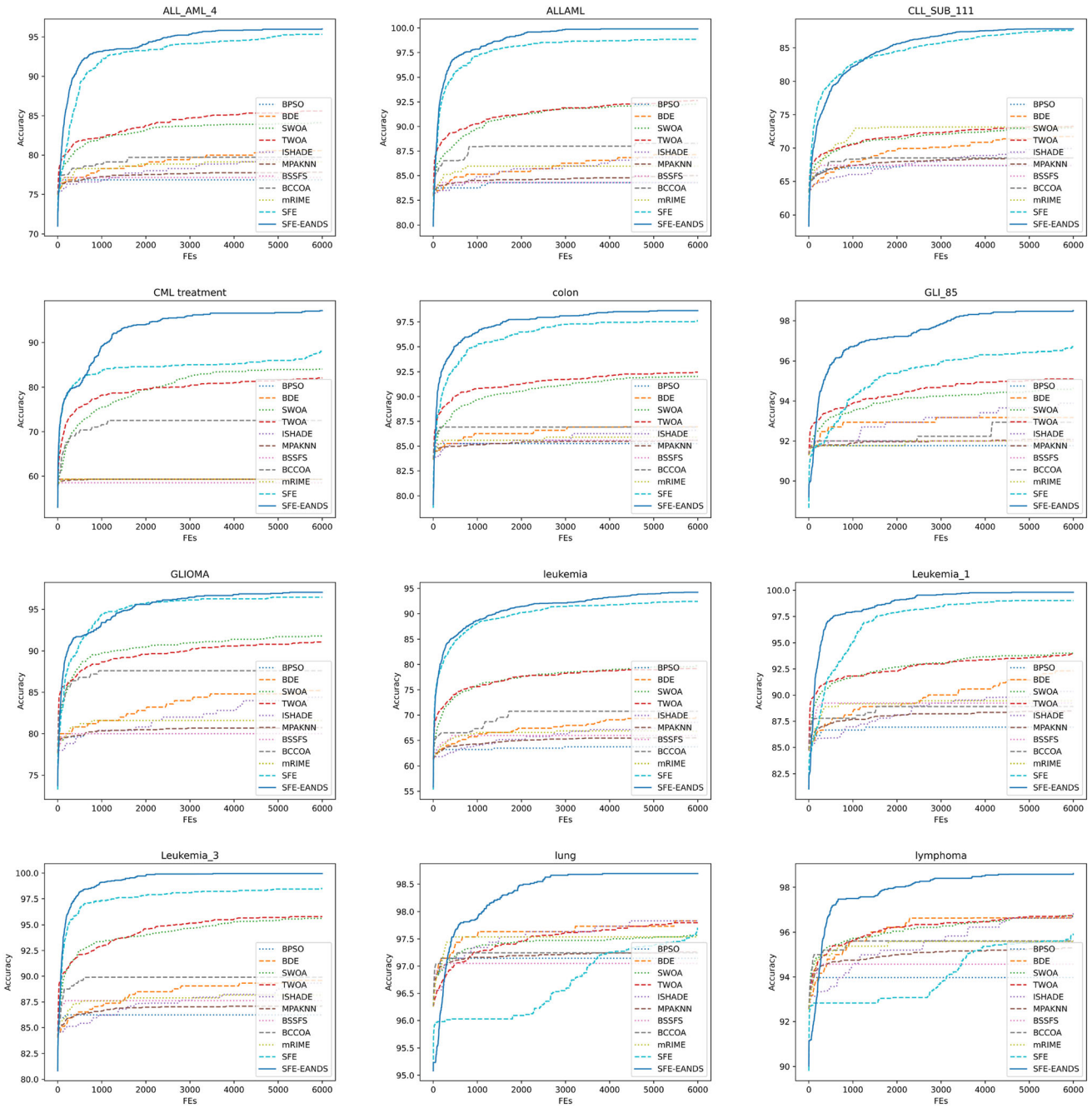


Fig. 10 Convergence curves of SFE-EANDS versus other algorithms with the KNN classifier (k = 1)

algorithm (BBOA) [49] have theoretically identical computational complexity with SFE and SFE-EANDS which is equal to $O(T' \cdot NP \cdot (D + C))$, where T' is the maximum iteration of the population-based approaches and NP is the population size (i.e., $T = N \cdot T'$). However, the computational complexity of BGA and BGWO is $O(T' \cdot NP \cdot (D + C + NP \cdot \log NP))$ in the best case and $O(T' \cdot NP \cdot (D + C + NP^2))$ in the worst case since the sort operator is necessary for these techniques. Therefore, our proposed SFE-EANDS is superior to some EC techniques in the computational complexity aspect.

5.2 Performance analysis based on classification accuracy

As one of the most concerning metrics to evaluate the effectiveness of a feature selection approach, classification accuracy can directly reflect the performance of an algorithm to find the global optimum. From the experimental

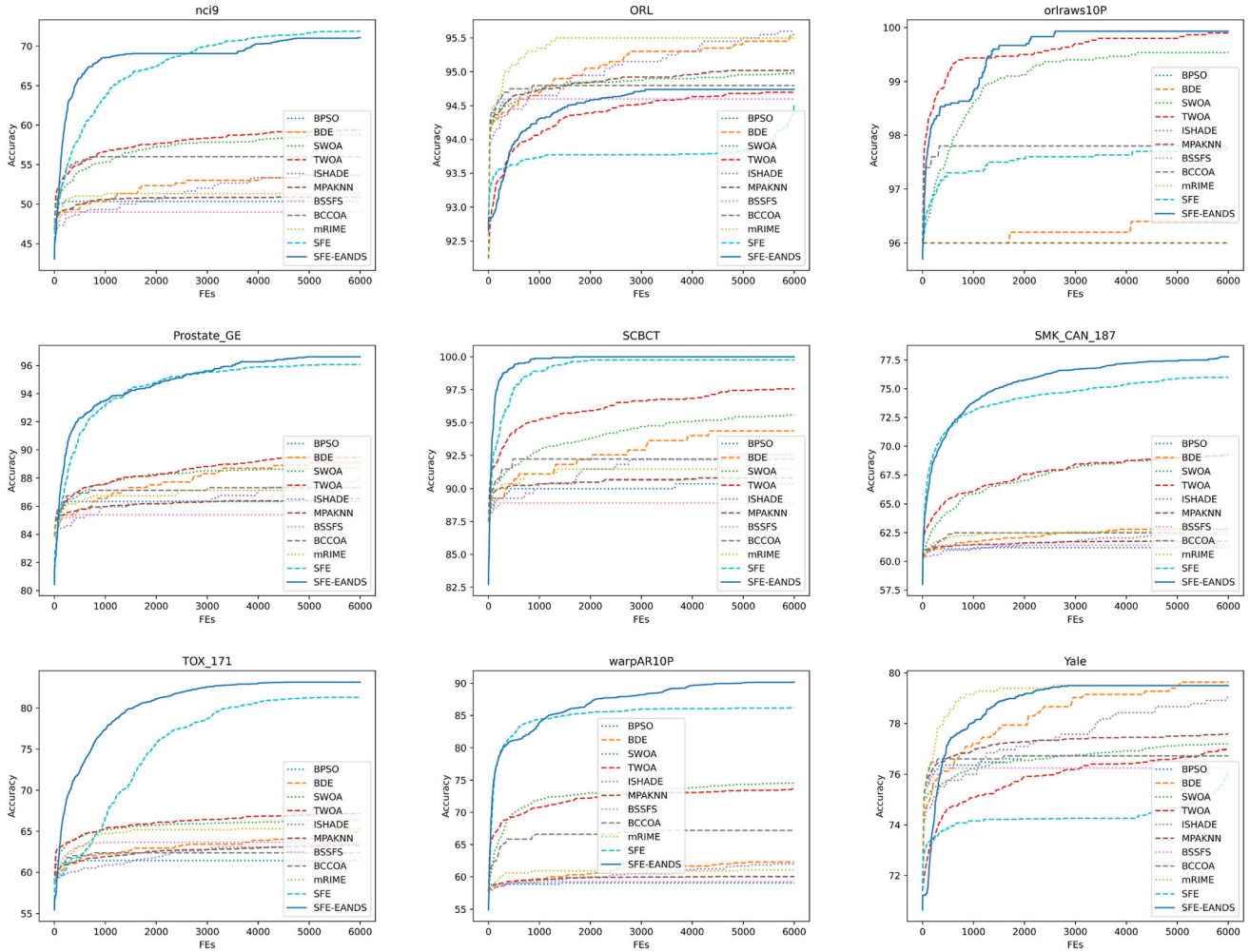


Fig. 11 Convergence curves of SFE-EANDS versus other algorithms with the KNN classifier (k = 1) (Continued)

