This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

A Fast Hybrid Feature Selection Based on Correlation-Guided Clustering and Particle Swarm Optimization for High-Dimensional Data

Xian-Fang Song, Yong Zhang, *Member, IEEE*, Dun-Wei Gong, *Member, IEEE*, and Xiao-Zhi Gao

***Abstract*—The "curse of dimensionality" and the high computational cost have still limited the application of the evolutionary algorithm in high-dimensional feature selection (FS) problems. This article proposes a new three-phase hybrid FS algorithm based on correlation-guided clustering and particle swarm optimization (PSO) (HFS-C-P) to tackle the above two problems at the same time. To this end, three kinds of FS methods are effectively integrated into the proposed algorithm based on their respective advantages. In the first and second phases, a filter FS method and a feature clustering-based method with low computational cost are designed to reduce the search space used by the third phase. After that, the third phase applies oneself to finding an optimal feature subset by using an evolutionary algorithm with the global searchability. Moreover, a symmetric uncertainty-based feature deletion method, a fast correlation-guided feature clustering strategy, and an improved integer PSO are developed to improve the performance of the three phases, respectively. Finally, the proposed algorithm is validated on 18 publicly available real-world datasets in comparison with nine FS algorithms. Experimental results show that the proposed algorithm can obtain a good feature subset with the lowest computational cost.**

***Index Terms*—Clustering, feature selection (FS), hybrid search, particle swarm optimization (PSO).**

I. Introduction

FEATURE selection (FS) is an important dimension reduction method, which has been applied in many real problems, such as bioinformatics and image processing [1]–[4]. For a dataset with noise, irrelevant or redundant features generally slow down the speed of a learning algorithm, and even reduce its accuracy. By eliminating irrelevant and redundant features, an FS algorithm can reduce the size of features, shorten the learning time, and/or improve the classification performance of the algorithm [5], [6].

Existing FS algorithms can be divided into three categories: 1) filter; 2) wrapper; and 3) embedded [2], [7]. The difference between the filter and wrapper is whether a classifier is used as the evaluation criterion of the feature subset. Compared with the wrapper method, the filter method is independent of learning algorithms, having the advantage of low computational complexity [8], [9]. This makes it more suitable for high-dimensional FS problems. However, its performance is often worse than the wrapper and embedded methods due to the lack of a follow-up learning algorithm. The wrapper utilizes a learning algorithm as a black box to score feature subsets. Since it can use a powerful search strategy to seek optimal feature subsets, this kind of method is generally more efficient than the filter. Until now, many effective search strategies have been proposed, such as the sequential forward-selection algorithm (SFS) [10], sequential backward-selection algorithm (SBS) [11], and Plus-l take-away-r algorithm (PTA) [12], to name a few. However, most of them still suffer from shortcomings, such as the local convergence and/or high computational cost [13].

To improve the global search capability of an FS algorithm, recently evolutionary computation (EC) has been applied to FS problems [14]–[16]. Particle swarm optimization (PSO) [17]–[19] is a relatively new EC algorithm [20]. Compared with other ECs, such as the genetic algorithm, PSO has the advantages of concise implementation and fast convergence in most cases [21]. Now, PSO-based FS algorithms have received much attention [13], [22]. Part of representative algorithms include the bare-bones PSO (BPSO)-based FS algorithm [23]; the multiobjective PSO-based FS algorithms [21], [24], [25]; the discretization-based FS algorithm [26]; the self-adaptive algorithm [27]; the set-based PSO algorithm [28]; the adaptive potential PSO algorithm [29]; and so on. Most of them employ PSO to search good features and adopt a predetermined learning system (such as a classifier) to evaluate the goodness of selected features. However, because of the need to repeatedly evaluate the classification performance of different feature subsets, these methods often have high computational cost, limiting their applications in high-dimensional data [22], [26].

Because of being able to integrate the advantages of different FS methods, hybrid FS algorithms have also been

Manuscript received March 28, 2020; revised July 25, 2020, November 1, 2020, and January 18, 2021; accepted February 17, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61876185 and Grant 51875113; and in part by the Scientic Innovation 2030 Major Project for New Generation of AI under Grant 2020AAA0107300, Ministry of Science and Technology of the People's Republic of China. This article was recommended by Associate Editor M. Zhang. *(Corresponding authors: Yong Zhang; Dun-Wei Gong.)*

Xian-Fang Song, Yong Zhang, and Dun-Wei Gong are with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China (e-mail: songxf0614@126.com; yongzh401@126.com; dwgong@vip.163.com).

Xiao-Zhi Gao is with the School of Computing, University of Eastern Finland, 70210 Kuopio, Finland (e-mail: xiao.z.gao@gmail.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2021.3061152.

Digital Object Identifier 10.1109/TCYB.2021.3061152

2168-2267 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

studied [30]–[33]. Existing hybrid methods can be divided into two categories: 1) filter–wrapper hybrid and 2) filter-clustering hybrid. Most of those methods adopt the two-phase hybrid way. The first phase applies oneself to removing irrelevant features and the second phase is devoted to finding out an optimal feature subset from that reduced feature space. In addition, filter methods have been used to improve the performance of evolutionary feature section algorithms. However, the existing hybrid algorithms still face the following challenges.

1) The capability of reducing the search space is still limited when tackling high-dimensional data. Taking the dataset CNS as an example, after removing irrelevant features by using the mutual information maximization method [34], the size of the remaining features is still relatively large, whose value is more than 6000.
2) Since it is difficult to determine an appropriate threshold to judge whether a feature is useless, part algorithms often wrongly delete features, which are valid but have a weak correlation.
3) Because of the need of calculating the similarity between features, part methods still face the challenge of high-computational cost.

Different from these two-phase hybrid approaches, this article develops a novel three-phase hybrid FS algorithm (HFS-C-P). After removing irrelevant features, in the second phase, a fast feature clustering method is proposed to further reduce the search space of PSO. Following that, the third phase can fast select representative features from a small number of feature clusters. The main contributions of this article are as follows.

1) Proposing a novel three-phase hybrid FS method, that is, filter-clustering-evolutionary wrapper algorithm, which effectively integrates the advantages of three kinds of FS algorithms. Compared with the existing filter–wrapper hybrid strategy, the newly added second phase, that is, the feature clustering, can obviously reduce the search space of the subsequent wrapper method. Compared with the existing filter-clustering hybrid strategy, the newly added third phase, that is, the evolutionary wrapper method with global search capability, can find better representative features from such feature clusters.
2) Developing a fast correlation-guided feature clustering strategy (FCFC). Traditional methods need to compare the similarity between all features one by one; however, FCFC needs only to compare the similarity between features and known cluster centers. It can not only reduce obviously the computation cost of clustering features but also does not need to specify the clustering number in advance.
3) Proposing an improved BPSO algorithm suitable for the integer FS model. Two problem-specific operators, that is: a) the relevance-guided swarm initialization and b) the difference-based adaptive disturbance, are developed to improve the performance of PSO.

The remainder of this article is organized as follows. Section II introduces basic and related works. Section III introduces the proposed algorithm. Section IV validates the performances of the proposed algorithm. Section V gives the conclusions.

II. Related Work

A. Feature Selection Problems

The purpose of FS is to choose some key features from the original feature set, reducing the learning cost while optimizing given performance measures. Assuming that a dataset *Data* has D features and L instances, and its original feature set is F, an FS problem in the context of classification can be represented: selecting as few features as possible to make the given performance index $H(\cdot)$ optimal. The expression is as follows:

$$\begin{aligned} \max\ & H(X) \\ \text{s.t.}\ & X = (x_1, x_2, \ldots, x_D) \\ & x_i \in \{0, 1\},\ i = 1, 2, \ldots, D \end{aligned} \tag{1}$$

where $x_i = 1$ represents that the ith feature is chosen into the feature subset X; otherwise, it is not selected.

B. Particle Swarm Optimization

In the PSO, each particle represents a potential solution of the optimized problem. In each iteration, a particle updates its position by learning two empirical positions. One is the optimal position found by the particle itself, that is, the local or personal leader (*Pbest*). Another is the optimal position found by the particle's neighborhoods, that is, the global leader (*Gbest*). Without loss of generality, assuming that the personal and global leaders of the ith particle $X_i = (x_{i,1}, \ldots, x_{i,D})$ are $Pb_i = (pb_{i,1}, \ldots, pb_{i,D})$ and $Gb_i = (gb_{i,1}, \ldots, gb_{i,D})$, respectively, this particle is updated as follows:

$$v_{i,j}^{t+1} = wv_{i,j}^{t} + c_1 r_1 \left(pb_{i,j}^{t} - x_{i,j}^{t}\right)) + c_2 r_2 \left(gb_{i,j}^{t} - x_{i,j}^{t}\right) \tag{2}$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} \tag{3}$$

where t is the number of iterations, w is the inertia weight, c_1 and c_2 are the two learning factors, and r_1 and r_2 are two random numbers within [0, 1].

Kennedy [35] proposed a simpler version of PSO, that is,, the BPSO. BPSO adopts a Gauss distribution without w, c_1 and c_2 to update the particles. Taking the ith particle as example, the new formula is as follows:

$$x_{i,j}^{t+1} = \begin{cases} G\left(0.5\left(pb_{i,j}^{t} + gb_{i,j}^{t}\right), \left|pb_{i,j}^{t} - gb_{i,j}^{t}\right|\right), & \text{rand} < 0.5 \\ pb_{i,j}^{t}, & \text{otherwise} \end{cases} \tag{4}$$

where $G(a, b)$ is a Gaussian distribution with mean a and variance b.

C. Existing Feature Selection Approaches

1) Evolutionary Feature Selection: The EC-based FS approach is called evolutionary FS for short. Until now, many typical evolutionary algorithms have been applied to FS problems, including the genetic algorithm [33], [36]; shuffled frog leaping algorithm [37]; ant colony algorithm [14], [38]; artificial bee colony algorithm [39]; differential evolution [15], [40]; firefly algorithm [41]; salp swarm algorithm [42]; and whale optimization algorithm [43] to name a few. In 2015, Diao and Shen gave a detailed

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

overview on existing nature-inspired FS approaches [22], and Xue *et al.* [13] surveyed EC approaches applied to FS. Hancer *et al.* [44] introduced a comprehensive survey on FS approaches for clustering, where some latest research results in evolutionary FS were also surveyed.

