# Binary Particle Swarm Optimisation for Feature Selection: A Filter Based Approach

Liam Cervante, Bing Xue, Mengjie Zhang
School of Engineering and Computer Science
Victoria University of Wellington
PO Box 600, Wellington 6140, New Zealand
Email: {Liam.Cervante, Bing.Xue, Mengjie.Zhang} @ecs.vuw.ac.nz

Lin Shang
State Key Laboratory of Novel Software Technology
Department of Computer Science and Technology
Nanjing University, Nanjing 210046, China
Email: shanglin@nju.edu.cn

***Abstract*—Based on binary particle swarm optimisation (BPSO) and information theory, this paper proposes two new filter feature selection methods for classification problems. The first algorithm is based on BPSO and the mutual information of each pair of features, which determines the relevance and redundancy of the selected feature subset. The second algorithm is based on BPSO and the entropy of each group of features, which evaluates the relevance and redundancy of the selected feature subset. Different weights for the relevance and redundancy in the fitness functions of the two proposed algorithms are used to further improve their performance in terms of the number of features and the classification accuracy. In the experiments, a decision tree (DT) is employed to evaluate the classification accuracy of the selected feature subset on the test sets of four datasets. The results show that with proper weights, two proposed algorithms can significantly reduce the number of features and achieve similar or even higher classification accuracy in almost all cases. The first algorithm usually selects a smaller feature subset while the second algorithm can achieve higher classification accuracy.**

## I. Introduction

In many problems such as classification, a large number of features are introduced into the dataset to well describe the target concepts. However, the large number of features causes the problem known as "the curse of dimensionality", which is a major obstacle in classification. Meanwhile, the presence of less relevant or highly correlated features often decrease the classification performance. Feature selection is an essential and widely used technique to deal with the large data size problem [1]. For a given classification task, feature selection can be described as follows: given the original set $G$ consisting of $n$ available features, find a feature subset $F$ consisting of $m$ relevant features, where $m < n$ and $F \subset G$ without replacement [2]. Feature selection reduces the number of features through eliminating irrelevant and redundant features, and thus results in enhanced efficiency and increased classification accuracy [1].

A feature selection algorithm explores the search space of different feature combinations to optimise the classification performance. Evaluation criterion and search strategy are two key parts in feature selection. According to the evaluation criterion, feature selection algorithms can be categorized into wrapper approaches and filter approaches. In a wrapper approach, a learning algorithm is used as part of the evaluation function to determine the fitness of the selected feature subset. Wrappers can usually achieve better results than filter approaches, but the main drawbacks are their computational deficiency and loss of generality [3]. In a filter approach, feature selection is done as a preprocessing procedure and the search process is independent of a learning algorithm. Therefore, the performance of a filter approach relies mainly on the goodness of the evaluation criterion. Many different criteria have been used in filter approaches, including information measures [4], dependency measures [5], consistency measures [6], and distance measures [7]. Compared with wrappers, filter approaches are argued to be computationally less expensive and more general [1].

In feature selection, the size of the search space for $n$ features is $2^n$. So in most situations, it is impractical to conduct an exhaustive search for feature selection [3]. Therefore, the search strategy can significantly influence the results of a feature selection approach. Many search techniques have been applied in feature selection such as greedy search, but most of them usually suffer from the problem of becoming stuck in local optima and/or high computational cost [8, 9]. Therefore, a computationally cheap global search technique is needed to develop a good feature selection algorithm.

Evolutionary computation techniques are well-known for their global search ability, and have been applied to feature selection problems. These includes particle swarm optimisation (PSO) [10, 11], genetic algorithms (GAs) [6] and genetic programming (GP) [12]. Compared with GAs and GP, PSO is easier to implement, has fewer parameters, computationally less expensive, and can converge more quickly [13]. Due to these advantages, PSO has been used as a promising method for feature selection problems [10, 11]. However, most of existing PSO based feature selection algorithms are wrapper approaches, which may obtain low performance in other learning algorithms and also sometimes are practically impossible because of the high computational cost. Few studies have been conducted on using fuzzy sets and rough sets theories in PSO based filter feature selection algorithms [14, 15]. Information theory is one of the most important theories that are capable to measure the relevance between features and class labels [1]. However, not much work has been conducted to investigate the use of information theory in a PSO based feature selection approach.

### A. Goals

This paper aims to develop a filter based feature selection approach using PSO and information theory with the expectation of selecting a small number of features to achieve similar or even higher classification accuracy than using all features. To achieve this goal, we will develop two new filter feature selection algorithms based on PSO and two information measurements for finding a subset of features for classification. The two new feature selection methods will be examined on four benchmark datasets with different numbers of features and instances. Specifically, we will

- develop a filter feature selection algorithm based on PSO and the mutual information of each pair of features, which is used to evaluate the relevance and redundancy in the selected feature subset, and investigate whether this algorithm can select a small number of features to achieve better performance than using all features and can outperform conventional approaches;
- develop a filter feature selection algorithm based on PSO and the entropy of each group of features, which is applied to evaluate the relevance and redundancy in the selected feature subset, and investigate whether this algorithm can outperform the method of using all features, conventional approaches and the first proposed algorithm;
- investigate whether using different weights for relevance and redundancy in the first algorithm could further reduce the number of features and improve the classification performance; and
- investigate whether using different weights for relevance and redundancy in the second algorithm can further increase the performance in terms of the number of features and classification performance.

