Applied Soft Computing Journal 169 (2025) 112574

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Feature selection in high-dimensional classification via an adaptive multifactor evolutionary algorithm with local search

Zhihui Li [a], Hong Li [a,*], Weifeng Gao [a], Jin Xie [a], Adam Slowik [b]

[a] School of Mathematics and Statistics, Xidian University, Xi'an 710126, China
[b] Department of Electronics and Computer Science, Koszalin University of Technology, Koszalin, Poland

ARTICLE INFO

Keywords:
Feature selection
Evolutionary multitasking
Multifactor optimization
Local search
High-dimensional dataset classification
Knowledge transfer

ABSTRACT

As datasets grow in dimension and sample size, feature selection becomes increasingly important in machine learning. Features are often associated with multiple tasks, so adopting a multi-task optimization framework in feature selection can improve its classification performance. Multifactor optimization provides a powerful evolutionary multi-tasking paradigm capable of simultaneously handling multiple related optimization tasks. Taking inspiration from these, this article proposes a parameter adaptive multifactor feature selection algorithm (AMFEA). To help the algorithm escape from local optima, AMFEA uses a local search strategy to assist the algorithm in finding the global optimum. In addition, AMFEA has designed an adaptive knowledge transfer parameter matrix that dynamically adjusts parameter sizes based on the population's fitness to control the frequency of knowledge transfer between tasks. This effectively transfers knowledge between different tasks and helps the algorithm converge quickly. Experimental results on 18 high-dimensional datasets show that AMFEA significantly improves classification accuracy compared with evolutionary algorithms and traditional feature selection methods.

1. Introduction

Feature selection is crucial in machine learning as it can effectively improve the overall efficiency of algorithms and reduce model overfitting. There are three main types of feature selection methods: filtered, wrapped, and embedded [1]. The filtered feature selection method is independent of the model and evaluates and ranks features before model training [2]. It selects the most valuable features based on their association with class labels, aiming to refine the model and reduce complexity by eliminating redundant and noisy features [3]. Standard criteria for assessing the importance of features include mutual information [4], analysis of variance [5], the Chi-square test [6], and Pearson's correlation coefficient [7]. The filtering method simplifies the model by calculating the importance of features and selecting important ones, thereby improving the model's predictive performance.

However, filtering methods cannot capture the interaction between features and model performance [8]. Moreover, in practical applications, filtering methods are easily affected by changes in data distribution and feature relationships. Unlike the filtered approach, the wrapped approach integrates the feature selection process into model training, assessing its impact on performance by gradually adding or removing features [9]. Recursive feature culling, for example, is a commonly used wrapped approach that iteratively refines a subset of features by training the model and removing the least important features [10]. Although these methods can optimize feature subsets based on model performance [11], they are computationally heavy and slow compared to filtered methods [2]. Embedded methods typically utilize random forest algorithms [12] or regression regularization techniques (such as minimum absolute contraction and selection operators) to balance feature importance and model complexity [13]. However, traditional embedded methods usually only focus on the overall correlation of features and may ignore the relevant features [14].

The analysis of most feature selection methods mentioned above shows that they have the disadvantages of extensive computation and that it is easy to ignore the correlation between features. In addition, traditional feature selection methods often encounter local optimal solutions [15]. Heuristic algorithms have become an effective tool for feature selection due to their flexibility in data representation and ability to search a vast space, aiming to solve these problems [16]. A variety of heuristic algorithms have been used, such as genetic algorithm [17], particle swarm optimization [18], differential evolution [19], grasshopper optimization algorithm [20], whale swarm algorithm [21] and grey

* Corresponding author.
E-mail addresses: lizhihui@stu.xidian.edu.cn (Z. Li), lihong@mail.xidian.edu.cn (H. Li), gfw@xidian.edu.cn (W. Gao), jxie@xidian.edu.cn (J. Xie), adam.slowik@tu.koszalin.pl (A. Slowik).

https://doi.org/10.1016/j.asoc.2024.112574
Received 16 May 2024; Received in revised form 11 November 2024; Accepted 25 November 2024
Available online 3 December 2024

wolf optimization [22]. Feature selection can be modeled to single-objective and multi-objective optimization problems. Studies in [23,24] describe how heuristic algorithms are applied to single-objective and multi-objective feature selection. When using the single-task single-objective heuristic algorithm for feature selection, the sample and feature space can be divided or clustered. For example, Song et al. [25] proposed a single-objective particle swarm optimization algorithm. Xu et al. [26] developed a single-task multi-objective algorithm that uses repeated analysis to filter redundant solutions and modifies the select reserved solution method. These algorithms prove the effectiveness of heuristic algorithms in feature selection. However, they do not account for situations where multiple related tasks may share specific characteristics.

To address the limitations of feature selection, this paper proposes an evolutionary multi-task optimization approach. This method utilizes the knowledge of similar tasks to process multiple tasks simultaneously, thereby improving performance in the iterative process [27]. Because traditional filtered methods can quickly obtain important feature subsets without training the model, they are often combined with multi-task optimization frameworks to help algorithms select feature subsets with high classification performance. For example, Chen et al. [28] used Relief-F to generate feature weights and knee point selection to generate different tasks. They adopted the PSO variant of the evolutionary multi-tasking algorithm for feature selection. However, using the Relief-F method to generate different tasks results in a lack of diversity in feature subsets and low classification accuracy. Thus, Chen et al. [15] introduced randomness when using Relief-F to generate populations of different tasks. Although this approach improves the diversity of feature subsets, it also preserves fewer relevant features. Moreover, both algorithms use knee point selection methods when generating tasks. However, the inflection point selection method will select more features with higher computational complexity. In addition, current multi-task feature selection algorithms typically use fixed and identical knowledge transfer parameters. The same parameters will hinder the rapid transmission of useful information between tasks, making the algorithm inefficient in transferring knowledge between tasks.

This paper proposes an adaptive multifactor evolutionary algorithm (AMFEA) for feature selection in high-dimensional data. In AMFEA, the filtered approach generates different tasks and searches for features with higher classification performance through knowledge transfer between tasks. In addition, AMFEA adopts a local search strategy to help the population eliminate local optima. To facilitate knowledge transfer more effectively, AMFEA has been designed an adaptive parameter matrix. The main contributions of this article are listed as follows:

- A novel multifactor evolutionary algorithm is proposed for feature selection in high-dimensional datasets. This approach generates distinct tasks using filter-based methods. It searches a subset of features that can improve classification performance through knowledge transfer in a search space of different tasks from the dataset.
- An adaptive parameter matrix RMP is designed to control the probability of knowledge transfer between tasks. The matrix can be dynamically adjusted according to population information, effectively controlling negative transfer between tasks and facilitating efficient knowledge transfer during positive transfer.
- A local search strategy is developed to help the population escape local optimal solutions, assisting the algorithm in finding the optimal subset of features in the search space.

The remainder of this article is organized as follows. Section 2 discusses the related work on multi-task optimization and feature selection. Section 3 introduces the details of the proposed AMFEA algorithm. Section 4 presents experimental results and comparisons. Finally, Section 5 concludes this article by providing a summary.

2. Related works

This section introduces evolutionary multitasking and evolutionary feature selection methods, and also describes the multifactor evolutionary algorithm used in this article.

2.1. Evolutionary multitasking

Multi-task optimization learns multiple related tasks through information sharing. This approach can lead to better results than training a single task individually and can help alleviate model overfitting and improve the model's generalization ability. In multi-task optimization, tasks are solved through collaborative search and knowledge transfer. Similarity is required between tasks to prevent negative knowledge transfer [29]. Evolutionary algorithms are often used in multitask optimization due to their ability to represent data flexibly, as multitask Bayesian optimization is limited to continuous search space and cannot be applied to combinatorial search space [30]. Therefore, evolutionary algorithms can optimize multiple tasks in multitasking optimization. The framework capable of performing multi-tasks concurrently, with a total of K tasks, can be represented as follows:

$$\begin{aligned} &\min f_i(X_i), i = 1, 2, \ldots, K \\ &s.t.\ X_i = [x_i^1, x_i^2, \ldots, x_i^{D_i}] \end{aligned} \tag{1}$$

where $f_i(X_i)$ is the objective function of the task i, $X_i = [x_i^1, x_i^2, \ldots, x_i^{D_i}]$ is the solution of the objective function corresponding to task i, and D_i is the dimension of X_i.

Multi-task optimization can efficiently search multiple decision spaces while handling multiple related tasks [15]. There are some common definitions in multitasking optimization containing K tasks, including the following:

Definition 1 (*Factorial Cost*)**.** The factorial cost of individual P_i in task T_k is denoted as Ψ_k^i, and is calculated as $\Psi_k^i = \lambda \cdot \delta_k^i + f_k^i$, where λ is the penalty factor, δ_k^i is the total value of constraint violations, and f_k^i is the objective function value.

Definition 2 (*Factorial Rank*)**.** The rank of individual P_i in task T_k is denoted as r_k^i, obtained from the factorial cost. In the case of the same rankings, a random tie-breaking method is used to solve the problem, where the two individuals with the same rank become k-counterparts.

Definition 3 (*Scalar Fitness*)**.** For all tasks of individual P_i, there is a corresponding factorial rank r_k^i, and the scalar fitness of individual P_i is the best factorial rank between optimized tasks, calculated as $\phi_i = \frac{1}{\min_{k\in\{1,\ldots,K\}}\{r_k^i\}}$.

