Hybridising Particle Swarm Optimisation with Differential Evolution for Feature Selection in Classification

Ke Chen[1,2], Bing Xue[2], Mengjie Zhang[2], and Fengyu Zhou[1*]
[1]*School of Control Science and Engineering*, *Shandong University*, Jinan 250061, China
{chenke_zixf, fyzhou_sdu}@163.com
[2]*School of Engineering and Computer Science*, *Victoria University of Wellington*
PO BOX 600, Wellington 6140, New Zealand
{bing.xue, mengjie.zhang}@ecs.vuw.ac.nz

***Abstract*—Classification has been widely studied due to its practical applications. Feature selection aims to improve the classification accuracy by selecting a small feature subset from the original full feature set. However, identification of relevant features is not trivial due to the large search space. Particle swarm optimisation (PSO) is an efficient meta-heuristic algorithm which has shown to be promising in feature selection. However, traditional PSO uses its personal best experience and its historical best experience to determine its search direction, but this learning strategy may limit its performance for feature selection due to the premature convergence. Therefore, the potential of PSO needs to be further explored. In this paper, a new evolutionary learning algorithm termed hybridising PSO with differential evolution (HPSO-DE) is proposed to develop new feature selection methods. In HPSO-DE, differential evolution is applied to breed promising and efficient exemplars for PSO to guide its search, which is expected to not only preserve the diversity of the population but also guide particles to fly to promising areas. HPSO-DE is compared with three classic PSO variants and five traditional feature selection methods on 15 classification problems. The results show that the proposed algorithm can effectively achieve a higher classification accuracy with a smaller feature subset than the compared methods.**

***Index Terms*—Particle swarm optimisation, feature selection, differential evolution, classification**

I. Introduction

Classification is an important and widely-studied data mining task [1], and its purpose is to assign each instance a class label according to the information presented by its features. In a classification problem, it often contains a large quantity of features, but not all of them are useful to help in class prediction. Many features may be irrelevant and redundant, and these features may even degrade the performance of a learning algorithm due to the large search space [2], [3]. Feature selection is an effective and important data preprocessing technique to address above issue by choosing a subset of relevant features from the original features.

Recently, many feature selection approaches have been presented by researches, which can be generally classified into two categories according to the different evaluation criteria, that is, wrapper-based approaches and filter-based approaches [4]. Filter-based approaches determine features according to their intrinsic characteristics (e.g. correlation, distance, information theoretic, consistency) [5]. Wrapper-based approaches use the predicted accuracy of a learning algorithm to assess the selected feature subset. In general, wrapper-based approaches can obtain higher classification accuracy than filter-based approaches. However, these wrapper-based approaches are computationally expensive, especially when the search space is large and complex [6]. On the contrary, the filter-based approaches are typically computationally less expensive, but often obtain lower classification accuracy [7].

Feature selection can be viewed as a combinatorial optimisation problem [7]. Suppose a dataset contains n original features, the feature selection process tends to choose one from 2^n possible feature combinations. Therefore, it is almost impossible to perform an exhaustive search when the number of features is large. A lot of heuristic search approaches have been used to address this problem, such as sequential backward or forward search techniques [8]. However, these methods may fall into local optima and need long computational time during the feature selection process.

So as to better solve feature selection, an algorithm with strong global search ability is required to improve search efficiency [9]. Evolutionary computation (EC) techniques have been proved to have strong global search ability and can effectively search in complex spaces to achieve optimal or near-optimal solutions. Particle swarm optimisation (PSO) [10] is one of EC techniques, which has been widely and successfully adopted to address many practical applications due to its efficiency and simplicity of implementation. A review of EC-based feature selection approaches was completed in [4]. Compared with other EC methods, PSO is preferred since it has a natural representation for feature selection.

In canonical PSO, each particle denotes a potential solution in the search space. During the search process, all particles in the population keep learning from the personal best position (*pbest*) and the global best position (*gbest*) to update their position and velocity. This learning strategy is easy to implement but may cause particle oscillation if *pbest* and *gbest* are situated on different sides of the current location [11].

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

In addition, if *pbest* and *gbest* are situated in the same local optimum, which may lead to premature convergence during the evolutionary process. During the past two decades, a host of variants have been designed to enhance the search performance of canonical PSO. According to the different objectives to be addressed, these PSO variants can be generally categorized into four cases, that is, topology adjustments [12], parameter adjustments [13], hybrid methods [14], and novel learning strategies [15]. However, many PSO variants fail to maintain their improvement to problems with different characteristics, especially when the search space is complex and large. Therefore, improving the overall performance of PSO is still a challenging problem for wider applications.

