Community-Grouping Based Particle Swarm Optimisation Algorithm for Feature Selection

Jianfeng Qiu†, Jiangchuan Wan†, Lei Zhang†, Fan Cheng†
and Yongkang Luo†
†*Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Anhui University, China*
School of Computer Science and Technology, Anhui University, Hefei 230039, China
Email: qiujianf@ahu.edu.cn, 695261708@qq.com, zl@ahu.edu.cn, chengfan@mail.ustc.edu.cn(*Corresponding author*)
2457248761@qq.com

***Abstract*—As a frequently-used dimensionality reduction technique in machine learning, feature selection has attracted interests in the last decade. Since feature selection is essentially a combinatorial optimization problem, how to search the valuable feature subset is a challenging optimization task. Particle swarm optimization (PSO) algorithm and its variations have shown their competitiveness in solving feature selection problem. However, they have been proven to be easily trapped into the local optimal in high-dimensional space due to their intrinsic characteristic of quick convergence. To this end, an effective binary particle swarm optimization algorithm, named CBPSOFS, is proposed for feature selection, where a community-grouping based adaptive updating strategy is designed to avoid trapping into the local optimum and enhance the performance of PSO algorithm in feature selection. To be specific, the correlationship among features is used to construct the feature network, where multiple feature groups are obtained by dividing the achieved feature network. Considering that a community usually contains multiple similar features, the proposed adaptive updating strategy utilizes these feature groups to make the similar features not be included in the same particle so as to maintain the diversity of the population in the evolution. In addition, an information gain based initialization strategy and a history information based resetting strategy are also developed to improve the quality of obtained feature subset. Experimental results on several real world datasets have demonstrated the effectiveness of CBPSOFS in feature selection when compared with the several state-of-the-art baselines.**

***Index Terms*—feature selection; binary particle swarm optimisation; community grouping; population diversity;**

I. Introduction

With the rapid development of data acquisition, a large number of high-dimensional datasets with redundant, irrelevant and even noisy features have been produced in different fields [1]–[3]. These features not only increase the training time but also decrease the classification accuracy of the trained classifier [4]. Feature selection (FS) is to select the optimal feature subset to achieve similar or better performance than using all features, which is regarded as a combination optimization problem [5]. Since the search space increases exponentially with the number of the available features, how to obtain an optimal feature subset is a challenging task.

During the past decade, a plenty of FS algorithms have been developed, which can be simply classified into two categories: filter and wrapper methods [5]. Filter methods usually detect the feature subset according to the intrinsic characteristics of the data. Compared with the filter methods, wrapper based methods can identify better feature subsets than filter based methods since a classification algorithm is employed to evaluate each candidate feature subset. To this end, in this paper, we focus on the wrapper based methods.

In wrapper methods, identifying the optimal feature subset by exhaustively searching has been considered to be effective in small datasets. Due to the huge search space, the search efficiency decreases dramatically on large datasets, which is the main limitation of this category of methods. To this end, greedy based heuristic algorithms, e.g., sequential forward selection (SFS) [6] and sequential backward selection (SBS) [7] etc, have been proposed to overcome the limitation mentioned above. However, these algorithms may easily converge to local optimal since no enough feature subsets are evaluated which affects the quality of the chosen feature subsets [5].

Due to the good global search ability, a series of evolutionary computation (EC) based FS algorithms have been developed in recent years, and they can yield more effective solutions [8]–[11]. Among these algorithms, PSO based FS algorithms have attracted much attention due to its simpleness, effectiveness and less parameters. However, these algorithms could easily be trapped into the local optimal area in searching the optimal feature subset when confronted with large search space. Therefore, many variations of PSO algorithm have been suggested to tackle this problem. For example, Chuang et al. [12] proposed a variation of PSO algorithm for feature selection, where a resetting strategy was developed when *gbest* has kept the same value for a number of iterations. Xue et al. [9] proposed three different strategies for updating personal history best and global best particles, which can effectively avoid to be trapped into the local optimal. Moradi et al. [13] designed a local search operator to select the salient feature subset to avoid the stagnation happening, which can be used to guide the evolve process effectively. Recently, Qiu [10] developed an adaptive chaotic jump strategy integrated into an improved particle swarm optimization (BBPSO [14]) for feature selection to prevent the stagnated particles from being trapped into local optimal. In addition, several different updating mechanisms developed for escaping the local optima were integrated into PSO algorithm for feature selection [15], [16]. Empirical results of these PSO variants have justified their competitiveness in overcoming the local optimal in solving FS problem.

In this paper, we continue this research line by proposing a novel community-grouping based adaptive updating strategy under the framework of binary particle swarm optimization algorithm for feature selection. Different from these algorithms mentioned above, the proposed CBPSOFS utilizes the correlationship among features to construct the feature network and the feature communities obtained by dividing the feature network are further used to guide the population evolution. In summary, the main contributions can be summarized as follows.

- A community-grouping based adaptive updating strategy is suggested for PSO, which can avoid to be trapped into the local optimal and improve the quality of the identified feature subset. The suggested strategy utilizes the feature community information to guide the particles evolve and maintain the diversity of the population in the evolution, where feature community is obtained by the feature network constructed by the correlationship among the features.
- Based on the suggested adaptive updating strategy, an effective binary particle swarm optimization algorithm, termed CBPSOFS, is proposed for feature selection. In addition, an information gain based initialization strategy and a history information based particle resetting strategy are also suggested, with which the performance of CBPSOFS can be further improved.
- We compare the proposed CBPSOFS with several state-of-the-arts on eight datasets with different characteristics. The experimental results demonstrate the effectiveness of the proposed algorithm in terms of the number of the selected features and the accuracy using the selected feature subset.

