New Fitness Functions in Binary Particle Swarm Optimisation for Feature Selection

Bing Xue
School of Engineering
and Computer Science
Victoria University of Wellington
Wellington, New Zealand
Email: Bing.Xue@ecs.vuw.ac.nz

Mengjie Zhang
School of Engineering
and Computer Science
Victoria University of Wellington
Wellington, New Zealand
Email: Mengjie.Zhang@ecs.vuw.ac.nz

Will N. Browne
School of Engineering
and Computer Science
Victoria University of Wellington
Wellington, New Zealand
Email: Will.Browne@ecs.vuw.ac.nz

***Abstract*—Feature selection is an important data preprocessing technique in classification problems. This paper proposes two new fitness functions in binary particle swarm optimisation (BPSO) for feature selection to choose a small number of features and achieve high classification accuracy. In the first fitness function, the relative importance of classification performance and the number of features are balanced by using a linearly increasing weight in the evolutionary process. The second is a two-stage fitness function, where classification performance is optimised in the first stage and the number of features is taken into account in the second stage. K-nearest neighbour (KNN) is employed to evaluate the classification performance in the experiments on ten datasets. Experimental results show that by using either of the two proposed fitness functions in the training process, in almost all cases, BPSO can select a smaller number of features and achieve higher classification accuracy on the test sets than using overall classification performance as the fitness function. They outperform two conventional feature selection methods in almost all cases. In most cases, BPSO with the second fitness function can achieve better performance than with the first fitness function in terms of classification accuracy and the number of features.**

I. Introduction

Classification is one of the major tasks in machine learning and data mining, involving the prediction of class labels based on information about different features. In classification, datasets often have a large number of features. However, not all of the features are useful for classification. Irrelevant and redundant features may even reduce the classification performance. Meanwhile, a large number of features leads to the curse of dimensionality, which is a major obstacle in classification problems. Feature selection is an effective treatment for this situation [1]. Feature selection is a process of choosing a subset of relevant features from a large number of original features. The selected feature subset should be sufficient to describe the target concepts. By eliminating irrelevant and redundant features, feature selection could improve classification performance, make learning and executing processes faster, and/or simplify the structure of the learned classifiers [2].

Existing feature selection methods can be broadly classified into two categories: filter approaches and wrapper approaches. The search process in filter methods is independent of a learning algorithm and they are argued to be computationally less expensive and more general than wrapper approaches [2]. On the other hand, wrapper approaches search for the best feature subset using a learning algorithm as part of the evaluation function. By considering the performance of the selected feature subset on a particular learning algorithm, wrappers can usually achieve better results than filter approaches [3].

Feature selection is a difficult problem, especially when the number of available features is large, because the size of search space grows exponentially with the number of features. Therefore, it is impractical to search the whole space exhaustively in most situations [3]. In order to avoid exhaustive search, greedy algorithms have been introduced to solve feature selection problems such as sequential forward selection (SFS) [4] and sequential backward selection (SBS) [5]. However, such greedy approaches usually suffer from the problem of becoming stuck in local optima and/or high computational cost. Therefore, an efficient global search technique is needed to develop a good feature selection algorithm.

Recently, different evolutionary computation techniques, which are well-known for their global search ability, have been applied to feature selection problems, such as particle swarm optimisation (PSO) [6, 7, 8], genetic algorithms (GAs) [9] and genetic programming (GP) [10]. PSO is based on the idea of swarm intelligence and inspired by social behaviour of birds flocking or fish schooling. Compared with GAs and GP, PSO is easier to implement, has fewer parameters, computationally less expensive, and can converge more quickly [11]. Due to these advantages, PSO has been used as a promising method for feature selection problems [6, 7, 8]. However, feature selection problems have two goals, which are maximising the classification accuracy and minimising the number of features. Most of existing feature selection approaches, including PSO based approaches, aim to maximise the classification performance only, so studies on addressing feature selection as a multi-objective problem are rare. Therefore, it is needed to develop a feature selection approach using PSO to simultaneously maximise the classification accuracy and minimise the number of features selected.