and statistical results in Table 2, SFE-EANDS performs significantly better than BPSO, BDE, SWOA, TWOA, ISHADE, and MPAKNN on most datasets, and the significant inferiority is only observed in the ORL dataset compared with BDE and ISHADE. However, in the ORL dataset, SFE-EANDS still has better performance than SFE, thus, we infer this inferiority is due to the original SFE is not good at dealing with this specific feature selection task. Besides, from the average classification accuracy on 21 high-dimensional datasets in Fig. 12a, SFE-EANDS is better than BPSO with 14.83%, BDE with 11.93%, SWOA with 7.57%, TWOA with 7.42%, ISHADE with 12.54%, MPAKNN with 14.04%, BSSFS with 14.55%, BCCOA with 11.58%, and mRIME with 13.08%. The above explanation is sufficient to prove the efficiency of SFE-EANDS compared with advanced population-based feature selection techniques in practice, and we owe this superiority to the original architecture design of the SFE algorithm, which has high-performance selection and non-selection operators and outstanding trades-off between exploration and exploitation.

Compared with the original SFE algorithm, the improvement of SFE-EANDS is also obvious. In the summary of the statistical analysis, SFE-EANDS is significantly better than SFE in 15 cases and approximately equal to SFE in 6 cases, and no significant deterioration is observed. 1.70% average improvement of SFE-EANDS is also shown in Fig. 12a. Especially, in the CML treatment dataset, a great improvement of 8.98% exists from SFE to SFE-EANDS. These inspiring improvements practically prove the introduction of our proposed two strategies (i.e., external archive and normalized distance-based selection) is effective in alleviating the influence from the basin of attractions in the original SFE algorithm. The combination of these two strategies encourages the algorithm to explore unknown sub-regions while the information of the global optimum can be saved, which can further improve the performance of the SFE algorithm.

Table 3 The mean scale of the optimal feature subset on 21 high-dimensional datasets with 30 independent trial runs

Dataset	BPSO	BDE	SWOA	TWOA	ISHADE	MPAKNN	BSSFS	BCCOA	mRIME	SFE	SFE-EANDS
ALL_AML_4	3.59e+03	3.60e+03	4.37e+02	3.13e+02	3.55e+03	3.60e+03	3.05e+03	1.19e+03	3.52e+03	**4.43e+01**	2.25e+02
ALLAML	3.60e+03	3.54e+03	3.37e+02	3.05e+02	3.54e+03	3.63e+03	2.74e+03	1.91e+03	3.56e+03	**3.92e+01**	1.58e+02
CLL_SUB_111	5.69e+03	5.68e+03	8.82e+02	4.98e+02	5.65e+03	5.67e+03	4.79e+03	3.29e+03	5.69e+03	**6.83e+01**	2.38e+02
CML treatment	6.27e+03	6.28e+03	3.69e+01	1.85e+01	6.30e+03	6.30e+03	6.55e+03	7.22e+01	6.33e+03	**3.93e+01**	1.92e+02
colon	9.77e+02	9.85e+02	1.45e+02	8.90e+01	9.90e+02	9.97e+02	8.30e+02	7.08e+02	9.90e+02	**1.64e+01**	7.78e+01
GLI_85	1.12e+04	1.12e+04	3.14e+03	1.49e+03	1.12e+04	1.12e+04	1.02e+04	4.94e+03	1.11e+04	**1.70e+02**	4.24e+02
GLIOMA	2.21e+03	2.20e+03	1.78e+02	2.08e+02	2.20e+03	2.21e+03	1.80e+03	4.73e+02	2.21e+03	**2.68e+01**	1.32e+02
leukemia	3.59e+03	3.55e+03	6.20e+01	3.82e+01	3.57e+03	3.56e+03	2.85e+03	7.04e+02	3.57e+03	**4.56e+01**	1.28e+02
Leukemia_1	2.63e+03	2.65e+03	4.77e+02	3.20e+02	2.63e+03	2.69e+03	1.67e+03	1.72e+03	2.64e+03	**3.78e+01**	1.49e+02
Leukemia_3	3.52e+03	3.53e+03	2.03e+02	2.53e+02	3.52e+03	3.53e+03	2.26e+03	1.04e+03	3.54e+03	**3.58e+01**	1.42e+02
lung	1.67e+03	1.64e+03	9.90e+02	3.29e+02	1.64e+03	1.63e+03	1.78e+03	1.43e+03	1.66e+03	**7.77e+01**	1.36e+02
lymphoma	2.03e+03	2.00e+03	5.82e+02	3.12e+02	2.01e+03	2.02e+03	1.32e+03	1.27e+03	2.00e+03	**8.95e+01**	1.61e+02
nci9	4.83e+03	4.85e+03	5.17e+02	4.26e+02	4.81e+03	4.85e+03	4.50e+03	6.16e+02	4.85e+03	**5.80e+01**	2.97e+02
ORL	5.06e+02	5.00e+02	4.35e+02	1.48e+02	5.00e+02	5.11e+02	5.15e+02	4.87e+02	4.96e+02	**6.15e+01**	7.32e+01
orlraws10P	5.14e+03	5.15e+03	1.91e+02	2.23e+02	5.18e+03	5.15e+03	5.12e+03	2.25e+03	5.15e+03	2.56e+03	**1.28e+02**
Prostate_GE	2.97e+03	2.97e+03	8.45e+02	3.79e+02	2.98e+03	2.99e+03	2.76e+03	1.77e+03	2.98e+03	**4.07e+01**	1.41e+02
SRBCT	1.15e+03	1.15e+03	3.14e+02	1.09e+02	1.15e+03	1.14e+03	1.09e+03	5.84e+02	1.15e+03	**1.66e+01**	2.04e+01
SMK_CAN_187	1.00e+04	1.00e+04	1.87e+02	1.06e+02	1.00e+04	9.99e+03	7.22e+03	4.89e+03	1.00e+04	**1.00e+02**	2.78e+02
TOX_171	2.84e+03	2.88e+03	6.45e+02	2.88e+02	2.85e+03	2.87e+03	2.22e+03	1.96e+03	2.92e+03	**6.87e+01**	1.31e+02
warpAR10P	1.19e+03	1.19e+03	4.63e+01	7.09e+01	1.18e+03	1.19e+03	1.13e+03	1.95e+02	1.18e+03	**3.20e+01**	8.75e+01
Yale	5.09e+02	5.17e+02	3.70e+02	1.46e+02	5.19e+02	5.13e+02	5.48e+02	4.87e+02	5.02e+02	**5.61e+01**	6.91e+01
Ave. scale	3.62e+03	3.62e+03	5.25e+02	2.89e+02	3.62e+03	3.63e+03	3.09e+03	1.52e+03	3.62e+03	1.76e+02	**1.61e+02**

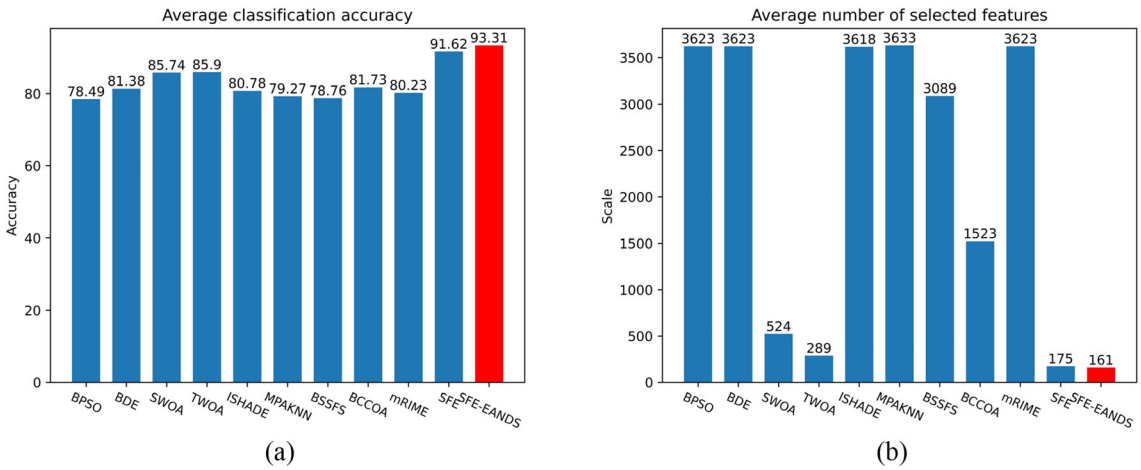


Fig. 12 The average information of numerical experiments. **a** The average classification accuracy on 21 datasets. **b** The average number of selected features on 21 datasets

The convergence information is provided in Figs. 10 and 11. As a single-agent optimization technique, the initial classification accuracy of SFE and SFE-EANDS is inferior to the population-based approaches, but the remarkable convergence speed accelerates SFE and SFE-EANDS to win compared algorithms rapidly, which also indicates the effectiveness of the selection and non-selection operators. In addition, SFE-EANDS has a higher convergence speed than SFE, and the final classification accuracy is also better than SFE, These phenomena also provide experimental support to prove the effectiveness of our proposed two strategies.