Until now, a lot of PSO-based FS algorithms have been developed. Tran *et al.* [26] proposed a new PSO-based method for the discretization and FS of high-dimensional data. In the algorithm, a new representation of particles was developed to reduce the search space, and a new fitness function was used to evaluate candidate solutions. Their experimental results showed that the algorithm can select less than 5% of features for all the test datasets. Zhang *et al.* [24] proposed an improved multiobjective PSO for cost-sensitive FS problems by combining an effective hybrid operator, the crowding distance, and the external archive. This algorithm can obtain a set of nondominated feature subsets instead of a single feature subset. After that, Tran *et al.* [45] proposed a variable-length PSO-based FS method by combining several new operators, including the population division, the feature ranking, and the particle length changing mechanism. This algorithm can help the swarm jump out of local optima, and reduce the search space by using the length changing mechanism. However, it updates independently each dimension of a particle, without fully considering the relationship between different dimensions. Chen *et al.* [46] presented a hybrid PSO FS algorithm based on a spiral-shaped mechanism. In the algorithm, a logical map sequence was used to enhance the diversity of the swarm, and a location update model was proposed to improve the quality of the swarm. However, since it cannot remove redundant features effectively, the size of the optimal feature subset obtained is still relatively large when dealing with high-dimensional data. Overall, those algorithms above all improve the capability of PSO in dealing with FS problems. However, since the dimension of the search space is not significantly reduced, in essence, most of them still face the problem of "curse of dimensionality."

2) Clustering-Based Feature Selection: As a typical unsupervised analysis approach, clustering technology has been employed to deal with FS problems. Song *et al.* [8] proposed a two-phase clustering-based FS algorithm. The first phase used a minimum spanning tree to cluster all features, while the second phase selected the most representative feature from each cluster to form a feature subset. Liu *et al.* [47] proposed a clustering approach based on the minimum spanning tree, in which a measure with information variation was used to evaluate the relevance and redundancy between features. First, a minimum spanning tree was created based on the correlation between features, then those long edges were removed to form clusters. After that, a feature with maximal relevance was selected from each cluster. Because of the need to calculate the correlation between all features in advance for forming a complete graph, the two algorithms above still have high computational cost when dealing with high-dimensional data. Jie *et al.* [48] presented density peak clustering, in which the information distance between features was used as a clustering measure. This algorithm can effectively cluster features, but it needs to set the number of clusters in advance. Chatterjee *et al.* [49] developed a filter method based on K-means. In the method, the K-means technology was used to cluster features, and then the features in each cluster were sorted according to their importance. Finally, a certain proportion of features was selected from each cluster. The algorithm can cluster features effectively. However, it needs to run the K-means technology many times for determining the best K value. As we know, the K-means usually has high computational cost on high-dimensional data. Most of all, in all the above methods, each feature cluster selects independently its own representative feature without considering the combination performance of those selected features.

3) Hybrid Feature Selection: The hybrid FS approach can combine the advantages of different FS methods [50]. Until now, many hybrid FS approaches have been proposed. Combining the filter FS methods and wrapper FS methods, Apolloni *et al.* [51] proposed two hybrid FS algorithms, both of which were combined by a differential evolution-based wrapper method and a rank-based filter method. Huang *et al.* [52] designed a two-phase FS algorithm by combining the binary state transition algorithm and ReliefF. With the help of some new operators, these algorithms both show good global convergence on low-dimensional data. However, since the search space of wrapper FS methods still includes lots of irrelevant features, their ability to process high-dimensional data is still limited. Moreover, Lu *et al.* [34] proposed a hybrid FS algorithm by combining the mutual information maximization and an adaptive GA. On the basis of mutual information maximization, the number of features was reduced to a predetermined value, 300, for all the test datasets. This algorithm can effectively reduce the dimension of original data by removing redundancy features. However, setting the same size (300) for all data with different feature sizes is arbitrary, and the size of 300 is still relatively large in many cases. Ansari *et al.* [53] proposed a PSO-based hybrid FS method. In the filter phase, the recursive feature elimination was used to remove irrelevant features. In the wrapper phase, a binary PSO was applied to obtain the optimal feature subset. Wu *et al.* [54] proposed a hybrid FS method with binary quantum PSO. In this algorithm, the maximum information coefficient was used to remove irrelevant features before executing PSO. All these algorithms can delete a lot of irrelevant features before executing wrapper FS methods, but they still face the problem of the "curse of dimensionality" when tackling data with a large number of relevant features.

In recent years, some scholars began to study filter-cluster hybrid FS approaches. Using the minimum spanning tree to cluster relevant features, and the filter method to select the most relevant feature from each cluster, Song *et al.* [8] proposed a fast clustering-based feature subset selection algorithm. Chatterjee *et al.* [49] proposed a filter-cluster method based on K-means. After clustering all features by the K-means, a certain proportion of features was selected from each cluster to form the final feature subset. This kind of hybrid method can obviously reduce the feature search space to a certain extent. However, they just simply select the top feature from each cluster to form the final feature subset, which

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

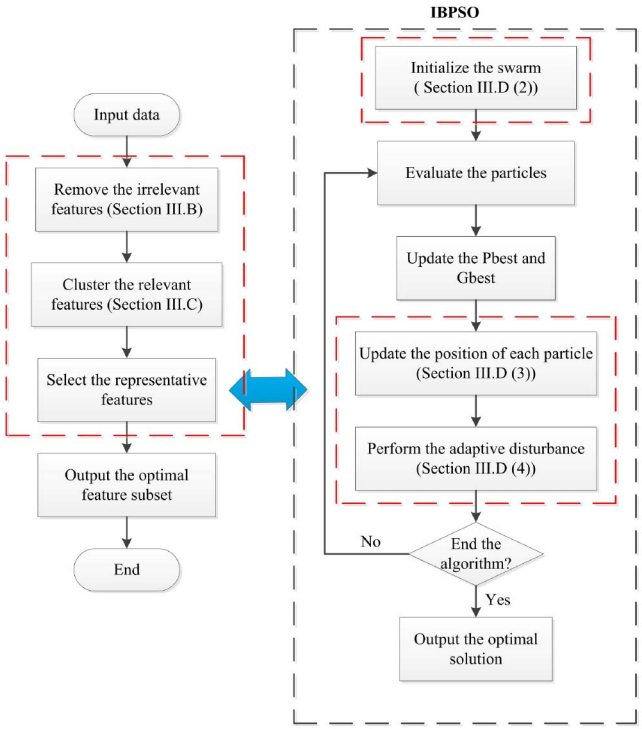

Fig. 1. Framework of the proposed hybrid FS algorithm.

ignores the complex coupling relationship between different features.

III. Proposed Algorithm

A. General Framework and Key Definitions

The "curse of dimensionality" and high-computational cost are two main challenges when using EC to deal with high-dimensional FS problems. One of the most direct and effective methods to overcome the two challenges is to reduce the search space of EC. In view of this, this article proposes a fast three-phase hybrid FS algorithm. Fig. 1 shows its general framework. The framework consists of three phases, that is: 1) removing irrelevant features; 2) clustering relevant features; and 3) selecting representative features. First, the first phase deletes features, which are irrelevant or weakly related to the current classification task. Next, the second phase divides the remaining relevant features into multiple groups according to their similarity. After that, the last phase selects representative features from those feature clusters by employing an evolutionary algorithm, thus forming the final feature subset. This article selects PSO as the global search approach in the third phase. The main contributions of this article have been marked with red lines in Fig. 1.

In this framework, since the number of feature clusters is generally far less than the size of original features, both the search space and the evaluation cost of PSO are obviously reduced in the third phase. Moreover, since those features belonging to different clusters are relatively independent of each other to some extent, selecting features from different clusters can also improve the speed of PSO to find the optimal selection subset.

Of course, the proposed framework can be extended to other evolutionary algorithms, such as the genetic algorithm. Specifically, in the third phase of the proposed framework, other evolutionary algorithms can also be used to select representative features from those feature clusters, instead of PSO. But their update rules on individuals must be suitable for integer encoding.

1) Main Challenges: The proposed framework can effectively reduce the search space of PSO used in the third phase and, at the same time, pose some challenges to the first and second phases, as follows.

1) How to evaluate the relevant degree between a feature and the classification task in the first phase? Even if we can obtain the relevant degree, it is difficult to correctly distinguish irrelevant features from all features. Taking the datasets, Driv_face and Yale_64, as examples, the biggest relevant degree of a feature (i.e., the C-relevance value defined by Definition 1) is 0.1199 for Driv_face, while the smallest relevant degree of a feature is 0.3473 for Yale_64. This indicates that using a fixed threshold to distinguish irrelevant features is no longer feasible.
2) How to divide similar features into the same cluster quickly and accurately? The existing clustering methods [8], [47] usually need to calculate the similar degrees between all features before clustering. Taking data with D features as an example, they need to run the correlation analysis $(D-1)!$ times for obtaining similar degrees between all features. The computational complexity increases exponentially as the size of features increases.

2) Key Definitions: In many FS algorithms [8], [45], [55], symmetric uncertainty (SU) has been employed to describe the correlation between features and class labels or between features, because it can well describe the nonlinear relationship between random variables. This article also uses SU to evaluate the correlation between features and class labels. First, we give the SU method and three existing measures, that is: 1) the C-relevance; 2) the S-relevance; and 3) the F-Correlation [8]. Next, three new measures are proposed for trying to provide a theoretical basis for the proposed algorithm.

The SU is defined as follows:

$$SU(X,Y)=2\frac{H(X)-H(X|Y)}{H(X)+H(Y)} \tag{5}$$

where $H(X)$ and $H(Y)$ are the entropies of variables X and Y, respectively, and $H(X|Y)$ is the conditional entropy of X when Y is known. Assuming that $p(x)$ and $p(y)$ are the prior probabilities of X and Y, $p(x|y)$ is the posterior probability of X when Y is given, and we have

$$H(X)=-\sum_{x\in X}p(x)\log_2 p(x) \tag{6}$$

$$H(Y)=-\sum_{y\in Y}p(y)\log_2 p(y) \tag{7}$$

$$H(X|Y)=-\sum_{y\in Y}p(y)\sum_{x\in X}p(x|y)\log_2 p(x|y). \tag{8}$$

Letting F be the original feature set, and $f_i \in F$ be the ith feature, three key definitions are described as follows.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Definition 1 (C-Relevance): The relevance between a feature $f_i \in F$ and the class labels C, that is, the C-relevance between f_i and C, is calculated by $SU(f_i, C)$.

Definition 2 (S-Relevance): For a feature $f_i \in F$ and the class labels C, if the value of $SU(f_i, C)$ is greater than a predetermined threshold ρ_0, we call that f_i is a strong relevant feature with respect to C.

Definition 3 (F-Correlation): The correlation between two features f_i and $f_j (f_i \neq f_j)$, called the F-correlation, can be denoted by $SU(f_i, f_j)$.

Definition 4 (W-Correlation): For any two features f_i and f_j, if the value of $|SU(f_i, C) - SU(f_j, C)|$ is greater than a predetermined threshold ρ_1, we believe that the two features have weak correlation.

As the conclusion given in [8], redundant features are assembled in a cluster, and a representative feature can be taken out of the cluster. In order to cluster redundant features in a group, we give the definition of feature redundancy.

Definition 5 (F-Redundancy): For any two features f_i and f_j, if they satisfy that $|SU(f_i, C) - SU(f_j, C)| \leq \rho_1$ and $SU(f_i, f_j) \geq \min(SU(f_i, C), SU(f_j, C))$, then f_i and f_j are mutually redundant.