Definition 4 (*Skill Factor*)**.** The skill factor indicates the task in which an individual P_i performs best among different tasks, denoted as τ_i. It is computed as $\tau_i = \arg\min_{k\in\{1,\ldots,K\}} r_k^i$. Two individuals are considered strong counterparts if their factorial rank and skill factor are equal.

The above four concepts are the basis of multitasking optimization, and different definitions serve different purposes. Factorial cost determines how individuals interact with each other, while factorial rank determines which solutions are chosen to continue the iteration; scalar fitness assigns a task to the individual; skill factor categorizes and ranks the population [31].

2.2. Feature selection

The problem of feature selection is typically defined as follows. For a given dataset, S, with d features, feature selection entails the extraction of a subset of features, F, containing n features, where $n < d$. The subset

F represents the best features selected from S, typically the subset that optimizes a given performance evaluation function [16].

Feature selection is a computationally complex NP-hard problem [32]. Heuristic algorithms are known for their simplicity, flexibility and ability to avoid local optimality. Therefore, heuristic algorithms often solve feature selection problems [33]. The solution of a heuristic algorithm is usually represented as a set of binary encoded vectors [34]. The performance of this vector requires the use of a fitness function to calculate $fitness(\cdot)$, which is represented as follows:

$$\begin{aligned} &\min fitness(X) \\ &s.t.\ X = \{x_1, x_2, \ldots, x_d\} \\ &x_i \in \{0,1\}; \forall i \in \{1,2,\ldots,d\} \end{aligned} \tag{2}$$

where X represents a solution with d features, and $x_i = 1$ indicates that the ith feature is selected, otherwise the ith feature is discarded.

2.3. Multifactor evolutionary algorithm

Multifactor evolutionary algorithm (MFEA) is a genetic algorithm based on multiple factors, which utilizes the similarity between tasks to promote knowledge transfer and solve different tasks simultaneously. The MFEA was first proposed in [35] and has been used in many fields. For example, Ban et al. [36] used the MFEA to solve the traveling salesman problem with a time window. Similarly, Hu et al. [37] applied the MFEA to solve the complex three-dimensional path planning problem.

However, despite many improvements in improving the performance of the MFEA, there are still gaps when applying it to feature selection problems. At present, MFEA and its variants often face the challenges of high computational complexity and local optimality. In addition, existing methods typically use only one crossover operator for knowledge transfer, which may affect its effectiveness [38]. Moreover, in practical applications, the dynamic adjustment method of transfer parameters is often computationally large and time-consuming, and the effect may not be satisfactory [39].

In feature selection problems, the MFEA must deal with many features and tasks, which requires more efficient knowledge transfer and search strategies. However, the current MFEA used in feature selection problems may not fully use the correlation between features and tasks, resulting in locally optimal solutions and possibly missing important features.

In order to make up for these shortcomings, our research proposes an improved algorithm based on MFEA. We introduced different crossover operators and dynamic knowledge transfer probability matrices. The knowledge transfer between tasks needs to be carried out in a unified search space [40]. This ensures efficient knowledge transfer, reduces negative transfer effects, and simplifies the search for task-specific solutions. Since MFEA adopts the basic structure of evolutionary algorithms, the crossover and mutation operators in MFEA are also similar to evolutionary algorithms, but compared to evolutionary algorithms, crossover and mutation in MFEA occur only in specific cases and follow the principle of random or selective mating [41]. In MFEA, the random mating probability (rmp) determines whether crossover and mutation will occur between parents. This search method is conducive to exploring the search space. Since the probability that an individual will perform well on all tasks is low, the MFEA uses the method of evaluating individuals in tasks where they are probably to perform well. Transferring knowledge between tasks could facilitate information exchange. The degree of knowledge transfer between task i and task j can be controlled by the random mating probability ($rmp_{ij} \in [0,1]$). If $rmp_{ij} = 0$, knowledge transfer between tasks is not allowed. The greater the value of rmp_{ij}, the more frequent the knowledge transfer.

3. Proposed method

This section provides a detailed introduction to the adaptive multifactor evolutionary algorithm (AMFEA) proposed in this article for feature selection. Firstly, we introduce the algorithmic process in Section 3.1. Then, we present the fitness function used for evaluating the feature subset. Next, we will briefly introduce the knowledge transfer method used in this article and the local search strategy that helps algorithms improve search performance. Finally, we present the adaptive method for knowledge transfer probability design.

3.1. Overall algorithm

Algorithm 1 is the pseudo-code of AMFEA proposed in this paper. The input is the original data set and related parameters in the algorithm, and the output is the optimal subset of features obtained through iteration. AMFEA randomly initializes a subset of features obtained by different filter-based methods, similar to most evolutionary algorithms. The proposed algorithm uses Relief-F, variance of terms (TV), and Spearman correlation coefficient to calculate the importance of features for initial feature selection. Then, filter out the more important features according to a certain threshold, and each method corresponds to a task. To enhance the population's diversity, individuals in each task population will also randomly select some features when selecting important features for different tasks based on the threshold value. This article constructs four different tasks to complete feature selection. The construction method of Task 1 is based on the original feature set, the remaining three tasks are generated by filtering methods (Relief-F, TV, and Spearman correlation coefficient). Candidate solutions are randomly generated using feature subsets of different tasks, the fitness values are calculated, and the skill factors are assigned (lines 4–5). Then, the proposed multifactor optimization method is used to iterate the population, and a knowledge transfer strategy is employed to enable the population to search for features with good classification performance across different tasks (lines 6–23).

Assign skill factors to the population and evaluate them using the fitness function, then randomly select two individuals (line 7). Suppose the skill factors of two individuals are identical. In such situation, AMFEA use the simulated binary crossover operator with the polynomial mutation operator. If the skill factors differ, knowledge transfer between tasks should be carried out by **Algorithm** 2 (lines 8–14). Then, the newly generated individuals should be evaluated using fitness functions and merged into the population (line 15). To determine the best individual, sort the fitness values of all individuals in ascending order (line 16). The RMP matrix is then updated by **Algorithm** 4 (line 17). Suppose the fitness value of the optimal individual in the current population is not superior to that of the previous generation. In that case, a local search is performed by **Algorithm** 3 (lines 18–20). Finally, the optimal subset of features for the population is selected in ascending order of fitness values (line 22).

3.2. Fitness function

After generating the initial population, it is necessary to evaluate individuals using fitness functions. The fitness function determines the evolutionary direction of the population. A well-designed fitness function can select a higher-quality subset of features. The feature subsets are evaluated using Eq. (3) as the fitness function.

$$fitness = \alpha \cdot ER + (1-\alpha) \cdot \frac{|F|}{|S|} \tag{3}$$

where ER represents the classification error rate, $|F|$ represents the number of features selected in the feature subset, and $|S|$ represents the number of all features in the dataset. α determines the proportion of classification accuracy and feature selection quantity in the fitness

Algorithm 1 Adaptive multifactor evolutionary algorithm (AMFEA) for feature selection

Input: Training dataset D, the number of tasks K, initial parameter matrix RMP, the maximum number of iterations G_{Max} and the population size NP.
Output: The optimal feature subset FS.
Set $G = 0$.
Initialize K populations $(P_1, P_2, ..., P_K)$ of size $\frac{NP}{K}$ for all tasks.
For each individual p_i in the population P_i $(i = 1, 2, ..., K)$, assign skill factor τ_i to p_i and calculate fitness value $fitness_i^G$.
while $G < G_{Max}$ **do**
  Randomly select two individuals p_i and p_j, and their skill factors are τ_i and τ_j respectively.
  if $\tau_i = \tau_j$ **then**
    Perform intra-task crossover for individuals p_i and p_j, and assign skill factors τ_i.
    Perform mutation for individuals p_i and p_j, and assign skill factors τ_i.
  else
    Perform inter-task crossover for individuals p_i and p_j according to **Algorithm 2**.
    Perform mutation for individuals p_i and p_j, and assign skill factors τ_i and τ_j.
  end if
  Calculate fitness values for new individuals and merge them into the population.
  Obtain the fitness value $fitness_{best}^G$ of the best individual p_{best}^G from the population.
  The RMP matrix is updated based on the results of the offsprings knowledge transfer, following **Algorithm 4**.
  if $fitness_{best}^G \geq fitness_{best}^{G-1}$ **then**
    Execute local search according to **Algorithm 3**.
  end if
  $G = G + 1$.
  Rank the populations by fitness values to obtain the optimal subset of features F.
end while

function, which is limited within the range of (0,1). To ensure high classification accuracy of the algorithm, we set $\alpha = 0.999999$ [42].

In feature selection, classification accuracy is prioritized over the number of selected features. Therefore, the coefficient of classification error rate is larger in the design of the fitness function. Furthermore, the dataset imbalance becomes more pronounced as the dataset size increases. This article uses the balanced classification error rate, as show in Eq. (4) to calculate the classification error rate.

$$ER = 1 - \frac{1}{L}\sum_{i=1}^{L} TPR_i \tag{4}$$

where L is the number of categories in the dataset, TPR_i is the proportion of instances correctly recognized by the algorithm in class i.

3.3. Knowledge transfer strategy

The proposed AMFEA is used to optimize individuals in a population generated by different tasks. Knowledge transfer is an essential component of AMFEA. Due to the fact that implicit transfer can achieve knowledge transfer through cross between individuals, AMFEA uses this method [43]. The concrete implementation method is to select two individuals from different tasks from the group randomly, perform cross operation on these two individuals. However, the randomness of this method is very strong. Therefore, to avoid this situation, AMFEA uses a knowledge transfer strategy different from the original MFEA strategy. **Algorithm** 2 gives pseudocode for the knowledge transfer strategy.