## II. Background

This section provides some background information about PSO, entropy and mutual information in information theory, and also reviews typical related work on feature selection.

### A. Particle Swarm Optimisation (PSO)

PSO is an evolutionary computation technique proposed by Kennedy and Eberhart in 1995 [16]. PSO simulates the social behaviour such as birds flocking and fish schooling. In PSO, a population, also called a *swarm*, of candidate solutions are encoded as particles in the search space. PSO starts with the random initialisation of a population of particles. The whole swarm move in the search space to search for the best solution by updating the position of each particle based on the experience of its own and its neighbouring particles [17]. During movement, the current position of particle $i$ is represented by a vector $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$, where $D$ is the dimensionality of the search space. The velocity of particle $i$ is represented as $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$, which is limited by a predefined maximum velocity, $v_{max}$ and $v_{id}^t \in [-v_{max}, v_{max}]$. The best previous position of a particle is recorded as the personal best called *pbest* and the best position obtained by the population thus far is called *gbest*. Based on *pbest* and *gbest*, PSO searches for the optimal solution by updating the velocity and the position of each particle according to the following equations:

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \tag{1}$$

$$\begin{aligned} v_{id}^{t+1} &= w * v_{id}^t + c_1 * r_{1i} * (p_{id} - x_{id}^t) \\ &+ c_2 * r_{2i} * (p_{gd} - x_{id}^t) \end{aligned} \tag{2}$$

where $t$ denotes the $t$th iteration in the search process. $d \in D$ denotes the $d$th dimension in the search space. $w$ is inertia weight. $c_1$ and $c_2$ are acceleration constants. $r_{1i}$ and $r_{2i}$ are random values uniformly distributed in [0, 1]. $p_{id}$ and $p_{gd}$ represent the elements of *pbest* and *gbest* in the $d$th dimension.

PSO was originally proposed for solving problems in real-number search spaces. However, many optimisation problems, such as feature selection, occur in a discrete search space. For this reason, Kennedy and Eberhart [18] developed a binary particle swarm optimisation (BPSO) for discrete problems. In BPSO, Equation (2) is still applied to update the velocity, where $x_{id}$, $p_{id}$ and $p_{gd}$ are restricted to 1 or 0. The velocity in BPSO indicates the probability of the corresponding element in the position vector taking value 1. A sigmoid function $s(v_{id})$ is introduced to transform $v_{id}$ to the range of (0, 1). BPSO updates the position of each particle according to the following formulae:

$$x_{id} = \begin{cases} 1, \text{ if } rand() < s(v_{id}) \\ 0, \text{ otherwise} \end{cases} \tag{3}$$

where

$$s(v_{id}) = \frac{1}{1 + e^{-v_{id}}} \tag{4}$$

where $rand()$ is a random number selected from a uniform distribution in [0,1].

### B. Entropy and Mutual Information

Information theory developed by Shannon [19] provides a way to measure the information of the random variables with entropy and mutual information.

The entropy is a measure of the uncertainty of random variables. Let $X$ be a random variable with discrete values, its uncertainty can be measured by entropy $H(X)$, which is defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \tag{5}$$

where $p(x) = Pr(X = x)$ is the probability density function of $X$. Note that entropy does not depend on actual values, but just the probability distribution of the random variable.

For two discrete random variables $X$ and $Y$ with their probability density function $p(x, y)$, the joint entropy $H(X, Y)$ is defined as

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

When a certain variable is known and others are unknown, the remaining uncertainty is measured by the conditional entropy. Assume that variable $Y$ is given, the conditional entropy $H(X|Y)$ of $X$ with respect to $Y$ is

$$H(X|Y) = -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log_2 p(x|y) \quad (7)$$

where $p(x|y)$ is the posterior probabilities of $X$ given $Y$. From this definition, if $X$ completely depends on $Y$, then $H(X|Y)$ is zero, which means that no more other information is required to describe $X$ when $Y$ is known. Otherwise, $H(X|Y) = H(X)$ denotes that knowing $Y$ will do nothing to observe $X$.

The information shared between two random variables is defined as mutual information. Given variable $X$, how much information one can gain about variable $Y$, which is mutual information $I(X;Y)$.

$$\begin{aligned} I(X;Y) &= H(X) - H(X|Y) \\ &= H(Y) - H(Y|X) \\ &= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log_2 \frac{p(x,y)}{p(x)p(y)} \end{aligned} \quad (8)$$

According to Equation 8, the mutual information $I(X;Y)$ will be large if two variables $X$ and $Y$ are closely related. Otherwise, $I(X;Y) = 0$ if $X$ and $Y$ are totally unrelated.

### C. Recent Work Related to Feature Selection

Many filter feature selection algorithms have been proposed and typical algorithms are reviewed in this section.

#### 1) Classical Feature Selection Approaches

The FOCUS algorithm is a classical filter feature selection algorithm, which starts with an empty feature subset and exhaustively examines all subsets of features, then selects the minimal subset of features. However, the FOCUS algorithm performs an exhaustive search to determine the best feature subset, which is computationally expensive.

