Connection Science
Vol. 00, No. 00, Month 200x, 1–23

RESEARCH ARTICLE

A Multi-Objective Particle Swarm Optimisation for Filter Based Feature Selection in Classification Problems

(*Received 00 Month 200x; final version received 00 Month 200x*)

Feature selection has the two main objectives of minimising the classification error rate and the number of features. Based on binary particle swarm optimisation (BPSO), we develop two novel multi-objective feature selection frameworks for classification, which are *NSBPSO* and *CMDBPSO*. Four multi-objective feature selection methods are then developed by applying mutual information and entropy as two different filter evaluation criteria in each of the proposed frameworks. The proposed algorithms are examined and compared with a single objective method on eight benchmark datasets. Experimental results show that the proposed multi-objective algorithms can evolve a set of solutions that use a smaller number of features and achieve better classification performance than using all features. In most cases, NSBPSO achieves better results than the single objective algorithm and CMDBPSO outperforms all other methods mentioned above. This work represents the first study on multi-objective BPSO for filter based feature selection.

Keywords: Feature Selection; Particle Swarm Optimisation; Multi-Objective Optimisation; Filter Approaches

1. Introduction

In classification, features that are used to describe the problem can significantly influence the classification performance. Without prior knowledge, relevant features are usually difficult to determine. Therefore, a large number of features are often involved, but not all of them are useful for classification. Irrelevant and redundant features may even reduce the classification performance due to the unnecessarily large search space. Feature selection is a data pre-processing technique to select only the relevant features for classification, which aims to reduce the number of features for classification and simultaneously increase the classification performance (Dash and Liu 1997).

Feature selection is a difficult problem because there can be complex interaction between features. An individually relevant (redundant or irrelevant) feature may become redundant (relevant) when working together with other features. A good feature subset should be a group of complementary features that span over the diverse properties of the classes to properly discriminate them. In order to find such a good feature subset, an evaluation criterion is needed to determine the goodness of the selected feature subsets, which is a key factor in feature selection. Based on the evaluation criterion, existing feature selection approaches can be broadly classified into two categories: wrapper approaches and filter approaches. Wrapper approaches include a learning/classification algorithm as part of the evaluation function to determine the goodness of the selected feature subsets. Wrappers can often achieve better results than filter approaches. As each evaluation involves a training and testing classification process, wrappers are usually more computational expensive. Meanwhile, wrapper approaches have the drawback of the loss of generality because the selected feature subset may not work well when using other

ISSN: 0954-0091 print/ISSN 1360-0494 online

DOI: 10.1080/09540090xxxxxxxxxxxx
http://www.informaworld.com

learning/classification algorithms than their internal algorithm (Kohavi and John 1997). Filter approaches use statistical characteristics of the data for evaluation and the feature selection search process is independent of a learning/classification algorithm. Compared with wrappers, filter approaches are argued to be computationally less expensive (Dash and Liu 1997).

The feature selection task is challenging also because of the large search space, which is 2^n for n features. So in most situations, it is impractical to conduct an exhaustive search for feature selection (Kohavi and John 1997). In order to solve this problem, a variety of search techniques have been applied to feature selection such as exhaustive search, complete search, greedy search, heuristic search and random search (Dash and Liu 1997, Dash and Lee 2003, Whitney 1971). However, most existing feature selection methods still suffer from a variety of problems, such as stagnation in local optima and high computational cost (Unler and Murat 2010, Liu et al 2011). In order to better address feature selection problems, an efficient global search technique is needed. Evolutionary computation techniques are well-known for their global search ability. Particle swarm optimisation (PSO) (Kennedy and Eberhart 1995, Shi and Eberhart 1998) is a relatively recent evolutionary computation technique. Compared with other evolutionary computation algorithms such as genetic algorithms (GAs) and genetic programming (GP), PSO is computationally less expensive and can converge more quickly (Engelbrecht 2007). Therefore, PSO has been used as an effective technique in many fields, including feature selection (Unler and Murat 2010, Liu et al 2011, Chuang et al 2008, Huang and Dun 2008, Mohemmed et al 2009).

Feature selection has the two main objectives of minimising both the classification error rate and the number of features. These two objectives are usually conflicting and the optimal decision needs to be made in the presence of a trade-off between the two objectives. However, most existing feature selection approaches are single objective algorithms and belong to wrapper approaches, which are "less general" than filter approaches, and often do not achieve good results because of the high computational cost. There has been no work conducted to use PSO to develop a multi-objective, filter based feature selection approach to date.

1.1. *Goals*

The overall goal of this paper is to develop a multi-objective, filter based feature selection approach to classification based on PSO and information theory to search for a set of non-dominated solutions (feature subsets), which are expected to contain a small number of features and achieve similar or even better classification performance than using all features. To achieve this goal, we will develop two information measurements (mutual information and entropy) and two multi-objective binary PSO (BPSO) frameworks, which are *NSBPSO* using the idea of non-dominated sorting and *CMDBPSO* using the ideas of crowding, mutation and dominance. Four multi-objective feature selection algorithms will be proposed by applying the two information measurements in each of the two frameworks. These proposed feature selection algorithms will be examined and compared with a single objective BPSO on eight benchmark problems of varying difficulty. Specifically, we will investigate

- whether a single objective BPSO approach with the two information measurements can select a small number of features and improve the classification performance over using all features;
- whether NSBPSO based multi-objective feature selection algorithms can evolve a smaller number of features and achieve better classification performance than the single objective approach; and

- whether CMDBPSO based multi-objective feature selection algorithms can evolve a set of good feature subsets, and can outperform all other algorithms mentioned above.

1.2. *Organisation*

The remainder of the paper is organised as follows. Section 2 provides background information. Section 3 describes the proposed BPSO based multi-objective feature selection algorithms. Section 4 describes experimental design and Section 5 presents experimental results with discussions. Section 6 provides conclusions and future work.

2. Background

This section provides background about PSO, multi-objective optimisation, mutual information and entropy, and also reviews typical related work on feature selection.

2.1. *Particle Swarm Optimisation (PSO)*

PSO is an evolutionary computation technique proposed by Kennedy and Eberhart (Kennedy and Eberhart 1995, Shi and Eberhart 1998). PSO is based on swarm intelligence and motivated by social behaviours such as birds flocking and fish schooling. The underlying phenomenon of PSO is that knowledge is optimised by social interaction in the swarm where thinking is not only personal but also social.

In PSO, each solution can be represented as a particle in the swarm. A particle has a position in the search space and a vector $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$ represents the position of particle i, where D is the dimensionality of the search space. Particles move in the search space to search for the optimal solutions. So, particle i has a velocity represented as $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$. During the movement, each particle updates its position and velocity according to the experience of its own and its neighbours. The best previous position of the particle is recorded as the personal best *pbest* and the best position obtained by the swarm thus far is called *gbest*. Based on *pbest* and *gbest*, PSO updates the velocity and the position of each particle to search for the optimal solutions. The updating formulae are shown by Equations 1 and 2.

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

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_1 * (p_{id} - x_{id}^{t}) + c_2 * r_2 * (p_{gd} - x_{id}^{t}) \tag{2}$$

where t represents the tth iteration in the evolutionary process. $d \in D$ represents the dth dimension in the search space. w is inertia weight, which is to control the impact of the previous velocities on the current velocity. c_1 and c_2 are acceleration constants. r_1 and r_2 are random values uniformly distributed in [0, 1]. p_{id} and p_{gd} denote the elements of *pbest* and *gbest* in the dth dimension. The velocity is limited by a predefined maximum velocity, v_{max} and $v_{id}^{t+1} \in [-v_{max}, v_{max}]$. The algorithm stops when a predefined criterion is met, which could be a good fitness value or a predefined maximum number of iterations. Figure 1 shows the basic steps of a PSO algorithm.

PSO was originally proposed to address continuous problems (Kennedy and Eberhart 1995). Later, Kennedy and Eberhart (Kennedy and Eberhart 1997) developed a binary PSO (BPSO) to solve discrete problems. In BPSO, 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 is used to transform v_{id} to the range of (0, 1). BPSO updates the position of each particle according to the following formula:

$$x_{id} = \begin{cases} 1, \text{ if } rand() < \frac{1}{1+e^{-v_{id}}} \\ 0, \ otherwise \end{cases} \tag{3}$$

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

2.2. *Multi-Objective Optimisation*

A problem can be called a multi-objective problem when optimal decisions need to be taken in the presence of trade-offs between two or more conflicting objectives. Multi-objective optimisation involves minimising or maximising multiple conflicting objective functions. In mathematical terms, the formulae of a multi-objective minimisation problem can be written as follows:

$$minimise \ \ F(x) = [f_1(x), f_2(x), \ ... \ , f_k(x)] \tag{4}$$

subject to:

$$g_i(x) \leq 0, \ i = 1, 2, \ ... \ m \tag{5}$$

$$h_i(x) = 0, \ i = 1, 2, \ ... \ l \tag{6}$$

where x is the vector of decision variables, $f_i(x)$ is a function of x, k is the number of objective functions to be minimised, $g_i(x)$ and $h_i(x)$ are the constraint functions.

In multi-objective optimisation, the quality of a solution is explained in terms of trade-offs between conflicting objectives. Let y and z be two solutions of the above k-objective minimisation problem. If the following conditions are met, one can say y dominates z (or z is dominated by y, or y is better than z):

$$\forall i : f_i(y) \leq f_i(z) \quad and \quad \exists j : f_j(y) < f_j(z) \tag{7}$$

where $i, j \in \{1, 2, 3, ...k\}$. Take a two-objective minimisation problem (shown in Figure 2) as an example, x_1 dominates both x_2 and x_3. For the case that neither x_2 dominates x_3 nor x_3 dominates x_2, x_2 and x_3 are called non-dominated solutions or trade-off solutions of each other. When a solution is not dominated by any other solutions, it is referred as a Pareto-optimal solution. The set of all Pareto-optimal solutions forms the trade-off surface in the search space, the *Pareto front*. A multi-objective algorithm is designed to search for a set of non-dominated solutions.