Algorithm 2 Knowledge transfer strategy

Input: Randomly select two individuals, p_i and p_j, from the population whose skill factors are τ_i and τ_j, respectively
Output: Offspring individuals O_i and O_j with their skill factors.
Compare the fitness values of p_i and p_j, where the individual with lower fitness value is denoted as B and the other as W.
if $rand < RMP(\tau_i, \tau_j)$ **then**
  Using Eq. (5), inter-task crossover between B and W to obtain offspring O_i, O_j.
  Assign skill factors τ_i, τ_j randomly to offspring O_i, O_j.
else
  Select an individual B_1 with the same τ_i as B , but the individual is different from B.
  Perform intra-task crossover for B_1 and B.
  Assigning a skill factor to the resulting individual O_i is τ_i.
  Select an individual W_1 with the same τ_j as W, but the individual is different from W.
  Perform intra-task crossover for W_1 and W.
  Assigning a skill factor to the resulting individual O_j is τ_j.
end if

Two individuals with different skill factors, p_i and p_j, were randomly selected from the population. Their skill factors are denoted as τ_i and τ_j respectively. Individuals are then compared according to their fitness values. Suppose the fitness value of p_i is higher than that of p_j. In that case, the individuals with the higher fitness value will perform inter-task knowledge transfer to produce offspring according to the following Eq. (5).

$$O_i = \sum_{k\in[1,4]\neq i,j} p_k \times g_k + p_j \times g_j \tag{5}$$

where O_i represents the offspring generated by individuals with higher fitness values through inter-task crossover. p_k represents the best-performing individual in the population generated by task i, and $g_i, i \in [1, 4]$ represents the weight corresponding to task i. The skill factor of the offspring individual O_i obtained from this is a randomly selected skill factor for p_i or p_j. The crossover operation is then executed for individuals p_i and p_j to complete the knowledge transfer process for individuals with a poor fitness value in p_i and p_j. If $rand < RMP(\tau_i, \tau_j)$, non-$p_i$ and non-p_j individuals are selected from the population generated by the same task for each individual. The intra-task crossover operation is performed, and the skill factor is assigned to the resulting progeny individuals.

3.4. Local search strategy

This algorithm uses a local search strategy to enhance its search capability. Due to the evolutionary algorithm's strong randomness, this paper considers using a more deterministic method to calculate the importance of features to help individuals in the population select more essential features. However, all the subsets are generated by the filtering method. In that case, the population lacks diversity, so some features can be added or deleted to change the search direction. The addition and deletion of features are considered according to the importance of features, which can combine the certainty of the filter methods and ensure the diversity of evolutionary algorithms. Relief-F is suitable for a variety of data sets in traditional filtering methods, especially when the relationship between feature and target variables is complex. Therefore, the local search method designed in AMFEA uses the Relief-F to help the algorithm select more important features. **Algorithm** 3 gives pseudocode for the local search.

The importance of the features is first evaluated using Relief-F. The most crucial unselected feature is identified and incorporated into the

Algorithm 3 Local search strategy

Input: Currently best subset of features F.
Output: New subset of features F^*.
Evaluate feature importance using the Relief-F approach.
for each feature in F **do**
  Add the most significant of the unselected features in F.
  Remove the least important of the selected features in F.
  if fitness score of the new individual F^* is better than that of F **then**
    Output the new subset of features F^*.
  else
    if less than 10 repetitions **then**
      Repeat feature additions and deletions 10 times.
    else
      Output feature subset F^*.
    end if
  end if
end for

optimal feature subset. Remove the feature with the lowest importance ranking in current set. Repeat the addition and removal of features until the new individual fitness value obtained is better than the original subset of best features. Assuming that after ten repetitions, the fitness value of the new individual still does not improve, the original feature subset will be output.

3.5. Adaptive RMP matrix of knowledge transfer probability

In many multitasking algorithms, the probability of knowledge transfer is the same and fixed. However, the similarity between different tasks is different, and different parameters should be designed. At the beginning of knowledge transfer, knowledge transfer between similar tasks can obtain more useful information. However, as knowledge transfer occurs continuously, effective transfer decreases, so we should make the knowledge transfer parameters for this task smaller. A calculation method of knowledge transfer probability parameter matrix is designed in [39]. However, if this method is used in feature selection problems, it will be found that it will consume many computing resources and take too long time. Therefore, this paper designs knowledge transfer probability as a symmetric matrix that can change adaptively with population evolution. In order to enable better knowledge transfer between different tasks, the knowledge transfer probability matrix used by AMFEA is expressed as follows:

$$RMP = \begin{bmatrix} rmp_{11} & rmp_{12} & \cdots & rmp_{1K} \\ rmp_{21} & rmp_{22} & \cdots & rmp_{2K} \\ \vdots & \vdots & & \vdots \\ rmp_{K1} & rmp_{K2} & \cdots & rmp_{KK} \end{bmatrix} \tag{6}$$

The matrix RMP is a $K \times K$ symmetric matrix since the knowledge transfer probability between task i and task j is equal to the knowledge transfer probability between task j and task i ($rmp_{ij} = rmp_{ji}$). Additionally, the knowledge transfer probability of the same task equals 1 ($rmp_{ii} = 1, i = 1, 2, \ldots, K$). In order to achieve efficient and fast knowledge transfer between tasks, AMFEA adjusts the parameters in the RMP matrix adaptively by comparing individual fitness values. If knowledge transfer between tasks results in offspring with lower fitness than the parent. In this case, this parameter should be reduced. On the contrary, it increases the parameter between these two tasks. **Algorithm** 4 outlines the proposed adaptive method for the RMP matrix.

The fitness values of the offspring are compared to those of the parents and recorded. If the number of times that the offspring outperforms the parent is greater than the number of times that the parent

Algorithm 4 Adaptive RMP matrix of knowledge transfer probability

Input: The offspring individuals $O_1, O_2, ..., O_m$ obtained after the knowledge transfer of tasks i, j.
Output: $RMP = (rmp_{ij})_{K \times K}$.
Calculate the fitness values of $O_1, O_2, ..., O_m$ and compare their fitness values with their parent individuals.
Record the number of times t_1,t_2 that these offspring outperform and underperform their parents
if $t_1 > t_2$ **then**
  if $rmp_{ij} < 0.7$ **then**
    $rmp_{ij} = rmp_{ji} \times 1.1$.
  else
    $rmp_{ij} = rmp_{ji} \times 1.01$.
  end if
else
  if $rmp_{ij} > 0.3$ **then**
    $rmp_{ij} = rmp_{ji} \times 0.9$.
  else
    $rmp_{ij} = rmp_{ji} \times 0.99$.
  end if
end if

outperforms the offspring, the probability of knowledge transfer between task i and task j increases; otherwise, it will decrease. However, when rmp_{ij} exceeds the recommended range of [0.3, 0.7], the increase or decrease of rmp_{ij} will be limited in order to avoid a high or low probability of knowledge transfer. To better control the increase or decrease of knowledge transfer probability, reducing the degree of change is recommended when rmp_{ij} exceeds the suggested range of values. This is because if the degree of change is not adjusted, the knowledge transfer probability may exceed 1.