5.3 Performance analysis based on the scale of selected features

The scale of selected features is also another important metric to evaluate the effectiveness of the feature selection approach since the main objective of the feature selection task is to minimize the scale of selected features and maximize the classification accuracy. Compared with BPSO, BDE, SWOA, TWOA, ISHADE, and MPAKNN, our proposed SFE-EANDS inherits the superiority of the original SFE and has smaller scales in most datasets. Although TWOA also performs well on some datasets, the average number of selected features on 21 datasets is shown in Fig. 12b, which also reveals the efficiency of SFE-EANDS in reducing the scale of selected features. And we owe this excellent feature reduction capacity to the efficiency of the non-selection operator. The non-selection operator focuses on removing the selected features, while the other operators, such as the crossover and mutation in BDE, velocity, and position update in BPSO, may transfer unselected features into the selected ones. These search strategies will expand the scale of search space and further decelerate the convergence of optimization.

Compared with the original SFE algorithm, the proposed SFE-EANDS has larger scales of selected features than the original SFE on most datasets. During the optimization, the combination of external archive and normalized distance-based selection sacrifices some priority of the objective of reducing the selected feature and endowed a more powerful exploration and exploitation ability to the proposed SFE-EANDS. This balance between minimizing the scale of selected features and maximizing the classification accuracy is acceptable. However, SFE-EANDS still has a smaller average scale on 21 datasets, which is visualized in Fig. 12b. Here, we concentrate on the orlraws10P dataset, the average scale of the feature subset selected by SFE is $2.56e+03$, while SFE-EANDS only selects $1.28e+02$ features, and the deterioration in the orlraws10P dataset is responsible for the phenomenon that SFE has better performance on the number of selected features in most datasets but performs worse than SFE-EANDS in the performance indicator of the average number of selected features. Besides, SFE-EANDS selects fewer features but achieves significantly better classification accuracy in the orlraws10P dataset, which also shows the superiority of our proposed SFE-EANDS in reducing the number of selected features.

5.4 Performance analysis based on various classifiers

The classifier also plays an important role in the feature selection task, thus, we run a set of numerical experiments to investigate the performance of SFE-EANDS when facing various classifiers. Table 4 summarizes the experimental and statistical results. Compared with other KNNs

Table 4 The mean and std of optimal classification accuracy on 21 high-dimensional datasets among four classifiers in 30 independent trial runs (P.: our proposed SFE-EANDS; No.: no feature selection technique)

Dataset		KNN (k=1)		KNN (k=3)		KNN (k=5)		Logis		MLP		SVC	
		P	No	P	No	P	No	P	No	P	No	P	No
ALL_AML_4	Mean	**96.02**	71.81	95.30≈	74.76	93.28+	70.48	95.04≈	84.48	92.70+	76.01	91.82+	68.95
	Std	**1.82**	0.00	1.89	0.00	1.94	0.00	1.55	0.00	2.67	4.26	2.60	0.00
ALLAML	Mean	**99.91**	80.38	99.82≈	79.05	99.81≈	76.38	99.11+	92.86	98.80+	85.55	99.20≈	85.90
	Std	**0.34**	0.00	0.47	0.00	0.61	0.00	0.77	0.00	1.56	4.01	0.94	0.00
CLL_SUB_111	Mean	**87.84**	56.88	86.87+	53.12	84.07+	54.03	80.16+	76.56	82.19+	61.49	83.14+	67.55
	Std	**2.80**	0.00	3.01	0.00	3.52	0.00	3.91	0.00	4.34	3.18	3.75	0.00
CML treatment	Mean	**97.20**	51.33	94.36+	51.33	91.93+	39.33	87.51+	55.33	89.93+	51.40	92.60+	59.33
	Std	**2.51**	0.00	3.36	0.00	3.51	0.00	3.29	0.00	5.32	9.39	3.19	0.00
colon	Mean	**98.65**	78.72	97.82≈	83.59	96.75+	78.59	94.37+	83.59	94.21+	79.17	93.53+	77.05
	Std	**1.33**	0.00	1.53	0.00	1.27	0.00	1.24	0.00	1.48	2.27	1.49	0.00
GLI_85	Mean	**98.51**	88.24	98.35≈	83.53	97.96+	82.35	95.45+	89.41	94.16+	78.12	94.82+	85.88
	Std	**1.00**	0.00	0.78	0.00	1.00	0.00	1.28	0.00	1.47	4.01	1.44	0.00
GLIOMA	Mean	**97.07**	76.00	93.87+	78.00	93.27+	76.00	94.73+	76.00	93.33+	72.80	92.07+	70.00
	Std	**1.91**	0.00	1.63	0.00	2.03	0.00	1.90	0.00	2.09	2.56	2.34	0.00
leukemia	Mean	**94.23**	53.52	92.46+	60.48	91.73+	61.81	86.32+	74.48	88.83+	63.43	89.76+	71.71
	Std	**2.67**	0.00	2.13	0.00	1.72	0.00	2.66	0.00	2.78	5.53	1.81	0.00
Leukemia_1	Mean	**99.82**	80.29	99.77≈	78.76	99.25≈	80.38	98.88+	92.86	98.24+	88.17	98.13+	73.52
	Std	**0.47**	0.00	0.63	0.00	0.87	0.00	0.99	0.00	2.30	3.52	1.10	0.00
Leukemia_3	Mean	99.96	80.29	**100.00** ≈	81.90	**100.00** ≈	84.67	98.09+	92.86	99.44≈	86.66	99.39≈	84.48
	Std	0.24	0.00	**0.00**	0.00	**0.00**	0.00	1.37	0.00	0.85	4.49	0.86	0.00
lung	Mean	98.69	95.06	**98.73** ≈	96.06	98.58≈	93.60	98.14≈	95.57	97.80+	88.78	96.93+	92.12
	Std	0.39	0.00	**0.33**	0.00	0.44	0.00	0.49	0.00	0.76	1.72	0.49	0.00
lymphoma	Mean	**98.62**	88.68	95.38+	89.58	94.86+	89.53	96.08+	88.63	92.89+	83.77	88.86+	77.16
	Std	**1.01**	0.00	0.57	0.00	0.72	0.00	1.12	0.00	2.28	2.30	1.74	0.00
nci9	Mean	71.06	43.33	69.00+	45.00	69.78≈	48.33	**74.67** −	61.67	61.00+	44.00	66.28+	31.67
	Std	4.62	0.00	4.34	0.00	4.05	0.00	**3.71**	0.00	4.78	4.36	4.25	0.00
ORL	Mean	94.74	93.00	90.17+	86.25	87.49+	82.00	**97.33** −	97.25	95.28≈	94.88	95.62 −	95.00
	Std	0.55	0.00	0.78	0.00	1.20	0.00	**0.32**	0.00	0.48	0.52	0.51	0.00
orlraws10P	Mean	99.93	96.00	99.57≈	93.00	99.40≈	89.00	**100.00** ≈	99.00	99.97≈	96.10	**100.00** ≈	99.00
	Std	0.25	0.00	0.50	0.00	0.66	0.00	**0.00**	0.00	0.18	2.17	**0.00**	0.00
Prostate_GE	Mean	**96.63**	81.48	95.98≈	79.52	95.75+	77.48	94.80+	91.24	94.57+	85.37	94.72+	86.38
	Std	**1.46**	0.00	1.03	0.00	1.29	0.00	1.77	0.00	2.02	2.27	1.43	0.00
SRBCT	Mean	**100.00**	85.27	**100.00** ≈	83.45	**100.00** ≈	74.36	**100.00** ≈	98.18	**100.00** ≈	96.16	**100.00** ≈	96.36
	Std	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.00	**0.00**	0.98	**0.00**	0.00
SMK_CAN_187	Mean	77.79	57.77	77.91≈	57.27	77.62≈	61.51	72.54+	60.40	72.44+	57.21	**78.06** ≈	63.09
	Std	1.94	0.00	1.80	0.00	1.82	0.00	1.96	0.00	2.56	3.50	**2.06**	0.00
TOX_171	Mean	**83.16**	53.23	79.92+	58.52	76.97+	56.79	81.03+	61.97	75.74+	59.93	80.86+	60.84
	Std	**2.26**	0.00	2.87	0.00	4.37	0.00	3.42	0.00	4.70	2.65	2.44	0.00
warpAR10P	Mean	90.18	56.92	85.62+	49.23	79.79+	46.15	**92.54** ≈	88.46	86.95+	78.92	88.08+	67.69
	Std	3.92	0.00	3.83	0.00	3.63	0.00	**1.46**	0.00	2.40	1.98	2.41	0.00
Yale	Mean	79.49	71.52	77.74+	65.45	77.33+	64.24	**85.68** −	85.45	79.82≈	76.30	80.87≈	76.97
	Std	1.67	0.00	1.56	0.00	1.72	0.00	**1.50**	0.00	1.92	1.91	1.16	0.00
+/≈/− summary		–		10/11/0		13/8/0		13/5/3		16/5/0		14/6/1	