Feature selection has two main conflicting objectives, which are minimising both the number of features and the classification error rate. Therefore, feature selection can be expressed as a two-objective minimisation problem.

2.3. *Entropy and Mutual Information*

Entropy and mutual information in information theory are able to measure the information of random variables (Shannon and Weaver 1949). 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 $H(X)$ defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x) \tag{8}$$

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

$$H(X,Y) = -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log_2 p(x,y) \tag{9}$$

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

$$H(X|Y) = -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log_2 p(x|y) \tag{10}$$

where $p(x|y)$ is the posterior probabilities of X given Y. $H(X|Y) = 0$ means that X completely depends on Y and no more other information is required to describe X when Y is known. $H(X|Y) = H(X)$ denotes that knowing Y will do nothing to observe X.

Mutual information defines the information shared between two random variables. Given variable X, mutual information $I(X;Y)$ is how much information one can gain about variable Y.

$$\begin{aligned} I(X;Y) =& H(X) - H(X|Y) \\ =& H(Y) - H(Y|X) \\ =& -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log_2 \frac{p(x,y)}{p(x)p(y)} \end{aligned} \tag{11}$$

According to Equation 11, the mutual information $I(X;Y)$ will be large if two variables X and Y are closely related. $I(X;Y) = 0$ if X and Y are totally unrelated.

2.4. *Related Work on Feature Selection*

A number of feature selection algorithms have been proposed in recent years, which can be classified into wrapper approaches and filter approaches (Dash and Liu 1997). Typical feature selection algorithms are reviewed in this section.

2.4.1. Wrapper Feature Selection Algorithms

Sequential forward selection (SFS) (Whitney 1971) and sequential backward selection (SBS) (Marill and Green 1963) are two commonly used wrapper feature selection methods. SFS (SBS) starts with no features (all features), then candidate features are sequentially added to (removed from) the initial feature subset until the further addition (removal) does not increase the classification performance. The limitation of SFS and SBS is that once a feature is selected (eliminated) it cannot be eliminated (selected) later, which is so-called nesting effect (Yusta 2009). This limitation can be overcomed by combining both SFS and SBS into one algorithm. Therefore, the "plus-l-take away-r" method was proposed by Stearns (1976), which performs l times forward selection followed by r times backward elimination. The challenge is to determine the optimal values of (l, r). To address this challenge, Pudil et al (1994) proposed two floating feature selection algorithms, namely sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS). SFFS and SBFS automatically determine the values for (l, r). These two floating methods are regarded to be at least as good as the best sequential method, but they still suffer from the problem of stagnation in local optima (Yusta 2009).

Evolutionary computation techniques have been applied to address feature selection problems, such as GAs, GP, ant colony optimisation (ACO) and PSO. Hamdani et al (2007) developed a multi-objective feature selection algorithm using non-dominated sorting based multi-objective genetic algorithm II (NSGAII) and K nearest neighbours (KNN), but the performance of the proposed algorithm has not been compared with any other feature selection algorithm. Zhu et al (2007) proposed a hybrid wrapper and filter feature selection algorithm (WFFSA) based on a memetic algorithm, i.e. a combination of GA and local search. In WFFSA, GA adds or deletes a feature based on the ranked individual features. Three different local search strategies, namely improvement first strategy, greedy strategy and sequential strategy were investigated in WFFSA. Experiments show that WFFSA outperformed GA and other methods. This work also shows that a good balance between local search and genetic search can improve the search quality and efficiency of WFFSA. Muni et al (2006) developed a multi-tree GP algorithm for feature selection (GPmtfs) to simultaneously select a feature subset and design a classifier using the selected features. For a c-class problem, each classifier in GPmtfs has c trees. Comparisons suggest GPmtfs achieved better results than SFS, SBS and other methods. However, the number of features selected increases when there are (synthetically added) noisy features. ACO has also been applied in wrapper based feature selection, Gao et al (2005) proposed a wrapper feature selection algorithm based on ACO to network intrusion detection.

PSO has recently gained more attention for solving feature selection problems. Azevedo et al (2007) proposed a wrapper feature selection algorithm using PSO and support vector machine (SVM) for personal identification in a keystroke dynamic system. However, the proposed algorithm obtained a relatively high false acceptance rate, which should be low in most identification systems. Mohemmed et al (2009) proposed a hybrid method (PSOAdaBoost) that incorporates PSO with an AdaBoost framework for face detection. PSOAdaBoost aims to search for the best feature subset and determine the decision thresholds of AdaBoost simultaneously, which speeds up the training and increases the accuracy of weak classifiers in AdaBoost. Huang and Dun (2008) also proposed a similar feature selection method but used two versions of PSO. BPSO is used to search for the optimal feature subset and continuous PSO is used to simultaneously optimise the parameters in the kernel function of SVM. Xue et al (2012) proposed a feature selection approach based on BPSO and new fitness functions, which included both the number of features and the classification performance. Compared with the basic fitness function, the proposed algorithms further reduced the number of features and increase the classification performance.

The performance of PSO may be improved by properly setting the value of the inertia weight to balance its local search and global search. Yang et al (2008) proposed two feature selection algorithms, which are based on two inertia weights strategies to properly balance the local search and global search of PSO. The two proposed algorithms outperformed other algorithms, such as SFS, "plus-l-take away-r" , SFFS, a sequential GA and different hybrid GAs. Chuang et al (2008) developed a strategy for *gbest* in PSO for feature selection in which *gbest* will be reset to zero if it maintains the same value after several iterations. Chuang et al (2011) applied the so-called catfish effect to PSO for feature selection, which is to introduce new particles into the swarm by re-initialising the worst particles when *gbest* has not improved for a number of iterations. The authors claimed that the introduced catfish particles could help PSO avoid premature convergence and lead to better results than sequential GA, SFS, SFFS and other methods.

Liu et al (2011) introduced a multi-swarm PSO algorithm to search for the op-

timal feature subset and optimise the parameters of SVM simultaneously. Experiments show that the proposed feature selection method achieved higher classification accuracies than grid search, standard PSO and GA. However, the proposed algorithm is computationally more expensive than the other three methods because of the large population size and complicated communication rules between different subswarms. Based on PSO, Unler and Murat Unler and Murat (2010) proposed a feature selection algorithm with an adaptive selection strategy, where a feature is chosen not only according to the likelihood calculated by PSO, but also to its contribution to the features already selected. Experiments suggest that the proposed method outperforms the tabu search and scatter search algorithms.

Esseghir et al (2010) proposed a filter-wrapper feature selection method based on PSO, which aims to integrate the strengths of both filters and wrappers. The proposed filter-wrapper scheme encodes the position of each particle with a score, which reflects feature-class dependency levels evaluated by a filter criterion. The fitness of a particle is the classification accuracy achieved by the selected features. Experimental results show that the proposed method could achieve slightly better performance than BPSO based filter algorithm. As the proposed approach uses the wrapper scheme, one would have expected that a wrapper approach was compared in the experiments.

2.4.2. Filter Feature Selection Algorithms

The Relief algorithm (Kira and Rendell 1992) is a classical filter feature selection algorithm. Relief assigns a weight to each feature 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. Cardie (1993) proposed a filter based feature selection algorithm that uses a decision tree (DT) algorithm to select a subset of features for a nearest neighbour algorithm, because DT use only relevant features that are required to completely classify the training set and remove all other features. The FOCUS algorithm (Almuallim and Dietterich 1994), a filter algorithm, exhaustively examines all possible feature subsets, then selects the smallest feature subset. However, the FOCUS algorithm is computationally inefficient because of the exhaustive search.

Evolutionary computation techniques have also been applied to filter based feature selection. Based on GAs, Chakraborty (2002) proposed a feature selection algorithm using a fuzzy sets based fitness function. However, PSO with the same fitness function in (Chakraborty 2008) achieved better performance than this GA based algorithm. Neshatian and Zhang (2009) proposed a GP relevance measure (GPRM) to evaluate and rank subsets of features in binary classification tasks, and GPRM is also efficient in terms of feature selection to select a small number of features and achieve higher classification performance than using all features. Ming (2008) proposed a feature selection method based on ACO and rough sets theory (Pawlak 1982). The proposed algorithm starts with the features included in the core of the rough sets. Forward selection is adopted into the proposed method to search for the best feature subset. Experimental results show that the proposed algorithm achieved better classification performance with fewer features than a C4.5 based feature selection algorithm. However, experiments did not compare the proposed method with other commonly used feature selection algorithms.

Based on BPSO, Iswandy and Koenig (2006) developed a multi-objective, filter based feature selection algorithm. The proposed algorithm employed different weights to linearly combine three objectives, which were evaluated by three filter criteria, into a single fitness function. Experimental results show that the proposed algorithm outperformed other methods on several benchmark problems. The proposed algorithm aimed to select only one feature subset instead of a set of non-

dominated solutions. Wang et al (2007) proposed a filter feature selection algorithm based on an improved BPSO and rough sets theory. The goodness of a particle is assigned as the dependency degree between class labels and selected features, which is measured by rough sets. 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.

A variety of feature selection approaches have been proposed, but most of them are single objective algorithms and not much work has been conducted to treat a feature selection task as a multi-objective problem. Although Hamdani et al (2007) developed a multi-objective, wrapper, feature selection algorithm based on NSGAII, there is no comparisons with other feature selection methods to judge its performance. Many studies have shown that PSO is a powerful technique for feature selection, but the use of PSO for multi-objective, filter based feature selection has not been investigated. Moreover, most existing approaches are wrappers, which are computationally expensive and "less general" than filter approaches. A relatively small number of PSO 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. There are a variety of other measures that can be used in a filter based feature selection approach. Therefore, the investigation of a PSO based multi-objective, filter based feature selection approach is still an open issue and the work conducted in this paper is the first effort in this area.