The remainder of this paper is organized as follows. We first give the preliminary of feature selection and the related work in Section II. Section III presents the details of the proposed algorithm and the empirical results by comparing CBPSOFS with the state-of-the-arts are reported in Section IV. Section V concludes the paper and discusses the future work.

II. The Preliminaries and Related Work

In this section, the preliminaries about feature selection are firstly described, and then the related works on feature selection with PSO algorithm are introduced.

A. Feature Selection Problem

Usually, feature selection is regarded as a combination optimization problem, which is formulated as follows. Given a dataset D with the features $F = \{f_1, f_2, \cdots, f_d\}$, there exists an optimal feature subset $F_x \subset F$, $|F_x| < d$ that can provide the better performance than using all ones. To guide the evolution of the population, a fitness function *Fitness* considering simultaneously the size of the selected feature subset and the classification performance on it is defined as follows, which is also adopted in the works of [15], [17], [18].

$$Min\,Fitness = \alpha \cdot ErrorRate + (1-\alpha) \cdot FeatureRate \quad (1)$$

where $ErrorRate = 1 - Accuracy(C, F_x)$, $FeatureRate = \frac{|F_x|}{d}$, C is a specific classification algorithm, $|F_x|$ represents the number of the features in the selected feature subset F_x, α is an user-specified parameter.

B. Related Work on PSO based Feature Selection

PSO, as a promising optimization technique, has been widely used for feature selection [11], [19], [20]. In the following, we briefly review some representative works in feature selection suggested to enhance the performance of PSO algorithm by tackling the premature convergence. A comprehensive survey for PSO based feature selection can be found in [5].

Chuang et al. [12] proposed an improved PSO algorithm with the aim to skip the local optimum by resetting the value of *gbest*. The motivation was inspired by the following fact that if the value of *gbest* has kept unchanged for several iterations, the population needed to be reset to enhance the ability of escaping the prematurity. Experimental results have demonstrated its effectiveness in overcoming local optimal. In recognizing the superiority of the resetting strategy, a variety of useful strategies have been proposed to maintain the diversity of the population by activating the stagnated particles. M.Vieira et al. [21] suggested a novel resetting swarm mechanism to avoid premature convergence. To be specific, when the global best solution has not changed for several iterations, all bits in the *gbest* were resetted as zero. However, in these methods, due to the lack of effective strategies for overcoming prematurity convergence, the population easily traps into the local optimal. To this end, Xue et al. [9] proposed three different *pbest* and *gbest* updating strategies to guide the search with the goal of avoiding to be trapped into the local optimal, among which *PSO (4-2)* was considered to be more competitive and promising in feature selection. Moradi et al. [13] recently developed a PSO based hybrid feature selection algorithm, named *HPSO-LS*. Specifically, for a given particle, the selected features were grouped into two subgroups, 'X_d' and 'X_s' based on *pearson correlation coefficient* [22]. Then two operators '*Add*' and '*Delete*' were designed to perform local search. The results showed that the suggested local search strategy can effectively guide the population evolve. Qiu [10] proposed an adaptive chaotic jump strategy integrated into BBPSO [14] algorithm, namely *BBPSO-ACJ*, where the chaotic behavior was employed to activate the stagnated particles and further enhance the search ability of the particles by overcoming the local optimal.

The PSO variants mentioned above have demonstrated their effectiveness in tackling the premature convergency, and in this paper we will continue this research line by suggesting a community-grouping based particle swarm optimization algorithm, named CBPSOFS, for feature selection. The basic idea of CBPSOFS is to utilize the relationship among features to generate a feature network and then divide the network into several communities. Since the features in the same community have more similar than the ones in different communities, a community-grouping based adaptive updating strategy is designed to guide the population evolution, which can balance the exploitation and exploration well. The details of the proposed algorithm is presented in next section.

III. The Proposed Algorithm CBPSOFS

In this section, we first present the framework of CBPSOFS. Then, we elaborate the suggested community-grouping based adaptive updating strategy, which is the key component of CBPSOFS. Finally, an information gain based initialization strategy and a history information based particles resetting strategy are given to further enhance the performance of CBPSOFS.

A. The General Framework of CBPSOFS

Before we introduce the suggested framework of CBPSOFS, the adopted binary encoding scheme is first described which is used to represent a candidate feature subset. Specifically, for each particle p_i, where $i \in \{1, 2, \cdots, N\}$ and N is the size of population, the j-th feature is selected by the particle p_i and the corresponding bit is set as 1, i.e. $p_{ij} = 1$, otherwise $p_{ij} = 0$. The proposed CBPSOFS adopts a similar framework with standard binary particle swarm optimization (BPSO) algorithm [23], which has been widely used as the basic framework for feature selection [24]–[26]. It mainly consists of three stages. In the first stage (Line 1-5), an information gain based initialization strategy is suggested to generate the population with good diversity. Then the correlationship among features are used to construct a feature network and several feature communities are generated by *Louvain* algorithm [27]. In the second stage (Line 6-9), the obtained feature communities are employed to design a community-grouping based adaptive updating strategy which make the similar features not be included in the same particle. In the third stage (Line 10-15), to further enhance the performance of the proposed algorithm, a history information based particles resetting strategy is also developed to activate the stagnated particles. The general framework of CBPSOFS is presented in Algorithm 1.