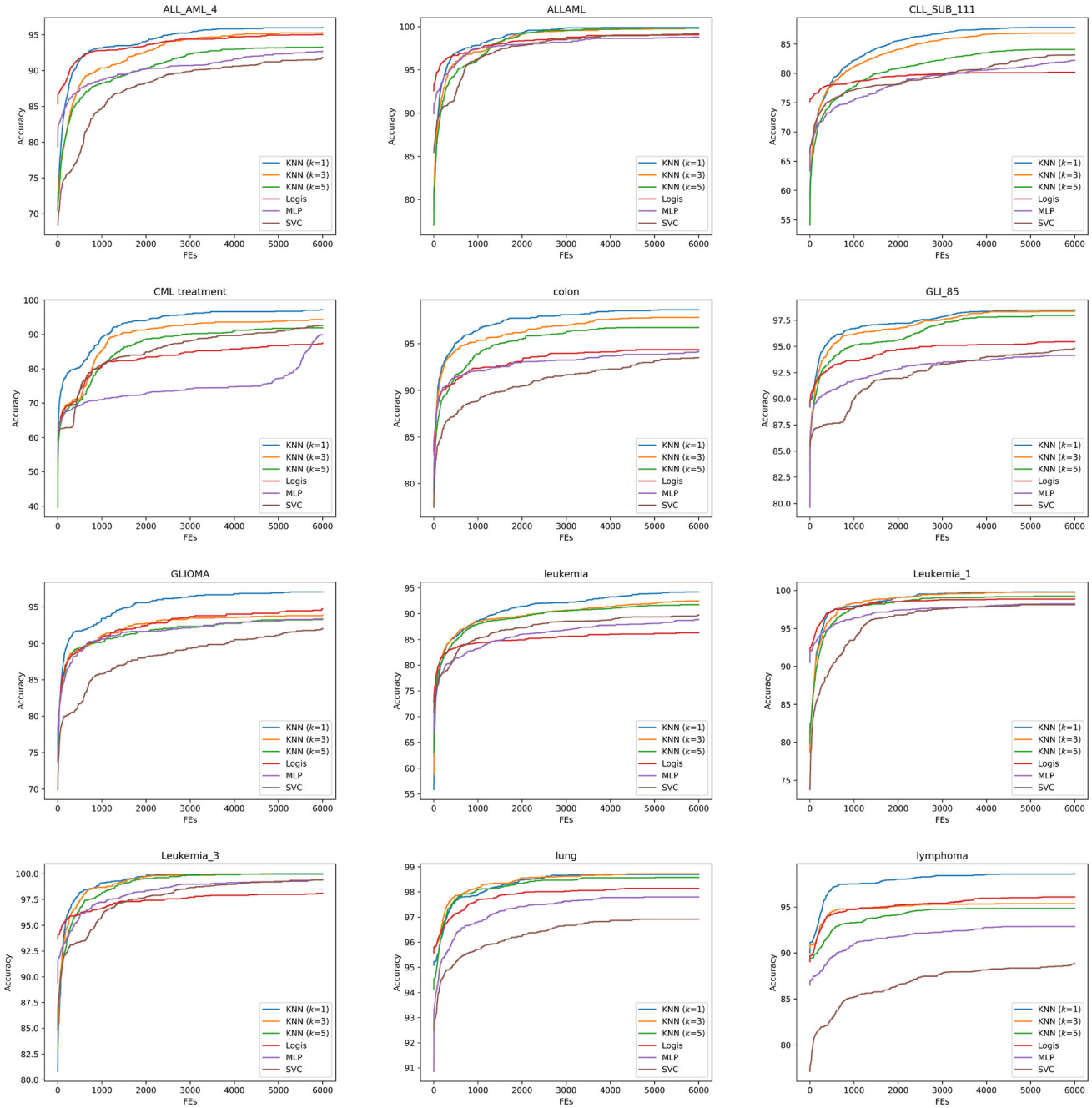


Fig. 13 Convergence curves of SFE-EANDS with various classifiers

with different hyper-parameter settings, KNN (k=1) is significantly better or at least approximately equal to KNN (k=3) and KNN (k=5). In the meantime, some significant deterioration also happens such as in the nci9 dataset between KNN (k=1) and the Logistic classifier, the ORL dataset between KNN (k=1) and the MLP, and the Yale dataset between KNN (k=1) and the SVC. Although the degeneration is observed from numerical experiments, KNN (k=1) still outperforms other classifiers on most high-dimensional datasets. We preliminarily prove that the KNN (k=1) is the best choice among the provided classifiers in these datasets. We recommend using SFE-EANDS cooperating with KNN (k=1) to deal with high-dimensional feature selection tasks together. Furthermore, how to determine the most suitable classifier for the specific feature selection algorithm is an interesting topic for future research.

Moreover, the classifier construction experiment with all features is also implemented to demonstrate the efficiency and effectiveness of our proposed SFE-EANDS.