On the basis of the method of Definition 5, all the original features can be divided into a number of homogeneous groups by checking the redundancy relationship between them. In this article, we call these feature groups *Ru*-feature clusters.

Definition 6 (Ru-Feature Cluster): For a S-Relevance feature f_i with respect to C, all its redundant features constitute a redundant feature set, called the *Ru*-feature cluster.

Like most of the clustering-based FS methods [8], [47], [56], after all the features are clustered into multiple groups according to their redundancy, some representative features should be selected from such feature clusters to form a final feature subset. Since such features in the same cluster are mutually redundant, selecting a representative feature from each feature cluster can obtain a good feature subset, as shown in the results of [8], [47], and [56].

Overall, the above definitions indicate the following.

1) Irrelevant features have no/weak correlation with the class labels C (from Definitions 1 and 2).
2) Two features that have a strong correlation with C should have close C-relevance values (from Definition 4).
3) After mutually redundant features are grouped into the same cluster (from Definition 5), selecting a representative from each feature cluster can constitute an optimal feature subset (from Definition 6 and the related conclusions in [8], [47], and [56]).

According to the above work, the three phases of the proposed framework are translated into the following three processes: 1) identifying and retaining S-relevance features; 2) dividing the S-relevance features into multiple *Ru*-feature clusters; and 3) selecting representative features from those *Ru*-feature clusters by a global search strategy.

B. First Phase: Removing Irrelevant Features

For a dataset S with D features, $F = \{f_1, f_2, \ldots, f_D\}$, first, the C-relevance value of each feature, $SU(f_i, C)$, $i = 1, 2, \ldots, D$, is calculated. Following that, the features whose C-relevance values are higher than a predetermined threshold ρ_0 are selected and saved into the set of strong relevance features F'.

Algorithm 1: Method of Removing Irrelevant Features

Input: The original feature set, $F = (f_1, f_2, \cdots, f_D)$, the set of class labels, C
Output: The set of strong relevance features, F'

```
Calculate the C-relevance value of each feature, SU(f_i, C), i = 1, 2 ··· , D;
Determine ρ_0 = min(0.1 * SU_max, SU_⌊D/ log D⌋−th);
for i=1:D do
    % from the first feature to the last one in the F;
    if SU(f_i, C) ≥ ρ_0 then
        Save the i-th feature into the set F';
    end
end
Output the set F'.
```

In the above method, threshold ρ_0 plays a key role in deciding S-relevance features. Traditionally, ρ_0 is set to be a very small constant. However, when dealing with some real data, the C-relevance values of all features are relatively large. Taking Yale_64 as an example, its C-relevance values are close to 0.5. To address the above issue, Yu and Liu [57] suggested that the threshold ρ_0 be set to the C-relevance value of the $\lfloor D/\log D \rfloor - \text{th}$ ranked feature in the dataset, that is, $\rho_0 = SU_{\lfloor D/\log D \rfloor - \text{th}}$, where $\lfloor \cdot \rfloor$ represents the rounding down function. In order to prevent removing relevant features by setting a high threshold ρ_0, we set threshold ρ_0 as follows:

$$\rho_0 = \min\left(0.1 * SU_{\max}, SU_{\lfloor D/\log D \rfloor - \text{th}}\right). \tag{9}$$

In the above formula, $SU_{\max}$ represents the maximal C-relevance value among all features. Furthermore, Algorithm 1 shows the pseudocode of the operator "removing irrelevant features."

C. Second Phase: Clustering Relevant Features

A good clustering method should be able to group similar features into the same cluster with lower computational cost. Researchers have proposed several typical strategies [8], [47], [56], [58] to improve the performance of feature clustering. However, since there is a need to calculate the correlation degrees between all features, most of them suffer from the disadvantage of high calculation cost.

As shown in Definition 4, if the $|SU(f_i, C) - SU(f_j, C)|$ value between the features f_i and f_j is greater than a predetermined threshold ρ_1, the two features should have weak correlation to a large extent. Calculating the C-relevance values of all D features in a dataset only needs to run the SU measure D times. This means that we can delete weakly relevant features relative to the current feature by using the W-correletion, before evaluating the redundancy between all features accurately.

Based on this, this article proposes an FCFC. The pseudocodes of FCFC are listed in Algorithm 2. FCFC employs the set of strong relevance features obtained by the first phase as an input, that is, $F' = \{f'_1, f'_2, \ldots, f'_{D'}\}$, $(D' \leq D)$. First, according to the C-relevance values, all the features in the set

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

F' are sorted from high to low, denoting the result after being sorted by $U_0 = \{f_1'', f_2'', \ldots, f_{D'}''\}$. After that, the two steps (i.e., the 6th and 15th lines in Algorithm 2) are used to select all features similar to the first feature. Since the first feature in set U_0 has the biggest correlation degree with the class labels, it is the most important feature among U_0. Based on this, the first feature f_1'' and its redundant features constitute the first feature cluster Cluster[1]. Following that, we reset $U_0 = U_0/\text{Cluster}^1$, and the two steps above are executed circularly until all the features in U_0 are divided into a cluster.

In Algorithm 2, the function of the 6th line is to remove features that have a weak correlation with the first feature in U_0, by using the C-relevance. The function of the 15th line is to find all features similar to the first feature in the set U_0 by using Definition 5, that is, all redundant features about the first feature. Taking the set $U_1 = U_0$ as an input, the methods described in the 6th and 15th lines are described as follows.

1) Remove Features Which Have Weak Correlation With the First Feature in U_1: For this goal, FCFC first calculates the difference in the C-relevance between the first feature $f_{1-\text{th}}''$ and remaining ones in the set U_1, denoted these differences by $dt(1-\text{th}, i)$, $i = 2, 3, \ldots, |U_1|$, where $|U_1|$ represents the size of U_1. For any feature $f_i'' \in U_1$, if the value of $dt(1-\text{th}, i)$ is greater than the predetermined threshold ρ_1, this feature is removed from set U_1. Based on the process above, we will obtain a smaller feature set U_1 ($|U_1| < |U_0|$). Since a larger difference in the C-relevance between two features indicates that they have lower correlation, we set the value of ρ_1 as follows:

$$\rho_1 = dt_{\max} \times \frac{\log(D')}{D'} \tag{10}$$

where $dt_{\max}$ is the maximal difference between any two features on the C-relevance.

2) Find All Features Similar to the First Feature From the Set U_1, and Save Them Into the Same Cluster: This step employs the output of the 6th line, U_1, as an input. Since a feature in the set U_1 still has the chance to be irrelevant with the first feature $f_{1-\text{th}}''$, we need to find all features similar to $f_{1-\text{th}}''$ from U_1 further. For any feature $f_i'' \in U_1/\{f_{1-\text{th}}''\}$, if $SU(f_{1-\text{th}}'', f_i'') \geq \min(SU(f_{1-\text{th}}'', C), SU(f_i'', C))$, the features f_1'' and f_i'' should be redundant to each other as Definition 5 shows. Hence, feature f_i'' is saved into the corresponding cluster of $f_{1-\text{th}}''$. By checking each feature in $U_1/\{f_{1-\text{th}}''\}$ in turn, we can find all features similar to $f_{1-\text{th}}''$, that is, its redundant features. After that, feature $f_{1-\text{th}}''$ and its redundant features compose a feature cluster Cluster^k.

Moreover, we give an example to explain lines 16–22 in Algorithm 2. Assuming $U_1 = \{f_3, f_1, f_4, f_{10}\}$, $SU(f_3, C) = 0.22$, $SU(f_1, C) = 0.20$, $SU(f_4, C) = 0.18$, and $SU(f_{10}, C) = 0.16$, the following steps will be performed on the basis of the best important feature in U_1, that is, the first element f_3: 1) calculate the correlation degrees between the first feature (f_3) and the remaining features, denoted as $SU(f_3, f_1) = 0.55$, $SU(f_3, f_4) = 0.15$, and $SU(f_3, f_{10}) = 0.04$; 2) check whether $\{f_1, f_4, f_{10}\}$ satisfy Definition 5 in turn. We find that only f_1 satisfies Definition 5, having $SU(f_3, f_1) > -\min(SU(f_3, C), SU(f_1, C))$. Therefore, f_1 is clustered into the corresponding feature cluster of f_3; and 3) remove the cluster $\{f_3, f_1\}$ from U_1, set $U_1 = U_1/\{f_3, f_1\} = \{f_4, f_{10}\}$, and return to 1) until all features are clustered into a cluster. Note that here the first element in U_1 becomes f_4.

Algorithm 2: Proposed FCFC Clustering Method

Input: The set of strong relevance features obtained by the first phase, F'

Output: The M feature clusters, $F' = \bigcup_{k=1}^{M} Cluster^k$

1 Sort all the features in F' by their C-relevance values, denoted the sorting results as U_0;
2 Let $k = 1$;
3 Set the temporary set $U_1 = U_0$;
4 Let the first feature in the U_1 to be $f_{1-\text{th}}''$, and its C-relevance value to be $SU(f_{1-\text{th}}'', C)$;
5 Initialize the k-th feature cluster to be Cluster$^k = f_{1-\text{th}}''$;
6 Remove features which have weak correlation with $f_{1-\text{th}}''$ from U_1 by the following method (the lines 7-14);
7 **for** i =2:$|U_1|$ **do**
8 % from the second feature to the last one in the U_1 ;
9 Calculate the C-relevance value of f_i'', $SU(f_i'', C)$;
10 Calculate the difference value, $dt(1-\text{th}, i) = \left|SU(f_{1-\text{th}}'', C) - SU(f_i'', C)\right|$;
11 **if** $dt(1-\text{th}, i) > \rho_1$ **then**
12 Remove f_i'' from U_1, i.e., $U_1 = U_1/\{f_i''\}$;
13 **end**
14 **end**
15 Find all features similar to $f_{1-\text{th}}''$ from U_1, and save them into the cluster Cluster^k, by the following method (the lines 16-22);
16 **for** i =2 : $|U_1|$ **do**
17 % from the second feature to the last one in the U_1;
18 Calculate the correlation degree between f_i'' and $f_{1-\text{th}}''$, denoted by $SU(f_{1-\text{th}}'', f_i'')$;
19 **if** $SU(f_{1-\text{th}}'', f_i'') \geq \min(SU(f_{1-\text{th}}'', C), SU(f_i'', C))$ **then**
20 Save f_i'' into Cluster^k, i.e., $Cluster^k = Cluster^k \cup \{f_i''\}$;
21 **end**
22 **end**
23 Set $U_0 = U_0/Cluster^k$;
24 If $|U_0| > 1$, let $k = k + 1$, return to the Step 3; otherwise, set the last cluster to be U_0, and output the clustering result.

Compared with the existing clustering methods, the proposed FCFC method has the following characteristics or advantages.