Algorithm 1 The General Framework of CBPSOFS

Input: D: the dataset, d: the number of features, C: the set of class label of the dataset, $Maxiter$: the maximum number of iterations; N: population size, ω: inertia weight, c_1, c_2: learning factors, v_{min}, v_{max}: the minimum and maximum velocity of the particles; m: the number of iteration of P_{pbest} not being updated;
Output: The final output optimal feature subset P_{gbest}

1: $P \leftarrow$ InfoGainBasedInitial (N, D, d, C);
2: $group \leftarrow$ Obtain different feature communities using *Louvain* algorithm;
3: $Fitness \leftarrow$ Calculate the fitness values of the particles in P using (1);
4: $P_{pbest} = P$; $P_{gbest} = \arg\min_{i \in 1,\ldots,N}\{Fitness(p_i)\}$;
5: $Counts = zeros(1, N)$;
6: **for** $iter = 1$ to $Maxiter$ **do**
7: $\quad P' \leftarrow$ Update the particles using the formulas in BPSO [23];
8: $\quad CR \leftarrow$ Calculate the adaptive updating probability value for each particle using the formulas (2) to (4);
9: $\quad P'' \leftarrow$ CommunityBasedAdaUpdate (P', $group$, CR, d);
10: $\quad$ Update P_{pbest} of the population;
11: $\quad Index \leftarrow$ Find the indexes of the particles which have not been updated;
12: $\quad Counts(Index) += 1$;
13: $\quad [P, Counts] \leftarrow$ HisInfoBasedParticleRest (N, d, P_{pbest}, P'', m, $Counts$);
14: $\quad P_{gbest} = \arg\min_{i \in 1,\ldots,N}\{Fitness(p_i)\}$;
15: **end for**

From the above explanation of Algorithm 1, we can find that there are three important components in CBPSOFS, that is, a community-grouping based adaptive updating strategy (Line 9), an information gain based initialization strategy (Line 1) and a history information based particle resetting strategy (Line 13). In the following, we will introduce them in detail.

B. A Community-Grouping Based Adaptive Updating Strategy

Due to the quick convergency of PSO, the population easily traps into local optimal especially in high-dimensional search space. To tackle this issue, a community-grouping based adaptive updating strategy is designed by employing the obtained feature communities information. The main idea is motivated by the following fact.

Similar particles in decision space may have been projected into the same or similar positions in objective space, which may lead them to be trapped into local optimal. Thus, employing the similarity to balance the exploration and exploitation strength adaptively is considered to be helpful for escaping the local optimal. In this paper, hamming distance is adopted for measuring the similarity and used for designing the adaptive updating probability. The proposed community-grouping based adaptive updating strategy is performed as follows.

Firstly, a weight matrix W with $d \times d$ is constructed based on *pearson correlation coefficient* [22], where the weight w_{ij} ($w_{ij} \in (0, 1)$) denotes the correlation between the i-th feature f_i and the j-th feature f_j, d is the number of features. Then the matrix W is transformed into a $0-1$ matrix $W^{'}$ controlled by using a threshold θ. f_i and f_j are considered as connected, $w^{'}_{ij} = 1$, if the value w_{ij} is larger than the predefined threshold θ. Otherwise is disconnected, $w^{'}_{ij} = 0$. To obtain feature community grouping information, *Louvain* algorithm [27] is employed to cluster on the matrix $W^{'}$ and a series of feature communities are achieved. The features in the same community have more similarities among them than the ones in different communities, which are considered as the reference information used for the following adaptive updating strategy.

Secondly, an adaptive updating probability is designed to balance exploration and exploitation in different evolutionary stages. Specifically, for any two particles p_i and p_j, the hamming distance between them is firstly defined as the sum of the positions which have different binary values in Eq. (2) and is normalized into $[0, 1]$.

$$Ham(p_i, p_j) = \frac{\sum_{k=1}^{d} \left| p_i^k - p_j^k \right|}{d} \tag{2}$$

To quantize the variation of the particle diversity, in this paper, we define the similarity of each particle based on the hamming distance, which is shown in Eq. (3).

$$Sim(p_i) = \frac{\sum_{j=1, j \neq i}^{N} \mid Ham(p_i, p_j) < \frac{iter}{Maxiter} * 0.5 \mid}{N} \tag{3}$$

For each particle p_i, Eq. (3) calculates the similarity with its neighbours, $iter$ is the current iteration and $Maxiter$ is the maximum number of iteration.

In BPSO, the particles converge quickly, which leads the population close to the local optimal in the early stage, and more exploration operators are required. While in the late stage, intensifying exploitation operation is helpful to find the potential optimal solution. Based on this fact, an adaptive updating probability is defined as Eq. (4) to balance the strength of exploration and exploitation.

$$CR(p_i) = (1 - Sim(p_i)) \times e^{-Sim(p_i)} \tag{4}$$

With the adaptive updating probability defined above, a community-grouping based adaptive updating operator is proposed. To be specific, firstly, one particle p_i is selected randomly, and the other particle p_j which has the maximum hamming distance with p_i, is selected as a candidate and updating particle. The updating operator is performed on the selected two particles by utilizing the found feature communities. Fig. 1 illustrates the process of the suggested adaptive updating operator for the selected particles.