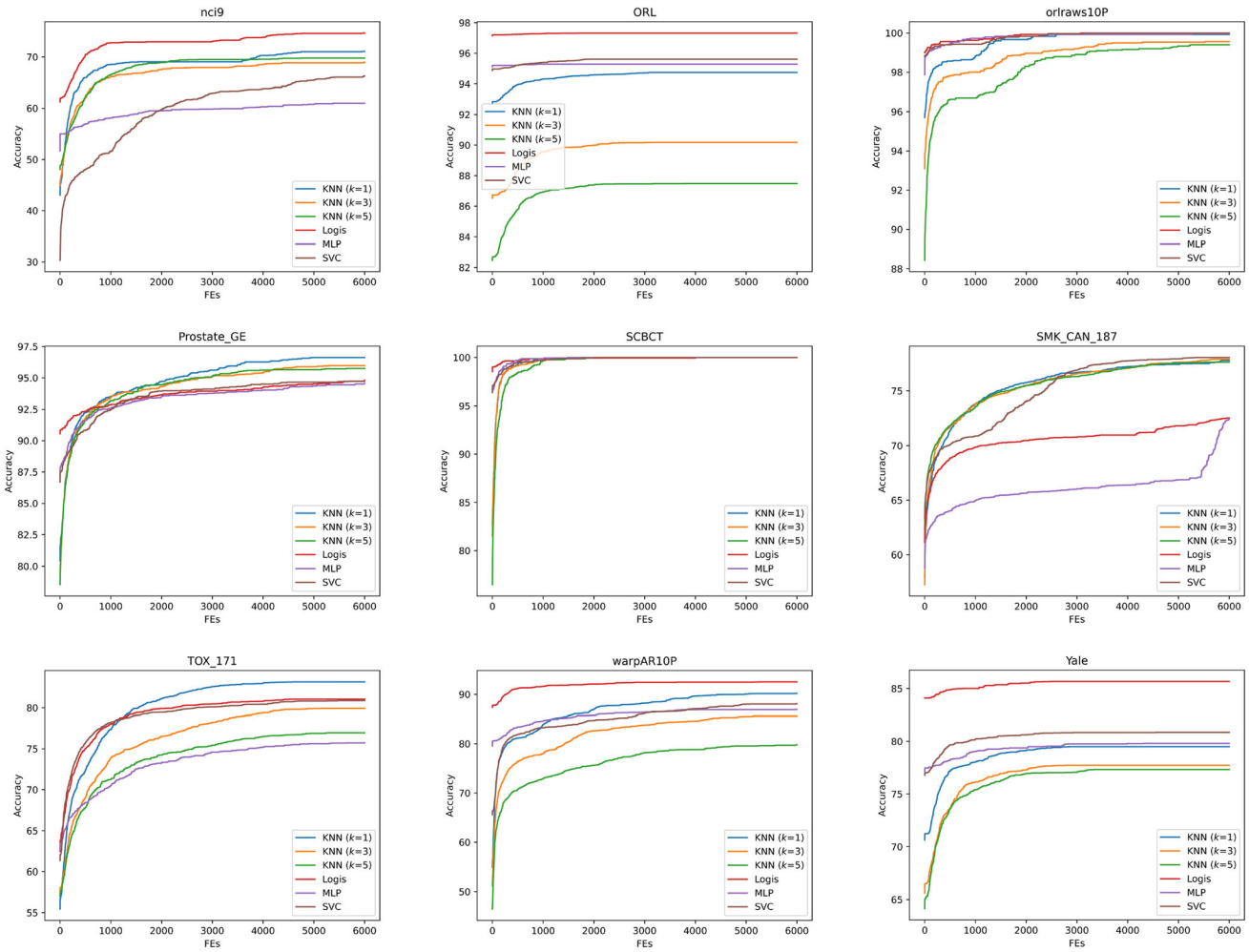


Fig. 14 Convergence curves of SFE-EANDS with various classifiers (Continued)

Experimental results in Table 4 reveal that SFE-EANDS is overwhelmingly superior to the whole feature selection in practice. However, the universal approximation theorem [50] states that neural networks can represent a wide variety of functions when given appropriate weights. As a kind of data compression technique, the feature selection approach must lose some information about the original datasets, thus, the whole feature dataset will construct a better classifier than the dataset with the feature selection approach theoretically and ideally. Nonetheless, real-world applications have computational resource limitations, and the universal approximation theorem does not provide a construction for the weights but merely states that such a construction is possible. Therefore, our proposed SFE-EANDS is a success for feature selection and data compression.

5.5 Open topics and future research

Through the above experimental results and analysis, our proposed SFE-EANDS is an efficient technique for the high-dimensional feature selection task. However, there are still some open topics that can further improve the performance of SFE-EANDS. Here, we list some open topics for future research.

5.5.1 Cooperating with EC approaches

This topic is also involved in the original paper of SFE [19] that SFE is suggested to cooperate with the PSO. First, the SFE algorithm is implemented, and once the convergence stays stagnation more than 1000 FEs, SFE-PSO switches to the PSO and searches in the sub-region for better solutions. Thus, it is reasonable and promising to combine SFE-EANDS with other EC approaches. Recently, many powerful and advanced EAs have been proposed, and the

Table 5 The mean scale of the optimal feature subset on 21 high-dimensional datasets with 30 independent trial runs

Dataset	KNN (k=1)	KNN (k=3)	KNN (k=5)	Logis	MLP	SVC
ALL_AML_4	2.25e+02	1.98e+02	1.70e+02	2.06e+02	**5.37e+01**	1.47e+02
ALLAML	1.58e+02	1.49e+02	1.36e+02	2.13e+02	**6.36e+01**	1.43e+02
CLL_SUB_111	2.38e+02	1.75e+02	1.33e+02	8.65e+02	**5.69e+01**	1.52e+02
CML treatment	1.92e+02	2.00e+02	1.45e+02	1.25e+02	**2.40e+01**	2.75e+01
colon	7.78e+01	6.98e+01	7.15e+01	8.35e+01	**3.14e+01**	3.37e+01
GLI_85	4.24e+02	4.02e+02	3.57e+02	4.86e+02	**1.04e+02**	1.97e+02
GLIOMA	1.32e+02	1.56e+02	1.37e+02	9.14e+01	**5.24e+01**	5.29e+01
leukemia	1.28e+02	1.32e+02	1.58e+02	1.50e+02	**5.25e+01**	1.41e+02
Leukemia_1	1.49e+02	1.50e+02	1.72e+02	1.90e+02	**5.57e+01**	1.48e+02
Leukemia_3	1.42e+02	1.21e+02	1.30e+02	2.12e+02	**5.68e+01**	1.17e+02
lung	1.36e+02	1.34e+02	1.35e+02	1.38e+02	**5.50e+01**	1.15e+02
lymphoma	1.61e+02	1.86e+02	1.75e+02	1.66e+02	1.09e+02	**1.03e+02**
nci9	2.97e+02	2.94e+02	3.36e+02	3.12e+02	3.37e+02	**1.43e+02**
ORL	7.32e+01	7.22e+01	**6.82e+01**	3.50e+02	3.79e+02	1.11e+02
orlraws10P	1.28e+02	1.98e+02	2.03e+02	1.12e+02	**9.94e+01**	1.02e+02
Prostate_GE	1.41e+02	1.58e+02	1.70e+02	3.13e+02	**6.35e+01**	1.67e+02
SRBCT	2.04e+01	2.24e+01	3.18e+01	2.71e+01	**1.54e+01**	2.25e+01
SMK_CAN_187	2.78e+02	2.13e+02	2.03e+02	1.45e+02	**2.85e+01**	5.65e+01
TOX_171	1.31e+02	9.85e+01	7.80e+01	1.18e+02	**4.64e+01**	1.26e+02
warpAR10P	8.75e+01	6.65e+01	**5.36e+01**	1.33e+02	7.13e+01	7.85e+01
Yale	6.91e+01	6.59e+01	**6.05e+01**	1.62e+02	1.09e+02	6.61e+01
Ave. scale	1.61e+02	1.55e+02	1.49e+02	2.19e+02	**8.88e+01**	1.07e+02

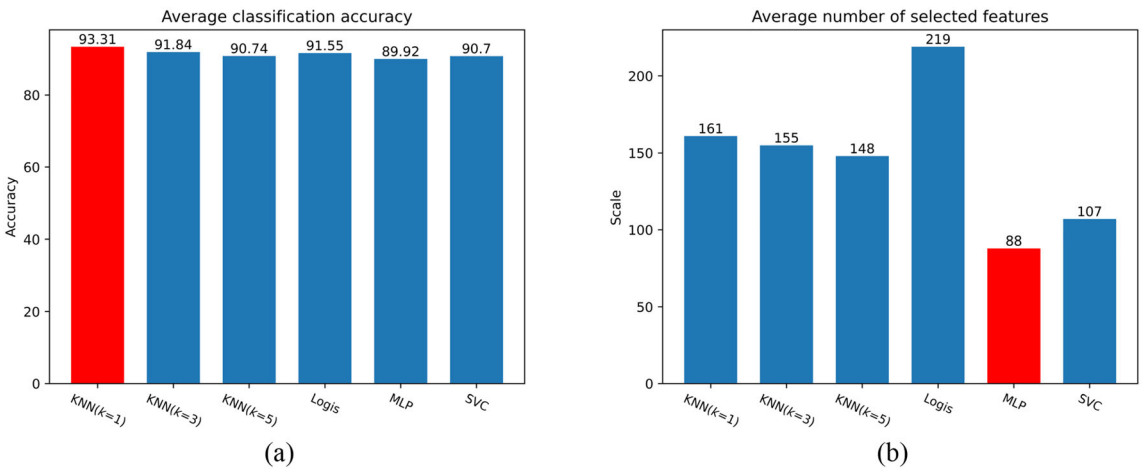


Fig. 15 The average information of numerical experiments. **a** The average classification accuracy of SFE-EANDS with various classifiers on 21 datasets. **b** The average number of selected features of SFE-EANDS with various classifiers on 21 datasets

collaboration of SFE-EANDS with these state-of-the-art EAs is expected to further improve the performance.