Differential evolution (DE) [16] is another efficient nature-inspired algorithm that finds the optimal solution for a task by repeatedly trying to enhance the quality of the current solutions based on the DE operators (i.e. mutation, crossover, and selection). Compared with PSO, DE can discover more useful information during the search process, which can improve the search performance of canonical PSO using such useful information.

A. Goals

The overall goal of this paper is to develop an effective PSO variant based on DE operators for feature selection problems. In the proposed algorithm, the DE operators are used to provide good guidance for each particle by breeding a promising and efficient exemplar. The proposed algorithm is expected to achieve a high classification accuracy with a small feature subset in a reasonable time. Specifically, this work has the following research objectives:

- Propose a new PSO variant that based on the differential evolution operator to improve the search performance of PSO,
- Design a new exemplar updating operator to help PSO fly out of the local optimum, and
- Verify the performance of the proposed algorithm in terms of classification accuracy and feature subset size by comparing with the state-of-the-art algorithms.

II. Background

A. Particle swarm optimisation and its variants

Particle swarm optimisation (PSO) is a nature-inspired optimisation algorithm originally introduced by Eberhart and Kennedy in 1995 [10], [17]. Each particle in canonical PSO has a location which denotes a potential solution in the search space. In the *D*-dimensional feasible search space, the location and the velocity of particle i can be represented by a position vector $\boldsymbol{X}_i = [X_{i1}, X_{i2}, \cdots, X_{iD}]$ ($X_{id} \in [X_{min}, X_{max}]$) and a velocity vector $\boldsymbol{V}_i = [V_{i1}, V_{i2}, \cdots, V_{iD}]$ ($V_{id} \in [V_{min}, V_{max}]$), respectively. The previously best location of the *i*-th particle is represented by $\boldsymbol{pbest}_i = [pbest_{i1}, pbest_{i2}, \cdots, pbest_{iD}]$, and the best location among the whole particles achieved so far is represented as $\boldsymbol{gbest} = [gbest_1, gbest_2, \cdots, gbest_D]$. At each step, the individual best location $pbest$ and the global best location $gbest$ are used to generate a new velocity for the *i*-th particle. The updating process for the *d*-th dimension of the *i*-th particle are represented as follows:

$$\begin{aligned} V_{id} =& \omega * V_{id} + c_1 * r_1 * (pbest_{id} - X_{id}) + \\ & c_2 * r_2 * (gbest_d - X_{id}) \end{aligned} \tag{1}$$

$$X_{id} = X_{id} + V_{id} \tag{2}$$

where ω denotes the inertia weight. c_1 and c_2 are acceleration constants. r_1 and r_2 are two uniform distributed values within the range [0, 1].

In the traditional PSO, each particle uses its personal best location (*pbest*) and global best location (*gbest*) to determine its position and velocity. This learning scheme may cause particle oscillation because the guidance of these two positions may be on different sides of X_i. In addition, since each particle of the population is learning from *pbest* and *gbest* even if *gbest* and *pbest* are situated in a same local optimum. This may cause the search of PSO to stagnate in the local region due to rapid convergence, especially when the number of local optimum solutions is large.

So as to address these deficiencies in canonical PSO, researchers have designed a variety of learning strategies to make better use of beneficial information to determine the search direction of PSO, and the widely used velocity updating formula is shown as follows [18]:

$$V_{id} = \omega * V_{id} + c * r * (EV_{id} - X_{id}) \tag{3}$$

where c is acceleration constant, and r is a uniform distributed number within the range of 0 to 1. Here, a exemplar vector $\boldsymbol{EV}_i = [EV_{i1}, EV_{i2}, \cdots, EV_{iD}]$ is constructed to guide the *i*-th particle during the search process. For example, a linear combination of *pbest* and *gbest* is used to form the exemplar vector EV_i in [18]. Eq. (4) shows the linear combination formula.

$$EV_{id} = \frac{c_1 * r_1 * pbest_{id} + c_2 * r_2 * gbest_d}{c_1 * r_1 + c_2 * r_2} \tag{4}$$

B. PSO for feature selection

In the last decades, a large number of PSO-based approaches have been developed and shown good results for feature selection problems [19]. In these approaches, different strategies have been designed to enhance the overall performance of canonical PSO during the evolutionary process.

Resetting the position of *gbest*, *pbest*, and particle is an effective way to overcome premature convergence in the large search space. In [20], a *gbest* update strategy was proposed to help PSO escape from local optima. In this method, *gbest* was created based on a Boolean function when the fitness value of *gbest* does not change within a given number of generations. In [21], *gbest* was reset to zero when PSO's search process stagnates, which encourages PSO to select a smaller feature subset. The experimental results showed that these methods could obtain higher classification accuracy in most cases. However, the modifications in these methods may ignore some useful information during the feature selection process.