Fig. 1 gives 10 features which have been divided into two feature communities A and B using *Louvain* algorithm, i.e., $A = \{f_1, f_2, f_3, f_5, f_7, f_9\}$ and $B = \{f_4, f_6, f_8, f_{10}\}$. As shown in Fig. 1 (a), the particle p_i is randomly selected from the population, and p_j is the particle with the maximum hamming distance with p_i. Let $crmax = max(CR(p_i), CR(p_j))$, $rand \in (0, 1)$ be a random number. If $rand < crmax$, the suggested community-grouping based updating process is performed in Fig. 1 (b) to (d).

More specific, we firstly select the features satisfying the condition $p_{i,k} \neq p_{j,k}$, where $p_{i,k}$ denotes the $k\text{-}th$ feature in the particle p_i, $k = 1, \cdots, d$. Then, we calculate and assign a ratio for each selected feature, taking the feature f_k as an example ($f_k \in A$), the ratio $BitCR$ is defined as follows.

$$BitCR = \begin{cases} 1 - \frac{|C_{jk}|+1}{|A|}, & p_{i,k} = 1 \\ 1 - \frac{|C_{ik}|+1}{|A|}, & p_{j,k} = 1 \end{cases} \tag{5}$$

where $|C_{ik}|$ ($|C_{jk}|$) is the number of the selected features in p_i (p_j) which belongs to the same community with the feature f_k. $|A|$ denotes the size of the community A. Usually, the large ratio means that the feature has more probability of being selected for performing crossover, which results in the features of the same community as few as possible being included in the same particle. In Fig. 1, since the generated random number is less than $BitCR(3)$ and $BitCR(6)$, the corresponding feature f_3 (shown in Fig. 1 (a)) and f_6 (shown in Fig. 1 (b)) are exchanged. Feature f_9 is not exchanged due to $rand > BitCR(9)$. The whole procedure of updating the particles in the population by utilizing the community information is presented in Algorithm 2.

Algorithm 2 CommunityBasedAdaUpdate

Input: P: the population;
  $group$: the feature community information;
  CR: the adaptive crossover probability;
  d: the number of the features;
Output: Q: the updated population;

```
while |Q| < |P| do
    randomly select a particle p_i, p_j is the particle with the maximum hamming distance with p_i;
    crmax = max(CR(p_i), CR(p_j));
    if rand < crmax then
        for k = 1 to d do
            if p_{i,k} ≠ p_{j,k} then
                Index = Find(group == group(k)) ; // Find the indexes of the features in the same community with feature f_k;
                if p_{i,k} == 1 then
                    IndSum = length(Find(p_{j,Index} == 1)); // Find the number of the selected features in particle p_j which belong to the same community with the feature f_k;
                    BitCR = 1-(IndSum+1)/length(Index);
                end if
                if p_{j,k} == 1 then
                    IndSum = length(Find(p_{i,Index} == 1));
                    BitCR = 1-(IndSum+1)/length(Index);
                end if
                if rand < BitCR then
                    p_{i,k} ↔ p_{j,k};
                    generating two updated particles p_i' p_j'
                    Q = Q ∪ {p_i', p_j'};
                end if
            end if
        end for
    end if
end while
```

C. The Information Gain based Initialization Strategy

Usually, random initialization is a frequently-used method to generate initial population. However, the quality of the initial population could be further improved if the prior information about the problem has been considered. To this end, an information gain based initialization strategy is proposed for generating the initial population with good diversity. Algorithm 3 presents the main procedure of the proposed initialization strategy, which includes two steps: the information gain calculation (step1, Line 1-4), and the population initialization with the calculated information gain (step2, Line 5-8).

In the first step, information gain (IG), as a popular and widely applied method [28], [29], is calculated and used to measure the correlation between features and class labels. To be specific, the information gain $I(f; C)$ between the feature f and class C is defined as follows.

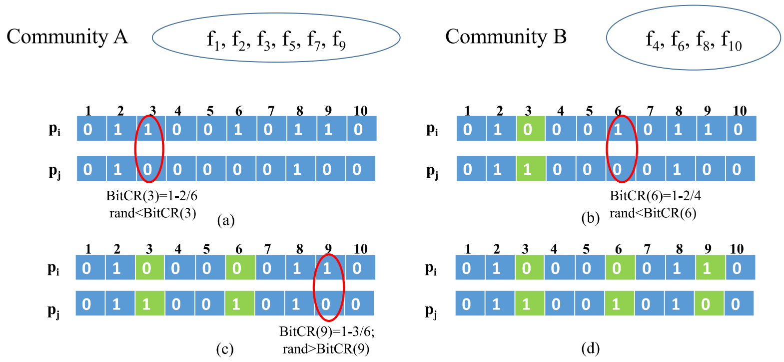

Fig. 1: An example to illustrate the updating strategy for the selected particles with feature community information.

$$I(f;C)=\sum_{k=1}^{n}\sum_{j=1}^{m}p(f(k),c_j)\log_2\frac{p(f(k),c_j)}{p(f(k))p(c_j)} \tag{6}$$

where n is the number of the possible values in feature f and m represents the number of class. In Eq. (6), if $I(f;C)$ is larger, the feature f and class label C are more correlated. Then, we sort the results and group them into two parts. The first half (*High-InfoGain* group) has a stronger correlation with the class labels than the remaining half (*Low-InfoGain* group).