5.5.2 Introducing some efficient techniques to SFE-EANDS

SFE-EANDS has a similar architecture to the original SFE algorithm, and we can introduce some efficient techniques to SFE-EANDS from three aspects: initialization, non-selection operator, selection operator, and objective function design, which corresponds to the sub-sections in Sect. 2.

Many pieces of literature state that the essence of the high-dimensional feature selection task is a sparse combinatorial optimization problem, and numerous redundant and irrelevant features exist in the original feature set

[51–54], thus, it is promising to adopt the sparse generation techniques such as sparse population sampling [55] to initialize the search agent for SFE-EANDS.

Parameter tuning is an important procedure in the non-selection operator since many parameters exist in the non-selection operator of SFE-EANDS: the upper and lower bound of the number of selected features UR_{min} and UR_{max}, and the linearly adaptive scheme of UR during the optimization. These parameters can also be optimized, and how to determine the best combination of these parameters is a promising topic to further develop SFE-EANDS.

The selection operator is activated and adds the features randomly when all features are removed from the feature subset. The efficiency of this stochastic feature restore strategy can be observed in the numerical experiments. Another research topic is that we can introduce the memory mechanism [56, 57] to SFE-EANDS and the more important feature has a high probability of being added by the selection operator, which is a potential technique to improve the performance of SFE-EANDS.

Finally, the objective function in feature selection tasks is also an interesting research topic since there are many formats of the objective function in wrapper-based feature selection tasks. Except for the classification accuracy which is adopted in this paper, we also list some common objective functions in literature.

$$Fit = w_1 \cdot err + w_2 \cdot (|S|/|N|) \quad s.t. w_1 + w_2 = 1 \tag{11}$$

The definition of the objective function in Eq. (11) is one of the most common instances for the feature selection task [58], where w_1 and w_2 are two control parameters, err indicates the classification error rate of a given classifier, $|S|$ and $|N|$ represent the number of the selected features and the number of the total features of a dataset. The minimization of this objective function can realize the feature selection process.

Similarly, paper [59] defines a maximizing objective function for feature selection tasks.

$$Fit = w_1 \cdot acc - w_2 \cdot (|S|/|N|) \quad s.t. w_1 + w_2 = 1 \tag{12}$$

where acc means the classification accuracy of a certain classifier, and the rest of the components have the identical definition with Eq. (11).

Furthermore, two main objectives are involved in the feature selection task: Maximizing the classification accuracy and minimizing the number of selected features, therefore, multi-objective optimization techniques can be adopted to solve feature selection tasks, and the objective function is expressed in Eq. (13).

$$Fit = \begin{cases} \min |S| \\ \min err \end{cases} \tag{13}$$

A simple idea is to separate two components in Eq. (11) independently, which is also the most popular multi-objectives function [60–63]. Besides, the reliability of the data [64] can also be formulated as an objective.

$$Fit = \begin{cases} \min \dfrac{\sum_{i=1}^{N} x_i e_i}{\sum_{i=1}^{N} x_i} \\ \max acc \end{cases} \tag{14}$$

where e_i in [0, 1] represents the reliability of the i^{th} feature x_i. More categories of objective function design can be found in [65]. Therefore, we can extend various objective functions to the SFE-EANDS algorithm and the further development of multi-objective SFE-EANDS is a potential topic for our future research.

6 Conclusion

This paper focuses on the shortage of the original SFE algorithm and introduces two strategies: the external archive and the normalized distance-based selection mechanism, and we name our proposed SFE-EANDS. The external archive saves the optimal solution found so far and the normalized distance-based selection mechanism allows an opportunity for a deteriorating solution to be accepted. The effectiveness and efficiency of these two strategies are practically proven in numerical experiments with 21 high-dimensional datasets. In addition, we investigate the robustness and scalability of SFE-EANDS with various classifiers, four common classifiers are embedded into the SFE-EANDS algorithm, and we suggest cooperating the SFE-EANDS algorithm with the KNN (k=1) based on the experimental results.

Author contributions Rui Zhong: Conceptualization, Methodology, Investigation, Writing - original draft, Writing - Review & Editing, and Funding Acquisition. Yang Cao: Formal Analysis, Validation, and Writing - Review & Editing. Essam H. Houssein: Investigation, Resources, and Writing - Review & Editing. Jun Yu: Methodology, Software, and Writing - review & editing. Masaharu Munetomo: Writing - review & editing, Supervision, and Project Administration.

Funding This work was supported by JSPS KAKENHI Grant Number 21A402 and 24K15098 and JST SPRING Grant Number JPMJSP2119.

Data availability The datasets of this research can be downloaded from http://csse.szu.edu.cn/staff/zhuzx/datasets.html, https://jundongl.github.io/scikitfeature/datasets.html, https://figshare.com/articles/

dataset/Microarray_data_rar/7345880/2, https://file.biolab.si/biolab/supp/bicancer/projections/info/gastricGSE2685.html, and the source code is open access in https://github.com/RuiZhong961230/SFE-EANDS.

Declarations

Conflict of interest The authors declare no Conflict of interest.