The Relief algorithm is another popular filter feature selection method that assigns a relevance weight to each feature [20]. The weight is intended to denote the relevance of the feature to the target concept. However, Relief does not deal with redundant features, because it attempts to find all relevant features regardless of the redundancy between them [21].

Decision trees (DT) use only relevant features that are required to completely classify the training set and remove all other features. Cardie [22] proposes a filter based feature selection algorithm that uses a DT to select a subset of features for a nearest neighbourhood algorithm. However, the features that are good (or not good) for DT are not necessarily useful (or not useful) for the nearest neighbour algorithm, which will lead to poor feature selection performance.

Sequential forward selection (SFS) [8] and sequential backward selection (SBS) [9] are two popular wrapper feature selection approaches. A greedy hill-climbing search strategy is applied in both approaches to search for the best feature subset. However, both SFS and SBS suffer from the so-called nesting effect and easily trapped into local optima.

#### 2) BPSO based Feature Selection Approaches

BPSO has recently gained more attention for solving feature selection problems. Based on BPSO, both filter and wrapper feature selection approaches have been proposed, but most of them are wrapper approaches. Some PSO based filter feature selection are reviewed in this section.

Chakraborty [14] proposes a BPSO based filter feature selection algorithm with a fuzzy sets based fitness function. The idea of the fuzzy sets for feature selection is to minimise the ambiguity associated within the class and maximise the ambiguity between the classes. The performance of BPSO is compared with that of GA in two benchmark datasets. Experimental results show that the BPSO based feature selection algorithm could achieve slightly higher classification accuracy and computationally less expensive than the GA based algorithm. However, only using two datasets in the experiment is not enough to verify the effectiveness of the proposed algorithm.

Since rough sets can handle imprecision, uncertainty and vagueness, Wang et al. [15] proposes a filter feature selection approach based on an improved BPSO (IBPSO) and rough sets theories. In IBPSO, the velocity is defined as a positive integer to determine how many bits in the position should be changed. According to the rough sets theories, the dependency degree of classes on features is measured and used to evaluate the fitness of each particle. Experiments show that the proposed algorithm is computationally less expensive than a GA based filter feature selection algorithm in terms of both memory and running time. This work also shows that the computation of the rough sets consumes most of the running time, which is a drawback of using rough sets in feature selection problems.

Yang et al. [11] propose a feature selection approach in which *gbest* of a particle will be reset after being identical for three iterations. A Boolean operator 'and(.)' will 'and' each bit of *pbest* of all particles in an attempt to create a new *gbest*. Experimental results show that the proposed method usually achieves better classification performance than GA and standard BPSO based feature selection approaches.

Based on a modified BPSO and a logistic regression model, Unler et al. [10] propose a wrapper feature selection algorithm. Social learning is introduced into BPSO to update the velocity of the particles. An adaptive feature selection strategy is developed in the proposed algorithm, where the features are selected not only according to the likelihood calculated by BPSO, but also according to their contribution to the subset of features already selected. Compared with tabu search and scatter search algorithms, the proposed algorithm can achieve better performance.

#### 3) Other Evolutionary Computation Methods for Feature Selection

GP, GAs and and ant colony optimisation (ACO) are also applied to feature selection problems.

Based on GP and a variation of naïve bayes (NB), Neshatian and Zhang [12] propose a feature selection approach, where a bit-mask representation is used for feature subsets and a set of operators are used as primitive functions. GP is

used to combine feature subsets and operators together to find the optimal subset of features. Experiments show that the dimensionality and processing time can be significantly reduced by the proposed algorithm.

Chakraborty [23] proposes a GA with fuzzy sets based fitness function to build a filter feature selection approach. This method have the same fitness function with BPSO based method in [14]. However, the performance of BPSO in [14] is better than that of this GA based algorithm.

Based on ACO and rough sets theory, He [24] proposes a filter based feature selection approach. The features included in the core of the rough sets is the starting point of the proposed method. Forward selection is adopted into the proposed method search for the best feature subset. Experimental results show that the proposed approach achieves better classification performance with fewer features than a C4.5 based feature selection approach. However, experiments do not compare the proposed method with other commonly used feature selection approaches.

BPSO has been shown to be an efficient search technique for feature selection by many existing studies. However, most of the existing approaches are wrappers, which are computationally expensive and less general than filter approaches. A relatively small number of BPSO based filter feature selection approaches have been proposed in which rough sets and fuzzy sets theories are mainly used to evaluate the fitness of the selected features. However, Wang et al. [15] has already shown the drawback of using rough sets theories. There are a variety of other measures that can be used in a filter based feature selection approach, which may achieve better performance than using rough sets and fuzzy sets theories. Therefore, investigation of an effective BPSO based filter feature selection algorithm is still an open issue and we make an effort in this paper.

## III. Proposed Filter Based Methods

In this section, two BPSO based filter feature selection approaches are proposed, where mutual information and entropy are applied to evaluate the relevance and the redundancy in the selected feature subsets.