A. Goals

The overall goal of this paper is to develop a new fitness function in PSO for feature selection in classification problems with the expectation of using a small number of features to achieve higher classification accuracy than using all features. To achieve this goal, we will develop two new fitness functions in PSO for feature selection for finding a good subset of features for classification. The two new fitness functions will be examined on ten benchmark datasets with different numbers of features and instances. Specifically, we will investigate

- whether the use of PSO with the overall classification performance as the fitness function can select a good subset of features for classification,
- whether using only a single fitness function that considers both the classification performance and the number of features can further reduce the number of features selected and improve the classification performance, and
- whether the use of two-stage training can further reduce the number of features selected and increase the classification performance.

B. Organisation

The remainder of the paper is organised as follow. Background information is provided in Section II. Section III describes the two proposed BPSO based feature selection approaches with new fitness functions. Section IV describes experimental design and Section V presents experimental results with discussions. Section VI provides conclusions and future work.

II. Background

A. Particle Swarm Optimisation (PSO)

PSO is a population based global search technique proposed by Kennedy and Eberhart in 1995 [12]. In PSO, each candidate solution of the problem is encoded as a particle moving in the search space. The whole swarm searches for the optimal solution by updating the position of each particle based on the experience of its own and its neighbouring particles [13]. Generally, a vector $x_i = (x_{i1}, x_{i2}, ..., x_{iD})$ is used in PSO to represent the position of particle i, where D is the dimensionality of the search space and a vector $v_i = (v_{i1}, v_{i2}, ..., v_{iD})$ represents the velocity of particle i. During the search process, the best previous position of each particle is recorded as the personal best called *pbest* and the best position obtained by the swarm thus far is called *gbest*. The swarm is initialised with a population of random solutions and searches for the best 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 tth iteration in the search process. $d \in D$ denotes the dth dimension in the search space. 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 dth dimension, respectively. w is inertia weight. The velocity v_{id}^{t} is limited by a predefined maximum velocity, v_{max} and $v_{id}^{t} \in [-v_{max}, v_{max}]$.

PSO was originally proposed as an optimisation technique to address continuous problems. However, many optimisation problems, such as feature selection, occur in a space featuring discrete, qualitative distinctions between variables and between levels of variables. To extend the implementation of the PSO algorithm, Kennedy and Eberhart [14] developed a binary particle swarm optimisation (BPSO) for discrete problems. The velocity in BPSO represents the probability of an element in the position taking value 1. Equation (2) is still applied to update the velocity while x_{id}, p_{id} and p_{gd} are restricted to 1 or 0. 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:

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

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. Recent Work Related to Feature Selection

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

1) Classical Feature Selection Approaches

The FOCUS algorithm [15] is a classical filter feature selection algorithm. It starts with an empty feature subset and exhaustively examines all subsets of features, and then selects the minimal subset of features that is sufficient to determine the class label for all instances in the training set. 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 [16]. The weight is intended to denote the relevance of the feature to the target concept. However, Relief does not deal with redundant features whose discriminative ability is covered by other features, because Relief attempts to find all relevant features regardless of the redundancy between them.

SFS [4] and SBS [5] are two commonly used wrapper feature selection approaches. Both of them use a greedy hill-climbing search strategy to search for the optimal feature subset. SFS starts with an empty set of features and iteratively adds one feature at one time until no improvement in classification accuracy can be achieved. By contrast, SBS sequentially removes features from a full candidate feature subset until the removal of further features does not increase the classification accuracy. Both SFS and SBS suffer from the so-called nesting effect, which means that once a feature is selected (discarded) it cannot be discarded (selected) later. Therefore, both SFS and SBS are easily trapped in local optima.

2) BPSO based Feature Selection Approaches

BPSO has recently gained more attention for solving feature selection problems. Chakraborty [17] proposes a BPSO based filter feature selection algorithm with a fuzzy sets based fitness function. 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.

Wang et al. [18] define the velocity in BPSO as the number of elements that should be changed in the position of a particle. The performance of the improved BPSO is compared with that of GA in a filter feature selection model based on rough sets theories. Experiments show that the improved BPSO algorithm is computationally less expensive than GA 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.

Based on BPSO, Unler and Murat [7] propose a wrapper feature selection algorithm with an adaptive selection strategy, where a feature is chosen not only according to the likelihood calculated by BPSO, but also to its contribution to the features already selected. Experimental results suggest that the proposed BPSO method outperforms the tabu search and scatter search algorithms.