1) Since two features that have a stronger correlation relationship usually have similar C-relevance values, deleting features that are irrelevant or weakly relevant with the first feature by the 6th line in Algorithm 2 can still guarantee the reliability of clustering results.
2) FCFC does not need to set the number of clusters in advance. Compared with the number of clusters, the correlation threshold ρ_1 is easier to set up by using (10).
3) FCFC has a lower computation complexity. Suppose that running the 6th line of Algorithm 2 every time can delete half of features from the current set U_1, and one Zth of the remaining features is similar to the first feature in U_1, the times of running the SU measure by FCFC are as follows:

$$\prod_{k=0}^{M-1} \frac{1}{2}\left(D - \frac{i \times D}{Z}\right) = \frac{D^M Z!}{2^M Z^M} \tag{11}$$

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

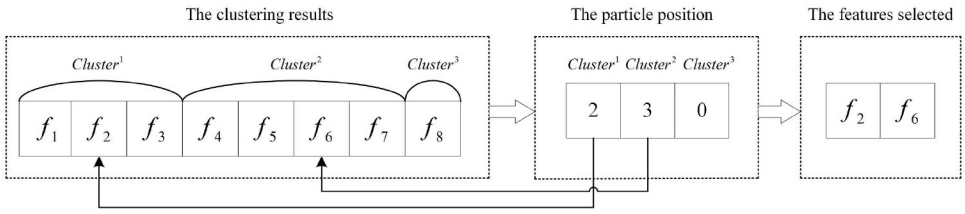

Fig. 2. Method of decoding a particle.

where M is the number of clusters. Considering the worst case that all the D features are unrelated to each other, that is, $M = Z = D$, the times of running the SU measure by FCFC are $[(D-1)!/(2^{D-2})]$. Conversely, since all features are involved, the existing clustering methods usually need to run the SU measure $(D-1)!$ times for determining the redundancy between all features.

D. Third Phase: Selecting Representative Features by Improved Integer PSO

1) Integer Optimization Model: After implementing the second phase of HFS-C-P, all the S-relevance features are divided into different clusters, where each cluster consists of redundant features to each other. The goal of this section is to produce a final feature subset by selecting representative features from those feature clusters. Suppose that the M clusters obtained by the second phase are $\text{Cluster}^k,\ k = 1, 2, \ldots, M$, the goal is to select M' representative features from M feature clusters ($M' \leq M$), making the objective function $H(\cdot)$ the best, namely, having the highest classification performance.

Furthermore, we employ the integer coding to describe a solution (feature subset) of the problem, with the following expression:

$$X = (x_1, x_2, \ldots, x_M),\ x_i \in \left\{0, 1, \ldots, \left|\text{Cluster}^i\right|\right\} \quad (12)$$

where $x_i = a$ indicates that the ath feature in Cluster^i is selected into the feature subset X; if $a = 0$, then the Cluster^i is not selected. Based on this, the optimization model described in (1) can be formulated as follows:

$$\begin{aligned} \max\ & H(X) \\ \text{s.t.}\ & X = (x_1, x_2, \ldots, x_M) \\ & x_i \in \left\{0, 1, \ldots, \left|\text{Cluster}^i\right|\right\},\ i = 1, 2, \ldots, M \\ & \bigcup_{i=1}^{M} \text{Cluster}^i = U,\ \bigcap_{i=1}^{M} \text{Cluster}^i = \Phi. \end{aligned} \quad (13)$$

2) Improved BPSO Algorithm: On one hand, since all particles can learn optimal solutions obtained by the swarm in time, PSO has the advantage of fast convergence in most cases, compared with other EC [21]. On the other hand, compared with the canonical PSO, BPSO in [35] does not require to tune control parameters, including inertia weight and acceleration coefficients for achieving good performance. Therefore, in this article, we extend the concept of BPSO to solve the integer optimization problem above and propose an improved BPSO algorithm (IBPSO). A relevance-guided swarm initialization operator and an adaptive disturbance are developed to improve the capability of IBPSO.

Algorithm 3: Proposed Initialization Operator

Input: The M feature clusters, $Cluster^j,\ j = 1, 2, \cdots M$
Output: The N particles, $X_i,\ i = 1, 2, \cdots N$

```
Calculate the selected probability for each cluster, pc_j, j = 1, 2 ··· , M;
for i= 1 :N do
    for j= 1:M do
        if rand(0, 1) < pc_j then
            Randomly select a feature from the Cluster^j, denoted by the a-th feature;
            Set X_{i,j} = a;
        end
        else
            Set X_{i,j} = 0;
        end
    end
end
Output the N particles, X_i, i = 1, 2, ··· N.
```

Algorithm 4: Proposed IBPSO Algorithm

1. **Step 1:** Set related parameters, including the size of swarm N, the maximum evaluation time FE_MAX and so on;
2. **Step 2:** Initialize the swarm according to the method in Section III-D (2);
3. **Step 3:** Calculate the fitness of each particle in the swarm, and update its *Pbest* and *Gbest*;
4. **Step 4:** Update all the particles according to Eq. (15);
5. **Step 5:** Perform the adaptive disturbance based on Eq. (16);
6. **Step 6:** If satisfy the stop condition, stop the algorithm and output the optimal solution; otherwise, go to Step 3.

1) *Particle Encoding Scheme:* For the integer problem descried in (13), the position of a particle is directly defined as $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,M}),\ x_{i,j} \in \{1, 2, \ldots, |\text{Cluster}^j|\}$. Fig. 2 shows an illustration to show the encoding and decoding of a particle.

2) *Relevance-Guided Swarm Initialization:* A good initial swarm can improve the convergence of an algorithm. As Definitions 1 and 2 show, irrelevant features have usually no/weak correlation with the class labels C. This means that the higher the C-relevance value of a feature, the greater the probability of the feature being selected into an initial particle should be. Hence, this article proposes a relevance-guided swarm initialization operator. Suppose that the highest C-relevance value obtained by the jth cluster is $Cv_j^{\max}$, M, and $Cv = \max(Cv_j^{\max} |\ j =$

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

$1, 2, \ldots, M$), the probability with which a feature is selected from the jth cluster is as follows:

$$pc_j = \frac{Cv_j^{\max}}{Cv},\ j = 1, 2, \ldots, M. \tag{14}$$

Taking the particle $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,M})$ as an example, the proposed initialization method can be depicted as follows: for the jth element of this particle, if a random number within [0, 1] is smaller than its corresponding probability pc_j, then randomly select a feature from the jth cluster and set its index as the value of this element. The above process is repeated until a number of particles are generated. Algorithm 3 shows the pseudocode of the proposed initialization operator.

3) *Integer Updating Rules:* Since the position of each particle is an integer vector, the updating rules in continuous BPSO [35] fit no longer. Therefore, we redefine them as follows:

$$x_{i,j} = \begin{cases} \left\lceil \frac{pb_{i,j}+gb_{i,j}}{2} \right\rceil + \lceil G(0,1) \times |pb_{i,j} - gb_{i,j}| \rceil, & \text{rand} > 0.5 \\ pb_{i,j}, & \text{otherwise} \end{cases} \tag{15}$$

where $G(0, 1)$ is a Gaussian distribution and $\lceil \cdot \rceil$ represents the upper rounding function.

4) *Difference-Based Adaptive Disturbance:* It is observed from (15) that if the *Pbest* of a particle happens to be close or equal to *Gbest* in the evolution process, obviously, this particle will stop updating at once because the value of $\lceil G(0, 1) \times |pb_{i,j} - gb_{i,j}| \rceil$ becomes 0. To improve the diversity of the swarm and help it to escape from local optima, we introduce an adaptive disturbance, called the difference mutation (DM). Checking each element of the particle $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,M})$ in turn, if a random number rand$' \in [0, 1]$ is smaller than the mutation probability pm, DM will reinitialize this element within its corresponding search space. To enhance the exploitation capability of the swarm, DM uses the difference between *Pbest* and *Gbest* to adjust dynamically the probability pm as follows:

$$pm(X_i) = 0.1 \times \frac{\sum_{j=1}^{M} eq\left(pb_{i,j}, gb_{i,j}\right)}{M} \tag{16}$$

where the function $eq(a, b)$ returns the value of 1 when a and b are equal; otherwise, returns the value of 0. By employing the above adaptive probability, each particle will obtain its own disturbance value based on the difference between its two leaders (*Pbest* and *Gbest*).

5) *Steps of IBPSO:* The steps of IBPSO are given in Algorithm 4. In step 3, we update *Pbest* and *Gbest* of each particle based on the fitness (i.e., classification accuracy) and the number of selected features. Especially, for a new position of a particle, if its fitness value is larger than that of the current *Pbest*, set the new position to be *Pbest*; if the new position and the current *Pbest* have the same fitness value, but the new position includes fewer features, set the new position to be *Pbest*. If a new *Pbest* is better than the current *Gbest*, *Gbest* is replaced by the new *Pbest*; if the new *Pbest* and the current *Gbest* have the same fitness value, but *Pbest* includes fewer features, *Gbest* is also replaced by the new *Pbest*.

TABLE I
BASIC INFORMATION OF THE 18 DATASETS

Datasets	# of features	# of samples	# of classes
arrhythmia	195	452	16
SCADI	205	70	7
GFE	301	743	2
Prostate	339	102	2
MFD	649	700	10
COIL20	1024	200	20
Yale_64	1024	165	15
Colon	2000	62	2
SRBCT	2308	83	4
WrapAR10P	2400	130	10
leukemia_small	3571	72	2
DBWorld	4703	64	2
DLBCL	5469	77	2
Driv_face	6400	606	3
leukemia_big	7128	72	2
CNS	7129	60	2
Lung	12600	203	2
Ovarian	15154	253	2

[1] GFE and MFD are the abbreviations of grammatical_facial_expression01 and MultipleFeaturesDigit, respectively.

E. Further Discussion

Since it is difficult to distinguish the feature that is actually "valid and good" but weakly correlated by using the limited size of training data, the first phase of the proposed algorithm does delete some good features that include useful information. First, to avoid deleting good features by mistake, (9) introduces a new method to determine the deletion threshold.

We use two indicators, that is: 1) the improvement degree in the classification accuracy (ID-AC) and 2) the ratio of wrongly deleted features to all deleted features (W-All), to evaluate the effectiveness of the proposed operator. Table A1 in Appendix A in the supplementary material lists the average ID-AC and W-All values obtained by readding those deleted features into optimal feature subsets for the 11 datasets. We can see that the influence of reselected features on the performance of the optimal solution is not very clear. Second, the number of those correctly deleted features often is far more than that of wrongly deleted features. Moreover, for all the nine datasets except arrhythmia and GFE, their average W-All values are smaller than 10%. Since the first and second phases of HFS-C-P significantly reduce the feature space, the processing efficiency of the third phase is improved significantly, making our proposed algorithm achieve a good result finally.