In the second step, we employ the information gain as a kind of prior knowledge to generate the initial population. To maintain the diversity of the initial population, a random integer $Krand$ is used to determine the number of the features in each individual, where $Krand \in (0, 0.6d]$ and d is the number of features of the dataset. To make more features with high correlation be selected, $ceil(rand(0.5,1)*Krand)$ features are chosen randomly from *High-InfoGain* group and the remaining ones are selected from *Low-InfoGain* group.

Algorithm 3 InfoGainBasedInitial

Input: N: the size of population; D: the dataset; C: the class label set; d: the number of features;
Output: P: Initial population;

for $i=1$ to d **do**
  $InforGain(i)$ = $I(f_i;C)$ //f_i denotes the i-th feature;
end for
$[High-InfoGain, Low-InfoGain]$ = $Sort(InfoGain)$; // Sort the information gain values $InfoGain$ in descending order and divide them into two parts: *High-InfoGain* and *Low-InfoGain*;
for $i=1$ to N **do**
  $p_{i,\{1,2,\cdots,d\}} \leftarrow zeros(1,d)$;
  $Krand$ ← a random integer generated from $(0, 0.6d]$; $ceil(rand(0.5,1)*Krand)$ features are selected from *High-InfoGain* randomly; $Krand - ceil(rand(0.5,1)*Krand)$ features are selected from *Low-InfoGain* randomly;
  $p_{i,\{j_1,j_2,\cdots,j_{Krand}\}} \leftarrow ones(1, Krand)$;
end for

D. The History Information-based Particle Resetting Strategy

To further enhance the performance of the proposed algorithm, a history information-based particle resetting strategy is suggested in Algorithm 4 to recover the diversity of the population, which is composed of two steps: one is to collect the preference information about each feature from its personal history optimal solutions (step1, Line 1-7), and the other is to activate the stagnated particles according to the achieved preference information (step2, Line 8-15).

In the first step, the selected frequency of each feature (preference information) is calculated from its personal history optimal solutions P_{pbest}. In the second step, the suggested resetting process for the stagnated particles is performed as follows: 1) the particles that keeps the same personal history optimal solutions for a successive m iterations are selected as candidate particles for performing the activation operation. 2) for each candidate particle, a roulette wheel based selection mechanism is employed to generate a new particle, where k features are chosen according to the achieved preference information.

Algorithm 4 HisInfoBasedParticleRest

Input: N: the size of population; d: the number of the features; P_{pbest}: the personal optimum solutions; P'': the updated population; m: the maximum number of P_{pbest} not being updated; $Counts$: the vector recording how many iterations of each feature not been updated;
Output: P: Reseted population;
  $Counts$: Updated vector;

1: $P \leftarrow P''$;
2: **for** $j=1$ to d **do**
3:   $Frequency(j)$ = $\sum_{i=1}^{N} p_{i,j}^{pbest}$;
4: **end for**
5: **for** $j=1$ to d **do**
6:   $GenProbability(j)=\frac{Frequency(j)}{\sum_{j=1}^{d}Frequency(j)}$;
7: **end for**
8: **for** $i=1$ to N **do**
9:   **if** $Counts(i) == m$ **then**
10:     $p_{i,\{1,\cdots,d\}} = zeros(1,d)$;
11:     k = $ceil(rand(0, \lceil 0.6d \rceil))$; //determining the feature number randomly
12:     $(f_{u1},\cdots,f_{uk})$ = $RoulettewheelSelect(GenProbability)$; //choosing k features using a roulette wheel based selection mechanism according to $GenProbability$;
13:     $p_{i,\{u1,\cdots,uk\}} = ones(1,k)$;
     $Counts(i)=1$;
14:   **end if**
15: **end for**

IV. Experimental Evaluation

In this section, we first give the experimental settings including datasets, baselines, evaluation metrics and parameter settings. Then, we empirically verify the performance of the proposed CBPSOFS by comparing it with several baselines for feature selection.

A. Experimental Setting

1) Dataset: In our experiments, the eight real-world datasets are employed to verify the performance of the proposed CBPSOFS, which can be download from UCI machine learning repository[1]. The detailed characteristics of these datasets are depicted in Table I.

TABLE I: The eight datasets with different characteristics

No	DataSet	# Features	# Instances	# Classes
1	Wine	13	178	3
2	Australian	14	690	2
3	Parkinsons	22	195	2
4	German	24	1,000	2
5	Wdbc	30	569	2
6	Ionosphere	34	351	2
7	Spectf	44	267	15
8	Multi_features	649	2,000	10

To evaluate the performance of each candidate feature subset, 10-fold cross-validation is employed in these experiments. In addition, K nearest neighbor (KNN) algorithm is adopted to evaluate the classification performance and K is set to 5 in the following experiments to simplify the calculation.