Inertia weight is the most important parameter in BPSO, which can improve the performance by properly balancing its local search and global search. Yang et al. [6] propose two BPSO based wrapper feature selection approaches by developing two strategies to determine the inertia weight of BPSO. Experiments show that the two proposed algorithms can outperform other methods, including sequential forward search, plus and take away, sequential forward floating search, sequential GA and different hybrid GAs.

In order to keep the diversity of the population in BPSO, Yang et al. [8] propose a strategy to set *gbest* during the search process. In the proposed algorithm, when *gbest* is identical after three iterations, a Boolean operator `and(.)′ will `and′ each bit of *pbest* of all particles in an attempt to create a new *gbest*. Experiments show that feature subset selected by the proposed method usually achieves higher classification accuracy than that of GA and standard BPSO.

Chuang et al. [19] propose a feature selection algorithm based on an improved BPSO in which all the elements of *gbest* will be reset to 0 if it maintains the same value after several iterations. Experiments with cancer-related gene expression datasets show that the proposed BPSO outperforms the approach proposed in [8] in most cases.

Alba et al. [20] develop a feature selection algorithm based on a geometric BPSO with a support vector machine (SVM) as the learning algorithm in a wrapper approach. In geometric BPSO, *pbest*, *gbest* and the current position of a particle are used as three parents in a three-parent mask-based crossover operator to create a new position for the particle instead of using the position update equation. Experiments show that the proposed algorithm could achieve higher classification accuracy than GA with SVM in most cases, but the performance of geometric BPSO is not compared with that of standard BPSO.

Based on PSO and SVM, Huang et al. [21] propose a wrapper feature selection method in which BPSO is used to search the optimal subset of features and continuous PSO is used to simultaneously optimise the parameters in the kernel function of SVM. Experiments show that the proposed algorithm could determine the parameters, search the optimal feature subset simultaneously and also achieve good classification performance. However, the authors do not compare the performance of the proposed method with other approaches.

Liu et al. [22] introduce a multi-swarm BPSO (MSPSO) algorithm to search for the optimal feature subset and optimise the parameters of SVM simultaneously. Experiments show that the proposed feature selection method could achieve higher classification accuracy than grid search, standard BPSO and GA. However, the proposed algorithm is computationally more expensive than other three methods because of the large population size and complicated communication rules between different subswarms.

3) Other Evolutionary Computation Techniques for Feature Selection

Besides PSO, other different evolutionary computation algorithms have been applied to feature selection problems, such as GAs, GP and and ant colony optimisation (ACO). Based on GA, Yuan et al. [9] propose a two-phase feature selection approach using both filter and wrapper methods. In the filter phase, GA was employed for feature selection with an inconsistency criterion to evaluate the fitness of solutions to remove irrelevant features. The wrapper phase starts with a feedforward neural network whose input nodes are features in the optimal feature subset obtained in the first phase. However, without considering feature interactions, features that would form the best feature subset may be removed in the first phase.

Neshatian and Zhang [10] propose a feature selection approach based on GP and a variation of naïve bayes (NB). A bit-mask representation is used for feature subsets and a set of operators are used as primitive functions. GP combines these feature subsets and operators together to find the optimal subset of features. Experiments on a highly unbalanced face detection problem show that the proposed algorithm can achieve a significant reduction in dimensionality and processing time.

He [23] proposes a filter approach for feature selection based on ACO and rough set theory. The proposed method starts with the features included in the core of the rough set and forward selection is adopted to search for the best subset of features. Experiments show that the proposed approach achieves higher accuracy 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.

Many studies have shown that BPSO is an efficient search technique for feature selection. However, most of the existing approaches are proposed to maximise the classification performance and not much work has been conducted solving a feature selection task as a multi-objective problem. Therefore, development of a feature selection algorithm using BPSO to simultaneously maximise the classification accuracy and minimise the number of features is still an open issue.

III. Proposed BPSO based Feature Selection Approaches

In this section, a BPSO based feature selection approach with the overall classification performance as the fitness function is described. Two new fitness functions are proposed to develop two BPSO based feature selection algorithms to further improve the classification performance and reduce the number of features selected.