### A. BPSO with Paired Evaluation for Feature Selection

Mutual information is defined as the information shared between two random variables, which can be used in feature selection to evaluate the relevance between features and class labels. However, in feature selection, because of the interactions between features, the combination of $m$ individually good features may not be the best combination of $m$ features. Therefore, it is necessary to reduce the redundancy among features and select a subset of features with minimal redundancy to each other and maximal relevance to class labels. For this reason, both relevance and redundancy are included in the fitness function to guide BPSO to search for the best feature subset, which can be represented by Equation 9.

$$Fitness_1 = D_1 - R_1 \tag{9}$$

**Algorithm 1:** The BPSO based feature selection algorithm

```
1  begin
2      divide Dataset into a Training set and a Test set;
3      initialise the position and velocity of each particle;
4      while Maximum Iterations or the stopping
       criterion is not met do
5          evaluate fitness of each particle according to
           Equation 9 on the Training set;
6          for i=1 to P do
7              update the pbest of particle i;
8              update the gbest of particle i;
9          end
10         for i=1 to Population Size do
11             for d=1 to Dimensionality do
12                 update the velocity of particle i according
                   to Equation 2;
13                 update the position of particle i according
                   to Equations 3 and 4;
14             end
15         end
16     end
17     calculate the classification accuracy of the selected
       feature subset on the test set;
18     return the position of gbest (the selected feature
       subset);
19     return the training and test classification accuracies;
20 end
```

where

$$D_1 = \sum_{x \in X} I(x; c),$$

$$R_1 = \sum_{x_i, x_j \in X} I(x_i, x_j).$$

where $X$ is the set of selected features and $c$ is the class label. Each selected feature and the class labels are treated as discrete random variables. $D_1$ uses pair wise calculations to calculate the mutual information between each feature and the class labels, which determine the relevance of the selected feature subset to the class labels. $R_1$ evaluates the mutual information shared by each pair of selected features, which indicates the redundancy contained in the selected feature subset. $Fitness_1$ is a maximisation function to maximise the relevance $D_1$ and simultaneously minimise the redundancy $R_1$ in the selected feature subset.

Algorithm 1 shows the pseudo-code of using BPSO with paired evaluation for feature selection. The representation of a particle in BPSO is a $n$-bit binary string, where $n$ is the number of available features in the dataset and also the dimensionality of the search space. In the binary string, "1" represents that the feature is selected and "0" otherwise.

### B. BPSO with Group Evaluation for Feature Selection

Feature interaction is one of the reasons that make feature selection a challenging problem. Feature interaction can be

in two-way or multi-way. Therefore, the relevance and redundancy among features can also be in two-way or multi-way. $Fitness_1$ evaluates the two-way relevance and redundancy by evaluating mutual information in pairs of features. The multi-way relevance and redundancy should be evaluated in groups of features. Therefore, we propose a new BPSO based filter feature selection algorithm with the fitness function of evaluating the selected features as a whole rather than a pair of features. The fitness function is defined in Equation 10.

$$Fitness_2 = D_2 - R_2 \tag{10}$$

where

$$D_2 = IG(c|X)$$

$$R_2 = \frac{1}{|S|} \sum_{x \in X} IG(x|\{X/x\})$$

where $X$ and $c$ have the same meaning as in $Fitness_1$ (Equation 9). Each selected feature and the class label are also treated as discrete random variables. $D_2$ evaluates the information gain in $c$ given information of the features in $X$, which show the relevance between the selected feature subset and the class labels. $R_2$ evaluates the joint entropy of all the features in $X$, which indicates the redundancy contained in the selected feature subset. $Fitness_2$ is a maximisation function to maximise the relevance $D_2$ and simultaneously minimise the redundancy $R_2$ among selected features.

Both $D_2$ and $R_2$ involve the calculation of a single discrete feature given information about a set of discrete features. Taken $D_2$ as the example,

$$\begin{aligned} D_2 &= IG(c|X) \\ &= H(c) - H(c|X) \\ &= H(c) - (H(c \cup X) - H(X)) \\ &= H(c) + H(X) - H(c \cup X) \end{aligned}$$

where $H(X)$ is the joint entropy of all the features in $X$. If $X = W, Y, Z$, then

$$H(W, Y, Z) = -\sum_{w \in W} \sum_{y \in Y} \sum_{z \in Z} p(wyz) \log_2 p(wyz).$$

The representation of a particle in this algorithm is the same as the $n$-bit binary string described in Section III-A. Algorithm 1 also can be used to show the pseudo-code of this algorithm by replacing the Equation 9 with Equation 10 in Line 5.

### *C. Different Weights for Relevance and Redundancy in Two Proposed Algorithms*

In the two proposed feature selection methods, the relevance and redundancy are equally important in the two fitness functions (Equations 9 and 10). In order to investigate whether using different weights that show the relative importance for the relevance and redundancy in the two proposed algorithms can further improve the performance, a parameter is introduced into each of the fitness function, which can be seen in Equations 11 and 12.