2) Comparison Algorithms: We compare the proposed CBPSOFS with four popular baselines, namely, BPSO [23], PSO(4-2) [9], HPSO-LG [13] and BBPSO-ACJ [10]. Among them, BPSO is a classical binary PSO and widely applied as a baseline for developing BPSO based feature selection algorithms. PSO(4-2) is an improved PSO algorithm, where the proposed updating strategies for $pbest$ and $gbest$ could avoid to be trapped into local optimal effectively. In HPSO-LG, two basic local search operators were integrated into PSO to enhance the performance of PSO for feature selection. BBPSO-ACJ is a novel feature selection algorithm based on BBPSO [14] where an adaptive chaotic jump strategy is proposed to overcome the premature problem.

3) Evaluation Metrics and Parameter Setting: There are two metrics are utilized for evaluating the quality of the obtained particle, where each particle denotes a selected feature subset. The first metric is mean classification accuracy of the classifier C, which is trained on the selected feature subset and calculated in Eq. (7):

$$MeanAccuracy = \frac{1}{k \times M} \sum_{j=1}^{k} \sum_{i=1}^{M} C_j(Pre_i, Real_i) \tag{7}$$

where k is the number of independent running times of the algorithm, M is the size of testing dataset, Pre_i is the label of the $i\text{-}th$ testing instance predicted by the classifier C_j trained using the selected feature subset F_j in the $j\text{-}th$ run and $Real_i$ is the real label of the $i\text{-}th$ instance. When Pre_i and $Real_i$ is identical, $C(Pre_i, Real_i)$ equals to 1, otherwise equals to 0. The other metric is the average size of the selected feature subsets during the k times and can be defined as follows.

$$MeanSize = \frac{1}{k} \sum_{j=1}^{k} |F_j| \tag{8}$$

where F_j is the best feature subset obtained in the $j\text{-}th$ run, $|F_j|$ represents the number of the selected features.

For fair comparisons, we adopt the recommended parameter values suggested in their original papers [9], [10], [13], [23].In CBPSOFS, the parameters are set as w =0.48, c1 = c2 = 2, vmax = 4.0, m = 5 and θ = 0.6 respectively. In all the following experiments, we fix the population size N = 30, the maximum number of iteration is set as 100, $\alpha = 0.8$ is the recommended parameter value in [15]. To further verify the significance level of any two algorithms, we employ T-test with significance level 0.05 to whether the proposed CBPSOFS is significantly better than other baselines. All algorithms are performed 10 runs independently using Matlab environment for each dataset.

B. The Comparison Results Between CBPSOFS and Baselines

In this section, the comparison results are presented to demonstrate the effectiveness of the proposed CBPSOFS, which are shown in Table II and Table III. The best results are marked in bold.

From Table II, we can find that the proposed CBPSOFS shows its superiority over the other baselines on most of the datasets. Similarly, in Table III, CBPSOFS further demonstrates its effectiveness in the mean classification accuracy. Specifically, CBPSOFS is ranked the first on 6 datasets and in the second on 2 datasets. The super performance of the proposed algorithm is attributed to the fact that in the procedure of feature selection, the features are divided into different communities and the similar features are not included in the same particle, which avoids to be trapped into local optimal.

Table IV shows the results of the T-test statistical test of the proposed algorithm compared with the baselines. When p-value is less than 0.05, there exists a significant difference between the two algorithms. It can be seen from Table IV that the proposed CBPSOFS is obviously better than the comparison algorithms. From the analysis above, we can conclude that the proposed CBPSOFS algorithm is a promising feature selection method on the real datasets.

C. Effectiveness of the proposed strategies in CBPSOFS

In CBPSOFS, three different strategies are proposed and in the following, we investigate the influence of these strategies on the performance of CBPSOFS. Fig. 2 presents the best fitness values of these algorithms with different iterations on each dataset. Specifically, *BPSO* and *CBPSOFS* denote BPSO algorithm and the proposed algorithm respectively. *BPSO+C* represents the BPSO with the proposed community-grouping based adaptive updating strategy, *BPSO+C+Init* denotes the

[1]https://archive.ics.uci.edu/ml/

TABLE II: The mean and standard deviation values of $Feature\ number$ obtained by five comparison algorithms. The best results are highlighted in bold.

	CBPSOFS		HPSO-LG		PSO(4-2)		BBPSO-ACJ		BPSO	
Data set	MeanSize	Std	MeanSize	Std	MeanSize	Std	MeanSize	Std	MeanSize	Std
Wine	3.5	0.85	8.3	1.06	9	1.25	5.9	1.2	**3.3**	0.48
Australian	**1.5**	0.67	7	1.56	4.4	1.33	7.2	2.35	2.6	1.27
Parkinsons	**4.8**	1.32	11.8	1.93	8.6	3.75	10.4	2.12	**4.8**	1.13
German	**2.5**	1.78	12.1	2.38	7.1	3.54	9.8	3.99	5.3	1.49
Wdbc	**4.6**	0.84	16.3	2.63	7.1	3.84	14.75	3.24	5.1	1.37
Ionosphere	4.5	1.84	11	4.27	9	4.64	**3.2**	0.92	5.7	1.26
Spectf	**6.3**	2.91	15.1	8.10	9.7	4.8	8	6.02	10.5	2.64
Multi_features	**270.3**	6.38	322.1	15.07	325.7	16.23	320.9	2.96	309.3	8.46

TABLE III: The mean and standard deviation values of $accuracy$ obtained by five comparison algorithms. The best results are highlighted in bold.