References

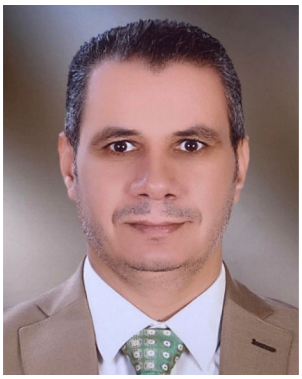
1. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Comput. Electric. Eng. **40**(1), 16–28 (2014). https://doi.org/10.1016/j.compeleceng.2013.11.024
2. Khaire, U.M., Dhanalakshmi, R.: Stability of feature selection algorithm: a review. J. King Saud Univ. Comput. Inform. Sci. **34**(4), 1060–1073 (2022). https://doi.org/10.1016/j.jksuci.2019.06.012
3. Liu, H.: Feature Selection, pp. 402–406. Springer, Boston (2010). https://doi.org/10.1007/978-0-387-30164-8_306
4. Chauhan, D., Shivani, Cheng, R.: Competitive swarm optimizer: a decade survey. Swarm Evolu. Comput. **87**, 101543 (2024). https://doi.org/10.1016/j.swevo.2024.101543
5. Zhong, R., Yu, J.: Gene-targeting multiplayer battle game optimizer for large-scale global optimization via cooperative coevolution. Clust. Comput. **27**, 12483–12508 (2024). https://doi.org/10.1007/s10586-024-04600-6
6. Kratsios, A., Hyndman, C.: Neu: a meta-algorithm for universal uap-invariant feature representation. J. Mach. Learn. Res. **22**(92), 1–51 (2021)
7. Yarotsky, D.: Universal approximations of invariant maps by neural networks. Constr. Approx. **55**, 407–474 (2022). https://doi.org/10.1007/s00365-021-09546-1
8. Chauhan, D.: Shivani: offline learning-based competitive swarm optimizer for non-linear fixed-charge transportation problems. Swarm Evol. Comput. **88**, 101608 (2024). https://doi.org/10.1016/j.swevo.2024.101608
9. Leardi, R., Boggia, R., Terrile, M.: Genetic algorithms as a strategy for feature selection. J. Chemom. **6**(5), 267–281 (1992). https://doi.org/10.1002/cem.1180060506
10. Khushaba, R.N., Al-Ani, A., Al-Jumaily, A.: Differential evolution based feature subset selection. In: 2008 19th International Conference on Pattern Recognition, pp. 1–4 (2008). https://doi.org/10.1109/ICPR.2008.4761255
11. Liu, Y., Wang, G., Chen, H., Dong, H., Zhu, X., Wang, S.: An improved particle swarm optimization for feature selection. J. Bionic Eng. **8**(2), 191–200 (2011). https://doi.org/10.1016/S1672-6529(11)60020-6
12. Emary, E., Zawbaa, H.M., Hassanien, A.E.: Binary grey wolf optimization approaches for feature selection. Neurocomputing **172**, 371–381 (2016). https://doi.org/10.1016/j.neucom.2015.06.083
13. Mafarja, M., Jaber, I., Ahmed, S.: Whale optimization algorithm for high-dimensional small-instance feature selection. In: 2018 Fifth International Symposium on Innovation in Information and Communication Technology (ISIICT), pp. 1–6 (2018). https://doi.org/10.1109/ISIICT.2018.8613293
14. Pourpanah, F., Shi, Y., Lim, C.P., Hao, Q., Tan, C.J.: Feature selection based on brain storm optimization for data classification. Appl. Soft Comput. **80**, 761–775 (2019). https://doi.org/10.1016/j.asoc.2019.04.037
15. Abdel-Basset, M., Mohamed, R., Chakrabortty, R.K., Ryan, M.J., Mirjalili, S.: An efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection. Comput. Indust. Eng. **153**, 107078 (2021). https://doi.org/10.1016/j.cie.2020.107078
16. Too, J., Mirjalili, S.: A hyper learning binary dragonfly algorithm for feature selection: a covid-19 case study. Knowl.-Based Syst. **212**, 106553 (2021). https://doi.org/10.1016/j.knosys.2020.106553
17. Chauhan, D., Yadav, A.: A comprehensive survey on artificial electric field algorithm: theories and applications. Archiv. Comput. Methods Eng. **31**, 2663–2715 (2024). https://doi.org/10.1007/s11831-023-10058-3
18. Köppen, M.: The curse of dimensionality. In: 5th Online World Conference on Soft Computing in Industrial Applications (WSC5), vol. 1, pp. 4–8 (2000)
19. Ahadzadeh, B., Abdar, M., Safara, F., Khosravi, A., Menhaj, M., Suganthan, P.: Sfe: a simple, fast and efficient feature selection algorithm for high-dimensional data. IEEE Transactions on Evolutionary Computation **PP**, 1–1 (2023). https://doi.org/10.1109/TEVC.2023.3238420
20. Blessie, E.C., Karthikeyan, E.: Sigmis: a feature selection algorithm using correlation based method. J. Algorithms Comput. Technol. **6**, 385–394 (2012). https://doi.org/10.1260/1748-3018.6.3.385
21. Beraha, M., Metelli, A.M., Papini, M., Tirinzoni, A., Restelli, M.: Feature selection via mutual information: New theoretical insights. 2019 International Joint Conference on Neural Networks (IJCNN), 1–9 (2019). https://doi.org/10.1109/IJCNN.2019.8852410
22. Yang, C., Hou, B., Ren, B., Hu, Y., Jiao, L.: Cnn-based polarimetric decomposition feature selection for polsar image classification. IEEE Trans. Geosci. Remote Sens. **57**(11), 8796–8812 (2019). https://doi.org/10.1109/TGRS.2019.2922978
23. Cherrington, M., Thabtah, F., Lu, J., Xu, Q.: Feature selection: Filter methods performance challenges. In: 2019 International Conference on Computer and Information Sciences (ICCIS), pp. 1–4 (2019). https://doi.org/10.1109/ICCISci.2019.8716478
24. Kumar, V.: Feature selection: a literature review. Smart Comput. Rev. (2014). https://doi.org/10.6029/smartcr.2014.03.007
25. Zhong, R., Zhang, C., Yu, J.: Chaotic vegetation evolution: leveraging multiple seeding strategies and a mutation module for global optimization problems. Evolu. Intell. (2024). https://doi.org/10.1007/s12065-023-00892-6
26. Nguyen, B.H., Xue, B., Zhang, M.: A survey on swarm intelligence approaches to feature selection in data mining. Swarm Evol. Comput. **54**, 100663 (2020). https://doi.org/10.1016/j.swevo.2020.100663
27. Abdel-Basset, M., Mohamed, R., Chakrabortty, R.K., Ryan, M.J., Mirjalili, S.: An efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection. Comput. Indust. Eng. **153**, 107078 (2021). https://doi.org/10.1016/j.cie.2020.107078
28. Pudjihartono, N., Fadason, T., Kempa-Liehr, A.W., O'Sullivan, J.M.: A review of feature selection methods for machine learning-based disease risk prediction. Frontiers Bioinform. (2022). https://doi.org/10.3389/fbinf.2022.927312
29. Lu, M.: Embedded feature selection accounting for unknown data heterogeneity. Expert Syst. Appl. **119**, 350–361 (2019). https://doi.org/10.1016/j.eswa.2018.11.006
30. Liu, H., Zhou, M., Liu, Q.: An embedded feature selection method for imbalanced data classification. IEEE/CAA J. Automat. Sinica **6**(3), 703–715 (2019). https://doi.org/10.1109/JAS.2019.1911447
31. Muthukrishnan, R., Rohini, R.: Lasso: A feature selection technique in predictive modeling for machine learning. In: 2016 IEEE International Conference on Advances in Computer Applications