3. Proposed Multi-Objective Feature Selection Approaches

This section briefly describes the two filter criteria to be used in this paper, which are based on mutual information and entropy (Cervante et al 2012). Then we propose two new multi-objective BPSO feature selection frameworks that form the new algorithms to address feature selection problems with the goal of minimising the number of features and maximising the relevance between features and class labels, which is evaluated by the two filter criteria.

3.1. *Mutual Information and Entropy for Feature Selection*

Mutual information shows the relevance between two random variables. In classification problems, the classification performance can be increased by maximise the relevance between each feature and the class labels. The number of features needed for classification can be reduced by minimising the redundancy between features, which can be shown by the mutual information between features. Therefore, mutual information can be used to develop a filter feature selection algorithm. We proposed a BPSO based filter feature selection algorithm (BPSOfsMI) based on mutual information in an attempt to maximise the relevance between features and class labels and minimise the redundancy among features (Cervante et al 2012). In BPSOfsMI, each selected feature and the class labels are treated as discrete random variables and the fitness function can be shown by Equation 12.

$$F_1 = Rel_1 - Red_1 \tag{12}$$

where

$$Rel_1 = \sum_{x \in X} I(x; c), \quad and \quad Red_1 = \sum_{x_i, x_j \in X} I(x_i, x_j)$$

where Rel_1 calculates the relevance (mutual information) between each feature and the class labels, which determine the relevance of the selected feature subset to the class labels. Red_1 evaluates the mutual information shared by each pair of selected features, which indicates the redundancy contained in the selected feature subset. X is the set of selected features and c is the class labels. F_1 aims to maximise the

relevance Rel_1 and simultaneously minimise the redundancy Red_1 in the selected feature subset.

Feature selection can be achieved by using mutual information to find the two-way relevance and redundancy between features. However, mutual information could not handle multi-way complex feature interaction, which is one of the challenges in feature selection. Entropy in information theory can measure the relevance between a group of features, based on which, we proposed a feature selection measurement to discover multi-way relevance and redundancy among features (Cervante et al 2012). Further, we developed a single objective filter feature selection algorithm (BPSOfsE) (Cervante et al 2012) based on BPSO and the proposed entropy measurement. Equation 13 was used as the fitness function in BPSOfsE.

$$F_2 = Rel_2 - Red_2 \tag{13}$$

where

$$Rel_2 = IG(c|X) \quad and \quad Red_2 = \frac{1}{|S|}\sum_{x\in X} IG(x|\{X/x\})$$

where Rel_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. Red_2 evaluates the joint entropy of all the features in X, which indicates the redundancy in the selected feature subset. X and c have the same meaning as in Equation 12. F_2 aims to maximise the relevance Rel_2 and minimise the redundancy Red_2 among selected features.

Both Rel_2 and Red_2 involve the calculation of a single discrete feature given information of a set of discrete features. Taken Rel_2 as the example,

$$\begin{aligned} Rel_2 &= IG(c|X) \\ &= H(c) - H(c|X) \\ &= H(c) - (H(c\cup X) - H(X)) \\ &= H(c) + H(X) - H(c\cup X) \end{aligned}$$

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

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

3.1.1. Different Weights for Relevance and Redundancy in BPSOfsMI and BPSOfsE

The relevance and redundancy are equally important in the two fitness functions (Equations 12 and 13). In order to investigate the influence of different relative importances for the relevance and redundancy, a parameter α is introduced into Equation 12 in BPSOfsMI (shown as α_1) and Equation 13 in BPSOfsE (shown as α_2), which can be seen in Equations 14 and 15.

$$F_1 = \alpha_1 * Rel_1 - (1-\alpha_1) * Red_1 \tag{14}$$

$$F_2 = \alpha_2 * Rel_2 - (1-\alpha_2) * Red_2 \tag{15}$$

where α_1 and α_2 are constant values in $[0, 1]$, which show the relative importance of the relevance in two fitness functions. $(1-\alpha_1)$ and $(1-\alpha_2)$ show the relative

Algorithm 1: Pseudo-Code of NSfsMI and NSfsE

```
begin
    divide Dataset into a Training set and a Test set, initialise the swarm (Swarm);
    while Maximum Iterations is not met do
        evaluate two objective values of each particle;  /* number of features and
        the relevance (Rel_1 in NSfsMI and Rel_2 in NSfsE) on the Training
        set */
        identify the particles (nonDomS) (non-dominated solutions in Swarm);
        calculate crowding distance particles in nonDomS and then sort them;
        for i=1 to Population Size (P) do
            update the pbest of particle i;
            randomly select a gbest for particle i from the highest ranked solutions in
            nonDomS;
            update the velocity and the position of particle i;
        end
        add the original particles Swarm and the updated particles to Union;
        identify different levels of non-dominated fronts F = (F_1, F_2, F_3, ...) in Union;
        empty the Swarm for the next iteration;
        i = 1;
        while |Swarm| < P do
            if (|Swarm| + |F_i| <= P) then
                calculate crowding distance of each particle in F_i;
                add F_i to Swarm;
                i = i + 1;
            end
            if (|Swarm| + |F_i| > P) then
                calculate crowding distance of each particle in F_i;
                sort particles in F_i;
                add the (P - |Swarm|) least crowded particles to Swarm;
            end
        end
    end
    calculate the classification error rate of the solutions (feature subsets) in the F_1
    on the test set;
    return the solutions in F_1 and their testing classification error rates;
end
```

importance of the reduction of the redundancy. We assume the relevance is more important than the redundancy, so α_1 or α_2 is set to be larger than $(1-\alpha_1)$ or $(1-\alpha_2)$. When $\alpha_1 = 0.5$ $(1-\alpha_1 = 0.5)$ and $\alpha_2 = 0.5$ $(1-\alpha_2 = 0.5)$, Equations 14 and 15 are the same as Equations 12 and 13, where the relevance and redundancy are equally important.

The representation of a particle in BPSOfsMI and BPSOfsE is a n-bit binary string, where n is the number of available features in the dataset. n is also the dimensionality of the search space. In the binary string, "1" represents that the feature is selected and "0" otherwise.

3.2. *New Algorithms: NSfsMI and NSfsE*

Experiments on BPSOfsMI and BPSOfsE show that mutual information and entropy can be effective criteria for filter feature selection (Cervante et al 2012). However, weights in the fitness functions of BPSOfsMI and BPSOfsE need to be predefined. Based on BPSO, we develop a multi-objective filter feature selection approach using mutual information (or entropy) with the objectives of minimising the number of features and maximising the relevance between features and the class labels to explore the Pareto front of a feature selection problem.

PSO was originally proposed as a single objective optimisation technique. To

extend PSO for multi-objective optimisation, one of the most important tasks is to determine a good leader (*gbest*) for each particle from a set of potential non-dominated solutions. Li (Li 2003) introduces the concepts from a popular evolutionary multi-objective technique, NSGAII (Deb et al 2002), into PSO to develop a continuous multi-objective PSO algorithm and achieves promising results on the optimisation of several benchmark functions. However, this idea has never been applied to feature selection problems.

In this study, we develop a binary multi-objective PSO framework (NSBPSO) for filter feature selection based on the idea of non-dominated sorting in NSGAII. Based on NSBPSO, two filter multi-objective feature selection algorithms are developed, which are NSfsMI and NSfsE. NSfsMI and NSfsE employ Rel_1 and Rel_2 to evaluate the relevance between features and class labels, respectively.

Figure 3 shows the flowchart of two NSBPSO based multi-objective feature selection algorithms, NSfsMI and NSfsE. The main idea is to use non-dominated sorting (Step 7) to select a *gbest* for each particle and update the swarm in the evolutionary process. As shown in Figure 3, in each iteration, the algorithm firstly identifies the non-dominated solutions in the swarm and calculates the crowding distance, then all the non-dominated solutions are sorted according to the crowding distance (Step 2). In Step 3, a *gbest* is randomly selected from the least crowded solutions (the highest ranked part) of the sorted non-dominated solutions. In Step 4, all the particles in the swarm are copied to a union. After determining the *gbest* and *pbest* for each particle, the new velocity and the new position of each particle are calculated according the Equations 2 and 3 and the new positions of all particles are added into the union (Step 5). The two objective values of each particle are evaluated in Step 6, where the relevance is evaluated by Rel_1 in CMDfsMI and by Rel_2 in CMDfsE. Step 7 shows the non-dominated sorting procedure. Specifically, the non-dominated solutions in the union are called the first non-dominated front, subsequently excluded from the union. Then the non-dominated solutions in the new union are called the second non-dominated front. The following levels of non-dominated fronts are identified by repeating this procedure. Step 8 shows the process of updating the swarm for the next iteration. Specifically, particles are selected from the top levels of the non-dominated fronts, starting from the first front. If the number of solutions needed is larger than the number of solutions in the current non-dominated front, all the solutions are added into the next iteration. Otherwise, the solutions in the current non-dominated front are ranked according to the crowding distance and the highest ranked solutions are added into the next iteration. Steps 2 to 8 are repreated until the termination criteria is met. The algorithm returns the first non-dominated Pareto front in the Union.

3.3. *New Algorithms: CMDfsMI and CMDfsE*

The performance of a PSO algorithm can be influenced by the diversity of the swarm. NSBPSO has a potential limitation of quickly losing the diversity of the population during the evolutionary process. Many new particles in the next iteration may be identical. Because new particles are selected from the combination of current particles and the updated particles, all non-dominated particles that share the same solution will be kept into the new swarm. In order to better address feature selection problems, we develop another binary multi-objective PSO framework (CMDBPSO) for feature selection using the ideas of crowding, mutation and dominance (Sierra and Coello 2005).