IV. EXPERIMENTS AND ANALYSIS

A. Datasets

We carry out the compassion experiments on 18 real datasets [8], [54], [59], which can be download from http://featureselection.asu.edu/datasets.php, http://portals.broadinstitute.org/cgi-bin/cancer/datasets.cgi, and http://archive.ics.uci.edu/ml/. Table I lists the basic information of these datasets.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE II
PARAMETER SETTINGS

Algorithms	Parameter settings
ReliefF [60]	The nearest neighbor number k=10.
BPSO [4]	The acceleration coefficients $c_1 = c_2 = 2$, the inertia weight w=1.
BBPSO [23]	The acceleration coefficients $c_1 = c_2 = 2$, the inertia weight w=1.
Rc-BBFA [41]	The jump probability $\sigma = 0.1 - 0.09 * (t/T)$.
SaPSO [59]	The selection probability of the j-th strategy p_j=0.2, the number of particles p_s=100, the empirically LP=10, the threshold θ=0.6.
HPSO-SSM [46]	The jump probability p=0.8.
CBFS [49]	Cluster number $K \in [2,\ 0.05 * D]$.
BPSO+filter [53]	The percentage of features at the top of ranking k=[0.1,0.2], the acceleration coefficients c_1=c_2=0.5, the inertia weight w=0.9.
HI-BQPSO [54]	The learning probability $\eta_i = 0.5$, the coefficient $\beta = (0.8 - 0.6) * \frac{T_{\max} - t}{T_{\max}} + 0.6$.

TABLE III
Ac AND t^* (UNIT: MIN) VALUES OBTAINED BY HFS-C-P, HFS-C-P-K, AND HFS-C-P-F (SVM)

Algorithms	arrhythmia		SCADI		GFE		Prostate		MFD		COIL20	
	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*
HFS-C-P-K	61.23	**16.944**	83.49	6.558	82.67	**12.173**	92.24	5.678	81.51	**23.777**	**100.00**	**80.416**
HFS-C-P-F	67.45	25.324	88.90	4.783	82.81	46.315	95.74	4.847	99.19	156.436	99.38	655.118
HFS-C-P	**67.68**	31.723	**89.68**	**3.314**	**85.13**	43.571	**97.49**	**3.550**	**99.40**	94.522	**100.00**	237.147
Algorithms	Yale_64		Colon		SRBCT		WrapAR10P		leukemia_small		DBWorld	
	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*
HFS-C-P-K	73.71	**13.460**	92.11	6.243	**100.00**	10.113	**100.00**	21.599	**100.00**	24.274	90.95	37.646
HFS-C-P-F	75.86	88.298	89.17	32.665	98.57	29.995	92.11	48.850	**100.00**	52.157	97.39	12.232
HFS-C-P	**79.52**	35.169	**92.47**	**5.974**	**100.00**	**8.716**	**100.00**	**13.104**	**100.00**	**5.274**	**97.57**	**4.571**
Algorithms	DLBCL		Driv_face		leukemia_ big		CNS		Lung		Ovarian	
	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*	$Ac(\%)$	t^*
HFS-C-P-K	91.25	56.948	97.92	631.918	96.03	261.233	77.48	229.377	88.20	2588.515	**100.00**	4645.440
HFS-C-P-F	98.58	71.841	94.46	2148.358	99.33	160.297	81.03	184.860	95.34	1567.898	**100.00**	48.254
HFS-C-P	**100.00**	**5.821**	**98.23**	**72.400**	**100.00**	**6.119**	**85.91**	**7.400**	**98.01**	**26.316**	**100.00**	**9.826**

B. Comparison Algorithms and Parameter Settings

We choose one filter algorithm, four evolutionary FS algorithms, one clustering-based FS algorithm, and three hybrid PSO algorithms as comparison algorithms. Furthermore, we also compare the proposed HFS-C-P with the full set, with the purpose of evaluating the performance of HFS-C-P. The nine comparison algorithms for FS are described as follows.

1) ReliefF FS (ReliefF) [60].
2) Binary PSO FS (BPSO) [4].
3) Binary bare bones PSO FS (BBPSO) [23].
4) Return-cost-based binary firefly FS (Rc-BBFA) [41].
5) Self-adaptive PSO for large-scale FS (SaPSO) [59].
6) Hybrid PSO with spiral-shaped mechanism (HPSO-SSM) [46].
7) Clustering-based FS framework (CBFS) [49].
8) Hybrid filter–wrapper FS method (BPSO+filter) [53].
9) Hybrid binary quantum PSO (HI-BQPSO) [54].

All experiments in this article are carried out on Intel core i7-8700 CPU, 3.2 GHz, 16.00-GB RAM. The maximum evaluation time *FE_MAX* is set to 5000 for all evolutionary algorithms and the running times of each algorithm is 30. The remaining parameters of these comparison algorithms are set according to suggestions of corresponding literature, as shown in Table II.

Since the proposed algorithm and the seven evolutionary FS algorithms belong to the wrapper, a classifier is needed to evaluate the fitness of individuals. There are many effective classifiers, such as support vector machines (SVM), *K*-nearest neighbor (*K*-NN), neural networks, and C4.5-based decision trees. SVM is a popular supervised learning algorithm, which has been applied to many wrapper methods. Based on this, we choose SVM as the classifier in the experiment. This article employs a ten-fold cross-validation method to demonstrate the effectiveness of all algorithms. The entire samples of a dataset are randomly divided into ten categories for training and testing. Since the employed classifier may influence the selection of features, we also adopt the *K*-NN to verify the performance of the proposed algorithm, instead of SVM.

C. Analyses of the Initial Swarm Operator

This experiment demonstrates the influence of the proposed relevance-guided swarm initialization on the performance of HFS-C-P. The HFS-C-P algorithm without the initialization is selected as a comparison algorithm, denoted by HFS-C-P-R. Fig. A1 in Appendix B in the supplementary material shows the average classification accuracy obtained by HFS-C-P and HFS-C-P-R. It reports that HFS-C-P achieves the highest average *Ac* values for all the 18 datasets with the help of the relevance-guided swarm initialization. In addition, since the first and second phases of HFS-C-P have achieved better initial feature subsets for COIL20, SRBCT, and Ovarian, HFS-C-P has obtained 100% classification accuracy before using PSO.

D. Analyses of the FCFC Clustering Strategy

This experiment analyzes the effectiveness of the proposed clustering strategy, FCFC. Here, the two clustering strategies, that is: 1) *K*-means and 2) FAST proposed in [8], are selected as the comparison methods. We denote HFS-C-P by *K*-means as HFS-C-P-K and denote HFS-C-P by FAST as HFS-C-P-F, respectively.

Table III lists the average values of *Ac* and t^* obtained by HFS-C-P, HFS-C-P-K, and HFS-C-P-F, where *Ac* and t^* represent the average classification accuracy and the average running time, respectively. It can be seen that: 1) for all the datasets, HFS-C-P achieves the highest *Ac* values with the help

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE IV
Ac VALUES OBTAINED BY HFS-C-P AND THE NINE COMPARISON ALGORITHMS (SVM)

Datasets	Full set	ReliefF	BPSO	BBPSO	Rc-BBFA	SaPSO	HPSO-SSM	CBFS	BPSO+filter	HI-BQPSO	HFS-C-P
arrhythmia	36.87(+)	55.91(+)	63.12(+)	67.38(=)	67.47(=)	59.29(+)	64.38(+)	52.55(+)	59.80(+)	66.91(+)	**67.68**
SCADI	80.29(+)	83.44(+)	87.29(+)	88.26(+)	88.43(+)	83.45(+)	86.82(+)	82.56(+)	83.71(+)	88.92(+)	**89.68**
GFE	68.45(+)	79.76(+)	74.69(+)	75.67(+)	75.09(+)	73.68(+)	78.94(+)	66.92(+)	82.57(+)	83.82(+)	**85.13**
Prostate	66.83(+)	78.75(+)	93.02(+)	95.98(+)	95.96(+)	88.22(+)	91.48(+)	88.65(+)	89.83(+)	95.59(+)	**97.49**
MFD	36.52(+)	98.31(+)	99.06(+)	99.23(=)	99.26(+)	98.56(+)	98.71(+)	98.63(+)	95.91(+)	98.07(+)	**99.40**
COIL20	**100.00**(=)	**100.00**(=)	**100.00**(=)	**100.00**(=)	**100.00**(=)	**100.00**(=)	**100.00**(=)	**100.00**(=)	93.58(+)	96.74(+)	**100.00**
Yale_64	71.59(+)	72.44(+)	77.49(+)	78.05(+)	78.10(+)	75.07(+)	75.24(+)	74.43(+)	67.06(+)	71.73(+)	**79.52**
Colon	72.46(+)	76.17(+)	90.52(+)	90.79(+)	90.70(+)	83.04(+)	87.40(+)	77.14(+)	88.11(+)	89.73(+)	**92.47**
SRBCT	42.77(+)	96.10(+)	94.94(+)	96.03(+)	96.18(+)	98.93(+)	99.01(=)	98.29(+)	97.73(+)	98.78(+)	**100.00**
WrapAR10P	53.22(+)	91.47(+)	86.43(+)	88.24(+)	86.90(+)	93.05(+)	95.56(+)	88.22(+)	89.31(+)	92.33(+)	**100.00**
leukemia_small	53.22(+)	94.36(+)	96.21(+)	97.67(+)	97.95(+)	99.09(+)	99.09(=)	97.26(+)	99.62(=)	100.00(=)	**100.00**
DBWorld	81.31(+)	92.31(+)	90.44(+)	92.40(+)	92.60(+)	90.63(+)	90.44(+)	90.19(+)	93.71(+)	96.38(+)	**97.57**
DLBCL	75.57(+)	88.55(+)	82.46(+)	83.16(+)	82.57(+)	91.54(+)	95.85(+)	93.54(+)	97.78(+)	99.29(+)	**100.00**
Driv_face	91.79(+)	97.46(+)	97.89(+)	98.14(=)	97.86(+)	97.64(+)	97.79(+)	97.04(+)	96.17(+)	94.31(+)	**98.23**
leukemia_ big	67.13(+)	88.72(+)	74.40(+)	75.02(+)	74.15(+)	94.89(+)	97.69(+)	94.31(+)	100.00(=)	100.00(=)	**100.00**
CNS	65.72(+)	64.27(+)	76.23(+)	74.77(+)	74.55(+)	71.87(+)	72.48(+)	64.03(+)	83.69(+)	80.88(+)	**85.91**
Lung	68.81(+)	48.28(+)	53.28(+)	60.55(+)	55.36(+)	78.93(+)	93.71(+)	83.08(+)	87.02(+)	96.67(+)	**98.01**
Ovarian	64.96(+)	96.59(+)	89.70(+)	94.97(+)	90.09(+)	97.91(+)	99.43(=)	96.74(+)	99.55(=)	100.00(=)	**100.00**

of the proposed clustering strategy and 2) for 13 out of the 18 datasets, HFS-C-P also shows the shortest running time t^* with the help of the proposed clustering strategy. Although the proposed clustering strategy spends higher running time than the K-means for the remaining datasets with fewer features, it markedly improves the *Ac* values of HFS-C-P. Therefore, FCFC is effective to improve the performance of HFS-C-P.