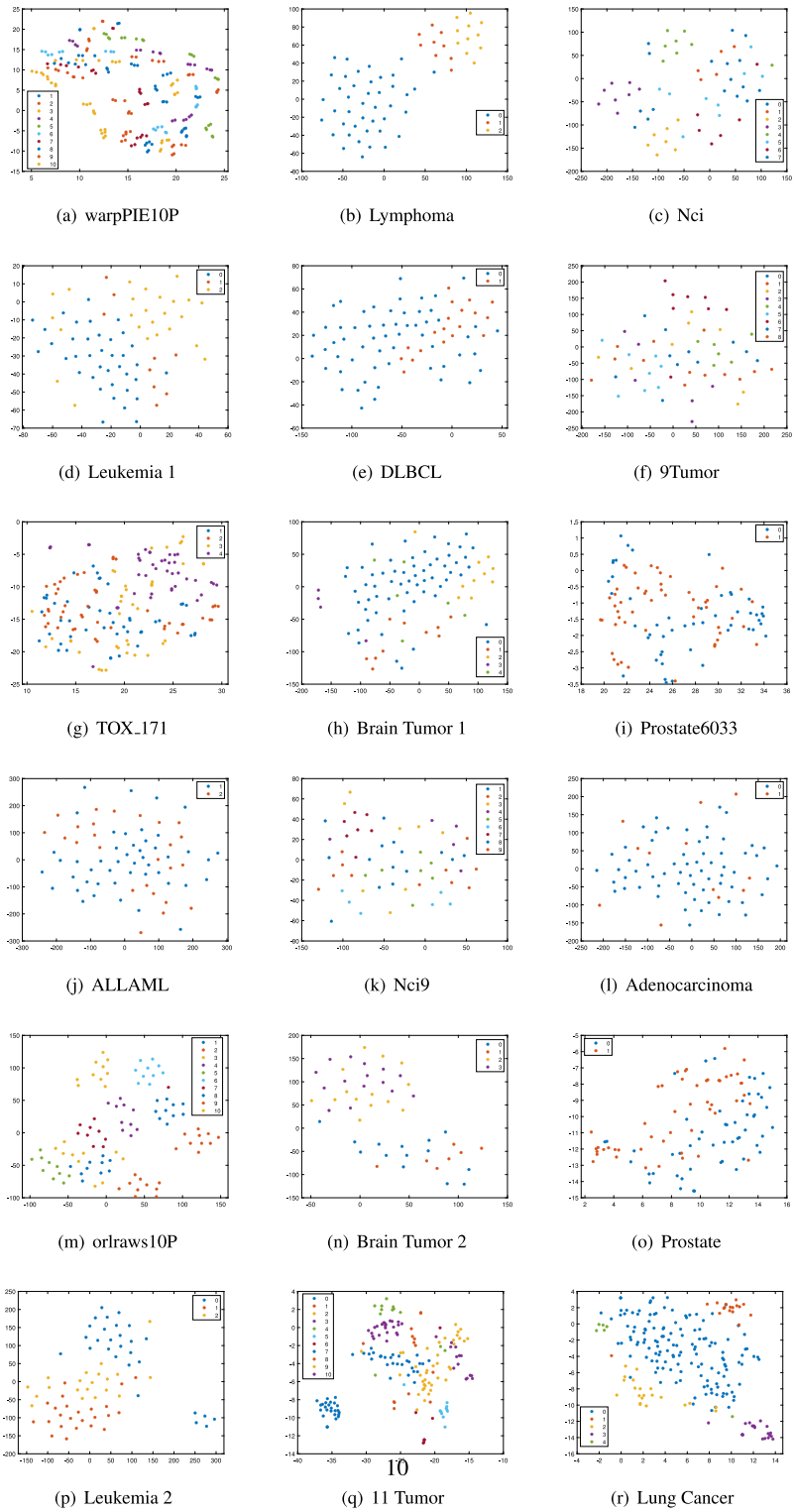
3.6. Computational complexity analysis

The computational complexity of AMFEA was analyzed according to the method used in [16]. As shown in **Algorithm** 1, AMFEA consists of three main parts: multi-task population generation, multi-task optimization, and result output. In the process of multi-task population generation, we use the original data set to randomly generate the initial population of one task and then generate the initial population of K tasks by K different filtering methods. The training set of K task generated by the filtering methods contains D features, and the required time complexity is $O(K \times D)$. The time complexity required for population initialization is $O(K \times N_i)$, where N_i is the size of the population for the ith task. The total time complexity required to perform the population initialization process is $O(K \times N)$, where N is the total population size. There are K tasks in algorithm, so the time complexity to generate these populations is $K \times O(D \times N_i)$. Therefore, we propose that the worst time complexity of AMFEA is $Max\{O(K \times D), O(K \times N), O(D \times N_i)\}$.

4. Experiments and analysis

4.1. Datasets

To evaluate the performance of the proposed AMFEA, it undergoes testing on 18 publicly available datasets, each containing thousands of features. The datasets can be accessed through the following link: https://github.com/gelin123/MF_CSO_Materials. The properties of these datasets are shown in Table 1. All experiments were performed on a computer with an AMD Ryzen 5 3.7 GHz CPU and 16 GB RAM using MATLAB R2023b running on Windows 11. Thirty independent runs were performed on each dataset.



(a) warpPIE10P (b) Lymphoma (c) Nci

(d) Leukemia 1 (e) DLBCL (f) 9Tumor

(g) TOX_171 (h) Brain Tumor 1 (i) Prostate6033

(j) ALLAML (k) Nci9 (l) Adenocarcinoma

(m) orlraws10P (n) Brain Tumor 2 (o) Prostate

(p) Leukemia 2 (q) 11 Tumor (r) Lung Cancer

Fig. 1. Visualization of eight datasets.

Since the data set used in the experiment contained more features than samples. In order to visually observe the distribution of the data set, we used a nonlinear dimensionality reduction algorithm, namely the T-distribution random neighborhood embedding algorithm [44]. Fig. 1 shows the distribution of each data set. As can be seen from Fig. 1, the distribution of most data sets is chaotic, and some data sets are unbalanced, which makes the task of classification and feature selection more difficult. For example, the distribution of data sets TOX_171, BrainTumor1, and Nci9 shows multiple dense clusters, and the boundaries between clusters are not obvious, which may make it difficult for traditional feature selection methods to select feature subsets with high classification performance.

Table 1
Properties of adopted datasets.

Dataset	Features	Instances	Classes	Fields
warpPIE10P	2420	210	10	Face Image
Lymphoma	5026	62	3	Biology
Nci	5244	61	8	Biology
Leukemia 1	5327	72	3	Biology
DLBCL	5469	77	2	Biology
9Tumor	5726	60	9	Biology
TOX_171	5748	171	4	Biology
Brain Tumor 1	5920	90	5	Biology
Prostate6033	6033	102	2	Biology
ALLAML	7129	72	2	Biology
Nci9	9712	60	9	Biology
Adenocarcinoma	9868	76	2	Biology
orlraws10P	10 304	100	10	Face Image
Brain Tumor 2	10 367	50	4	Biology
Prostate	10 509	102	2	Biology
Leukemia 2	11 225	72	3	Biology
11 Tumor	12 533	174	11	Biology
Lung Cancer	12 600	203	5	Biology

4.2. Comparative methods and parameter settings

To evaluate the AMFEA, several feature selection methods were compared.

(1) PS-NSGA is a multi-objective genetic algorithm designed for feature selection [45]. It employs various techniques to improve efficiency and convergence, including precision first domination operators, fast bit variation, mutation retry, and combination operators. In addition, it employs a solution selection strategy to optimize a subset of features.

(2) SM-MOEA is a feature selection algorithm based on a guidance matrix. It uses guidance matrices to guide the evolution of populations in multi-objective feature selection algorithms for high-dimensional data [46]. It combines the reduction and individual repair operators, the strategy initialization, updating of the matrix to improve feature subsets' search efficiency and classification performance.

(3) PSO-EMT is a multi-task feature selection method based on PSO [28]. It adopts a novel crossover operator, assortative mating, with a variable-range strategy.

(4) MTPSO is a multi-task PSO feature selection algorithm for high-dimensional data. It transforms problems into related low-dimensional tasks through task-generation strategies and knowledge transfer mechanisms [15].

(5) MF-CSO is a high-dimensional feature selection algorithm based on a multitasking algorithm [16]. It uses filtered methods to generate multiple related task populations and selects features through a competitive group optimizer.

(6) MOEA/D-FS is an algorithm that uses a decomposition-based multi-objective evolutionary algorithm (MOEA/D) for feature selection [47].

(7) CSO-FS is a competitive swarm optimizer for feature selection [48].

To guarantee accuracy and impartiality, the same parameters were used as those in [16]. Due to its fewer parameters than other classifiers, KNN classifiers are widely used to the feature selection problems. To avoid the influence of noise on the KNN classifier, we choose the KNN classifier with $k = 1$ as all feature selection methods for comparison. The maximum number of iterations is 70, and the population size is 300. Table 2 shows the parameter settings for each comparison algorithm. In the AMFEA algorithm, μ_S represents the threshold value of Spearman's method, μ_T represents the threshold value of the TV method, and μ_R is the percentage of selected features in the total set sorted by importance using Relief-F. To reduce the effects of randomness, we independently performed the proposed algorithm 30 times on all datasets. To prevent feature selection bias, we used 10-fold cross-validation.

4.3. Performance comparison and discussion

This section mainly introduces the experimental results between AMFEA and other feature selection algorithms.

4.3.1. Comparisons with EA-based feature selection methods

Table 3 shows each algorithm's average classification accuracy and standard deviation after 30 runs. Table 4 shows each algorithm's average number of feature selections in these 30 runs. The results of the experiment are also subjected to the Friedman test, and the ranking from the test is shown in the last row of the table.

• AMFEA Versus FULL: FULL is a direct classification method without feature selection, which is used to illustrate the importance of feature selection on data sets. In Table 3, the classification accuracy of FULL on the 9Tumor and Nci9 data sets is 36.67% and 41.33%, respectively, which does not reach the classification accuracy of 50%. On these two datasets, the average classification accuracy of AMFEA is 96.05% and 95.00%, respectively, which increases by 59.38% and 53.67% compared to FULL. Although FULL achieves a good classification accuracy rate on Lymphoma and Leukemia 2 datasets, the classification accuracy rate of AMFEA reaches 100.00% and 99.89% respectively. According to the Friedman test ranking of FULL's in Table 3, it can be seen that it is not good to classify data sets directly. Redundant features will affect the classification of data sets, so it is necessary to select features in data sets.

• AMFEA Versus SM-MOEA: AMFEA has better classification accuracy than SM-MOEA, and identifying fewer features. In Tables 3 and 4, although SM-MOEA selects fewer than 10 features in the Brain Tumor 1 and Brain Tumor 2 data sets, the classification accuracy of AMFEA is better than that of SM-MOEA. Moreover, the classification accuracy on these two data sets is improved by 20.51% and 41.64%, respectively. In addition, AMFEA not only selects fewer features than SM-MOEA in the Adenocarcinoma data set but also improves the classification accuracy by 20.46% compared with SM-MOEA. According to the ranking of Friedman test in Tables 3 and 4, although AMFEA has slightly higher features selection than SM-MEA algorithm, it can achieve better classification accuracy than SM-MEA algorithm.

• AMFEA Versus PS-NSGA: In Tables 3 and 4, the classification accuracy of PS-NSGA on data sets warpPIE10P, Lymphoma, TOX_171, and orlraws10P is higher than 90%. However, AMFEA algorithm on these data sets is 99.86%, 100%, 99.93% and 100%, respectively. AMFEA not only greatly improves the original data sets with poor classification accuracy but also significantly improves the data sets with good classification effects. In addition, according to the ranking of the Friedman test in Table 4, AMFEA selects fewer features than PS-NSGA while ensuring a higher classification accuracy.