Based on the CMDBPSO framework, we further develop two filter multi-objective feature selection algorithms, CMDfsMI and CMDfsE. Rel_1 is employed in CMDfsMI to evaluate the relevance and Rel_2 is used in CMDfsE to measure

Algorithm 2: Pseudo-Code of CMDfsMI and CMDfsE

```
1  begin
2      divide Dataset into a Training set and a Test set; initialise the swarm;
3      initialise the set of leaders LeaderSet and Archive
4      calculate the crowding distance of each member in LeaderSet;
5      while Maximum Iterations is not met do
6          for each particle do
7              select a leader (gbest) from LeaderSet for each particle by using a
               binary tournament selection based on the crowding distance;
8              update the velocity and the position of particle i according to
               Equations 2 and 3;
9              apply bit-flip mutation;
10             evaluate two objective values for each particle;   /* number of
               features and the relevance (Rel_1 in CMDfsMI and Rel_2 in
               CMDfsE) on the Training set */
11             update the pbest of each particle;
12         end
13         identify the non-dominated solutions (particles) to update LeaderSet;
14         send leaders to Archive;
15         calculate the crowding distance of each member in LeaderSet;
16     end
17     calculate the classification error rate of the solutions in Archive on the test
       set;
18     return the solutions in Archive and their training and test classification error
       rates;
19 end
```

the relevance between the selected features and the class labels. The objectives of CMDfsMI and CMDfsE are minimising the number of features and maximising the relevance between the selected features and the class labels. Algorithm 2 shows the brief pseudo-code of CMDfsMI and CMDfsE. Basically, CMDfsMI and CMDfsE follow the basic steps of the PSO algorithm except for the steps related to the selection of *gbest*, mutation and dominance. According to Algorithm 2, in order to address the main issue of determining a good leader (*gbest*), CMDfsMI and CMDfsE employ a leader set to keep the non-dominated solutions as the potential leaders for each particle. The maximum size of the leader set is usually set as the number of individuals in the population. A crowding factor is employed to decide which non-dominated solutions should be added into the leader set and kept during the evolutionary process. Binary tournament selection based on the crowding factor is applied to choose a leader (*gbest*) for each particle from the leader set. A bit-flip mutation operator is adopted to keep the diversity of the swarm and improve the search ability of the algorithm. An archive is used to keep the non-dominated solutions and a dominance factor is adopt to determine the size of archive, which is also the number of non-dominated solutions that CMDfsMI or CMDfsE reports.

4. Experimental Design

4.1. *Datasets*

Table 1 shows the eight datasets used in the experiments, which were chosen from the UCI machine learning repository (Frank and Asuncion 2010). The datasets were selected to have various numbers of features, classes and instances and they are used as representative samples of the problems that the proposed algorithms can address. As mutual information and entropy are mainly used for discrete variables,

the data in the selected datasets are categorical values.

In the experiments, all the instances in each dataset are randomly divided into two sets: 70% as the training set and 30% as the test set. The 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 KNN, NB, and DT. As DT is a very commonly used learning algorithm, it is selected in this study to calculate the classification accuracy of the selected features according to Equation 16:

$$Error\ rate = \frac{FP + FN}{TP + TN + FP + FN} \tag{16}$$

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

4.2. *Parameter Settings*

Parameter settings for BPSOfsMI, BPSOfsE, NSfsMI and NSfsE are shown in Table 2. These values are chosen based on the common settings in the literature (Shi and Eberhart 1998, Van Den Bergh 2001). Five different values for α_1 in BPSOfsMI and α_2 in BPSOfsE are used in the experiments, which are 0.9, 0.8, 0.75, 0.6 and 0.5. In the CMDfsMI and CMDfsE, w is a random value in [0.1,0.5], c_1 and c_2 are random values in [1.5, 2.0], and the mutation rate is $1/n$, where n is the number of available features (dimensionality). These values are based on the settings of an equivalent algorithm in the literature (Sierra and Coello 2005).

In order to examine the classification performance of BPSOfsMI (BPSOfsE), a statistical significant test, T-test, is performed with a significance level of 0.05 (95% confidence interval) between the 40 classification accuracies achieved by BPSOfsMI (BPSOfsE) and the classification accuracy obtained by using all features. For each dataset, BPSOfsMI and BPSOfsE obtain a single solution in each of the 40 independent runs. Multi-objective algorithms, NSfsMI, NSfsE, CMDfsMI, and CMDfsE obtain a set of non-dominated solutions in each run. In order to compare these two kinds of results, the 40 solutions (from each of the 40 runs) that resulted from BPSOfsMI and BPSOfsE is presented in next section. The 40 sets of feature subsets achieved by each multi-objective algorithm are firstly combined into one union set. In the union set, for the feature subsets that contain the same number of features (e.g. m), their classification error rates were averaged. The average classification error rate is assigned as the average classification performance of the subsets with m features. Therefore, a set of average solutions is obtained by using the average classification error rates and the corresponding numbers of features (e.g. m). The set of average solutions is called the *average* Pareto front and presented in next section. Besides the average Pareto front, the non-dominated solutions in the union set are also presented in next section.

5. Results and Discussions

This section provides the experimental results and discussions. Results of BPSOfsMI and BPSOfsE with different weights in the fitness functions are shown in Tables 3 and Tables 4. Figures 4 and 5 show the comparisons between BPSOfsMI (BPSOfsE) and the proposed multi-objective algorithms.

5.1. *Results of BPSOfsMI and BPSOfsE*

Tables 3 and 4 show the experimental results of BPSOfsMI and BPSOfsE with 5 different weights, α (α_1 in BPSOfsMI and α_2 in BPSOfsE), in the fitness functions. In the tables, "All" means that all of the available features are used for classification. "MeanNo." represents the average size of the feature subsets evolved by each algorithm in 40 independent runs. "BestAcc" indicates the best test accuracy. "MeanAcc" and "StdDevAcc" show the average and the standard deviation of the 40 test accuracies. "T-test" shows the result of the T-test, where "+" ("-") indicates that the classification performance of BPSOfsMI or BPSOfsE is significantly better (worse) than that of all features. "=" means they are similar.

Tables 3 and 4 show that BPSOfsMI with *mutual information* and BPSOfsE with *entropy* as the evaluation criteria can reduce the number of features and achieve similar or even better classification performance than using all features in some cases. The classification performances of BPSOfsMI and BPSOfsE are slightly worse than that of using all features, but their best classification accuracy could be the same or higher than using all features. BPSOfsMI or BPSOfsE with a large α (e.g. 0.9) usually evolved a subset with more features and achieved higher classification accuracy than with a small α (e.g. 0.5). This is because when α is large, the relevance (Rel_1 or Rel_2) is more important and the redundancy (Red_1 or Red_2) is less important than when α is small. The redundancy indirectly influences the number of features selected. Therefore, with a large α, the fitness function guides BPSOfsMI or BPSOfsE to search for a subset with high classification performance and more features. In contrast, the fitness function with a small α guides BPSOfsMI or BPSOfsE to search for a subset including fewer features. While a small α can always reduce the number of features, a large α does not always increase the classification performance. For example, in the Leddisplay dataset, the classification performance is the same in the five different α values, which means that the feature subsets still have redundancy. More detailed discussions can be seen in the literature (Cervante et al 2012).

Results in Tables 3 and 4 show that in order to achieve the optimal feature subset, an appropriate weight value α_1 or α_2 need to be predefined. In next section, we will investigate the use of BPSO for multi-objective feature selection to search for the non-dominated solutions. In the five different α_1 and α_2 values, the number of features is the most important when $\alpha_1 = \alpha_2 = 0.5$ and the classification performance is the most important when $\alpha_1 = \alpha_2 = 0.9$. Therefore, BPSOfsMI and BPSOfsE with weights (α_1 or α_2) values of 0.5 and 0.9 are used for comparison in the next section to examine the performance of multi-objective feature selection algorithms.

5.2. *Results of NSfsMI and CMDfsMI*

Figure 4 shows the Pareto front solutions obtained by NSfsMI and CMDfsMI in the filter feature selection objective space, where *mutual information* is used as the evaluation criterion. In filter based multi-objective feature selection approaches, the performance of these Pareto front solutions should be evaluated by its classification performance on the unseen test data. Therefore, the solutions used in Figures 4 are the Pareto front solutions obtained in the *mutual information* space, but their classification performances shown in the figures was evaluated by DT on the test data. Figures 4 compares the results of NSfsMI, CMDfsMI, and BPSOfsMI with $\alpha_1 = 0.5$ and $\alpha_1 = 0.9$, which employ *mutual information* to evaluate the relevancy and redundancy between a pair of features. On the top of each chart, the numbers in the brackets show the number of the available features and the classification

error rate using all features. In each chart, the horizontal axis shows the number of features selected and the vertical axis shows the classification error rate. In Figure 4, "NSfsMI-A" stands for the average Pareto front resulting from NSfsMI in the 40 independent runs. "NSfsMI-B" represents the non-dominated solutions resulting from NSfsMI in the 40 independent runs. $\alpha_1 = 0.5$ means the 40 solutions of BPSOfsMI with $\alpha_1 = 0.5$ and $\alpha_1 = 0.9$ means the 40 solutions of BPSOfsMI with $\alpha_1 = 0.9$.

In some datasets, BPSOfsMI and BPSOfsE may evolve the same feature subset in different runs and they are shown in the same point in the chart. Therefore, although 40 results are presented, there may be less than 40 distinct points shown in a chart. For "NSfsMI-B" and "CMDfsMI-B", each of these non-dominated solution sets may also have duplicate feature subsets. They are also shown in the same point in the chart. This is also the case for Figure 5.

5.2.1. Results of NSfsMI

According to Figure 4, in the Mushroom and Spect datasets, the average Pareto fronts of NSfsMI (NSfsMI-A) contains two or more solutions that selected a smaller number of features and achieved a lower classification error rate than using all features.

For the same number of features, there are a variety of combinations of features with different classification performance. In different runs, NSfsMI may select the same number of features with the same fitness evaluated by mutual information (Equation 12), but the same (or better) goodness measured by Equation 12 does not necessarily result to the same (or better) classification performance. Therefore, they may have different classification error rates. Although NSfsMI obtained a set of non-dominated solutions in each run, the average solutions in the average Pareto front may dominate each other (This also happens in CMDfsMI, NSfsE and CMDfsE). In almost all datasets, the non-dominated solutions (NSfsMI-B) include one or more feature subsets, which included less than 50% of the available features and achieved better classification performance than using all features. For example, in the Spect dataset, one non-dominated solution selected 11 features from 22 available features and the classification error rate was decreased from 33.75% to 25.00%.