A. Basic Fitness Function: Error Rate

Feature selection can be solved by BPSO as a single objective problem to minimise the classification error rate (maximise the classification accuracy) in a wrapper approach. The goal is to see whether BPSO can select a subset of features to achieve higher classification accuracy than using all available features and the results also could be used to as a baseline to compare the performance of newly developed approaches. The fitness function (See Equation 5) is to minimise the classification error rate (maximise the classification accuracy) obtained by the selected feature subset during evolutionary training process.

$$Fitness_1 = ErrorRate \tag{5}$$

where $ErrorRate$ is determined according to Equation 6:

$$ErrorRate = \frac{FP + FN}{TP + TN + FP + FN} \tag{6}$$

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

Algorithm 1 shows the pseudo-code of using BPSO 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. New Fitness Function: Error Rate and #Features

The feature subset selected by BPSO may still contain potential redundancy, because the basic fitness function (Equation 5) does not intend to minimise the number of features. We hypothesize that the same classification performance could be achieved by a smaller feature subset. In order to address this problem, a new fitness function is proposed with the goals of maximising the classification performance (minimise the classification error rate) and minimising the number of features. The formula of the new fitness function is shown in Equation 7.

$$Fitness_2 = \alpha_t * \frac{\#Features}{\#All\ Features} + (1 - \alpha_t) * \frac{ErrorRate}{Error_0} \tag{7}$$

where

$$\alpha_t = \alpha_{max} * \frac{t}{T} \tag{8}$$

where $\alpha_t \in [0, 1]$. t denotes the tth iteration in the search process. $\#Features$ represents the number of features selected. $\#All\ Features$ stands for the number of all the available features. $ErrorRate$ is the classification error rate obtained by the selected feature subset. $Error_0$ is the error rate obtained by using all the available features for classification on the training set. α_{max} is the predefined maximum value of α_t and $\alpha_{max} \in [0, 1]$. T represents the predefined maximum iterations of the BPSO evolutionary process.

In the fitness function (Equation 7), α_t and $(1 - \alpha_t)$ show the relative importance of the number of features and the classification error rate. $(1 - \alpha_t)$ is set larger than α_{max}, because the classification performance is assumed always more important than the number of features. A linearly increasing α_t indicates that classification error rate dominates the fitness function at the beginning of the evolutionary process and the size of feature subset becomes more and more important at the latter stages. However, the number of features is usually much larger than the classification error rate. In order to balance these two components, the size of feature subset is divided by the total number of features, which transforms the value to the range of (0, 1]. In some datasets, the classification error rate changes in a small range in the whole evolutionary training process. Therefore, the classification error rate is transformed in to [0, 1] by dividing the error rate obtained by using all available features. At the first few generations, $\frac{ErrorRate}{Error_0}$ may be larger than 1, but it will not influence the results because we want the classification error rate to dominate the fitness function at the beginning of the evolutionary process. When α_t increases to a relatively large value, BPSO is supposed to evolve the $ErrorRate$ to be smaller than $Error_0$.

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 5 with Equation 7 in Line 1.

C. New Fitness Function: A Two-Stage Approach

In the previous subsection (Section III-B), classification performance and the number of features are balanced by using a linearly increasing weight in the fitness function (Equation 7) in the evolutionary process, which is expected to solve the problem of selecting a redundant feature subset. A potential limitation of this fitness function (Equation 7) is that it may guide the PSO algorithm to search for a small feature subset with low classification performance instead of searching for a large feature subset with high classification performance. In order to overcome this limitation, we propose a two-stage feature selection approach, where the whole evolutionary process is equally divided into two stages. In first stage, the algorithm focuses on the optimisation of classification performance. In

Algorithm 1: The BPSO based feature selection algorithm

```
begin
    divide $Dataset$ into a Training set and a Test set;
    randomly initialise the position and velocity of each particle;
    while $maximumiterations$ or the stopping criterion is not met do
        evaluate fitness of each particle according to Equation 5 ;    /* $ErrorRate$ on the training set */
        for i=1 to $populationsize$ do
            update the $pbest$ of particle $i$;
            update the $gbest$ of particle $i$;
        for i=1 to $populationsize$ do
            for d=1 to $numberofavailablefeatures$ do
                update the velocity of particle $i$ according to Equation 2;
                update the position of particle $i$ according to Equations 3 and 4;
    calculate the classification accuracy of the selected feature subset on the test set;
    return the position of $gbest$ (the selected feature subset);
    return the training and test classification accuracies;
```