E. Comparison Results

1) Classification Results With SVM: In this section, the proposed HFS-C-P algorithm is compared with the nine comparison algorithms in terms of classification accuracy. In addition, we employ the Mann–Whitney U test to investigate whether there is a significant difference between HFS-C-P and compared algorithms. Table IV lists the average *Ac* values obtained by the nine FS algorithms with SVM, where "+" indicates that HFS-C-P is obviously superior to the comparison algorithm and "=" indicates that there is no significant difference between them.

According to the suggestion of literature [61], ReliefF selects the best {1/5*D, 2/5*D, 3/5*D, 4/5*D} features from the sorted original features in turn and takes the best *Ac* value obtained as the final output. CBFS also uses the same method to determine the number of selected features. CBFS also uses the same method to determine the number of selected features. From Table IV, it can be seen that except for the datasets, COIL20 and Lung, the *Ac* values of SVM are improved by the ten FS algorithms compared with the original Full set. Compared with the nine comparison algorithms, the proposed algorithm obtains the best *Ac* values for all the 18 datasets. The HFS-C-P achieves 100% classification accuracy on seven out of the 18 datasets (i.e., COIL20, SRBCT, WrapAR10P, leukemia_small, DLBCL, leukemia_big, and Ovarian) and shows better performance on all datasets. Moreover, it reports from their Mann–Whitney U results.

1) For all the datasets except COIL20, the *Ac* values of HFS-C-P are significantly larger than the Full set, ReliefF, BPSO, SaPSO, and CBFS.
2) HFS-C-P is significantly better than BBPSO for all the datasets except arrhythmia, MFD, COIL20, and Driv_face.
3) HFS-C-P is significantly better than Rc-BBFA for all the datasets except arrhythmia and COIL20.
4) HFS-C-P is significantly better than HPSO-SSM for all the datasets except COIL20, SRBCT, leukemia_small, and Ovarian.
5) HFS-C-P is significantly better than BPSO+filter for all the datasets except leukemia_small, and leukemia_big.
6) HFS-C-P is significantly better than HI-BQPSO on all the datasets except leukemia_small, leukemia_big, and Ovarian.
7) For dataset COIL20, there is no significant difference between all the algorithms except BPSO+filter and HI-BQPSO in terms of *Ac*. Overall, compared with the nine comparison algorithms, HFS-C-P is highly competitive for most of the datasets.

Due to the global search capability of PSO in HFS-C-P, HFS-C-P has better performance than the two filter algorithms: 1) ReliefF and 2) CBFS. Compared with BPSO, BBPSO, Rc-BBFA, SaPSO, and HPSO-SSM, the three hybrid algorithms (i.e., HFS-C-P, BPSO+filter, and HI-BQPSO) all show relatively good *Ac* values. This indicates the superiority of using the filter–wrapper hybrid way. Since HFS-C-P can obviously reduce the search space of the swarm by the proposed clustering technology, it shows also better performance than BPSO+filter and HI-BQPSO on most of the datasets.

Moreover, Table V lists the average number of selected features (d^*) by the nine comparison algorithms. It can be seen that: 1) for most of the datasets, the three hybrid algorithms: a) BPSO+filter; b) HI-BQPSO; and c) HFS-C-P, achieve less d^* values and 2) for most of the datasets, HFS-C-P obtains the third smallest d^* values, which are more than that of BPSO+filter and HI-BQPSO. The main reasons are as follows: 1) in BPSO+filter, 80%–90% features have been deleted by two different filter methods in the first phase and 2) in HI-BQPSO, only the first 100 important features were retained in the filter phase for all datasets. However, since many useful features were deleted directly, their classification accuracies are obviously less than HFS-C-P on most of the datasets. The proposed HFS-C-P algorithm can achieve a higher *Ac* value with a relatively small d^* value on most of the datasets, further indicating the superiority of the proposed algorithm.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE V
d^* VALUES OBTAINED BY HFS-C-P AND THE EIGHT COMPARISON ALGORITHMS (SVM)

Datasets	ReliefF	BPSO	BBPSO	Rc-BBFA	SaPSO	HPSO-SSM	CBFS	BPSO+filter	HI-BQPSO	HFS-C-P
arrhythmia	39.0	91.8	80.4	85.2	75.4	45.2	39.0	**8.0**	15.0	70.6
SCADI	41.0	99.2	91.0	95.2	83.6	32.2	123.0	**2.8**	7.2	18.0
GFE	121.0	146.4	122.8	146.6	123.4	**8.4**	121.0	9.3	11.4	54.2
Prostate	136.0	170.2	161.8	156.0	136.2	88.2	68.0	14.0	**12.4**	23.6
MFD	390.0	325.0	314.0	312.4	553.4	231.4	260.0	28.0	**15.6**	138.3
COIL20	205.0	424.6	267.2	418.2	418.4	109.4	615.0	**38.0**	25.4	80.6
Yale_64	410.0	509.6	507.0	507.2	505.8	450.0	205.0	54.0	**32.0**	357.4
Colon	400.0	977.0	805.2	1002.0	796.4	591.6	400.0	30.6	**29.4**	155.0
SRBCT	462.0	112.2	1099.0	1130.8	915.8	357.6	462.0	**19.6**	26.4	99.4
WrapAR10P	480.0	1174.6	1171.4	1172.4	949.4	186.6	480	82.4	**29.0**	204.6
leukemia_small	1429.0	1753.0	1749.0	1714.3	1447.2	985.2	715.0	23.4	**13.6**	74.1
DBWorld	1881.0	2445.0	2325.4	2337.4	1865.4	2332.2	941.0	30.2	**11.0**	81.4
DLBCL	1094.0	2717.2	2732.4	2725.0	2196.0	589.8	1094.0	36.2	**12.8**	227.0
Driv_face	1280.0	3184.8	2670.8	3178.0	2558.0	190.0	2560.0	211.0	**19.0**	773.7
leukemia_big	1426.0	3588.5	2924.3	3547.5	2831.8	655.2	1426.0	**48.2**	59.0	152.7
CNS	2852.0	3564.7	3542.3	3613.0	2843.4	1941.8	2852.0	622.0	**29.2**	367.6
Lung	2520.0	6295.4	5155.2	6295.8	5054.4	504.2	2520.0	275.0	**19.6**	522.4
Ovarian	3031.0	7577.0	6114.6	7495.4	6066.0	409.8	3031.0	277.0	**7.0**	42.5

TABLE VI
RUNNING TIME CONSUMED BY THE EIGHT EVOLUTIONARY FS ALGORITHMS ON THE 18 DATASETS (UNIT: MIN) (SVM)

Datasets	BPSO	BBPSO	Rc-BBFA	SaPSO	HPSO-SSM	BPSO+filter	HI-BQPSO	HFS-C-P
arrhythmia	45.261	37.678	50.230	32.004	36.599	**12.385**	29.060	31.723
SCADI	4.257	4.102	4.133	3.777	3.866	4.852	3.592	**3.314**
GFE	63.442	54.760	61.710	48.897	45.235	**13.198**	24.227	43.571
Prostate	4.296	3.772	4.658	3.587	4.239	3.224	3.819	**3.081**
MFD	184.512	182.960	178.706	140.214	146.563	**15.316**	22.124	94.522
COIL20	872.028	578.876	833.159	368.854	585.545	**55.616**	80.580	237.147
Yale_64	36.384	28.882	35.705	26.080	27.265	**7.808**	9.414	22.693
Colon	14.760	10.184	14.766	12.231	11.238	3.311	**2.936**	5.974
SRBCT	18.049	15.014	24.462	15.801	11.948	6.993	6.817	**6.670**
WrapAR10P	40.483	36.485	46.341	27.048	29.617	9.066	**5.574**	12.494
leukemia_small	25.512	22.128	24.864	15.278	19.837	5.952	5.949	**5.160**
DBWorld	14.079	11.731	15.523	10.604	13.443	4.977	4.798	**4.644**
DLBCL	56.455	43.554	55.872	34.877	43.075	4.461	**2.950**	6.766
Driv_face	1576.630	1164.914	1082.781	886.338	1050.056	73.195	**11.973**	72.400
leukemia_ big	67.845	30.059	57.388	38.984	42.902	9.512	8.753	**6.278**
CNS	46.169	24.043	48.535	34.408	34.679	4.416	**3.147**	7.874
Lung	564.316	405.192	564.050	343.779	357.687	48.521	**4.969**	16.595
Ovarian	1205.176	961.590	996.491	502.948	669.449	48.609	9.359	**7.859**

2) Classification Results With K-NN: In order to analyze the influence of a classifier on the performance of the proposed algorithm, K-NN is also selected as the classifier in this experiment, where $K = 3$. Table A2 of Appendix C in the supplementary material lists the Ac values of the ten algorithms with K-NN. It can be seen that: 1) for all the 18 datasets, the proposed algorithm obtains the best average Ac values and 2) for most of the datasets, the performance of HFS-C-P is significantly better than that of all the nine comparison algorithms, as shown in their Mann–Whitney U results. Taking HPSO-SSM as an example, its performance is significantly worse than that of the proposed HFS-C-P for all the datasets except MFD.

3) Running Time: This section compares the running time of the proposed algorithm with the seven evolutionary FS algorithms, with the purpose of evaluating the relationship between the performance improvement and the computation cost. To be fair, all experiments set the same maximal evaluation times, *FE_MAX*.

Table VI lists the average CPU time consumed by each algorithm on 18 datasets. It can be seen that:

1) for a dataset with more features or samples, the running time of the eight algorithms is relatively large. As we know, when processing high-dimensional data, the computational cost of an evolutionary FS algorithm is mainly consumed by the process of evaluating individuals. The computational cost of using a classifier to evaluate an individual depends mainly on the sizes of both selected features and samples;
2) for all the datasets, the three hybrid algorithms: a) BPSO+filter; b) HI-BQPSO; and c) HFS-C-P, have the shortest average CPU time. This is mainly attributed to the fact that BPSO+filter and HI-BQPSO delete most of the features in the filter phase, and HFS-C-P also reduces the search space by the clustering strategy. As mentioned above, the number of selected features directly determines the computational cost of an algorithm.

F. Further Analysis

1) Analyses of Two Key Parameters ρ_0 and ρ_1: This section discusses the influence of ρ_0 and ρ_1 on the performance of the proposed algorithm. First, the proposed method in (9) is verified by comparing with the traditional fixed-value method and the method proposed in [57]. In the traditional fixed-value method, the values of ρ_0 are set to be $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ in turn. In [57], $\rho_0 = SU_{\lfloor D/\log D \rfloor - \text{th}}$. Second, the method of ρ_1 proposed in (10) is compared with the traditional fixed-value method, where the value of ρ_1 is set to be $\{0.01, 0.02, 0.05, 0.1, 0.5\}$ in turn.