The results suggests that NSfsMI as a multi-objective algorithm can effectively search the solution space and automatically evolve a set of feature subsets to reduce the number of features and improve the classification performance.

5.2.2. Results of CMDfsMI

According to Figure 4, the average Pareto fronts of CMDfsMI (CMDfsMI-A) include two or more solutions that selected a smaller number of features and achieved better classification performance than using all features in *all* datasets (or similar classification performance only in the Connect4 dataset). In almost all cases (except for the Soybean Large dataset), CMDfsMI-B evolved feature subsets including less than one third of the available features and achieved better classification performance. For example, in the Spect dataset, CMDfsMI-B selected only one feature and decreased the classification error rate of 33.75% to 28.75%.

The results suggest that as a multi-objective algorithm, CMDfsMI can effectively explore the Pareto front of a feature selection problem to reduce both the classification error rate and the number of features needed for classification.

5.2.3. Comparisons between NSfsMI,CMDfsMI and BPSOfsMI

Comparing NSfsMI with BPSOfsMI, it can be seen that in most cases, NSfsMI (NSfsMI-B) achieved better classification performance than BPSOfsMI with $\alpha_1 =$

0.5 although the number of features are slightly larger. In most cases, NSfsMI (NSfsMI-B) outperformed BPSOfsMI with $\alpha_1 = 0.9$ in terms of both the number of features and the classification performance.

Comparing CMDfsMI with BPSOfsMI, in almost all cases, feature subsets evolved by CMDfsMI (CMDfsMI-B) achieved better performance than feature subsets evolved by BPSOfsMI with $\alpha_1 = 0.5$ and with $\alpha_1 = 0.9$ in terms of both the number of features and the classification performance.

The comparisons show that with *mutual information* in the fitness function, achieving better classification performance usually needs more features, but there are occasionally some feature subsets that include a smaller number of features and achieve better classification performance. NSfsMI and CMDfsMI can obtain non-dominated feature subsets that use a smaller number of features and achieve better classification performance. The results suggest that NSfsMI and CMDfsMI as multi-objective algorithms, could better explore the solution space than the single objective algorithm, BPSOfsMI.

5.3. *Results of NSfsE and CMDfsE*

Figure 5 shows the Pareto front solutions obtained by NSfsE and CMDfsE in the *entropy* space, but their classification performances shown in the figures was evaluated by DT on the test data. Figures 5 compares the results of NSfsE, CMDfsE, and BPSOfsE with $\alpha_2 = 0.5$ and $\alpha_2 = 0.9$, which employ *entropy* to evaluate the relevancy and redundancy of a group of features.

5.3.1. Results of NSfsE

According to Figure 5, in most cases, the average Pareto fronts of NSfsE (NSfsE-A) contains more than one solution that selected a smaller number of features and achieved better classification performance than using all features. In almost all datasets, NSfsMI-B reduced the classification error rate by only selecting around half of the available features. Taking the Spect dataset as an example, NSfsE reduced the classification error rate from 33.75% to 25.00% by selecting only 9 features from the 22 available features.

The results suggest that the proposed NSfsE with entropy as the evaluation criterion can automatically evolve a set of feature subsets to simultaneously reduce the number of features and improve the classification performance over using all features.

5.3.2. Results of CMDfsE

According to Figure 5, in all datasets, the average Pareto front of CMDfsE (CMDfsE-A) evolved feature subsets that selected a smaller number of features (less than half in most cases) and achieved better classification performance than using all features. In most cases, CMDfsMI-B increased the classification performance by selecting less than 25% of the available features.

The results in Figure 5 suggest that as a multi-objective algorithm, CMDfsE can automatically evolve a Pareto front of feature subsets, which reduce the classification error rate and substantially reduces the number of features used for classification.

5.3.3. Comparisons between NSfsE, CMDfsE and BPSOfsE

Comparing NSfsE with BPSOfsE, in most cases, NSfsE (NSfsE-B) achieved better classification performance than BPSOfsE with both $\alpha_2 = 0.5$ and $\alpha_2 = 0.9$ although the number of features is slightly larger. One can conclude that NSfsE

outperformed BPSOfsE when increasing the classification performance is considered more important than minimising the number of features.

Comparing CMDfsE with BPSOfsE, in almost all datasets, CMDfsE evolved a smaller number of features and achieved better classification performance than BPSOfsE with both $\alpha_2 = 0.5$ and $\alpha_2 = 0.9$. Only in the Connect4 datasets, CMDfsE achieved similar results to BPSOfsE with $\alpha_2 = 0.5$, but better results than BPSOfsE with $\alpha_2 = 0.9$.

The comparisons show with *entropy* as the evaluation criterion, the proposed multi-objective feature selection algorithms (NSfsE and CMDfsE) can better explore the search space and achieve better solutions than the single objective feature selection algorithm (BPSOfsE).

5.4. *Comparisons between Proposed Algorithms*

Comparing *mutual information* and *entropy*, Figures 4 and 5 show that BPSOfsE, NSfsE and CMDfsE, which use entropy generally achieved better classification performance than BPSOfsMI, NSfsMI and CMDfsMI, which use mutual information. For single objective algorithms, BPSOfsMI usually selected a smaller number of features than BPSOfsE using entropy. This suggests that the algorithms with entropy as the evaluation criterion can discover the multiple-way relevancy and redundancy among a group of features to further increase the classification performance. Because the evaluation is based on a group of features (instead of a pair of features), the number of features involved is usually larger in BPSOfsE than BPSOfsMI. However, the number of features in the proposed multi-objective algorithms is always smaller than single objective algorithms. For the proposed multi-objective algorithms, NSfsE and CMDfsE achieved better classification performance than NSfsMI and CMDfsMI with a similar number of features, which suggests that the NSfsE and CMDfsE with entropy can utilise the discovered multiple-way relevancy between features to increase the classification performance and as multi-objective methods, can simultaneously explore the search space more effectively to reduce the number of features.

Comparing NSfsMI and NSfsE with CMDfsMI and CMDfsE (the two proposed frameworks, NSBPSO and CMDBPSO), Figures 4 and 5 show that in almost all datasets, CMDfsMI and CMDfsE outperformed NSfsMI and NSfsE in terms of both the number of features and the classification performance. As discussed in Section 3.3, NSfsMI and NSfsE have a potential limitation of quickly losing the diversity of the swarm because of the updating mechanism. CMDfsMI and CMDfsE can address this limitation in NSfsMI and NSfsE by using crowding distance in the leader set to keep the diversity of the non-dominated solutions. Therefore, CMDfsMI and CMDfsE better search the space of solutions and achieve better classification performance using a smaller number of features than NSfsMI and NSfsE.

5.5. *Comparisons with a Rough Sets Based Algorithm*

The performances of the proposed algorithms are also compared with the filter feature selection approach (PSORSFS) proposed by Wang et al (2007), which is based on BPSO and rough sets. There are two datasets used in Wang et al (2007) and also in our paper, which are the Lymph and Mushroom datasets. Therefore, we can compare our proposed algorithms, NSfsMI, NSfsE, CMDfsMI and CMDfsE with PSORSFS based on the results of these two datasets.

Wang et al (2007) only gave the best solutions achieved by PSORSFS, which are 7 features with the classification error rate of 24.29% on the Lymph dataset and 4

features with the classification error rate of 0.3% on the Mushroom dataset. Note that the solution achieved by PSORSFS on the Mushroom dataset was claimed to be the (true) optimal solution found by an exhaustive search in the paper (Wang et al 2007).

Comparing their *best* solution with the *average* solutions achieved by our proposed algorithms, NSfsMI-A and NSfsE-A achieved similar or better classification performance than PSORSFS, although the number of features is slightly larger. CMDfsMI-A and CMDfsE-A outperformed PSORSFS in terms of both the number of features and the classification performance on the Lymph dataset and achieved similar classification performance with a slightly larger number of features on the Mushroom dataset.

Comparing the *best* solution of PSORSFS and the *best* solution of our proposed algorithms, NSfsMI-B and NSfsE-B outperformed PSORSFS on the Lymph dataset and achieved similar classification performance with a slightly larger number of features on the Mushroom dataset. CMDfsMI-B and CMDfsE-B outperformed PSORSFS on the Lymph dataset. On the Mushroom dataset, the best solution achieved by PSORSFS is included in CMDfsMI-B and CMDfsE-B. Moreover, CMDfsMI-B and CMDfsE-B include other non-dominated solutions, which can provide more options for users than PSORSFS in real-world applications.

5.6. *Comparisons with Exhaustive Search (Lymph)*

In order to further verify the performance of the proposed algorithms, an exhaustive search is performed on the Lymph dataset, which has a relatively small number of features and is possible to search the solutions space exhaustively, i.e, all possible solutions in the filter search space.

In both *mutual information* and *entropy* evaluation criteria spaces, the true Pareto front obtained by the exhaustive search contains two solutions, where the numbers of features are 1 and 2 and the corresponding testing classification error rates are 26.67% and 17.78%, respectively. The results obtained by NSfsMI and NSfsE can achieve the best classification performance (in the true Pareto front) although the number of features is relatively large. CMDfsMI and CMDfsE can achieve the true Pareto front in many runs although in some other runs, the number of features is slightly larger (e.g. 3 or 4). Note that CMDfsMI identifies the Pareto front solutions in the filter evaluation criterion space (training process), but chooses a different feature from the exhaustive search in some runs. Therefore, the Pareto front solution with 1 feature presented in Figure 4 has a lower testing classification error rate than the one reported by the exhaustive search (26.67%). There are different combinations for the feature subset that includes 2 features. CMDfsMI and CMDfsE usually obtained different combinations of features in different runs.