TABLE I
DATASETS

Dataset	Number of features	Number of classes	Number of instances
Wine	13	3	178
Vehicle	18	4	846
German	24	2	1000
World Breast Cancer -Diagnostic (WBCD)	30	2	569
Ionosphere	34	2	351
Lung Cancer	56	3	32
Hill-Valley	100	2	606
Musk Version 1 (Musk1)	166	2	476
Madelon	500	2	4400
Isolet5	617	2	1559

the second stage, the number of features is added into the fitness function. The second stage starts with the solutions achieved in the first stage, which ensures that the minimisation of the number of features is based on feature subsets with high classification performance. The fitness function used in this two-stage feature selection approach is shown in Equation 9.

$$Fitness_3 = \begin{cases} ErrorRate, \text{ Stage 1} \\ \alpha * \frac{\#Features}{\#All\ Features} + (1-\alpha) * \frac{ErrorRate}{Error_0}, \text{ Stage 2} \end{cases} \quad (9)$$

where α is constant values and $\alpha \in [0, 1]$. α shows the relative importance of the number of features and $(1 - \alpha)$ shows the relative importance of the classification error rate. $ErrorRate$, $\#Features$, $\#All\ Features$, $ErrorRate$, $Error_0$ are the same as the ones used in Section III-B. As the classification performance is assumed to be more important than the number of features, α is set to be smaller than $(1 - \alpha)$.

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 5 with Equation 9 in Line 1.

IV. Experimental Design

Ten benchmark datasets chosen from the UCI machine learning repository [24] are used in the experiments, which can be seen in Table I. The ten 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. For each dataset, the instances are randomly divided into two sets: 70% as the training set and 30% as the test set.

There are many learning algorithms that can be used here, such as K-nearest neighbour (KNN), NB, and decision tree (DT). One of the simplest learning algorithms, KNN, was selected as the learning algorithm in three BPSO based wrapper approaches. To simplify the evaluation process, we use K=5 in KNN (5NN). Classification accuracy is evaluated by 5NN implemented in Java machine learning library (Java-ML) [25].

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 $T = 100$. The fully connected topology is used in BPSO. These values are chosen based on the common settings in the literature [26]. As the maximum iteration is 100, in the two-stage approach, the first 50 iterations are the first stage and the last 50 iterations are the second stage. We assume the number of features is important in feature selection, but much less important than classification accuracy. Therefore, $\alpha_{max} = 0.2$ in Equation 7 and $\alpha = 0.2$ in Equation 9 in the second stage of the two-stage approach. For each dataset, each approach has been conducted for 40 independent runs.

V. Results and Discussions

Experimental results of three approaches on ten datasets are shown in Table II. In the table, "All" means that all of the available features are used for classification. "BPSO-Er" stands for the BPSO based feature selection approach with Equation 5 as the fitness function. "BPSO-ErNo" and "BPSO-2Stage" represent the two proposed feature selection approaches with Equation 7 and Equation 9 as fitness functions, respectively. "Ave-Size" represents the average size of the feature subsets selected by each algorithm in 40 runs. "Ave-Acc" shows the