• AMFEA Versus CSO-FS: From Table 3, it can be seen that among the 18 data sets, the classification accuracy of CSO-FS exceeds 90% only in the orlraws10P data set, while the accuracy of Nci9 in CSO-FS is only 43.19% at the lowest. However, AMFEA has a classification accuracy of 100% and 95% in these two data sets. In Table 4, although CSO-FS selected 4.6 features on the dataset warpPIE10P, its classification accuracy is only 48.67%, while AMFEA is 99.86%.

• AMFEA Versus MOEA/D-FS: This paper compares a classic multi-objective algorithm, MOEA/D, with a multi-task algorithm in a feature selection problem. According to Table 3, MOEA/D-FS has the highest classification accuracy of 99.00% for the dataset warpPIE10P. However, the classification accuracy of AMFEA on this dataset is 99.86%. MOEA/D-FS has the lowest classification accuracy of 43.83% on dataset 9Tumor, and AMFEA achieves a classification accuracy of 96.05% on this dataset. Table 4 shows that MOEA/D-FS is in the last order except FULL. Even on the warpPIE10P (dataset with the least feature selection), MOEA/D-FS selects an average of 1070.2 features. However, the average number selected by AMFEA is only 28.94, far lower than MOEA/D-FS.

Table 2
Parameter settings.

Algorithms	Parameters
SM-MOEA	Attenuation factor $\gamma = 0.1$
PS-NSGA	Mutation probability equals to 0.1, Mutation retry number equals to 1
PSO-EMT	$c_1 = c_2 = c_3 = 1.49445,\ \rho = 0.05,\ rmp = 0.6,\ m = 10,\ w = 0.9 - 0.5 \times \frac{iter}{max_{iter}}$
MTPSO	$c_1 = c_2 = c_3 = 1.49445,\ \rho = 0.05,\ rmp = 0.6,\ G = 10,\ w = 0.9 - 0.5 \times \frac{iter}{max_{iter}}$
CSO-FS	$r_1, r_2, r_3 \in \{0, 1\}$
MOEA/D-FS	$T = 0.8,\ \rho = 0.5$
MF-CSO	$r_1, r_2, r_3 \in \{0, 1\},\ g_1 = 0.1,\ g_2 = g_3 = g_4 = 0.45,\ p_{trans} = 0.5$
AMFEA	$g_1 = 0.1,\ g_2 = g_3 = g_4 = 0.45,\ \mu_S = 0.5,\ \mu_T = 0.05,\ \mu_R = 20\%,\ RMP^0 = \begin{bmatrix} 0.5 & \cdots & 0.5 \\ \vdots & & \vdots \\ 0.5 & \cdots & 0.5 \end{bmatrix}_{4\times4}$

Table 3
Average results of accuracy (%) obtained by 9 algorithms on the datasets.

Datasets	FULL	SM-MOEA	PS-NSGA	CSO-FS	MOEA/D-FS	PSO-EMT	MTPSO	MFCSO	AMFEA
warpPIE10P	84.51	98.52	98.57	48.67	99.00	99.17	99.16	99.18	**99.86(±0.25)**
Lymphoma	99.08	76.41	92.01	56.38	97.80	93.33	96.67	98.13	**100.00(±0.00)**
Nci	68.26	65.49	67.17	64.29	69.58	59.68	63.99	71.23	**98.90(±2.03)**
Leukemia 1	79.72	77.88	87.11	82.50	82.69	86.11	86.94	92.50	**99.71(±0.19)**
DLBCL	83.00	80.37	84.05	84.50	84.17	84.17	88.33	92.50	**99.94(±0.20)**
9Tumor	36.67	47.36	54.15	43.94	43.83	55.59	52.57	44.77	**96.05(±4.28)**
TOX_171	77.89	83.89	91.67	85.79	89.37	92.00	91.50	92.38	**99.93(±0.10)**
Brain Tumor 1	72.08	74.35	71.24	73.30	75.16	77.67	78.03	75.67	**94.86(±5.28)**
Prostate6033	81.31	81.72	83.97	84.42	80.33	80.50	84.00	87.18	**97.72(±1.32)**
ALLAML	77.46	81.17	86.89	82.78	83.00	90.90	91.08	96.24	**98.98(±1.51)**
Nci9	41.33	61.58	56.10	43.19	45.42	53.05	51.82	54.00	**95.00(±5.07)**
Adenocarcinoma	62.74	69.60	63.94	64.01	59.52	65.33	66.02	65.56	**90.06(±10.06)**
orlraws10P	77.82	93.72	93.30	92.33	95.00	96.60	97.00	96.30	**100.00(±0.00)**
Brain Tumor 2	62.50	57.21	66.33	72.40	64.00	66.67	73.33	74.58	**98.85(±1.30)**
Prostate	85.33	82.81	87.85	88.97	83.28	81.76	85.17	87.33	**97.52(±2.60)**
Leukemia 2	89.44	83.14	89.06	85.85	87.60	90.56	90.00	98.33	**99.89(±0.33)**
11 Tumor	71.42	70.29	77.81	72.39	74.38	78.54	80.00	79.69	**96.61(±3.68)**
Lung Cancer	78.05	87.06	86.50	78.45	78.83	84.60	84.28	87.56	**98.16(±2.03)**
Avg.Rank	7.39	6.67	5.28	6.56	6.25	4.86	4.11	2.89	**1.00**

Table 4
Average numbers of selected features (size) obtained by 9 algorithms.

Datasets	FULL	SM-MOEA	PS-NSGA	CSO-FS	MOEA/D-FS	PSO-EMT	MTPSO	MFCSO	AMFEA
warpPIE10P	2420	18	162.5	**4.6**	1070.2	115.47	227.76	103.93	28.94
Lymphoma	5026	71	**2.1**	2.61	1896.7	12	11.99	53.48	36.34
Nci	5244	**15.06**	125.23	403.17	2495.8	264.74	1467.47	273.46	110.34
Leukemia 1	5327	**9.16**	16.8	415.93	2535.4	242.5	910.54	229.6	46.62
DLBCL	5469	12.8	**11.2**	343.77	2629.6	154.4	1235.27	96.5	17.61
9Tumor	5726	**14.53**	192.2	339.15	2753.9	309.3	574.293	199.2	109.13
TOX_171	5748	**18.21**	94.4	585.43	2744.5	2867.8	879.863	1170.7	114.90
Brain Tumor 1	5920	**6.4**	61.11	139.28	2823	211.5	984.59	194.5	25.68
Prostate6033	6033	**9.89**	46.52	710.92	2857.3	342.34	1625.71	212.13	22.74
ALLAML	7129	**5.33**	17.9	401.87	3414.5	182.8	1955.12	297.6	26.10
Nci9	9712	**26.3**	167.7	560.133	4669.1	1333.7	689.98	2354.4	141.20
Adenocarcinoma	9868	11.49	55.95	533.773	4660.5	2084.56	297.77	523.31	**10.13**
orlraws10P	10 304	28.9	**27.93**	57.7	4899.4	807.88	1399.04	322.51	68.70
Brain Tumor 2	10 367	**9.27**	74.66	139.28	4987.8	211.5	1909.09	194.5	50.59
Prostate	10 509	**13.5**	63.2	1203.74	5065.88	132.1	2880.71	63.2	18.79
Leukemia 2	11 225	**8.56**	28.06	1109.66	5420.7	244.3	1605.6	373.4	60.07
11 Tumor	12 533	**46.44**	334.72	618.45	6096.3	884.3	2143.44	334.72	195.18
Lung Cancer	12 600	**20.69**	106.3	1163.09	6027.6	609.63	723.4	498.1	43.31
Avg.Rank	9.00	**1.56**	2.67	4.89	7.94	5.33	6.33	4.72	2.56

• AMFEA Versus PSO-EMT: Both PSO-EMT and AMFEA use evolutionary multitasking algorithms. According to the Friedman test ranking in Table 3, it can be seen that PSO-EMT not only has a higher ranking than the other four algorithms that do not use multitasking optimization, but also has a higher classification accuracy than those that directly classify. The results in Table 3 show that the classification accuracy of PSO-EMT in six data sets (warpPIE10P, Lymphoma, TOX_171, ALLAML, orlraws10P, Leukemia 2) is above 90%. However, in those six datasets, the classification accuracy of AMFEA exceeds 95%, and AMFEA only selects more features than PSO-EMT in the Lymphoma dataset. In comparison, for the other 5 data sets AMFEA selects fewer features than PSO-EMT.

• AMFEA Versus MTPSO: MTPSO is also an algorithm that uses a multi-task optimization framework. According to the Friedman test ranking in the Table 3, MTPSO performs better than single-task optimization on most datasets. However, AMFEA algorithm can obtain higher classification accuracy than the MTPSO algorithm on all data sets with less feature selection. Among the 18 data sets, the Nci data set has the largest increase in average classification accuracy, which increases by about 34.91%.