There are 18 features in the Lymph dataset. For exhaustive search, the number of evaluations is 262144 (2^{18}). While for the proposed algorithms, the number of evaluations is only 3000 (the population size 30 multiply by the maximum iterations 100). Therefore, comparisons suggests that the proposed algorithms can achieve good results (or the best results) by using a much smaller number of evaluations than exhaustive search. Of course, this experiment also suggest that there is still space to improve the proposed algorithms by reducing the number of selected features, which we will investigate in the future.

5.7. *Further Discussion*

In Figures 4 and 5, the solutions used in the charts are the Pareto front solutions obtained using the filter evaluation criteria, but their classification performances

shown in the figures was evaluated by DT on the test sets. Figure 4 shows the Pareto fronts achieved by the proposed algorithms (NSfsMI and CMDfsMI) using *mutual information* as the evaluation criterion. Figure 5 shows the Pareto fronts achieved by the proposed algorithms (NSfsE and CMDfsE) using *entropy* as the evaluation criterion.

As can be seen in Figures 4 and 5, some solutions in the *average* Pareto front (represented by "-A") dominate others although they are non-dominated solutions in the filter evaluation criterion space. This shows that the Pareto front in the filter evaluation criterion space on the training set does not necessarily involve the same subsets as the Pareto front in the DT-based evaluation on the test set. The main reason is that the goodness of a feature subset evaluated by *mutual information* or *entropy* on the training set does not necessarily show its exact classification performance on the test set. In addition, the true Pareto front achieved by exhaustive search in the two filter evaluation criteria objective space may not correspond to the true Pareto front of using DT-based evaluation on the test set. Feature subsets with the same (better or worse) filter goodness do not necessarily achieve exactly the same (better or worse) classification performance on the unseen test set evaluated by DT. For example, two feature subsets may have the same number of features (i.e. n), but different combinations of n features. These two feature subsets may have the same goodness values evaluated by the filter evaluation criterion on the training set. So they are non-dominated to each other. However, when using DT (or any other learning/classification algorithm) to evaluate their classification performances on the unseen test set, their classification performances may be (slightly) different. The feature subset with better classification performance will dominate the other one. This is also the case for other filter criteria and other learning/classification algorithms. Therefore, the Pareto front in the filter evaluation criterion space are usually not the same as the Pareto front in the DT-based evaluation.

Ideally, the proposed algorithms should identify the true Pareto front in each filter evaluation criterion space. As it is not possible to conduct exhaustive search for the datasets with large numbers of features to identify the true Pareto fronts, we take the Lymph dataset as an example (Section 5.6). The proposed multi-objective algorithms CMDfsMI and CMDfsE can identify the true Pareto fronts obtained by the exhaustive search, but NSfsMI and NSfsE can not. The main reason is that the swarm in NSfsMI and NSfsE may lose its diversity quickly due to the updating strategies (more details can be seen in Section 3.3). For the datasets with large numbers of features, CMDfsMI and CMDfsE might not achieve the true Pareto front. The main reason is that feature selection is a difficult problem due to the large search space and feature interaction. Most of the existing feature selection methods suffer from the problem of being stagnation in local optima and high computational cost. Heuristic search such as evolutionary computation methods (e.g. PSO) are introduced to address feature selection problems to search for satisfactory results. In our future work, we will further investigate multi-objective feature selection algorithms to achieve better performance.

6. Conclusions

The overall goal of this paper was to propose a multi-objective, filter feature selection approach based on BPSO and information theory to search for a set of non-dominated feature subsets, which reduced the number of features and achieved better classification performance than using all features. The goal was successfully achieved by developing two multi-objective BPSO frameworks (NSBPSO and CMDBPSO) and two information evaluation criteria (mutual information and

entropy). Hence four multi-objective algorithms (NSfsMI, NSfsE, CMDfsMI and CMDfsE) were proposed for feature selection. NSfsMI and CMDfsMI (NSfsE and CMDfsE) were examined and compared with a single objective BPSO based algorithm, BPSOfsMI using mutual information (BPSOfsE using entropy) with different weights for the classification performance and the number of features (represented by the redundancy between features).

Experimental results show that in almost all cases, the proposed multi-objective feature selection algorithms can automatically evolve a set of non-dominated feature subsets that include a smaller number of features and achieve better classification performance than using all features. NSfsMI and NSfsE achieved better results than the single objective algorithms (BPSOfsMI and BPSOfsE) although the number of features is slightly larger in many cases. In most datasets, CMDfsMI and CMDfsE outperformed (or achieved similar results in some cases) all other methods mentioned above in terms of both the number of features and the classification performance. Comparisons also show that the proposed multi-objective algorithms outperformed a rough sets based algorithm and CMDfsE can achieve the same solutions obtained by exhaustive search in most cases.

This work represents the first PSO based multi-objective algorithms for filter feature selection. Experimental results show the effectiveness of such algorithms, especially multi-objective PSO, with information theory for filter feature selection in classification. In future, we will further investigate the multi-objective PSO based filter algorithm to better explore the Pareto front of non-dominated solutions in feature selection problems. The claims that filter features selection methods are "more general" and less computational expensive than wrappers will be tested with the newly developed multi-objective filter based algorithms.

References

Dash, M., and Liu, H. (1997), "Feature selection for classification," *Intelligent Data Analysis*, 1(4), 131–156.

Kohavi, R., and John, G.H. (1997), "Wrappers for feature subset selection," *Artificial Intelligence*, 97, 273–324.

Dash, M., and Lee, H. (2003), "Consistency-Based Search in Feature Selection," *Artificial Intelligence*, 151(1–2), 155–176.

Whitney, A. (1971), "A Direct Method of Nonparametric Measurement Selection," *IEEE Transactions on Computers*, C-20(9), 1100–1103.

Unler, A., and Murat, A. (2010), "A discrete particle swarm optimization method for feature selection in binary classification problems," *European Journal of Operational Research*, 206(3), 528–539.

Liu, Y., Wang, G., Chen, H., and Dong, H. (2011), "An Improved Particle Swarm Optimization for Feature Selection," *Journal of Bionic Engineering*, 8(2), 191–200.

Kennedy, J., and Eberhart, R. (1995), "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942–1948.

Shi, Y., and Eberhart, R. (1998), "A modified particle swarm optimizer," in *IEEE International Conference on Evolutionary Computation (CEC'98)*, pp. 69–73.

Engelbrecht, A.P., *Computational intelligence: an introduction (2. ed.)*, Wiley (2007).

Chuang, L.Y., Chang, H.W., Tu, C.J., and Yang, C.H. (2008), "Improved binary PSO for feature selection using gene expression data," *Computational Biology and Chemistry*, 32(29), 29– 38.

Huang, C.L., and Dun, J.F. (2008), "A distributed PSO-SVM hybrid system with feature selection and parameter optimization," *Application on Soft Computing*, 8, 1381–1391.

Mohemmed, A., Zhang, M., and Johnston, M. (2009), "Particle Swarm Optimization based Adaboost for face detection," in *IEEE Congress on Evolutionary Computation (CEC'09)*, pp. 2494–2501.

Kennedy, J., and Eberhart, R. (1997), "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, pp. 4104–4108.

Shannon, C., and Weaver, W., *The Mathematical Theory of Communication*, Urbana:The University of Illinois Press (1949).

Marill, T., and Green, D. (1963), "On the effectiveness of receptors in recognition systems," *IEEE Transactions on Information Theory*, 9(1), 11–17.

Yusta, S.C. (2009), "Different metaheuristic strategies to solve the feature selection problem," *Pattern Recognition Letters*, 30, 525–534.

Stearns, S. (1976), "On selecting features for pattern classifier," in *Proceedings of the 3rd International Conference on Pattern Recognition*, Coronado, CA, pp. 71–75.

Pudil, P., Novovicova, J., and Kittler, J.V. (1994), "Floating Search Methods in Feature Selection," *Pattern Recognition Letters*, 15(11), 1119–1125.

Hamdani, T.M., Won, J.M., Alimi, A.M., and Karray, F. (2007), "Multi-objective Feature Selection with NSGA II," in *8th International Conference on Adaptive and Natural Computing Algorithms (ICANNGA'07) Part I*, Vol. 4431, pp. 240–247.

Zhu, Z.X., Ong, Y.S., and Dash, M. (2007), "Wrapper-Filter Feature Selection Algorithm Using a Memetic Framework," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 37(1), 70–76.

Muni, D., Pal, N., and Das, J. (2006), "Genetic programming for simultaneous feature selection and classifier design," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(1), 106–117.

Gao, H.H., Yang, H.H., and Wang, X.Y. (2005), "Ant colony optimization based network intrusion feature selection and detection," in *International Conference on Machine Learning and Cybernetics*, Vol. 6, pp. 3871–3875.

Azevedo, G., Cavalcanti, G., and Filho, E. (2007), "An approach to feature selection for keystroke dynamics systems based on PSO and feature weighting," in *IEEE Congress on Evolutionary Computation (CEC'07)*, pp. 3577–3584.

Xue, B., Zhang, M., and Browne, W.N. (2012), "New Fitness Functions in Binary Particle Swarm Optimisation for Feature Selection," in *IEEE Congress on Evolutionary Computation (CEC'12)*, pp. 2145–2152.

Yang, C.S., Chuang, L.Y., and Li, J.C. (2008), "Chaotic maps in binary particle swarm optimization for feature selection," in *IEEE Conference on Soft Computing in Industrial Applications (SMCIA '08)*, pp. 107–112.

Chuang, L.Y., Tsai, S.W., and Yang, C.H. (2011), "Improved binary particle swarm optimization using catfish effect for feature selection," *Expert Systems with Applications*, 38, 12699–12707.

Esseghir, M.A., Goncalves, G., and Slimani, Y. (2010), "Adaptive particle swarm optimizer for feature selection," in *international conference on Intelligent data engineering and automated learning (IDEAL'10)*, Berlin, Heidelberg: Springer Verlag, pp. 226–233.

Kira, K., and Rendell, L.A. (1992), "A Practical Approach to Feature Selection," *Assorted Conferences and Workshops*, pp. 249–256.