TABLE II
EXPERIMENTAL RESULTS

Dataset	Method	Ave-Size	Ave-Acc (Best-Acc)	Std-Acc
Wine	All	13	76.54	
	BPSO-Er	8.32	95.96 (97.53)	1.87E-2
	BPSO-ErNo	8.1	96.23 (98.77)	1.58E-2
	BPSO-2Stage	5.1	96.94 (100)	2.51E-2
Vehicle	All	18	83.86	
	BPSO-Er	9.28	84.3 (85.83)	61.9E-4
	BPSO-ErNo	7.68	84.34 (85.24)	60.3E-4
	BPSO-2Stage	7.3	84.47 (85.04)	54.1E-4
German	All	24	68	
	BPSO-Er	12.9	68.73 (72)	1.3E-2
	BPSO-ErNo	9.48	68.83 (71)	1.35E-2
	BPSO-2Stage	8.62	68.93 (73.67)	1.65E-2
WBCD	All	30	92.98	
	BPSO-Er	14.92	92.98 (92.98)	3.33E-16
	BPSO-ErNo	7.65	92.98 (92.98)	3.33E-16
	BPSO-2Stage	6.68	92.98 (92.98)	3.33E-16
Ionosphere	All	34	83.81	
	BPSO-Er	10.38	89.05 (93.33)	1.84E-2
	BPSO-ErNo	8.55	89.12 (94.29)	1.84E-2
	BPSO-2Stage	8.9	89.52 (93.33)	1.59E-2
Lung	All	56	70	
	BPSO-Er	26.92	72.5 (80)	5.36E-2
	BPSO-ErNo	23.5	73 (90)	5.57E-2
	BPSO-2Stage	22.22	73.25 (80)	6.08E-2
Hill-Valley	All	100	56.59	
	BPSO-Er	47.55	57.56 (60.71)	1.48E-2
	BPSO-ErNo	37.75	57.72 (60.16)	1.36E-2
	BPSO-2Stage	37.1	57.61 (60.44)	1.19E-2
Musk1	All	166	83.92	
	BPSO-Er	83.6	85.65 (89.51)	2.1E-2
	BPSO-ErNo	79.4	85.54 (90.91)	2.21E-2
	BPSO-2Stage	80.72	85.7 (89.51)	2.05E-2
Madelon	All	500	70.9	
	BPSO-Er	244.68	76.83 (78.85)	1.23E-2
	BPSO-ErNo	239.28	77.02 (79.49)	1.17E-2
	BPSO-2Stage	241.35	77.34 (79.62)	1.13E-2
Isolet5	All	617	98.45	
	BPSO-Er	303.14	98.5 (98.75)	17.4E-4
	BPSO-ErNo	297.02	98.58 (98.73)	8.72E-4
	BPSO-2Stage	302.55	98.57 (98.77)	9.43E-4

average test accuracy of the feature subsets selected by each algorithm in 40 runs and "Best-Acc" indicates the best test accuracy. "Std-Acc" represents the standard deviation of the 40 test accuracies achieved by each algorithm.

A. Results of BPSO with Basic Fitness Function

According to the results in Table II, it can be seen that in almost all the datasets, the feature subset selected evolved by "BPSO-Er" only includes half of the available features. With the selected feature subset, 5NN can achieve higher classification accuracy than using all features in almost all datasets (they are the same in the WBCD dataset). For example, in the Wine dataset, with all the 13 features, 5NN could achieve classification accuracy of 76.54% while with around 8 features, it can increase the classification accuracy to 95.96%. All the standard deviation values shown by "Std-Acc" are smaller than 0.03 except in the Lung dataset, which only has a small number of examples and the classification accuracy changes more than in a dataset with more examples. As can be seen in Table II, all the standard deviation values for all methods in all datasets are small, which indicates that all methods are considerably stable and statistical significant testing is not necessary here.

The results suggest that BPSO with overall classification error rate as fitness function can effectively select a subset of relevant features that contains around half of the available features and increase the classification performance.

B. Results of BPSO with New Fitness Function: Error Rate and #Features

According to the results ("BPSO-ErNo") shown in Table II, in most cases, the feature subsets evolved by BPSO contains fewer than half of the available features. With the selected feature subsets, the classifier can achieve higher classification accuracy than using all features in almost all datasets (except for the same accuracy in the WBCD dataset).

Comparing the results achieved by "BPSO-ErNo" with that of "BPSO-Er" , the average size of the feature subsets evolved by "BPSO-ErNo" is always smaller than that of "BPSO-Er" . The reduction of the average size is more than 10% in six of ten datasets and it is 48.7% in the WBCD dataset. The average classification accuracy achieved by the feature subsets resulted from "BPSO-ErNo" is higher than that of "BPSO-Er" in all the datasets. There is no obvious difference between the standard deviation values in two approaches in each dataset.