Appendix E in the supplementary material shows their corresponding results. It can be seen that: 1) with the help of

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE VII
Ac VALUES OBTAINED BY HFS-C-P, HFS-C-P/1, HFS-C-P/2, AND HFS-C-P/3

Datasets	arrhythmia	SCADI	GFE	Prostate	MFD	COIL20	Yale_64	Colon	SRBCT
HFS-C-P/1	66.56	88.64	84.49	97.31	99.11	**100.00**	79.48	92.44	**100.00**
HFS-C-P/2	67.17	88.31	78.53	95.03	99.14	**100.00**	76.70	89.69	99.92
HFS-C-P/3	50.10	77.57	67.68	76.20	98.06	**100.00**	73.28	78.73	96.13
HFS-C-P	**67.68**	**89.68**	**85.13**	**97.49**	**99.40**	**100.00**	**79.52**	**92.47**	**100.00**
Datasets	WrapAR10P	leukemia_small	DBWorld	DLBCL	Driv_face	leukemia_big	CNS	Lung	Ovarian
HFS-C-P/1	**100.00**	**100.00**	97.23	**100.00**	98.19	**100.00**	85.29	97.46	**100.00**
HFS-C-P/2	98.47	99.49	97.21	98.77	98.06	99.73	77.21	95.94	**100.00**
HFS-C-P/3	92.33	93.15	90.19	89.54	96.87	96.12	65.12	83.33	99.13
HFS-C-P	**100.00**	**100.00**	**97.57**	**100.00**	**98.23**	**100.00**	**85.91**	**98.01**	**100.00**

TABLE VIII
Ac VALUES OBTAINED BY ALL THE ALGORITHMS WITHOUT THE FS BIAS (SVM)

Datasets	ReliefF	BPSO	BBPSO	Rc-BBFA	SaPSO	HPSO-SSM	CBFS	BPSO+filter	HI-BQBPSO	HFS-C-P
arrhythmia	64.18(+)	66.34(+)	55.60(+)	66.83(=)	63.73(+)	66.49(=)	45.52(+)	44.18(+)	65.07(+)	**67.23**
SCADI	79.80(+)	80.30(+)	79.09(+)	80.68(+)	80.00(+)	80.45(+)	77.27(+)	76.77(+)	79.50(+)	**81.82**
(+) GFE	78.92(+)	80.81(+)	82.06(=)	80.54(+)	81.97(=)	77.58(+)	81.62(+)	77.23(+)	78.03(+)	**82.42**
Prostate	90.32(+)	89.35(+)	89.03(+)	90.97(+)	80.00(+)	92.26(+)	92.18(+)	84.52(+)	83.87(+)	**93.55**
MFD	96.65(+)	97.75(+)	97.37(+)	97.88(=)	97.93(=)	97.85(=)	97.37(+)	94.07(+)	96.94(+)	**98.06**
COIL20	**100.00**(=)	**100.00**(=)	**100.00**(=)	**100.00**(=)	**100.00**(=)	**100.00**(=)	**100.00**(=)	96.52(+)	96.36(+)	**100.00**
Yale_64	73.33(+)	74.55(+)	73.78(+)	73.33(+)	74.22(+)	73.78(+)	72.44(+)	61.33(+)	67.11(+)	**74.89**
Colon	73.68(+)	76.32(+)	75.79(+)	79.47(+)	77.37(+)	77.89(+)	73.68(+)	78.95(+)	77.89(+)	**81.05**
SRBCT	93.25(+)	93.06(+)	93.45(+)	93.75(=)	93.33(+)	93.94(=)	93.06(+)	94.05(=)	91.67(+)	**94.17**
WrapAR10P	82.50(+)	76.50(+)	81.50(+)	76.00(+)	78.50(+)	88.50(+)	85.50(+)	85.00(+)	91.50(+)	**94.75**
leukemia_small	90.48(+)	93.65(+)	94.08(=)	93.88(+)	**95.23**(-)	94.44(=)	92.38(+)	94.40(=)	92.38(+)	94.74
DBWorld	89.47(+)	89.47(+)	89.97(=)	90.30(=)	**92.11**(-)	89.55(=)	86.32(+)	89.47(+)	84.21(+)	90.58
DLBCL	78.26(+)	73.91(+)	89.57(+)	73.97(+)	75.65(+)	92.05(+)	86.96(+)	86.09(+)	85.22(+)	**92.61**
Driv_face	96.15(+)	96.31(+)	96.53(=)	96.38(+)	96.64(=)	96.52(=)	95.65(+)	95.24(+)	95.05(+)	**96.87**
leukemia_ big	90.48(+)	70.48(+)	92.52(+)	67.07(+)	79.05(+)	**94.05**(-)	80.00(+)	93.06(=)	91.90(+)	93.33
CNS	66.67(+)	66.67(+)	67.01(+)	66.67(+)	66.67(+)	66.67(+)	63.76(+)	62.22(+)	64.44(+)	**67.22**
Lung	96.72(+)	93.77(+)	97.21(+)	93.85(+)	95.08(+)	96.95(+)	95.08(+)	93.11(+)	93.93(+)	**97.70**
Ovarian	97.37(+)	96.32(+)	97.89(=)	94.21(+)	97.37(+)	98.03(=)	97.37(+)	97.56(+)	97.86(=)	**98.68**

the proposed method in (9), the proposed algorithm obtains the best average *Ac* values on 13 datasets and 2) with the help of the proposed method in (10), the proposed algorithm obtains the best average *Ac* values on 12 datasets. Overall, the proposed setting methods on the values of ρ_0 and ρ_1 are beneficial for improving the performance of the proposed algorithm on most of the datasets.

2) Contribution Analysis for Each Component in the Algorithm: This section analyzes the effectiveness of each component in the proposed three-phase FS algorithm. HFS-C-P/1, HFS-C-P/2, and HFS-C-P/3 represent the algorithms after removing the first phase, second phase, and third phase from HFS-C-P, respectively. Table VII shows *Ac* values obtained by HFS-C-P, HFS-C-P/1, HFS-C-P/2, and HFS-C-P/3. We can see that comparing with HFS-C-P/1, HFS-C-P/2, and HFS-C-P/3, HFS-C-P obtains the best *Ac* value for all the datasets. This indicates the importance of the three phases on improving the performance of HFS-C-P.

3) Analyses of the FS Bias: As mentioned in [62], using the same training data for FS and classification learning can eventuate the FS bias. The key issue for FS bias is there are no unseen data for the FS process, that is, the entire set of data is used during the FS process. This bias can exacerbate data overfitting and negatively affect classification performance [63]. This experiment is designed to analyze the FS bias of the proposed algorithm. Here, a dataset is divided into two independent parts before implementing FS, of which 70% is used as the training set and 30% is used as the testing set. Like [55], all the comparison algorithms are performed on the same training dataset and evaluated on the same testing data. In the evolutionary process of the eight population-based FS algorithms, cross-validation is used to evaluate the fitness of new particles or individuals.

Table VIII shows the average *Ac* values of all the ten algorithms without the FS bias, and Table A5 in the supplementary material lists the d^* values obtained by all the algorithms. It can be seen that:

1) for eight out of the 16 datasets, the *Ac* values of HFS-C-P are significantly better than that of all the nine comparison algorithms. For all the datasets except leukemia_small, DBWorld, and leukemia_big, the proposed algorithm obtains the best *Ac* values, which are 67.23%, 81.82%, 82.42%, 93.55%, 98.06%, 100.00%, 74.89%, 81.05%, 94.17%, 94.75%, 92.61%, 96.87%, 67.22%, 97.70%, and 98.68%, respectively. For the remaining three datasets, that is: a) leukemia_small; b) DBWorld; and c) leukemia_big, the proposed algorithm achieves the second-best *Ac* value. The possible reason is that some features are clustered into incorrect clusters in the first phase of HFS-C-P, because the number of samples in the three datasets is relatively small;
2) for 15 out of the 18 datasets, the proposed algorithm achieves the second or third smallest d^* values. For GFE and Yale_64, the proposed algorithm achieves the fourth-smallest d^* values. Since a large number of features are deleted in the filter phase, BPSO+filter or HI-BQPSO achieves the smallest d^* values on most of the datasets, but their *Ac* values are obviously less than that of HFS-C-P. A possible reason is that those features deleted in the filter phase may contain a lot of useful information. In summary, HFS-C-P still is a highly

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

competitive FS algorithm when considering the FS bias, compared with the nine comparison algorithms.

V. Conclusion

This article aimed to address the high-dimensional FS problem, which is still challenging due to the "curse of dimensionality" and/or the high computational complexity. It becomes critical and difficult to solve the problem of the "curse of dimensionality" by using a relatively low computational cost. This article proposed a new hybrid FS algorithm, called HFS-C-P. HFS-C-P developed the three-phase hybrid framework for the first time, which effectively integrates the advantages of three kinds of FS algorithms. With the help of the first and second phases, the search space of the PSO-based method in the third phase is reduced obviously, which greatly improves the search speed of the swarm. Second, the FCFC proposed in the second phase needs only to compare the similarity between each feature and the known cluster centers, which not only reduces obviously the computation cost of clustering features but also does not need to specify the clustering number. Moreover, both the relevance-guided swarm initialization and the difference-based adaptive disturbance make the proposed integer BPSO algorithm in the third phase more effective in solving FS problems.

We have compared the performance of the proposed algorithm with the nine well-known algorithms, including: 1) ReliefF; 2) BPSO; 3) Binary BPSO; 4) Rc-BBFA; 5) SaPSO; 6) HPSO-SSM; 7) BPSO+filter; 8) HI-BQPSO; and 9) CBFS, on the 18 publicly available real-world datasets. The experimental results showed that the proposed algorithm can achieve the best classification accuracy with relatively less running time on most of the datasets, indicating that it has strong competitiveness for solving high-dimensional FS problems.

Despite the promising results, there are still some limitations in the work. First, the feature clustering method plays an important role in HFS-C-P. More sophisticated clustering methods that do not need to set any threshold manually need to be designed. It may be a feasible way to adjust the threshold automatically based on the feedback result of the algorithm. Second, for data with a large number of samples, the proposed algorithm still faces the problem of highly computational cost. Reducing the calculation cost by new technologies, such as a surrogate model constructed by representative samples, needs to be discussed in the future. In addition, the proposed framework can be applied to multiobjective or multilabel FS problems, and specific algorithms should be studied further.

References

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

[2] M. Liang and X. Hu, "Feature selection in supervised saliency prediction," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 914–926, May 2015.

[3] W. Ding, C.-T. Lin, and W. Pedrycz, "Multiple relevant feature ensemble selection based on multilayer co-evolutionary consensus MapReduce," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 425–439, Feb. 2020.

[4] L. Y. Chuang, C.-S. Yang, K.-C. Wu, and C.-H. Yang, "Gene selection and classification using Taguchi chaotic binary particle swarm optimization," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13367–13377, Sep. 2011.

[5] J. Xu, B. Tang, H. He, and H. Man, "Semisupervised feature selection based on relevance and redundancy criteria," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 9, pp. 1974–1984, Sep. 2017.