(ICACA), pp. 18–20 (2016). https://doi.org/10.1109/ICACA.2016.7887916
32. Jiang, B.-N.: On the least-squares method. Comput. Methods Appl. Mech. Eng. **152**(1), 239–257 (1998). https://doi.org/10.1016/S0045-7825(97)00192-8
33. Wang, G., Sun, L., Wang, W., Chen, T., Guo, M., Zhang, P.: A feature selection method combined with ridge regression and recursive feature elimination in quantitative analysis of laser induced breakdown spectroscopy. Plasma Sci. Technol. **22**(7), 074002 (2020). https://doi.org/10.1088/2058-6272/ab76b4
34. Amini, F., Hu, G.: A two-layer feature selection method using genetic algorithm and elastic net. Expert Syst. Appl. **166**, 114072 (2021). https://doi.org/10.1016/j.eswa.2020.114072
35. Sylvester, E.V.A., Bentzen, P., Bradbury, I.R., Clément, M., Pearce, J., Horne, J., Beiko, R.G.: Applications of random forest feature selection for fine-scale genetic population assignment. Evol. Appl. **11**(2), 153–165 (2018). https://doi.org/10.1111/eva.12524
36. Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., Saeed, J.: A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. J. Appl. Sci. Technol. Trends **1**(2), 56–70 (2020). https://doi.org/10.38094/jastt1224
37. Wu, D., He, Y., Luo, X., Zhou, M.: A latent factor analysis-based approach to online sparse streaming feature selection. IEEE Trans. Syst. Man Cybernet.: Syst. **52**(11), 6744–6758 (2022). https://doi.org/10.1109/TSMC.2021.3096065
38. Vlad, S.E.: 24 - Basins of Attraction. In: Vlad, S.E. (ed.) Boolean Systems, pp. 221–236. Academic Press, New York (2023). https://doi.org/10.1016/B978-0-32-395422-8.00030-1
39. Ghosh, A., Das, S., Mallipeddi, R., Das, A.K., Dash, S.S.: A modified differential evolution with distance-based selection for continuous optimization in presence of noise. IEEE Access **5**, 26944–26964 (2017). https://doi.org/10.1109/ACCESS.2017.2773825
40. Zhong, R., Tu, B., Zhang, E., Munetomo, M.: Cooperative coevolutionary differential evolution with adjacent intensity matrix with linkage identification for large-scale optimization problems in noisy environments. Evolu. Intell. (2024). https://doi.org/10.1007/s12065-024-00941-8
41. Kalita, D., Singh, V., Kumar, V.: Two way threshold based intelligent water drops feature selection algorithm for accurate detection of breast cancer. Soft. Comput. **26**, 2277–2305 (2021). https://doi.org/10.1007/s00500-021-06498-3
42. Kennedy, J., Eberhart, R.C.: 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, pp. 4104–41085 (1997). https://doi.org/10.1109/ICSMC.1997.637339
43. Wang, L., Fu, X., Mao, Y., Ilyas Menhas, M., Fei, M.: A novel modified binary differential evolution algorithm and its applications. Neurocomputing **98**, 55–75 (2012). https://doi.org/10.1016/j.neucom.2011.11.033
44. Dong, H., Wang, X., Wang, X., Sun, J., Li, T.: A feature selection method based on adaptive differential evolution. In: 2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS), pp. 208–213 (2019). https://doi.org/10.1109/ICIS46139.2019.8940171
45. Elminaam, D.S.A., Nabil, A., Ibraheem, S.A., Houssein, E.H.: An efficient marine predators algorithm for feature selection. IEEE Access **9**, 60136–60153 (2021). https://doi.org/10.1109/ACCESS.2021.3073261
46. Sun, L., Si, S., Ding, W., Xu, J., Zhang, Y.: Bssfs: binary sparrow search algorithm for feature selection. Int. J. Mach. Learn. Cybern. **14**(8), 2633–2657 (2023). https://doi.org/10.1007/s13042-023-01788-8
47. Zhong, R., Zhang, C., Yu, J.: Cooperative coati optimization algorithm with transfer functions for feature selection and knapsack problems. Knowl. Inf. Syst. **66**, 6933–6974 (2024). https://doi.org/10.1007/s10115-024-02179-3
48. Abu Khurma, R., Braik, M., Alzaqebah, A., Gopal Dhal, K., Damaševičius, R., Abu-Salih, B.: Advanced rime architecture for global optimization and feature selection. J. Big Data **11**(1), 89 (2024). https://doi.org/10.1186/s40537-024-00931-8
49. Arora, S., Anand, P.: Binary butterfly optimization approaches for feature selection. Expert Syst. Appl. **116**, 147–160 (2019). https://doi.org/10.1016/j.eswa.2018.08.051
50. Kratsios, A., Papon, L.: Universal approximation theorems for differentiable geometric deep learning. J. Mach. Learn. Res. **23**(1) (2022)
51. Fu, G., Wu, Y.-J., Zong, M.-J., Pan, J.: Hellinger distance-based stable sparse feature selection for high-dimensional class-imbalanced data. BMC Bioinform. (2020). https://doi.org/10.1186/s12859-020-3411-3
52. Nie, F., Wang, Z., Tian, L., Wang, R., Li, X.: Subspace sparse discriminative feature selection. IEEE Trans. Cybern. **52**(6), 4221–4233 (2022). https://doi.org/10.1109/TCYB.2020.3025205
53. Cheng, F., Chu, F., Xu, Y., Zhang, L.: A steering-matrix-based multiobjective evolutionary algorithm for high-dimensional feature selection. IEEE Trans. Cybern. **52**(9), 9695–9708 (2022). https://doi.org/10.1109/TCYB.2021.3053944
54. Chakraborty, S., Das, S.: Detecting meaningful clusters from high-dimensional data: a strongly consistent sparse center-based clustering approach. IEEE Trans. Pattern Anal. Mach. Intell. **44**(6), 2894–2908 (2022). https://doi.org/10.1109/TPAMI.2020.3047489
55. Kropp, I., Nejadhashemi, A.P., Deb, K.: Benefits of sparse population sampling in multi-objective evolutionary computing for large-scale sparse optimization problems. Swarm Evol. Comput. **69**, 101025 (2022). https://doi.org/10.1016/j.swevo.2021.101025
56. Li, X., Yang, G.: Artificial bee colony algorithm with memory. Appl. Soft Comput. **41**, 362–372 (2016). https://doi.org/10.1016/j.asoc.2015.12.046
57. Črepinšek, M., Liu, S.-H., Mernik, M., Ravber, M.: Long term memory assistance for evolutionary algorithms. Mathematics (2019). https://doi.org/10.3390/math7111129
58. Tu, Q., Chen, X., Liu, X.: Hierarchy strengthened grey wolf optimizer for numerical optimization and feature selection. IEEE Access **7**, 78012–78028 (2019). https://doi.org/10.1109/ACCESS.2019.2921793
59. Rashid, A.N.M.B., Ahmed, M., Sikos, L.F., Haskell-Dowland, P.: A novel penalty-based wrapper objective function for feature selection in big data using cooperative co-evolution. IEEE Access **8**, 150113–150129 (2020). https://doi.org/10.1109/ACCESS.2020.3016679
60. Kozodoi, N., Lessmann, S.: Multi-objective particle swarm optimization for feature selection in credit scoring. In: Mining Data for Financial Applications: 5th ECML PKDD Workshop, MIDAS 2020, Ghent, Belgium, September 18, 2020, Revised Selected Papers, pp. 68–76. Springer, Berlin, Heidelberg (2020). https://doi.org/10.1007/978-3-030-66981-2_6
61. Zhang, Y., Gong, D.-W., Gao, X.-Z., Tian, T., Sun, X.-y.: Binary differential evolution with self-learning for multi-objective feature selection. Inform. Sci. **507**, 67–85 (2020). https://doi.org/10.1016/j.ins.2019.08.040
62. Nayak, S.K., Rout, P.K., Jagadev, A.K., Swarnkar, T.: Elitism based multi-objective differential evolution for feature selection: a filter approach with an efficient redundancy measure. J. King Saud Univ. Comput. Inform. Sci. **32**(2), 174–187 (2020). https://doi.org/10.1016/j.jksuci.2017.08.001
63. Rafie, A., Moradi, P., Ghaderzadeh, A.: A multi-objective online streaming multi-label feature selection using mutual information.

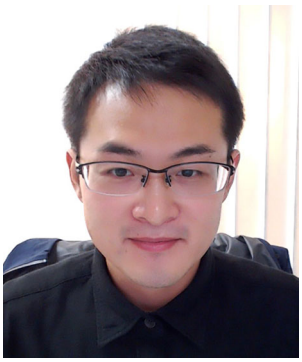
Expert Syst. Appl. **216**, 119428 (2023). https://doi.org/10.1016/j.eswa.2022.119428
64. Yong, Z., Dun-wei, G., Wan-qiu, Z.: Feature selection of unreliable data using an improved multi-objective pso algorithm. Neurocomputing **171**, 1281–1290 (2016). https://doi.org/10.1016/j.neucom.2015.07.057
65. Al-Tashi, Q., Abdulkadir, S.J., Rais, H.M., Mirjalili, S., Alhussian, H.: Approaches to multi-objective feature selection: a systematic literature review. IEEE Access **8**, 125076–125096 (2020). https://doi.org/10.1109/ACCESS.2020.3007291

Rui Zhong received the B.Eng. degree from Huazhong Agricultural University, China in 2019, the M.Eng. degree from Kyushu University, Japan in 2022, and a Ph.D. at Hokkaido University, Japan in 2024. He is now a Specifically Appointed Assistant Professor at Information Initiative Center, Hokkaido University. He has published more than 20 articles in high reputation journals. His research interests include Evolutionary Computation, Large-scale Global Optimization, and Meta-/Hyper-heuristics. His profile in google scholar can be found at https://scholar.google.com/citations?hl=en&user=xd1vrwIAAAAJ&view_op=list_works&sortby=pubdate.

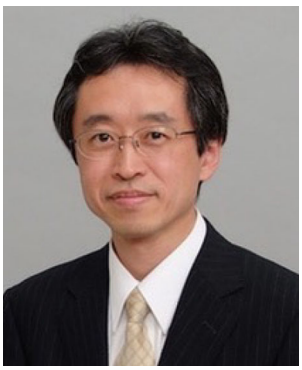
Yang Cao received the B.Eng. degree from Southwest Jiaotong University, China in 2015, the M.Eng. degree from Tokyo Metropolitan University, Japan in 2020, and a Ph.D. Student at Hokkaido University, Japan in 2024. His research interests include Evolutionary Computation, Artificial Intelligence, and Meta-/Hyper-heuristics.



Essam H. Houssein (Member, IEEE) received Ph.D. degree in computer science, in 2012. He is currently a Professor of Artificial Intelligence at the Faculty of Computers and Information, Minia University, Minia, Egypt. He is the founder and chair of the Artificial Intelligence Research (AIR) Group, Egypt. He is selected as a Highly Cited Researcher 2023, in 2024 Edition of the Ranking of Top Scientists in the field of Computer Science. He has published more than 240 scientific research articles in prestigious international journals. His research interests include Meta-heuristics Optimization Algorithms, Artificial Intelligence, WSN, Bioinformatics, Internet of Things, Artificial Intelligence, Image Processing, and Data Mining. He serves as a reviewer for more than 120 journals, such as Elsevier, Springer, and IEEE.



Jun Yu received a Bachelor degree from Northeastern University, China in 2014, and a Master degree and a doctorate from Kyushu University, Japan in 2017 and 2019, respectively. He is currently an Assistant Professor at Niigata University, Japan. His research interests include evolutionary computation, artificial neural networks, and machine learning.



Masaharu Munetomo received the B.A. degree in electrical engineering, the M.A. degree in information engineering, and the Ph.D. degree in information engineering from Hokkaido University, Sapporo, Japan, in 1991, 1993, and 1996, respectively., Since 2012, he has been a Full Professor with the Information Initiative Center and Graduate School of Information Science and Technology, Hokkaido University, where he is also the Director of the Information Initiative Center since 2019. He is involved in several research projects. He has authored over 200 journal and conference papers. His current research interests include evolutionary computation, machine learning and cloud computing.