The results show that by adding the size of the feature subset into the fitness function (Equation 7), it could guide the BPSO algorithm to search for a feature subset with a smaller number of features. By further removing the redundant or unnecessary features from the subset, the classification accuracy can be improved, the testing time can be reduced and the learnt classifiers can be simplified.

C. Results of BPSO with New Two-Stage Fitness Function

According to the results ("BPSO-2Stage") shown in Table II, two-stage feature selection approach selects a small number of features, which is slightly more than one third of the total number of features in many cases. With the selected feature subsets, the classifier can achieve higher classification accuracy than using all features in almost all cases (except for the WBCD dataset).

Comparing "BPSO-2Stage" with "BPSO-Er", "BPSO-2Stage" can evolve smaller feature subsets than "BPSO-Er" The reduction of the average size is more than 15% in seven of ten datasets and it is 55.2% in the WBCD dataset. With smaller feature subsets, "BPSO-2Stage" achieve better or same classification performance as "BPSO-Er" in almost all datasets (except for the WBCD dataset). The standard deviation values in the two approaches are similar.

According to the results, in most cases, "BPSO-2Stage" can evolve smaller feature subsets than "BPSO-ErNo". The reduction of the average size is 37% in the Wine dataset. The average classification accuracy resulted from "BPSO-2Stage" is higher or the same as that of "BPSO-ErNo" in almost all datasets. Only in the Hill-Valley and Isolet5 datasets is the average classification accuracy slightly lower (around 0.1% in Hill-Valley and 0.01% in Isolet5) in "BPSO-2Stage" than in "BPSO-ErNo", but the "BPSO-2Stage" obtains better results in terms of the best classification accuracy in the 40 runs. The standard deviation values is similar in these two approaches.

The results show that by using two-stage evolutionary training process, BPSO can further reduce the number of features selected and improve the classification performance in most cases. In the first stage, the fitness function (Equation 9) could guide BPSO to search for the feature subset with minimum classification error rate without considering the number of features selected. In the second stage, the fitness function (Equation 9) can guide BPSO to search for a smaller feature subset which could maintain the already achieved classification performance.

D. Further Analysis

Table II show that "BPSO-Er" can successfully select a subset of relevant features. "BPSO-ErNo" and "BPSO-2Stage" can further reduce the number of features selected because the number of features is included in the fitness functions.

In the WBCD dataset, the accuracies achieved by three methods are the same, which means larger feature subsets have redundant features. Therefore, we take the WBCD dataset as an example to analysis the difference and the similarity between the feature subsets evolved by BPSO with different fitness functions.

Considering the results in a typical run (where three methods share the same settings except the fitness function), the numbers of features selected by "BPSO-Er", "BPSO-ErNo" and "BPSO-2Stage" are 15, 7 and 6, respectively. The features selected by "BPSO-Er" are F3, F5, F6, F9, F10, F11, F13, F15, F19, F20, F22, F23, F25, F26 and F27, where Fi denotes the ith feature in the dataset. The features selected by "BPSO-ErNo" are F1, F3, F5, F22, F23, F25, F30. The features selected by "BPSO-2Stage" are F3, F9, F10, F13, F22, F23.

Comparing the selected features, it can be observed that all of the features selected by "BPSO-2Stage" are included in the feature subset evolved by "BPSO-Er". As in the first 50 generations, "BPSO-2Stage" and "BPSO-Er" share the same fitness function (See Equation 5 and Equation 9), they achieve the same results. This suggests because of the fitness function (Equation 9), which includes a secondary stage considering the number of features, "BPSO-2Stage" can effectively remove the redundant features from the feature subset resulted from the first stage. Two of the seven features chosen by "BPSO-ErNo" are not included in the features selected by "BPSO-Er", because they use the different fitness functions to guide the search of BPSO. In the other datasets, feature subsets evolved by "BPSO-2Stage" also share more similarity with the subsets evolved by "BPSO-Er" than "BPSO-ErNo".

Results suggest different fitness functions guide BPSO to search for different feature subsets and lead to various classification accuracies. "BPSO-2Stage" with a two-stage fitness function, which firstly optimise the objective of maximising the classification performance and then adds the objective of minimising the number of features in the second stage, can successfully remove the redundant features and achieve high classification accuracy. "BPSO-ErNo" can achieve high classification performance with a small number of features, although the results are not as good as "BPSO-2Stage".