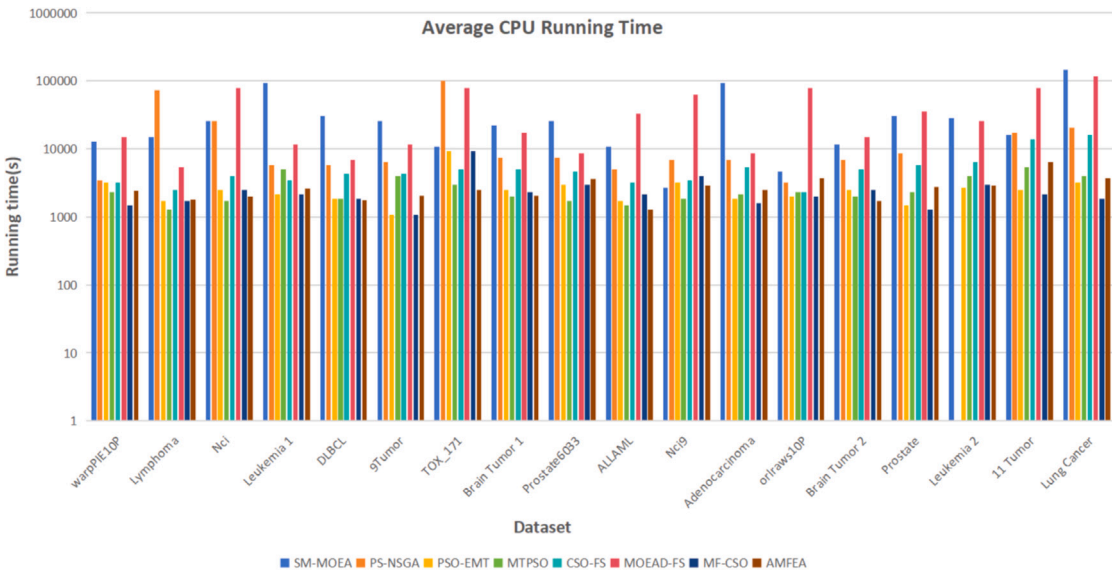


Fig. 2. Comparison of average running time between AMFEA and other feature selection algorithms.

• AMFEA Versus MFCSO: Compared with MFCSO, the multi-task feature selection algorithm, AMFEA has significantly improved in classification accuracy on all data sets. AMFEA selects fewer features than MFCSO because it uses a threshold value, not knee point selection for task generation. On dataset 9Tumor, MFCSO has the lowest classification accuracy, only 44.77%, but compared to MFCSO, AMFEA improves by 51.288%. Even on warpPIE10P, MFCSO's best-performing dataset, AMFEA improves by 0.68% over MFCSO.

Training time is also an essential criterion for evaluating algorithm efficiency. Therefore, Fig. 2 shows the average CPU runtime of AMFEA and other evolutionary algorithm-based feature selection algorithms over 30 runs. AMFEA has the shortest running time on five datasets (DLBCL, TOX_171, Brain Tumor 1, ALLAML, Brain Tumor 2). On most datasets, multi-task optimization algorithms (SM-MOEA, PS-NSGA, MOEA/D-FS, MF-CSO, and AMFEA) consume significantly less running time than traditional evolutionary algorithms. Multi-task optimization frameworks can effectively utilize the advantages of different tasks by combining important features selected from different tasks through knowledge transfer, reducing the time required for algorithms to calculate individual fitness values.

4.3.2. Comparisons with non-EA feature selection methods

To compare the AMFEA algorithm with feature selection algorithms that do not use evolutionary algorithms, we select three commonly used filtering feature selection algorithms (Relief-F, Spearman's correlation coefficient, and TV). In the experiment, in order to achieve better classification results using the three traditional filtered feature selection algorithms, we use the knee selection method in [16] to determine the selected features. We conduct experiments on 18 datasets with our algorithm. Table 5 presents the specific experimental results. AvgNF represents the average number of selected features, and AvgAcc represents the average classification accuracy.

Table 5 shows that AMFEA is more accurate than the three filtered methods on 17 data sets, except for warpPIE10P. In addition, the classification accuracy of AMFEA on Lymphoma and orlraws10P data sets reaches 100%, and the classification accuracy of AMFEA on Nci9, 9Tumor and Adenocarcinoma data sets with poor classification accuracy of the three filtering feature selection methods is also significantly improved. AMFEA also achieves good classification performance on the data set with a more complex distribution in Fig. 1. This is because AMFEA uses different filtering feature selection methods in task generation, iteratively combining the selected features with higher classification accuracy. Thus, AMFEA can combine the advantages of each filtered method to achieve better classification performance.

Table 5 shows that AMFEA selects fewer features than the filtered method. The traditional filtered method selects thousands of features, while AMFEA only selects dozens. Therefore, AMFEA has significant advantages in reducing the number of selected features and can combine important features from different filtered methods.

4.4. Analysis on the effect of three strategies on AMFEA

AMFEA has three main improvement components: (1) knowledge transfer strategy, (2) adaptive RMP matrix, and (3) local search strategy. To facilitate a comparative analysis of the performance of these three components, three distinct algorithms are developed, specifically, the AMFEA/KT algorithm employing the original knowledge transfer strategy of the MFEA algorithm, the AMFEA/AR algorithm featuring a fixed knowledge transfer parameter, and the AMFEA/LS algorithm excluding the use of local search strategy.

In order to compare the convergence performance between three variants and AMFEA, this section presents the changes in fitness values of four algorithms (AMFEA, AMFEA/AR, AMFEA/LS, and AMFEA/KT) on six complex distributed datasets (warpPIE10P, Nci, Leukemia 1, Brain Tumor 1, Adenocarcinoma, and 11Tumors) during the iteration process. The convergence curves of AMFEA and its variants at the highest classification accuracy are shown in Fig. 3, where the horizontal axis represents the number of iterations and the vertical axis represents the fitness function value. In the figure, AMFEA converges faster than the other three variants and tends to achieve lower fitness values on the first iteration. This shows that AMFEA can effectively obtain lower fitness values with fewer iterations. Among the other three variants, AMFEA/AR fails to achieve the same convergence effect as AMFEA after several iterations. And the parameter adaptive method can also effectively help the algorithm converge.

Moreover, it can be seen from the line chart that AMFEA/LS is the variant with a poor convergence effect among the three variants. And the final convergence of AMFEA/KT is the closest to AMFEA compared to the other three variants. However, AMFEA/KT can only reach the convergence degree of AMFEA after several iterations.

4.5. Comparisons with different classifiers

The commonly used classifiers in feature selection are KNN and SVM. In order to explore the performance of AMFEA under other

Table 5
Comparison between AMFEA and non-EA feature selection methods.

Datasets	Methods	AvgNF	AvgAcc	Datasets	Methods	AvgNF	AvgAcc
warpPIE10P	Relief-F	401.00	99.50	ALLAML	Relief-F	519.00	75.25
	Spearman	2189.70	73.83		Spearman	4695.30	98.33
	TV	2242.00	**100.00**		TV	611.00	70.83
	AMFEA	**28.94**	99.86		AMFEA	**26.10**	**98.98**
Lymphoma	Relief-F	343.00	99.08	Nci9	Relief-F	9709	50.00
	Spearman	2881.70	99.17		Spearman	8630.30	26.50
	TV	521.00	99.52		TV	7593.00	35.00
	AMFEA	**236.34**	**100.00**		AMFEA	**141.2**	**95.00**
Nci	Relief-F	548.00	83.57	Adenocarcinoma	Relief-F	3789.00	62.50
	Spearman	4069.00	66.81		Spearman	8539.20	47.98
	TV	397.00	75.23		TV	657.00	59.88
	AMFEA	**110.34**	**98.90**		AMFEA	**10.13**	**90.06**
Leukemia 1	Relief-F	436.00	93.67	orlraws10P	Relief-F	1013.00	**100.00**
	Spearman	1253.30	92.22		Spearman	507.60	56.00
	TV	421.00	88.19		TV	1161.00	92.00
	AMFEA	**37.75**	**99.71**		AMFEA	**68.7**	**100.00**
DLBCL	Relief-F	347.00	90.83	Brain Tumor 2	Relief-F	815.00	71.66
	Spearman	4252.30	93.83		Spearman	3567.00	77.92
	TV	458.00	88.33		TV	989.00	72.50
	AMFEA	**17.61**	**99.94**		AMFEA	**50.59**	**98.85**
9Tumor	Relief-F	403.00	63.33	Prostate	Relief-F	686.00	90.00
	Spearman	4766.50	30.45		Spearman	3836.30	89.50
	TV	380.00	55.00		TV	1242.00	80.50
	AMFEA	**109.13**	**96.05**		AMFEA	**18.19**	**97.52**
TOX_171	Relief-F	5474.00	85.25	Leukemia 2	Relief-F	1167.00	96.11
	Spearman	4153.90	84.13		Spearman	564.30	88.89
	TV	319.00	73.00		TV	734.00	92.78
	AMFEA	**114.9**	**99.93**		AMFEA	**60.07**	**99.89**
Brain Tumor 1	Relief-F	519.00	75.25	11 Tumor	Relief-F	829.00	82.13
	Spearman	2653.70	76.78		Spearman	7735.20	66.01
	TV	611.00	70.83		TV	697.00	78.64
	AMFEA	**25.68**	**94.86**		AMFEA	**144.77**	**96.61**
Prostate6033	Relief-F	518.00	90.00	Lung Cancer	Relief-F	881.00	83.50
	Spearman	5425.00	89.17		Spearman	6921.50	85.67
	TV	657.00	82.17		TV	563	85.47
	AMFEA	**22.74**	**97.72**		AMFEA	**43.31**	**98.16**

Table 6
Comparison between different classifiers.