[6] H. Liu, H. Motoda, and L. Yu, "Selective sampling approach to active feature selection," *Artif. Intell.*, vol. 159, nos. 1–2, pp. 49–72, Nov. 2004.

[7] P. P. Kundu and S. Mitra, "Feature selection through message passing," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4356–4366, Dec. 2017.

[8] Q. Song, J. Ni, and G. Wang, "A fast clustering-based feature subset selection algorithm for high-dimensional data," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 1, pp. 1–14, Jan. 2013.

[9] C. Lazar *et al.*, "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 9, no. 4, pp. 1106–1119, Jul./Aug. 2012.

[10] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Trans. Comput.*, vol. C-20, no. 9, pp. 1100–1103, Sep. 1971.

[11] T. Marill and D. M. Green, "On the effectiveness of receptors in recognition systems," *IEEE Trans. Inf. Theory*, vol. IT-9, no. 1, pp. 11–17, Jan. 1963.

[12] S. D. Stearns, "On selecting features for pattern classifiers," in *Proc. 3rd Int. Joint Conf. Pattern Recognit.*, 1976, pp. 71–75.

[13] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 606–626, Aug. 2016.

[14] S. Tabakhi and P. Moradi, "Relevance-redundancy feature selection based on ant colony optimization," *Pattern Recognit.*, vol. 48, no. 9, pp. 2798–2811, Sep. 2015.

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

[16] A. K. Das, S. Das, and A. Ghosh, "Ensemble feature selection using bi-objective genetic algorithm," *Knowl. Based Syst.*, vol. 123, pp. 116–127, May 2017.

[17] X. Xia *et al.*, "Triple archives particle swarm optimization," *IEEE Trans. Cybern.*, vol. 50, no. 12, pp. 4862–4875, Dec. 2020.

[18] X. Xia *et al.*, "An expanded particle swarm optimization based on multi-exemplar and forgetting ability," *Inf. Sci.*, vol. 508, pp. 105–120, Jan. 2020.

[19] X.-F. Song, Y. Zhang, D.-W. Gong, and X.-Y. Sun, "Feature selection using bare-bones particle swarm optimization with mutual information," *Pattern Recognit.*, vol. 112, Apr. 2021, Art. no. 107804.

[20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

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

[22] R. Diao and Q. Shen, "Nature inspired feature selection meta-heuristics," *Artif. Intell. Rev.*, vol. 44, no. 3, pp. 311–340, Jan. 2015.

[23] Y. Zhang, D. W. Gong, Y. Hu, and W. Q. Zhang, "Feature selection algorithm based on bare bones particle swarm optimization," *Neurocomputing*, vol. 148, pp. 150–157, Jan. 2015.

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

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

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

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

[28] A. P. Engelbrecht, J. Grobler, and J. Langeveld, "Set based particle swarm optimization for the feature selection problem," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 324–336, Oct. 2019.

[29] X. Huang, Y. Chi, and Y. Zhou, "Feature selection of high dimensional data by adaptive potential particle swarm optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2019, pp. 1052–1059.

[30] M. Kang, M. R. Islam, J. Kim, J. M. Kim, and M. Pecht, "A hybrid feature selection scheme for reducing diagnostic performance deterioration caused by outliers in data-driven diagnostics," *IEEE Trans. Ind. Electron.*, vol. 63, no. 5, pp. 3299–3310, May 2016.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[31] A. S. Ghareb, A. A. Bakar, and A. R. Hamdan, "Hybrid feature selection based on enhanced genetic algorithm for text categorization," *Expert Syst. Appl.*, vol. 49, pp. 31–47, May 2016.

[32] P. Bermejo, J. A. Gámez, and J. M. Puerta, "A GRASP algorithm for fast hybrid (filter-wrapper) feature subset selection in high-dimensional datasets," *Pattern Recognit. Lett.*, vol. 32, no. 5, pp. 701–711, Apr. 2011.

[33] A. A. Karizaki and M. Tavassoli, "A novel hybrid feature selection based on ReliefF and binary dragonfly for high dimensional datasets," in *Proc. 9th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, Oct. 2019, pp. 300–304.

[34] H. Lu, J. Chen, K. Yan, Q. Jin, Y. Xue, and Z. Gao, "A hybrid feature selection algorithm for gene expression data classification," *Neurocomputing*, vol. 256, pp. 56–62, Sep. 2017.

[35] J. Kennedy, "Bare bones particle swarms," in *Proc. IEEE Swarm Intell. Symp.*, Dec. 2003, pp. 80–87.

[36] T. Gangavarapu and N. Patil, "A novel filter-wrapper hybrid greedy ensemble approach optimized using the genetic algorithm to reduce the dimensionality of high-dimensional biomedical datasets," *Appl. Soft Comput.*, vol. 81, Aug. 2019, Art. no. 105538.

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

[38] S. Kashef and H. Nezamabadi-pour, "An advanced ACO algorithm for feature subset selection," *Neurocomputing*, vol. 147, pp. 271–279, Jan. 2015.

[39] Y. Zhang, S. Cheng, D.-W. Gong, Y. H. Shi, and X. C. Zhao, "Cost sensitive feature selection using two-archive multi-objective artifificial bee colony algorithm," *Expert Syst. Appl.*, vol. 137, pp. 46–58, Dec. 2019.

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

[41] Y. Zhang, X.-F. Song, and D.-W. Gong, "A return-cost-based binary firefly algorithm for feature selection," *Inf. Sci.*, vol. 418, pp. 561–574, Dec. 2017.

[42] H. Faris *et al.*, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems," *Knowl. Based Syst.*, vol. 154, pp. 43–67, Aug. 2018.

[43] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018.

[44] E. Hancer, B. Xue, and M. Zhang, "A survey on feature selection approaches for clustering," *Artif. Intell. Rev.*, vol. 53, no. 2, pp. 4519–4545, 2020.

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

[46] K. Chen, F.-Y. Zhou, and X.-F. Yuan, "Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection," *Expert Syst. Appl.*, vol. 128, pp. 140–156, Aug. 2019.

[47] Q. Liu, J. Zhang, J. Xiao, H. Zhu, and Q. Zhao, "A supervised feature selection algorithm through minimum spanning tree clustering," in *Proc. IEEE 26th Int. Conf. Tools Artif. Intell.*, Nov. 2014, pp. 264–271.

[48] C. Jie, S. Chao, Y. Sheng, S. Wang, and J. Luo, "Feature selection based on density peak clustering using information distance measure," in *Proc Int. Conf. Intell. Comput.,* Lecture Notes in Computer Science, vol. 10362, Jul. 2017, pp. 125–131.

[49] I. Chatterjee, M. Ghosh, P. K. Singh, R. Sarkar, and M. Nasipuri, "Clustering-based feature selection framework for handwritten indic script classification," *Expert Syst.*, vol. 36, no. 6, 2019, Art. no. e12459.

[50] N. Almugren and H. Alshamlan, "A survey on hybrid feature selection methods in microarray gene expression data for cancer classification," *IEEE Access*, vol. 7, pp. 78533–78548, 2019.

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

[52] Z. Huang, C. Yang, X. Zhou, and T. Huang, "A hybrid feature selection method based on binary state transition algorithm and ReliefF," *IEEE J. Biomed. Health Inform.*, vol. 23, no. 5, pp. 1888–1898, Sep. 2019.

[53] G. Ansari, T. Ahmad, and M. N. Doja, "Hybrid filter–wrapper feature selection method for sentiment classification," *Arab. J. Sci. Eng.*, vol. 44, pp. 9191–9208, Jul. 2019.

[54] Q. Wu, Z. Ma, J. Fan, G. Xu, and Y. Shen, "A feature selection method based on hybrid improved binary quantum particle swarm optimization," *IEEE Access*, vol. 7, pp. 80588–80601, 2019.

[55] X.-F. Song, Y. Zhang, Y.-N. Guo, X.-Y. Sun, and Y.-L. Wang, "Variable-size cooperative coevolutionary particle swarm optimization for feature selection on high-dimensional data," *IEEE Trans. Evol. Comput.*, vol. 24, no. 5, pp. 882–895, Oct. 2020.

[56] X. Zhao, W. Deng, and Y. Shi, "Feature selection with attributes clustering by maximal information coefficient," *Proc. Comput. Sci.*, vol. 17, pp. 70–79, May 2013.

[57] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, pp. 1205–1224, Oct. 2004.

[58] F. Pacheco, M. Cerrada, R.-V. Sánchez, D. Cabrera, C. Li, and J. V. de Oliveira, "Attribute clustering using rough set theory for feature selection in fault severity classification of rotating machinery," *Expert Syst. Appl.*, vol. 71, pp. 69–86, Apr. 2017.

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

[60] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, nos. 1–2, pp. 23–69, Oct. 2003.

[61] I. S. Oh, J.-S. Lee, and B.-R. Moon, "Hybrid genetic algorithms for feature selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 11, pp. 1424–1437, Nov. 2004.

[62] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artif. Intell.*, vol. 97, nos. 1–2, pp. 273–324, Dec. 1997.

[63] S. K. Singhi and H. Liu, "Feature subset selection bias for classification learning," in *Proc. 23rd Int. Conf. Mach. Learn.*, Jan. 2006, pp. 849–856.

Xian-Fang Song received the B.S. degree in automation from the Luoyang Institute of Science and Technology, Luoyang, China, in 2014. She is currently pursuing the Doctoral degree with the School of Information and Control Engineering, China University of Mining and Technology, Xuzhou, China.

Her research interests include intelligence optimization and feature selection.

Yong Zhang (Member, IEEE) received the B.Sc. and Ph.D. degrees in control theory and control engineering from the China University of Mining and Technology, Xuzhou, China, in 2006 and 2009, respectively.

He is a Professor with the School of Information and Control Engineering, China University of Mining and Technology. His research interests include intelligence optimization and data mining.

Dun-Wei Gong (Member, IEEE) received the Ph.D. degree in control theory and control engineering from China University of Mining and Technology in 1999.

He is a professor in the School of Information and Control Engineering, China University of Mining and Technology. He has been published in over 180 publications. His main research interests include intelligence optimization and control, and big data processing.

Xiao-Zhi Gao received the B.Sc. and M.Sc. degrees from the Harbin Institute of Technology, Harbin, China, in 1993 and 1996, respectively, and the D.Sc. (Tech.) degree from the Helsinki University of Technology (currently, Aalto University), Espoo, Finland, in 1999.

He has been working as a Professor with the University of Eastern Finland, Kuopio, Finland, since 2018. He has published more than 400 technical papers on refereed journals and conferences. His current research interests are nature-inspired computing methods with applications in optimization, data mining, machine learning, control, signal processing, and industrial electronics.

Authorized licensed use limited to: Univ of Calif Santa Barbara. Downloaded on June 22,2021 at 22:50:05 UTC from IEEE Xplore. Restrictions apply.