$$Fitness_1 = \alpha_1 * D_1 - (1 - \alpha_1) * R_1 \tag{11}$$

TABLE I
DATASETS

| Dataset | Datatype | Instances | Atributes | Classes |
|---|---|---|---|---|
| Chess | Categorical | 3196 | 36 | 2 |
| Splice | Categorical | 3190 | 61 | 3 |
| Spect | Categorical | 267 | 22 | 2 |
| Lymphography (Lymph) | Categorical | 148 | 18 | 4 |

$$Fitness_2 = \alpha_2 * D_2 - (1 - \alpha_2) * R_2 \tag{12}$$

where $\alpha_1$ and $\alpha_2$ are constant values and $\alpha_1,\ \alpha_2 \in [0, 1]$. $\alpha_1$ and $\alpha_2$ show the relative importance of the relevance in two fitness functions. $(1 - \alpha_1)$ and $(1 - \alpha_2)$ show the relative importance of the reduction of the redundancy. As the relevance is assumed to be more important than the redundancy, $\alpha_1$ or $\alpha_2$ is set to be larger than $(1 - \alpha_1)$ or $(1 - \alpha_2)$ in two fitness functions. Note that when $\alpha_1 = 0.5$ $(1 - \alpha_1 = 0.5)$ and $\alpha_2 = 0.5$ $(1 - \alpha_2 = 0.5)$, the algorithms actually are the same as without any weights (relevance and redundancy are equally important) in the fitness functions (See Equations 9 and 10).

## IV. EXPERIMENTAL DESIGN

Four benchmark datasets chosen from the UCI machine learning repository [25] are used in the experiments, which can be seen in Table I. The four datasets were selected to have different numbers of features, classes and instances as the representative samples of the problems that the proposed approaches can address. As the Chess and Splice datasets have a large number of instances, their instances are randomly divided into two sets: 70% as the training set and 30% as the test set. For the Spect and Lymph datasets with a small number of instances, 10-fold cross-validation is applied. The two proposed algorithms firstly run on the training set to select feature subsets and then the classification performance of the selected features will be calculated on the test set by a learning algorithm. There are many learning algorithms that can be used here, such as K-nearest neighbour (KNN), NB, and DT. A DT learning algorithm is selected in this study to calculate the classification accuracy of the selected features according to Equation 13:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

where TP, TN, FP and FN stand for true positives, true negatives, false positives and false negatives, respectively.

The parameters of BPSO are set as follows: inertia weight $w = 0.7298$, acceleration constants $c_1 = c_2 = 1.49618$, maximum velocity $v_{max} = 6.0$, population size $P = 30$, maximum iteration $T = 500$. The fully connected topology is used in BPSO. These values are chosen based on the common settings in the literature [26]. Five different values for $\alpha_1$ and $\alpha_2$ are used in the experiments, which are 0.9, 0.8, 0.75, 0.6 and 0.5. For each dataset, each approach has been conducted for 30 independent runs.

TABLE II
EXPERIMENTAL RESULTS OF TWO PROPOSED ALGORITHMS

| Dataset | Method | Ave-Size | Ave-Acc (Best-Acc) | Std-Acc |
|---|---|---|---|---|
| Chess | All | 36 | 0.985 | |
| | BPSO-P | 4.7 | 0.797 (0.902) | 0.027 |
| | BPSO-G | 15.7 | 0.970 (0.977) | 0.011 |
| Splice | All | 60 | 0.920 | |
| | BPSO-P | 8.1 | 0.781 (0.862) | 0.050 |
| | BPSO-G | 7.4 | 0.723 (0.877) | 0.094 |
| Lymph | All | 18 | 0.755 | |
| | BPSO-P | 3 | 0.711 (0.711) | 0.000 |
| | BPSO-G | 6.3 | 0.740 (0.778) | 0.017 |
| Spect | All | 22 | 0.809 | |
| | BPSO-P | 3.1 | 0.783 (0.794) | 0.002 |
| | BPSO-G | 4.5 | 0.812 (0.828) | 0.010 |

TABLE III
THE FIRST PROPOSED ALGORITHMS WITH DIFFERENT $\alpha_1$

| Dataset | $\alpha_1$ | Ave-Size | Ave-Acc (Best-Acc) | Std-Acc |
|---|---|---|---|---|
| Chess | All | 36 | 0.985 | |
| | 0.9 | 8.2 | 0.915 (0.940) | 0.026 |
| | 0.8 | 6.3 | 0.874 (0.938) | 0.055 |
| | 0.75 | 5.8 | 0.852 (0.938) | 0.053 |
| | 0.6 | 5.3 | 0.824 (0.938) | 0.049 |
| | 0.5 | 4.7 | 0.797 (0.902) | 0.027 |
| Splice | All | 60 | 0.920 | |
| | 0.9 | 19.9 | 0.927 (0.933) | 0.005 |
| | 0.8 | 13.4 | 0.927 (0.935) | 0.005 |
| | 0.75 | 12.1 | 0.918 (0.937) | 0.013 |
| | 0.6 | 9.2 | 0.841 (0.896) | 0.030 |
| | 0.5 | 8.1 | 0.781 (0.862) | 0.050 |
| Lymph | All | 18 | 0.755 | |
| | 0.9 | 8.3 | 0.755 (0.757) | 0.000 |
| | 0.8 | 5.5 | 0.752 (0.758) | 0.001 |
| | 0.75 | 4.9 | 0.744 (0.744) | 0.000 |
| | 0.6 | 3.3 | 0.751 (0.751) | 0.000 |
| | 0.5 | 3 | 0.711 (0.711) | 0.000 |
| Spect | All | 22 | 0.809 | |
| | 0.9 | 7.6 | 0.818 (0.824) | 0.006 |
| | 0.8 | 4.4 | 0.808 (0.817) | 0.005 |
| | 0.75 | 4.1 | 0.799 (0.809) | 0.005 |
| | 0.6 | 3.7 | 0.798 (0.802) | 0.004 |
| | 0.5 | 3.1 | 0.783 (0.794) | 0.002 |