E. Comparisons with Convential Feature Selection Methods

In order to show the performance of the proposed fitness functions, we comparing BPSO-Er, BPSO-ErNo and BPSO-2Stage with two conential feature selection methods, which are linear forward selection (LFS) [27] and greedy stepwise backward selection (GSBS) [28]. Taking the WBCD datasets as examples, LFS selects 10 features with a classification performance of 88.89% and GSBS selects 25 features and achieved a classification accuracy of 83.63%. Comparing the results in Table II, the average number of features in BPSO-Er, BPSO-ErNo and BPSO-2Stage is 14.92, 7.65 and 6.68, respectively. Their average classification accuracy is the same value, 92.98%. Other datasets have similar results, which show that in almost all cases, the classification performance achieved by BPSO-Er and BPSO-ErNo is higher than that of LFS, but the average number of features is larger (in some cases for BPSO-ErNo). In almost all cases, the average classification accuracy achieved by BPSO-2Stage is higher than that of LFS and the average size of the feature subsets is smaller in many cases. BPSO-Er, BPSO-ErNo and BPSO-2Stage outperformed GSBS in terms of both the classification accuracy and the number of features in almost all datasets.

VI. Conclusions and Future Work

The goal of this paper was to investigate a fitness function for a BPSO based feature selection approach to selecting a smaller number of features and achieving higher classification accuracy. This goal was successfully achieved by developing two new fitness functions, which are a linearly changing weights fitness function and a two-stage fitness function. Both of them included the optimisation of both classification performance and the number of features selected. The two fitness functions were examined and compared with a commonly used fitness function using overall classification performance as measurement in a BPSO based wrapper feature selection approach on ten problems of varying difficulty.

The results suggest that BPSO with overall classification performance as fitness function can improve the classification performance over the same classifier using all features. In almost all cases, BPSO with either of the two proposed fitness functions could achieve higher classification accuracy whilst using fewer features than BPSO with overall classification performance as the fitness function. BPSO with the two-stage fitness function outperforms the linearly changing weights fitness function in most cases in terms of the classification performance and the number of features selected.

BPSO with the proposed fitness functions can successfully reduce the number of features needed and achieve higher classification performance, but it is unknown whether the number of features selected could be further reduced without deteriorating or even increasing the classification performance. In the future, we will investigate a BPSO based evolutionary multi-objective feature selection approach to explore the Pareto front of non-dominated solutions, which can help users make a more informed choice of feature subsets. We also intend to investigate the use of a simple learning algorithm in the training process in a wrapper approach to obtain a good feature subset for a complicated learning algorithm, such as SVM and artificial neural network (ANN).

References

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

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

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

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

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] H. Almuallim and T. G. Dietterich, "Learning boolean concepts in the presence of many irrelevant features," *Artificial Intelligence*, vol. 69, pp. 279–305, 1994.

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

[17] 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.

[18] 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.

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

[20] E. Alba, J. Garcia-Nieto, and L. Jourdan, "Gene selection in cancer classification using pso/svm and ga/svm hybrid algorithms," in *IEEE Congress on Evolutionary Computation (CEC'07)*, 2007, pp. 284–290.

[21] C. L. Huang and J. F. Dun, "A distributed pso-svm hybrid system with feature selection and parameter optimization," *Application on Soft Computing*, vol. 8, pp. 1381–1391, 2008.

[22] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, and S. Wang, "An improved particle swarm optimization for feature selection," *Journal of Bionic Engineering*, vol. 8, no. 2, pp. 191–200, 2011.

[23] 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.

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

[25] T. Abeel, Y. V. de Peer, and Y. Saeys, "Java-ml: A machine learning library," *Journal of Machine Learning Research*, vol. 10, pp. 931–934, 2009.

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

[27] M. Gutlein, E. Frank, M. Hall, and A. Karwath, "Large-scale attribute selection using wrappers," in *IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09)*, 2009, pp. 332–339.

[28] R. Caruana and D. Freitag, "Greedy attribute selection," in *International Conference on Machine Learning (ICML'94)*, 1994, pp. 28–36.