Cardie, C. (1993), "Using Decision Trees to Improve Case-Based Learning," in *Proceedings of the Tenth International Conference on Machine Learning (ICML)*, pp. 25–32.

Almuallim, H., and Dietterich, T.G. (1994), "Learning Boolean Concepts in the Presence of Many Irrelevant Features," *Artificial Intelligence*, 69, 279–305.

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

Chakraborty, B. (2008), "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, pp. 1038–1042.

Neshatian, K., and Zhang, M. (2009), "Genetic Programming for Feature Subset Ranking in Binary Classification Problems," in *European Conference on Genetic Programming*, pp. 121–132.

Ming, H. (2008), "A Rough Set Based Hybrid Method to Feature Selection," in *International Symposium on Knowledge Acquisition and Modeling (KAM '08)*, pp. 585–588.

Pawlak, Z. (1982), "Rough sets," *International Journal of Parallel Programming*, 11, 341–356.

Iswandy, K., and Koenig, A. (2006), "Feature-Level Fusion by Multi-Objective Binary Particle Swarm Based Unbiased Feature Selection for Optimized Sensor System Design," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 365 –370.

Wang, X., Yang, J., Teng, X., Xia, W., and Jensen, R. (2007), "Feature selection based on rough sets and particle swarm optimization," *Pattern Recognition Letters*, 28(4), 459–471.

Cervante, L., Xue, B., Zhang, M., and Shang, L. (2012), "Binary Particle Swarm Optimisation for Feature Selection: A Filter Based Approach," in *IEEE Congress on Evolutionary Computation (CEC'2012) (to appear)*.

Li, X. (2003), "A Non-dominated Sorting Particle Swarm Optimizer for Multiobjective Optimization," in *Annual conference on Genetic and evolutionary computation (GECCO'03)*, pp. 37–48.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002), "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, 6(2), 182 –197.

Sierra, M.R., and Coello, C.A.C. (2005), "Improving PSO-Based Multi-objective Optimization Using Crowding, Mutation and epsilon-Dominance," in *EMO*, pp. 505–519.

Frank, A., and Asuncion, A., "UCI Machine Learning Repository," (2010).

Van Den Bergh, F. (2001), "An analysis of particle swarm optimizers," University of Pretoria, Faculty of Natural and Agricultural Science, Pretoria, South Africa.

Table 1. Datasets

Dataset	#Features	#Classes	#Instances
Lymphography (Lymph)	18	4	148
Mushroom	22	2	8124
Spect	22	2	267
Leddisplay	24	10	1000
Dermatology	34	6	366
Soybean Large	35	19	307
Chess	36	2	3196
Connect4	42	3	44473

Table 2. Parameter settings

Parameter	Value	Parameter	Value
w	0.7298	Population size	30
c_1	1.49618	Maximum iterations	500
c_2	1.49618	Topology	fully connected
v_{max}	6.0	Runs	40

.

Table 3. Results of BPSOfsMI with Different α_1

Dataset	α_1	MeanNo.	BestAcc	MeanAcc ± StdDevAcc	T-test
Lymph	All	18	82.22		
	α_1 =0.9	14	82.22	82.22±5.68E-14	=
	α_1 =0.8	9	82.22	82.11±69.3E-2	=
	α_1 =0.75	7	77.78	77.78±5.68E-14	-
	α_1 =0.6	6	77.78	77.78±5.68E-14	-
	α_1 =0.5	4	77.78	77.78±5.68E-14	-
Mushroom	All	22	100.0		
	α_1 =0.9	9.1	99.59	99.54±8.04E-2	-
	α_1 =0.8	4.5	98.88	98.81±5.88E-2	-
	α_1 =0.75	4.48	97.87	97.87±8.53E-14	-
	α_1 =0.6	3	97.87	97.87±8.53E-14	-
	α_1 =0.5	2	97.87	97.87±8.53E-14	-
Spect	All	22	66.25		
	α_1 =0.9	6	71.25	71.19±39E-2	+
	α_1 =0.8	4.1	71.25	70.75±1.5E0	+
	α_1 =0.75	4	71.25	71±1.56E0	+
	α_1 =0.6	3	71.25	71.25±0E0	+
	α_1 =0.5	3	71.25	71.19±27.2E-2	+
Leddisplay	All	24	100		
	α_1 =0.9	23.98	100	100±0E0	=
	α_1 =0.8	19	100	100±0E0	=
	α_1 =0.75	17	100	100±0E0	=
	α_1 =0.6	14.88	100	100±0E0	=
	α_1 =0.5	11.92	100	100±0E0	=
Dermatology	All	33	90.0		
	α_1 =0.9	30.12	90	90±0E0	=
	α_1 =0.8	17.65	90	89.09±91E-2	-
	α_1 =0.75	11.62	93.64	88.16±1.33E0	-
	α_1 =0.6	8.15	95.45	89.64±2.41E0	=
	α_1 =0.5	6.38	91.82	86.34±6.11E0	-
Soybeanlarge	All	35	90.73		
	α_1 =0.9	22.78	90.73	89.8±97.4E-2	-
	α_1 =0.8	13.8	91.22	86.72±2.2E0	-
	α_1 =0.75	9.68	88.78	84.11±2.49E0	-
	α_1 =0.6	7.4	86.34	80.33±3.53E0	-
	α_1 =0.5	5.72	84.39	76.89±3.97E0	-
Chess	All	36	98.44		
	α_1 =0.9	13.95	95.2	95.19±6.4E-2	-
	α_1 =0.8	10.02	95.2	94.76±42.5E-2	-
	α_1 =0.75	8.18	95.1	94.49±68.2E-2	-
	α_1 =0.6	6.7	94.99	94.25±1.19E0	-
	α_1 =0.5	6.1	94.99	93.32±1.66E0	-
Connect4	All	42	74.62		
	α_1 =0.9	9.68	70.36	69.4±60.7E-2	-
	α_1 =0.8	8	69.07	68.31±54.3E-2	-
	α_1 =0.75	7	68.73	67.6±54.2E-2	-
	α_1 =0.6	5.62	68.55	67.01±55.1E-2	-
	α_1 =0.5	5.12	67.52	66.51±51.2E-2	-

Table 4. Results of BPSOfsE with Different α_2

Dataset	α_2	MeanNo.	BestAcc	MeanAcc ± StdDevAcc	T-test
Lymph	All	18	82.22		
	α_2 =0.9	9.55	80	79.06±1.4E0	-
	α_2 =0.8	9.42	80	78.45±1.33E0	-
	α_2 =0.75	7.52	80	79.11±2.32E0	-
	α_2 =0.6	7	82.22	80.06±2.13E0	-
	α_2 =0.5	5.18	82.22	78.83±3.58E0	-
Mushroom	All	22	100.0		
	α_2 =0.9	5.88	100	99.74±8.77E-2	-
	α_2 =0.8	5.95	99.88	99.32±1.37E0	-
	α_2 =0.75	2.52	99.7	97.88±44.5E-2	-
	α_2 =0.6	2.02	97.76	97.76±9.95E-14	-
	α_2 =0.5	2.05	97.76	97.72±22.2E-2	-
Spect	All	22	66.25		
	α_2 =0.9	17.05	71.25	70.75 ± 1.5E0	+
	α_2 =0.8	15.3	71.25	68.59±1.75E0	+
	α_2 =0.75	12.85	71.25	66.44±2.06E0	=
	α_2 =0.6	9.62	72.5	68.25±1.08E0	+
	α_2 =0.5	7.42	71.25	68.5±3.53E0	+
Leddisplay	All	24	100		
	α_2 =0.9	9	100	100±0E0	=
	α_2 =0.8	9	100	100±0E0	=
	α_2 =0.75	9	100	100±0E0	=
	α_2 =0.6	9	100	100±0E0	=
	α_2 =0.5	9	100	100±0E0	=
Dermatology	All	33	90.0		
	α_2 =0.9	9.42	95.45	90.84±1.98E0	+
	α_2 =0.8	8.15	93.64	90.02±87.5E-2	=
	α_2 =0.75	7.68	93.64	90.27±1.31E0	=
	α_2 =0.6	6.52	93.64	89.32±1.59E0	-
	α_2 =0.5	6.28	92.73	89.16±2.94E0	=
Soybeanlarge	All	35	90.73		
	α_2 =0.9	20.72	88.29	82.94±2.88E0	-
	α_2 =0.8	18.9	85.85	81.4±2.96E0	-
	α_2 =0.75	17.28	87.8	80.51±3.65E0	-
	α_2 =0.6	15.82	88.78	81.23±3.97E0	-
	α_2 =0.5	13.68	89.27	83.74±3.28E0	-
Chess	All	36	98.44		
	α_2 =0.9	25.82	99.06	98.91±27E-2	+
	α_2 =0.8	22.62	99.37	99.07±15.6E-2	+
	α_2 =0.75	21.38	99.37	98.81±31.1E-2	+
	α_2 =0.6	19.22	99.06	98.36±40.2E-2	=
	α_2 =0.5	16.82	98.54	98.01±63.2E-2	-
Connect4	All	42	74.62		
	α_2 =0.9	37.92	75.9	74.66±73.9E-2	=
	α_2 =0.8	38.12	75.94	74.7±92.7E-2	=
	α_2 =0.75	38.1	76.89	74.62±1.05E0	=
	α_2 =0.6	37.75	78.41	74.64±1.28E0	=
	α_2 =0.5	36.75	78.38	74.48±1.28E0	=

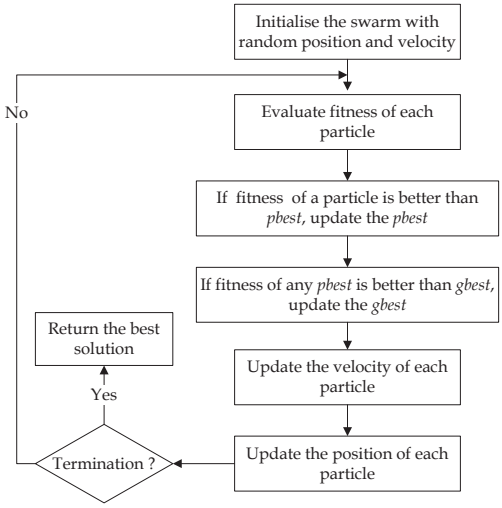

Figure 1. The Flowchart of PSO.