## V. RESULTS AND DISCUSSIONS

Experimental results are shown in Tables II, III and IV. Table II show the results of the proposed two approaches without weights in the fitness functions. In Table II, "Ave-Size" represents the average size of the feature subsets evolved by each algorithm in 30 independent runs. "Ave-Acc" shows the average test accuracy of the selected feature subsets in the 30 runs. and "Best-Acc" indicates the best test accuracy. "Std-Acc" represents the standard deviation of the 30 test accuracies. "All" means that all of the available features are used for classification. "BPSO-P" stands for the first algorithm, BPSO with paired evaluation for feature selection. "BPSO-G" represents the second proposed algorithm, BPSO with group evaluation for feature selection. Tables III and IV show the experimental results of the proposed two approaches with weights in the fitness functions. In these two tables, "Ave-Size", "Ave-Acc", "Best-Acc", "Std-Acc" and "All" have the same meaning as in Table II.

### *A. Results of BPSO with Paired Evaluation for Feature Selection*

According to the results ("BPSO-P") shown in Table II, it can be seen that in almost all the datasets, the feature subset selected evolved by "BPSO-P" reduces at least 83% of the available features. With the small selected feature subset, DT can achieve similar (or slightly worse) classification accuracy with using all features in the Lymph and Spect datasets. In the Chess and Splice dataset, the average classification accuracy drops more, but around 87% of the features are reduced.

As can be seen in Table II, all the standard deviation values in the four datasets shown by "Std-Acc" are smaller than 0.05, which shows that the first algorithm is stable.

The results suggest that BPSO with paired evaluation for feature selection can significantly reduce the number of features needed for classification and achieve similar classification performance with all features.

TABLE IV
THE SECOND PROPOSED ALGORITHMS WITH DIFFERENT $\alpha_2$

| Dataset | $\alpha_2$ | Ave-Size | Ave-Acc (Best-Acc) | Std-Acc |
|---|---|---|---|---|
| Chess | All | 36 | 0.985 | |
| | 0.9 | 24.7 | 0.986 (0.987) | 0.001 |
| | 0.8 | 22.6 | 0.986 (0.986) | 0.001 |
| | 0.75 | 22.4 | 0.985 (0.987) | 0.002 |
| | 0.6 | 19.1 | 0.977 (0.985) | 0.003 |
| | 0.5 | 15.7 | 0.970 (0.977) | 0.011 |
| Splice | All | 60 | 0.920 | |
| | 0.9 | 10.1 | 0.884 (0.931) | 0.034 |
| | 0.8 | 8.1 | 0.833 (0.928) | 0.059 |
| | 0.75 | 7.1 | 0.790 (0.916) | 0.068 |
| | 0.6 | 6.4 | 0.733 (0.877) | 0.092 |
| | 0.5 | 7.4 | 0.723 (0.877) | 0.094 |
| Lymph | All | 18 | 0.755 | |
| | 0.9 | 10.4 | 0.745 (0.785) | 0.016 |
| | 0.8 | 9.6 | 0.746 (0.804) | 0.022 |
| | 0.75 | 9.2 | 0.741 (0.785) | 0.022 |
| | 0.6 | 7.4 | 0.712 (0.744) | 0.017 |
| | 0.5 | 6.3 | 0.740 (0.778) | 0.017 |
| Spect | All | 22 | 0.809 | |
| | 0.9 | 17.2 | 0.812 (0.817) | 0.005 |
| | 0.8 | 15.6 | 0.811 (0.824) | 0.004 |
| | 0.75 | 14.3 | 0.807 (0.817) | 0.006 |
| | 0.6 | 5.2 | 0.809 (0.835) | 0.011 |
| | 0.5 | 4.5 | 0.812 (0.828) | 0.010 |

### *B. Results of BPSO with Group Evaluation for Feature Selection*

According to the results ("BPSO-G") shown in Table II, in three of the four cases, the feature subsets evolved by "BPSO-G" contains less than half of the available features. With the selected feature subset, the DT classifier can achieve similar or even better classification accuracy than using all features. For example, in the Spect dataset, the feature subsets resulted from "BPSO-G" consist of only 20% of the available features and can achieve average higher classification accuracy than using all features. Only in the Splice dataset is that the classification accuracy drops slightly, but around 88% of the features are reduced. All the standard deviation values in the four datasets are smaller than 0.1.

Comparing "BPSO-G" with "BPSO-P", in most cases, the average size of the feature subsets evolved by "BPSO-G" is slightly larger than that of "BPSO-P". However, the average classification accuracy achieved by the feature subsets resulted from "BPSO-G" is higher than that of "BPSO-P" in three of the four datasets (except for around 4% lower in the Splice

dataset). In general, the standard deviation values of "BPSO-P" are smaller than that of "BPSO-G", which means the first algorithm is more stable than the second algorithm. The reason might be that the second algorithm employs a more complicated fitness function than the first algorithm, which makes the search space more complicated.