Datasets	Methods	AvgNF	AvgAcc	Datasets	Methods	AvgNF	AvgAcc
warpPIE10P	AMFEA-SVM	40.14	98.79	ALLAML	AMFEA-SVM	38.37	99.38
	AMFEA	**28.94**	**99.86**		AMFEA	**26.10**	**99.65**
Lymphoma	AMFEA-SVM	69.82	99.96	Nci9	AMFEA-SVM	198.67	82.07
	AMFEA	**236.34**	**100.00**		AMFEA	**141.2**	**95.00**
Nci	AMFEA-SVM	155.06	90.16	Adenocarcinoma	AMFEA-SVM	49.89	74.47
	AMFEA	**110.34**	**99.08**		AMFEA	**10.13**	**90.06**
Leukemia 1	AMFEA-SVM	59.45	98.96	orlraws10P	AMFEA-SVM	370.10	99.93
	AMFEA	**37.75**	**99.94**		AMFEA	**68.7**	**100.00**
DLBCL	AMFEA-SVM	31.34	98.83	Brain Tumor 2	AMFEA-SVM	107.00	91.00
	AMFEA	**17.61**	**99.86**		AMFEA	**50.59**	**98.85**
9Tumor	AMFEA-SVM	215.13	79.79	Prostate	AMFEA-SVM	53.29	95.65
	AMFEA	**109.13**	**96.05**		AMFEA	**18.19**	**97.53**
TOX_171	AMFEA-SVM	278.46	95.10	Leukemia 2	AMFEA-SVM	84.51	98.15
	AMFEA	**114.9**	**99.93**		AMFEA	**60.07**	**99.89**
Brain Tumor 1	AMFEA-SVM	76.57	80.21	11 Tumor	AMFEA-SVM	235.00	91.06
	AMFEA	**25.68**	**94.86**		AMFEA	**144.77**	**97.79**
Prostate6033	AMFEA-SVM	62.57	94.54	Lung Cancer	AMFEA-SVM	102.60	94.74
	AMFEA	**22.74**	**97.72**		AMFEA	**43.31**	**98.16**

classifiers, the experimental results of AMFEA using the original KNN classifier and SVM as a classifier are compared. Table 6 shows the specific experimental results. AMFEA indicates that KNN is used as the classifier, and AMFEA-SVM indicates that SVM is used as the classifier. Table 6 shows that compared to KNN, the classification accuracy of AMFEA using SVM is significantly reduced. However, Tables 3, 4 and 6 indicate that regardless of which classifier the AMFEA algorithm uses, for the original dataset with lower classification accuracy, such as 9Tumor, AMFEA has been significantly improved. And SVM as a classifier has more features selected than KNN as a classifier.

The poor classification effect of SVM in high-dimensional data sets is because SVM usually finds an optimal hyperplane to separate different categories of data. However, the distance between high-dimensional data is more dispersed, which makes it difficult to determine the

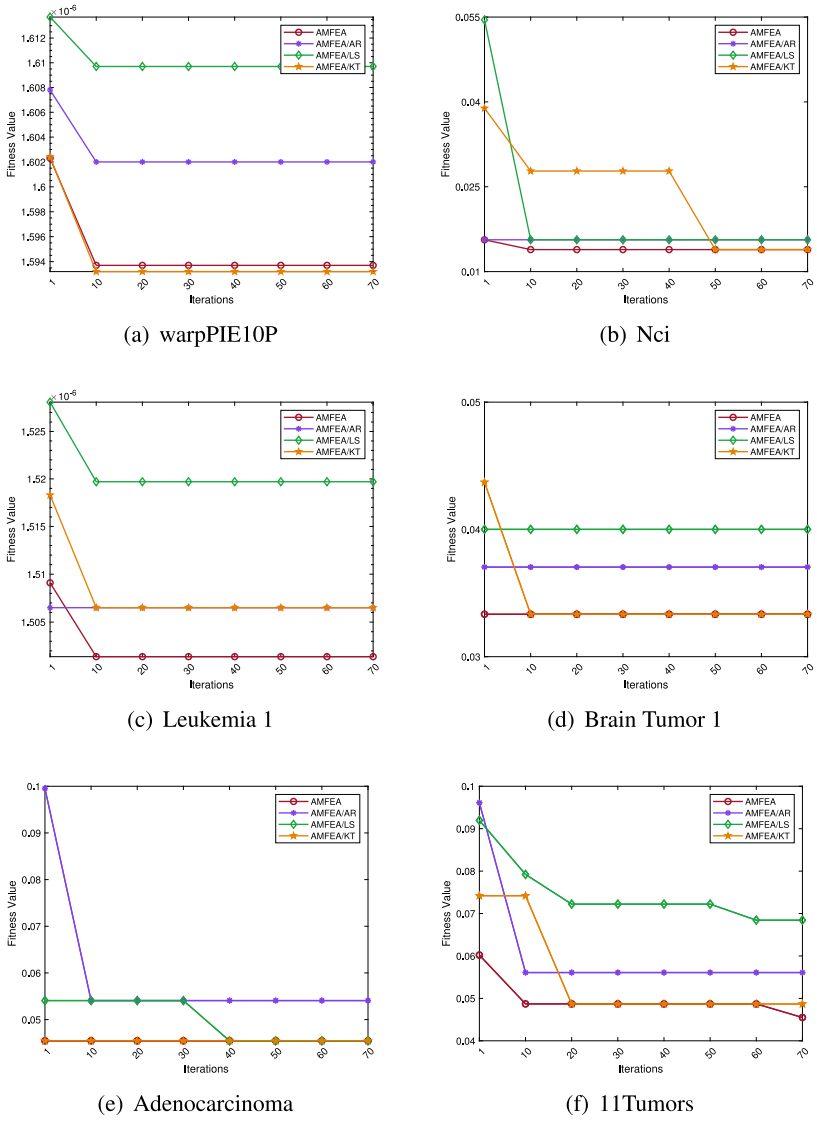


Fig. 3. Comparison of convergence of the algorithms on (a) warpPIE10P, (b) Nci, (c) Leukemia 1, (d) Brain Tumor 1, (e) Adenocarcinoma, and (f) 11 Tumors datasets.

hyperplane. Moreover, there are often more noise features, which will lead to the degradation of SVM classification performance. Compared with KNN, SVM has a longer training and prediction time. Therefore, the KNN classifier is widely used in feature selection because of its few parameters and good adaptability to data distribution.

5. Conclusion

With the continuous expansion of the data set, feature selection can effectively remove some irrelevant or redundant features, thereby reducing the complexity of the subsequent model, as well as the time and computing resource consumption of model training. This paper introduces a feature selection algorithm called AMFEA for the classification of high dimensional datasets. The evolutionary multitasking optimization method used in AMFEA could improve classification accuracy by selecting optimal subset of features, and performs better than other evolutionary feature selection algorithms. AMFEA uses filtered methods to generate tasks, utilizes multi-task optimization to select features with high classification performance, and adopts a local search strategy to search for feature subsets. The design of adaptive parameters promotes the effective transfer between tasks. We conducted experiments on 18 datasets and the results validated the effectiveness of AMFEA. The experiment verifies the advantages of AMFEA combined with traditional filtered feature selection method and multi-task optimization framework.

Although AMFEA shows strong classification performance, its computation time is relatively long because the algorithm needs to use traditional filtering feature selection methods during task generation. Therefore, the future research could focus on optimizing task-generation strategies to reduce run time.

In future work, we plan to address AMFEA's scalability by exploring more effective task-generation strategies and integration approaches. We also plan to study the application of AMFEA in other data domains, such as cancer prediction, by selecting gene features related to cancer to establish early cancer diagnosis models, to ensure its practicality. In addition, we will continue to explore the potential of evolving multi-task optimization methods to design feature selection algorithms with better performance.

CRediT authorship contribution statement

Zhihui Li: Writing – original draft, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Hong Li:** Writing – review & editing, Supervision, Project administration, Methodology, Funding acquisition. **Weifeng Gao:** Resources, Project administration, Funding acquisition. **Jin Xie:** Resources, Project administration, Funding acquisition. **Adam Slowik:** Writing – review & editing.