	CBPSOFS		HPSO-LG		PSO(4-2)		BBPSO-ACJ		BPSO	
Data set	MeanSize	Std	MeanSize	Std	MeanSize	Std	MeanSize	Std	MeanSize	Std
Wine	**0.9642**	0.0208	0.9623	0.0154	0.9358	0.04	0.9547	0.0438	0.9038	0.0338
Australian	**0.8758**	0.0195	0.8604	0.0074	0.8691	0.0276	0.8517	0.0188	0.8435	0.0131
Parkinsons	**0.9362**	0.02	0.8724	0.0475	0.8931	0.0513	0.9121	0.0410	0.8897	0.0604
German	0.7120	0.0134	0.7043	0.0367	0.6723	0.0190	**0.7213**	0.0273	0.7093	0.0210
Wdbc	0.9594	0.0052	**0.9635**	0.0146	0.9447	0.0210	0.9629	0.0096	0.9392	0.0237
Ionosphere	**0.8933**	0.0257	0.8819	0.0292	0.8781	0.0435	0.8771	0.0369	0.8680	0.0402
Spectf	**0.7888**	0.0190	0.7837	0.0475	0.7442	0.0545	0.7688	0.0667	0.7625	0.0183
Multi_features	**0.9637**	0.0064	0.9587	0.0082	0.9301	0.0201	0.9562	0.0087	0.9557	0.0069

TABLE IV: The calculated p-values from the T-test for the CBPSOFS versus other comparison algorithms.

P-value	BBPSO-ACJ	HPSO-LG	PSO(4-2)	BPSO
Wine	< 0.05	< 0.05	> 0.05	< 0.05
Australian	< 0.05	< 0.05	< 0.05	< 0.05
Parkinsons	< 0.05	< 0.05	< 0.05	< 0.05
German	< 0.05	< 0.05	< 0.05	< 0.05
Wdbc	< 0.05	< 0.05	< 0.05	> 0.05
Ionosphere	< 0.05	< 0.05	> 0.05	< 0.05
Spectf	< 0.05	< 0.05	< 0.05	< 0.05
Multi_features	< 0.05	< 0.05	< 0.05	< 0.05

BPSO+C algorithm with the proposed information gain based initialization strategy, and *BPSO+C+RS* is the *BPSO+C* algorithm with the proposed history information based resetting strategy for activating the stagnated particles.

The results shown in Fig. 2 demonstrate the superiority of the suggested strategies. Specifically, on most of the datasets, *BPSO+C* inherits the advantage of *BPSO* in convergency and obtains the better performance than *BPSO*. The reason is that the proposed community-based adaptive updating strategy can effectively escape local optimal by enhancing the diversity of the population. Moreover, we note that the proposed initialization strategy and the resetting strategy further improve the performance of the proposed CBPSOFS. The results shown in this figure have justified their effectiveness of jumping from the local optimal.

V. Conclusions and Future Work

In this paper, we proposed a community-grouping based PSO algorithm named CBPSOFS for feature selection. To be specific, a community-grouping based adaptive updating strategy is suggested, where the identified feature communities are used to maintain the diversity and avoid to be trapped into the local optimal. In addition, an information gain based initialization strategy and a history information based particles resetting strategy are developed to further enhance the performance of the propose algorithm. Finally, we demonstrate the effectiveness of the proposed algorithm on different datasets, and the experimental results clearly validate that CBPSOFS can get the feature subset with high quality. Since feature selection has been proven to be a multi-objective optimization problem [5], in the future, we would like to further explore the performance of the proposed community based updating strategy combined with different multi-objective evolutionary algorithm framework, e.g. NSGA-II [30], LMEA [31], KnEA [32] etc.

Acknowledgement

This work is supported by the Natural Science Foundation of China (Grant No.U1804262, 61976001 and 61876184), Humanities and Social Sciences Project of Chinese Ministry of Education (Grant No.18YJC870004), the Natural Science Foundation of Anhui Province (Grant No.1708085MF166, 1908085MF219) and the Opening Project of Jiangsu Key Laboratory of Data Science and Smart Software (Grant No.2019DS304)

References

[1] L. Wang, Y. Wang, and Q. Chang, "Feature selection methods for big data bioinformatics: A survey from the search perspective," *Methods*, vol. 111, pp. 21–31, 2016.

[2] X. Zhou, X. Gao, J. Wang, H. Yu, Z. Wang, and Z. Chi, "Eye tracking data guided feature selection for image classification," *Pattern Recognition*, vol. 63, pp. 56–70, 2017.

[3] N. Acharya and S. Singh, "An IWD-based feature selection method for intrusion detection system," *Soft Computing*, vol. 22, no. 13, pp. 4407–4416, 2018.