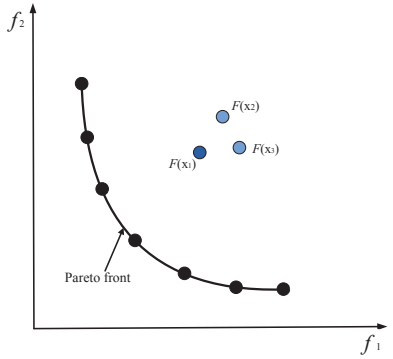

Figure 2. A Minimisation Problem with Two Objective Functions.

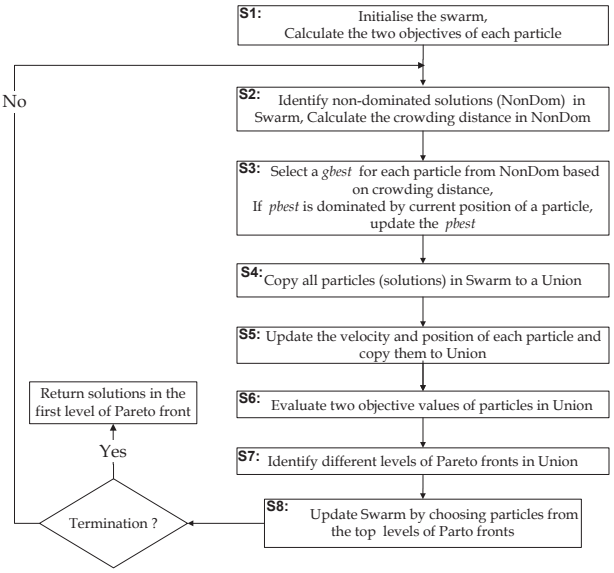

Figure 3. The Flowchart of NSfsMI and NSfsE.

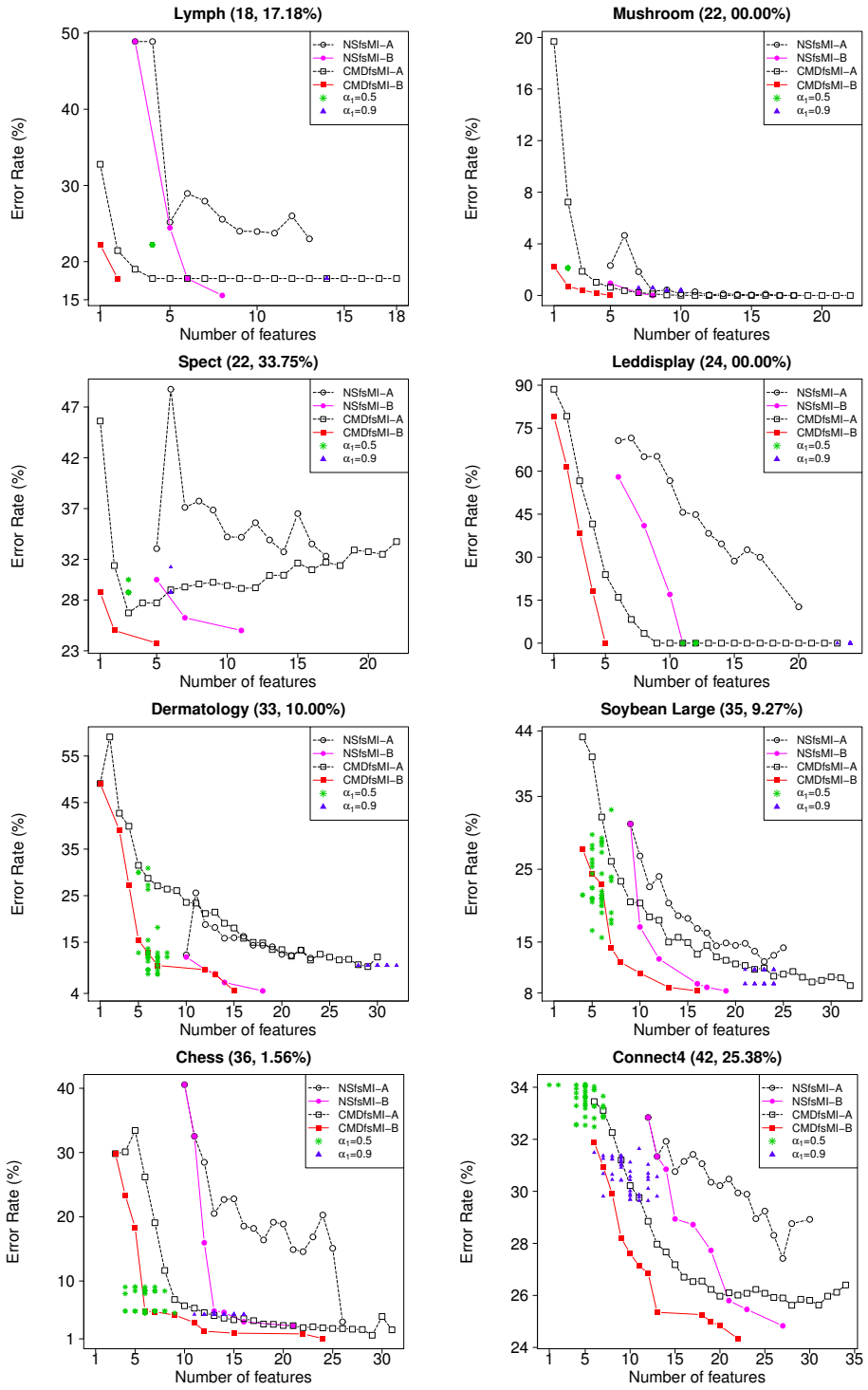

Figure 4. Experimental Results of BPSOfsMI, NSfsMI and CMDfsMI.

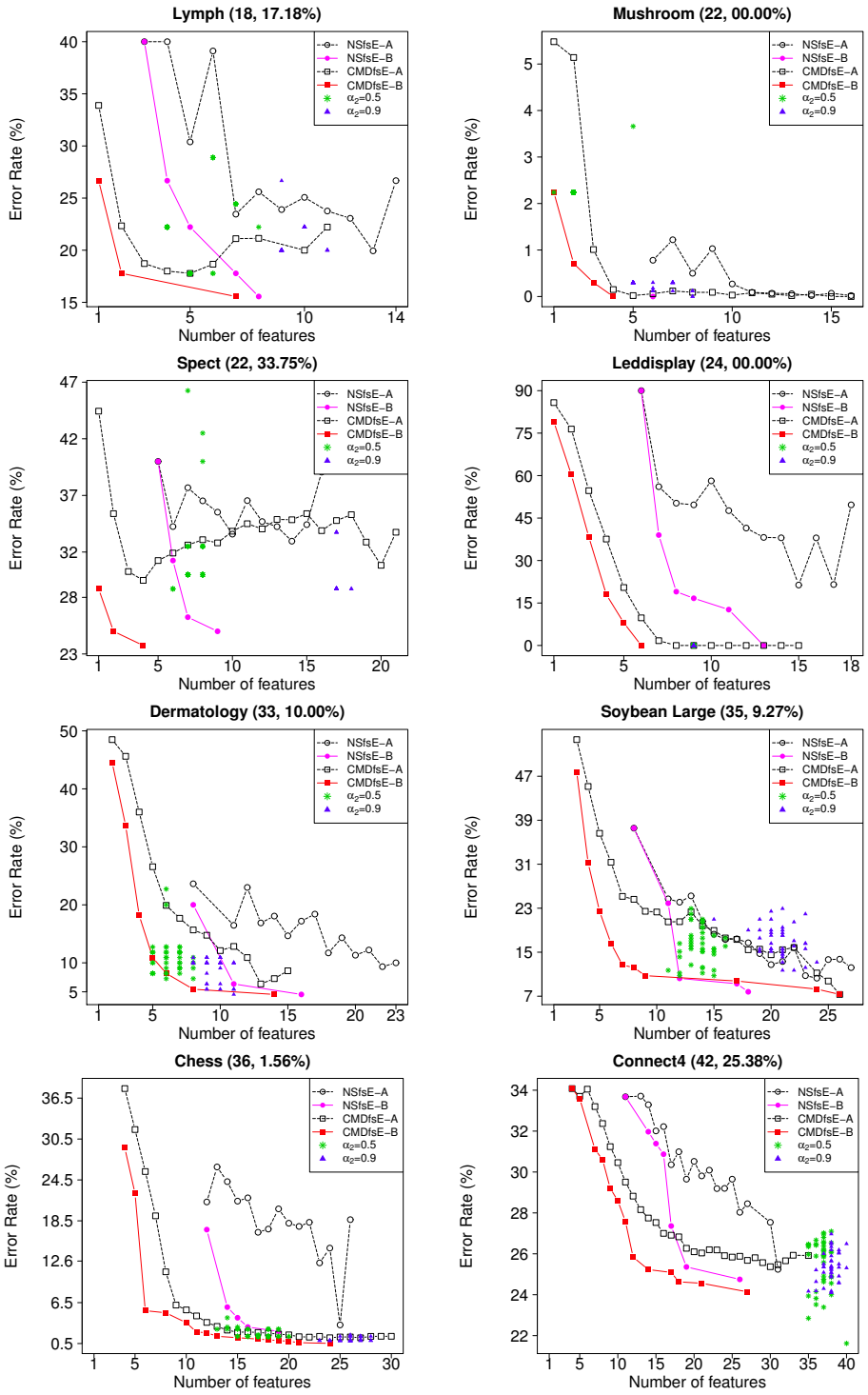

Figure 5. Experimental Results of BPSOfsE, NSfsE and CMDfsE.