The results show that "BPSO-G" can significantly reduce the number of features needed for classification and maintain the similar or even better classification accuracy than using all available features. In the Splice dataset, the reduction of the features is almost 90% in both "BPSO-P" and "BPSO-G", but the classification accuracy is slightly worse than using all features. Therefore, in the following two subsections, we will investigate whether the classification performance can be increased and the number of features can be further reduced by using different weights for the redundancy and relevance in two fitness functions.

### *C. Results of BPSO with Paired Evaluation for Feature Selection Using Different $\alpha_1$*

According to Table III, it can be seen that with at least one of the $\alpha_1$ values, BPSO can evolve a small number of features and achieve better classification than using all features in three of the four datasets. In the Chess dataset, where the classification accuracy using all features is already very high (0.985), the number of features is significantly reduced although the classification accuracy is slightly decreased.

In all datasets, BPSO with a large $\alpha_1$ usually evolves a subset with more features and achieve better classification performance than with a small $\alpha_1$. This is because when $\alpha_1$ is large, the relevance is more important and the redundancy, which indirectly influence the number of features, is less important than when $\alpha_1$ is small. In three of the four cases (except the Chess dataset), BPSO with $\alpha_1 = 0.9$ and $\alpha_1 = 0.8$ can obtain a feature subset with 20-40% of the available features and achieve better classification performance than using all features. All the standard deviation values in the four datasets are smaller than 0.6.

The results show that different weights for the relevance and redundancy in the fitness function can effectively influence the results evolved by BPSO in terms of the number of features selected and the classification performance. By using a proper value for $\alpha_1$, BPSO can evolve a feature subset with a small number of features and achieve better classification performance than using all features in almost all cases.

### *D. Results of BPSO with Group Evaluation for Feature Selection Using Different $\alpha_2$*

According to the results in Table IV, it can be seen that in three of the four datasets, with at least one of the $\alpha_2$ values, BPSO can evolve a small number of features and achieve better classification than using all features. Only in the Splice dataset, the classification accuracy is decreased, but the reduction of the number of features is at least 83%. However, the best accuracy ("Best-Acc") is higher than using all features when $\alpha_2 = 0.9$ and $\alpha_2 = 0.8$.

In all datasets, BPSO with a large $\alpha_2$ usually evolves a subset with more features and achieve better classification performance than with a small $\alpha_2$. The reason is the same as discussed in Section V-C. In almost all cases (except the Splice dataset), BPSO with $\alpha_2 = 0.9$ and $\alpha_2 = 0.8$ can obtain a feature subset with a small number of features and achieve better classification performance than using all features.

Comparing the results in Table IV with in Table III, for the same value of $\alpha$, neither of the two proposed algorithms consistently dominate the other one on four datasets. The first algorithm usually outperforms the second algorithm in terms of the number of features while the second algorithm achieve higher classification accuracy.

The results show that the number of features and classification accuracy are influenced by the value of $\alpha_1$ and $\alpha_2$ in the fitness functions. By using a proper value for $\alpha_1$ and $\alpha_2$, BPSO can evolve a feature subset with a small number of features and achieve better classification performance than using all features in almost all cases.

### *E. Further Analysis*

Results in Tables II, III and IV show that the first algorithm, BPSO with paired evaluation, usually selects a smaller number of features while the second algorithm, BPSO with group evaluation, achieves better classification performance. In order to further investigate the difference between two proposed algorithms, it is necessary to analyse the selected features.

Considering the Chess dataset as an example, "BPSO-P" and "BPSO-G" share the same parameter settings (except for the fitness function) and start with the same initialisation in each of the 30 runs. Fitness function in "BPSO-P" evaluate the mutual information of pairs of features, which is to discover the two-way relevance and redundancy caused by feature interaction. Fitness function in "BPSO-G" evaluate the entropy of groups of features, which is to discover the multi-way relevance and redundancy caused by feature interaction. Therefore, even starting with the same swarm, "BPSO-P" and "BPSO-G" obtain different solutions. Becuase of the discovery of the complicated multi-way relevance and redundancy, "BPSO-G" usually select more features and achieve better classification performance than "BPSO-P". The averge size of the feature subsets is 4.7 in "BPSO-P" and 15.7 in "BPSO-G". The average classification accuracy is 0.797 in "BPSO-P" and 0.970 in "BPSO-G".

Although the averge size 4.7 in "BPSO-P" is much smaller than 15.7 in "BPSO-G", in 27 of the 30 runs, not all features selected by "BPSO-P" are included in the subset resulted from "BPSO-G". Specifically, considering the results in a typical run, the numbers of features selected by "BPSO-P" and "BPSO-G" are 5 and 18, and the classification accuracies are 0.808 and 0.977. The features selected by "BPSO-P" are F9, F10, F16, F21, and F27, where F$i$ denotes the $i$th feature in the dataset. The features selected by "BPSO-G" are F1, F3, F6, F10, F14, F15, F16, F17, F18, F19, F21, F23, F25, F28, F29, F32, and F35.

The results suggest that using different measurements, the

relevant features or feature subset can be very different. The fitness functions will guide BPSO to search for different feature subsets and lead to various classification accuracies. It seems that there is a correlation between the number of features and the classification accuracy. We will investigate this in the future.