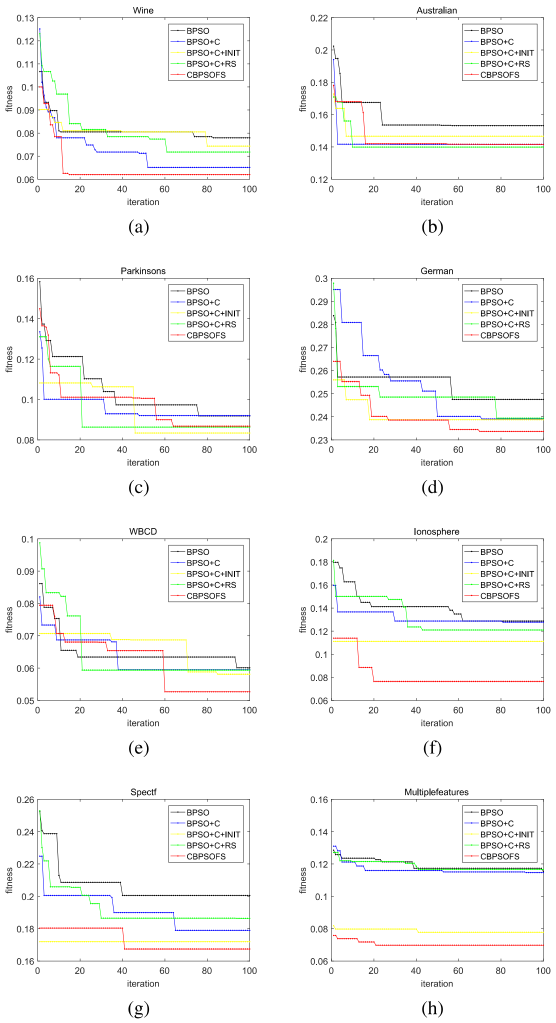

Fig. 2: The best fitness values of CBPSOFS and the baselines with different iterations.

[4] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[5] X. Bing, M. Zhang, W. N. Browne, and Y. Xin, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2016.

[6] A. W. Whitney, "A direct method of nonparametric measurement selection," *IEEE Transactions on Computers*, vol. 100, no. 9, pp. 1100–1103, 1971.

[7] T. Marill and D. Green, "On the effectiveness of receptors in recognition systems," *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 11–17, 1963.

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

[9] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms," *Applied Soft Computing*, vol. 18, pp. 261 – 276, 2014.

[10] C. Qiu, "Bare bones particle swarm optimization with adaptive chaotic jump for feature selection in classification," *International Journal of Computational Intelligence Systems*, vol. 11, no. 1, pp. 1–14, 2018.

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

[12] L. Y. Chuang, H. W. Chang, C. J. Tu, and C. H. Yang, "Improved binary pso for feature selection using gene expression data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29–38, 2008.

[13] P. Moradi and M. Gholampour, "A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy," *Applied Soft Computing*, vol. 43, pp. 117–130, 2016.

[14] J. Kennedy, "Bare bones particle swarms," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, 2003, pp. 80–87.

[15] J. Wei, R. Zhang, Z. Yu, R. Hu, J. Tang, C. Gui, and Y. Yuan, "A BPSO-SVM algorithm based on memory renewal and enhanced mutation mechanisms for feature selection," *Applied Soft Computing*, vol. 58, pp. 176–192, 2017.

[16] A. Maryam and M. B. Behrouz, "Optimizing multi-objective PSO based feature selection method using a feature elitism mechanism," *Expert Systems with Applications*, vol. 113, no. 15, pp. 499–514, 2018.

[17] M. M.Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, no. 18, pp. 302–312, 2017.

[18] H. Faris, M. M.Mafarja, A. A. Heidari, I. Aljarah, A. M.Al-Zoubi, S. Mirjalili, and H. Fujita, "An efficient binary salp swarm algorithm with crossover scheme for feature selection problems," *Knowledge-Based Systems*, vol. 154, no. 15, pp. 43–67, 2018.

[19] B. Tran, B. Xue, and M. Zhang, "Variable-length particle swarm optimization for feature selection on high-dimensional classification," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 3, pp. 473–487, 2018.

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

[21] S. M. Vieira, L. F. Mendonca, G. J. Farinha, and J. M. Sousa, "Modified binary PSO for feature selection using svm applied to mortality prediction of septic patients," *Applied Soft Computing*, vol. 13, no. 8, pp. 3494–3504, 2013.

[22] M. M. Kabir, M. Shahjahan, and K. Murase, "A new local search based hybrid genetic algorithm for feature selection," *Neurocomputing*, vol. 74, no. 17, pp. 2914–2928, 2011.

[23] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, 2002, pp. 4104–4108.

[24] B. Xue, S. Nguyen, and M. Zhang, "A new binary particle swarm optimisation algorithm for feature selection," in *European Conference on the Applications of Evolutionary Computation*. Springer, 2014, pp. 501–513.

[25] X. Teng, H. Dong, and X. Zhou, "Adaptive feature selection using v-shaped binary particle swarm optimization," *Plos One*, vol. 12, no. 3, pp. 1–22, 2017.

[26] S. P. Rajamohana and K. Umamaheswari, "Hybrid approach of improved binary particle swarm optimization and shuffled frog leaping for feature selection," *Computers & Electrical Engineering*, vol. 67, pp. 497–508, 2018.

[27] V. D. Blondel, J. L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics*, vol. 2008, no. 10, pp. 155–168, 2008.

[28] J. F. Qiao and H. G. Han, "Identification and modeling of nonlinear dynamical systems using a novel self-organizing RBF-based approach," *Automatica*, vol. 48, no. 8, pp. 1729–1734, 2012.

[29] J. Qiao, S. Li, and W. Li, "Mutual information based weight initialization method for sigmoidal feedforward neural networks," *Neurocomputing*, vol. 207, pp. 676–683, 2016.

[30] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[31] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "A decision variable clustering-based evolutionary algorithm for large-scale many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 1, pp. 97–112, 2016.

[32] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2014.