Funding

This work was supported in part by the National Nature Science Foundation of China under Grant 62276202 and 62106186, in part by the Ministry of Education joint fund of China under Grant 8091B03072304, in part by the China Postdoctoral Science Foundation, China under Grant 2023T160501, in part by the Joint Funds of the Zhejiang Provincial Natural Science Foundation of China under Grant LHZY24A010005, in part by National Key Laboratory of Science and Technology on Space Microwave of China under Grant HTKJ2024KL504008 and in part by the Fundamental Research Funds for the Central Universities of China, China under Grant QTZX22047.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] Peng Wang, Bing Xue, Mengjie Zhang, Jing Liang, A grid-dominance based multi-objective algorithm for feature selection in classification, in: 2021 IEEE Congress on Evolutionary Computation, CEC, 2021, pp. 2053–2060.
[2] Guangfen Wei, Jie Zhao, Yanli Feng, Aixiang He, Jun Yu, A novel hybrid feature selection method based on dynamic feature importance, Appl. Soft Comput. (ISSN: 1568-4946) 93 (2020) 106337.
[3] Jie Cai, Jiawei Luo, Shulin Wang, Sheng Yang, Feature selection in machine learning: A new perspective, Neurocomputing (ISSN: 0925-2312) 300 (2018) 70–79.
[4] Emrah Hancer, Bing Xue, Mengjie Zhang, Differential evolution for filter feature selection based on information theory and feature ranking, Knowl.-Based Syst. (ISSN: 0950-7051) 140 (2018) 103–119.
[5] Andrea Bommert, Xudong Sun, Bernd Bischl, Jörg Rahnenführer, Michel Lang, Benchmark for filter methods for feature selection in high-dimensional classification data, Comput. Statist. Data Anal. (ISSN: 0167-9473) 143 (2020) 106839.
[6] M. Cherrington, F. Thabtah, J. Lu, Q. Xu, Feature selection: filter methods performance challenges, in: 2019 International Conference on Computer and Information Sciences, ICCIS, Sakaka, Saudi Arabia, 2019, pp. 1–4.
[7] J. Benesty, J. Chen, Y. Huang, I. Cohen, Pearson correlation coefficient, in: Noise Reduction in Speech Processing, Springer Topics in Signal Processing, vol. 2, Springer Berlin Heidelberg, 2009, pp. 1–4.
[8] Marc Sebban, Richard Nock, A hybrid filter/wrapper approach of feature selection using information theory, Pattern Recognit. (ISSN: 0031-3203) 35 (4) (2002) 835–846.
[9] Girish Chandrashekar, Ferat Sahin, A survey on feature selection methods, Comput. Electr. Eng. (ISSN: 0045-7906) 40 (1) (2014) 16–28.
[10] Isabelle M. Guyon, et al., Gene selection for cancer classification using support vector machines, Mach. Learn. 46 (2002) 389–422.
[11] Mostafa Moradkhani, Ali Amiri, Mohsen Javaherian, Hossein Safari, A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm, Appl. Soft Comput. (ISSN: 1568-4946) 35 (2015) 123–135.
[12] M. Abdellatief, Y.M. Hassan, M.T. Elnabwy, L.S. Wong, R.J. Chin, K.H. Mo, Investigation of machine learning models in predicting compressive strength for ultra-high-performance geopolymer concrete: A comparative study, Constr. Build. Mater. (ISSN: 0950-0618) 436 (2024) 136884.
[13] Haoyue Liu, MengChu Zhou, Qing Liu, An embedded feature selection method for imbalanced data classification, IEEE/CAA J. Autom. Sin. 6 (3) (2019) 703–715.
[14] W. Zheng, S. Chen, Z. Fu, F. Zhu, H. Yan, J. Yang, Feature selection boosted by unselected features, IEEE Trans. Neural Netw. Learn. Syst. 33 (9) (2022) 4562–4574.
[15] K. Chen, B. Xue, M. Zhang, F. Zhou, Evolutionary multitasking for feature selection in high-dimensional classification via particle swarm optimization, IEEE Trans. Evol. Comput. 26 (3) (2022) 446–460.
[16] L. Li, M. Xuan, Q. Lin, M. Jiang, Z. Ming, K.C. Tan, An evolutionary multitasking algorithm with multiple filtering for high-dimensional feature selection, IEEE Trans. Evol. Comput. 27 (4) (2023) 802–816.
[17] Yongming Li, Sujuan Zhang, Xiaoping Zeng, Research of multi-population agent genetic algorithm for feature selection, Expert Syst. Appl. (ISSN: 0957-4174) 36 (9) (2009) 11570–11581.
[18] S. Gu, R. Cheng, Y. Jin, Feature selection for high-dimensional classification using a competitive swarm optimizer, Soft Comput. 22 (2018) 811–822.
[19] Yong Zhang, Dun-wei Gong, Xiao-zhi Gao, et al., Binary differential evolution with self-learning for multi-objective feature selection, Inform. Sci. (ISSN: 0020-0255) 507 (2020) 67–85.
[20] Majdi Mafarja, Ibrahim Aljarah, Ali Asghar Heidari, et al., Evolutionary population dynamics and grasshopper optimization approaches for feature selection problems, Knowl.-Based Syst. (ISSN: 0950-7051) 145 (2018) 25–45.
[21] M. Tubishat, M.A.M. Abushariah, N. Idris, et al., Improved whale optimization algorithm for feature selection in arabic sentiment analysis, Appl. Intell. 49 (2019) 1688–1707.
[22] E. Emary, Hossam M. Zawbaa, Aboul Ella Hassanien, Binary grey wolf optimization approaches for feature selection, Neurocomputing (ISSN: 0925-2312) 172 (2016) 371–381.
[23] Mehrdad Rostami, Kamal Berahmand, Elahe Nasiri, Saman Forouzandeh, Review of swarm intelligence-based feature selection methods, Eng. Appl. Artif. Intell. (ISSN: 0952-1976) 100 (2021) 104210.
[24] Q. Al-Tashi, S.J. Abdulkadir, H.M. Rais, S. Mirjalili, H. Alhussian, Approaches to multi-objective feature selection: A systematic literature review, IEEE Access 8 (2020) 125076–125096.
[25] X. Song, Y. Zhang, D. Gong, H. Liu, W. Zhang, Surrogate sample-assisted particle swarm optimization for feature selection on high-dimensional data, IEEE Trans. Evol. Comput. 27 (3) (2023) 595–609.
[26] H. Xu, B. Xue, M. Zhang, A duplication analysis-based evolutionary algorithm for biobjective feature selection, IEEE Trans. Evol. Comput. 25 (2) (2021) 205–218.
[27] S.-H. Wu, Z.-H. Zhan, K.C. Tan, J. Zhang, Orthogonal transfer for multitask optimization, IEEE Trans. Evol. Comput. 27 (1) (2023) 185–200.
[28] K. Chen, B. Xue, M. Zhang, F. Zhou, An evolutionary multitasking-based feature selection method for high-dimensional classification, IEEE Trans. Cybern. 52 (7) (2022) 7172–7186.
[29] B. Da, et al., Evolutionary Multitasking for Single-Objective Continuous Optimization: Benchmark Problems, Performance Metric, and Baseline Results, Technical Report, Nanyang Technological University, 2017.
[30] M. Zaefferer, T. Bartz-Beielstein, Efficient global optimization with indefinite Kernels, in: Proc. Int. Conf. Parallel Probl. Solving Nat., 2016, pp. 69–79.
[31] E. Osaba, J. Del Ser, A.D. Martinez, et al., Evolutionary multitask optimization: A methodological overview, challenges, and future research directions, Cogn. Comput. 14 (2022) 927–954.
[32] Huan Liu, Lei Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Trans. Knowl. Data Eng. 17 (4) (2005) 491–502.
[33] Tansel Dokeroglu, Ender Sevinc, Tayfun Kucukyilmaz, Ahmet Cosar, A survey on new generation metaheuristic algorithms, Comput. Ind. Eng. (ISSN: 0360-8352) 137 (2019) 106040.
[34] Tansel Dokeroglu, Ayça Deniz, Hakan Ezgi Kiziloz, A comprehensive survey on recent metaheuristics for feature selection, Neurocomputing (ISSN: 0925-2312) 494 (2022) 269–296.
[35] A. Gupta, Y.-S. Ong, L. Feng, Multifactorial evolution: toward evolutionary multitasking, IEEE Trans. Evol. Comput. 20 (3) (2016) 343–357.
[36] Ha Bang Ban, Dang Hai Pham, Solving optimization problems simultaneously: The variants of the traveling salesman problem with time windows using multifactorial evolutionary algorithm, PeerJ Comput. Sci. 9 (2023).
[37] H. Hu, Y. Zhou, T. Wang, X. Peng, A multi-task algorithm for autonomous underwater vehicles 3D path planning, in: 2020 3rd International Conference on Unmanned Systems, ICUS, Harbin, China, 2020, pp. 972–977.
[38] L. Zhou, et al., Toward adaptive knowledge transfer in multifactorial evolutionary computation, IEEE Trans. Cybern. 51 (5) (2021) 2563–2576.
[39] K.K. Bali, Y.-S. Ong, A. Gupta, P.S. Tan, Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II, IEEE Trans. Evol. Comput. 24 (1) (2020) 69–83.
[40] A. Gupta, Y.-S. Ong, L. Feng, K.C. Tan, Multiobjective multifactorial optimization in evolutionary multitasking, IEEE Trans. Cybern. 47 (7) (2017) 1652–1665.
[41] J. Rice, C.R. Cloninger, T. Reich, Multifactorial inheritance with cultural transmission and assortative mating. I. Description and basic properties of the unitary models, Am. J. Hum. Genet. 30 (1978) 618–643.
[42] G. Patterson, M. Zhang, Fitness functions in genetic programming for classification with unbalanced data, in: Proc. 20th Aust. Joint Conf. Artif. Intell., 2007, pp. 769–775.

[43] Ziying Tan, Linbo Luo, Jinghui Zhong, Knowledge transfer in evolutionary multitask optimization: A survey, Appl. Soft Comput. (ISSN: 1568-4946) 138 (2023) 110182.
[44] Laurens van der Maaten, Geoffrey Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.
[45] Y. Zhou, W. Zhang, J. Kang, X. Zhang, X. Wang, A problem-specific non-dominated sorting genetic algorithm for supervised feature selection, Inform. Sci. 547 (2021) 841–859.
[46] F. Cheng, F. Chu, Y. Xu, L. Zhang, A steering-matrix-based multiobjective evolutionary algorithm for high-dimensional feature selection, IEEE Trans. Cybern. (2021) 1–14.
[47] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[48] Y. Tian, X. Zheng, X. Zhang, Y. Jin, Efficient large-scale multiobjective optimization based on a competitive swarm optimizer, IEEE Trans. Cybern. 50 (8) (2020) 3696–3708.