## VI. Conclusions and Future Work

The goal of this paper was to investigate a filter feature selection approach based on BPSO and information theory to select a smaller number of features and achieve similar or even better classification performance. This goal was achieved by developing two new feature selection algorithms based on BPSO and two information measures, namely entropy and mutual information, In the first algorithm, mutual information of each pair of features is used to evaluate the relevance and redundancy of the selected feature subset. In the second algorithm, entropy of each group of features is employed to evaluate the relevance and redundancy of the selected feature subset. Two proposed algorithms were examined and compared with each other with different weights for relevance and redundancy on four problems of varying difficulty.

The results suggest that with proper weights, both of the proposed approaches can significantly reduce the number of features whilst achieve similar or even better classification accuracy in almost all cases. The first algorithm usually selected a smaller feature subset while the second algorithm can achieve higher classification accuracy. Neither of the two proposed approaches nominated the other one. Therefore, in the future, we will investigate a BPSO based evolutionary multi-objective filter feature selection approach to explore the Pareto front of non-dominated solutions, which can help users make a more informed choice to balance the number of features and the classification performance according to their requirements.

## Acknowledgment

This work is supported in part by the University Research Fund (URF 101154/2880) at Victoria University of Wellington and the National Natural Science Foundation of China (NSFC No. 61170180).

## References

[1] M. Dash and H. Liu, "Feature selection for classification," *Intelligent Data Analysis*, vol. 1, no. 4, pp. 131–156, 1997.

[2] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[3] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, pp. 273–324, 1997.

[4] P. Estevez, M. Tesmer, C. Perez, and J. Zurada, "Normalized mutual information feature selection," *IEEE Transactions on Neural Networks*, vol. 20, no. 2, pp. 189–201, 2009.

[5] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal of Machine Learning Research*, vol. 5, pp. 1205–1224, 2004.

[6] H. Yuan, S. S. Tseng, and W. Gangshan, "A two-phase feature selection method using both filter and wrapper," in *IEEE International Conference on Systems, Man, and Cybernetics (SMC'99)*, vol. 2, 1999, pp. 132–136.

[7] N. Kwak and C. H. Choi, "Input feature selection for classification problems," *IEEE Transactions on Neural Networks*, vol. 13, no. 1, pp. 143–159, 2002.

[8] A. Whitney, "A direct method of nonparametric measurement selection," *IEEE Transactions on Computers*, vol. C-20, no. 9, pp. 1100–1103, 1971.

[9] 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.

[10] A. Unler and A. Murat, "A discrete particle swarm optimization method for feature selection in binary classification problems," *European Journal of Operational Research*, vol. 206, no. 3, pp. 528–539, 2010.

[11] C. S. Yang, L. Y. Chuang, C. H. Ke, and C. H. Yang, "Boolean binary particle swarm optimization for feature selection," in *IEEE Congress on Evolutionary Computation (CEC'08)*, 2008, pp. 2093–2098.

[12] K. Neshatian and M. Zhang, "Dimensionality reduction in face detection: A genetic programming approach," in *24th International Conference Image and Vision Computing New Zealand (IVCNZ'09)*, 2009, pp. 391–396.

[13] J. Kennedy and W. Spears, "Matching algorithms to problems: an experimental test of the particle swarm and some genetic algorithms on the multimodal problem generator," in *IEEE Congress on Evolutionary Computation (CEC'98)*, 1998, pp. 78–83.

[14] B. Chakraborty, "Feature subset selection by particle swarm optimization with fuzzy fitness function," in *3rd International Conference on Intelligent System and Knowledge Engineering (ISKE'08)*, vol. 1, 2008, pp. 1038–1042.

[15] X. Wang, J. Yang, X. Teng, W. Xia, and R. Jensen, "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, vol. 28, no. 4, pp. 459–471, 2007.

[16] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948.

[17] J. Kennedy, R. C. Eberhart, and Y. Shi, *Swarm Intelligence*, ser. Evolutionary Computation Series. San Francisco: Morgan Kaufman, 2001.

[18] J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm algorithm," in *IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation.*, vol. 5, 1997, pp. 4104–4108.

[19] C. Shannon and W. Weaver, *The Mathematical Theory of Communication*. Urbana:The University of Illinois Press, 1949.

[20] K. Kira and L. A. Rendell, "A practical approach to feature selection," *Assorted Conferences and Workshops*, pp. 249–256, 1992.

[21] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," *Lecture Notes in Computer Science*, vol. 784, p. 171, 1994.

[22] C. Cardie, "Using decision trees to improve case-based learning," in *In Proceedings of the Tenth International Conference on Machine Learning (ICML)*, 1993, pp. 25–32.

[23] B. Chakraborty, "Genetic algorithm with fuzzy fitness function for feature selection," in *IEEE International Symposium on Industrial Electronics (ISIE'02)*, vol. 1, 2002, pp. 315– 319.

[24] H. Ming, "A rough set based hybrid method to feature selection," in *International Symposium on Knowledge Acquisition and Modeling (KAM '08)*, 2008, pp. 585–588.

[25] A. Frank and A. Asuncion, "UCI machine learning repository," 2010.

[26] F. Van Den Bergh, "An analysis of particle swarm optimizers," Ph.D. dissertation, Pretoria, South Africa, 2002.