Maintaining the diversity of the population is a useful way to enhance the search performance of canonical PSO. In [22], a PSO variant was proposed for feature selection, which aimed to explore untried areas of the search space. In this method, a mutation strategy was applied to maintain the population diversity. The experimental results showed that this method could achieve higher classification accuracy than other compared methods. However, this method may reduce the convergence performance of PSO within a predefined number of iterations.

Reducing the search space by removing redundant and irrelevant features is also presented to enhance PSO performance. In [23], an adaptive multi-subswarm PSO algorithm was designed for feature selection problems. This method divided the whole search space into smaller subspaces according to the importance of features, and then using PSO to select features from these smaller subspaces. The results showed that this method is more effective than other compared methods in terms of the classification accuracy and the number of selected features. However, the lack of information exchange between subspaces may reduce the search performance during the feature selection process.

Designing some local search strategies to exploit the fruitful regions is another effective way to improve PSO performance. In [24], a new PSO-based feature selection method was developed to address feature selection problems. In this method, a local search operator was embedded in PSO, which uses correlation information of each feature to conduct the feature selection process. The results on twelve datasets showed that this method could choose a smaller feature subset to obtain higher classification accuracy than using all available features and the compared feature selection approaches. However, this method requires more computing resources, especially when dealing with a large dataset.

In summary, although the existing PSO-based feature selection approaches have shown to be effective, the search performance of these approaches is still limited when applying PSO to solve complex feature selection problems. PSO for feature selection still needs further investigation.

III. THE PROPOSED METHOD

In this section, we describe the proposed PSO variant based on DE operators to address feature selection problems. The details of using DE operator to breed an exemplar are presented first. Then, a fitness function is presented, which is applied to assess the selected feature subset.

A. Differential evolution operators

As mentioned in Section II-A, the exemplar vector EV_i plays an important role to determine the search behaviour of particles. In this study, DE operators are used for breeding exemplars during the evolutionary process, which can provide a promising solution in the large search space. There are two potential benefits of using DE to breed exemplars. First, DE is expected to maintain the diversity of the population to prevent the premature convergence of PSO. Second, these exemplars constructed by DE can improve the search effectiveness of PSO by providing good guidance for each particle during the evolutionary process. The implementation of the DE operators is described step by step below.

1) Mutation operation: In the proposed algorithm, the mutation operation is applied to produce a mutant vector M_i on the *i*-th particle. According to [12], DE/best/1 is most suitable to discover useful information during the search process. This is because its composition takes into account the requirements of maintaining the diversity of population and improving convergence performance. DE/best/1 is shown as follows:

$$M_{id} = X_{best,d} + F * (X_{r1,d} - X_{r2,d}) \tag{5}$$

where r_1 and r_2 are random unequal integers obtained for each mutant vector within $[1, NP]$, and NP denotes the number of particles. The scale factor F is a parameter, which is used to scale the difference vector during the search process. X_{best} denotes the current best individual in the whole population.

In this paper, DE/best/1 is used to generate M_i and it can be expressed as follows:

$$M_{id} = gbest_d + F * (pbest_{r1,d} - pbest_{r2,d}) \tag{6}$$

During the search process, the mutant vector M_i may exceed the pre-determined boundary of the search space. Therefore, a boundary handling strategy is adopted to address this problem.

$$M_{id} = \begin{cases} X_{min}, & if\ M_{id} < X_{min} \\ X_{max}, & if\ M_{id} > X_{max} \end{cases} \tag{7}$$

2) Crossover operation: For each particle, the crossover operation is conducted on *pbest*$_i$ and M_i to produce a new trial vector U_i. This updating process is described as follows:

$$U_{i,d} = \begin{cases} M_{id}, & if\ rand_j \leq CR\ or\ j = j_{rand} \\ pbest_{id}, & otherwise \end{cases} \tag{8}$$

where j_{rand} is an integer randomly chosen from $[1, D]$ and $rand_j$ is a random number within the range $[0, 1]$. The crossover probability CR is a user-defined number, which is used to control the number of bits copied from the mutant vector M_i. Here, the condition $j = j_{rand}$ in Eq. 8 is adopted to assure that the new trial vector U_i has at least one parameter different from its corresponding original vector $pbest_i$.

3) Selection operation: After the mutation and crossover operations, a selection operation is conducted to determine whether U_i or *pbest*$_i$ survives for the next generation. This operation is expressed as follows:

$$EV_i = \begin{cases} U_i, & if\ f(U_i) \leq f(pbest_i) \\ pbest_i, & otherwise \end{cases} \tag{9}$$

where $f(U_i)$ and $f(pbest_i)$ denote the fitness values of U_i and *pbest*$_i$, respectively. It is noted that the number of function evaluations consumed by U_i will be counted in the new algorithm.

As can be seen from Eq (9), the new trial exemplar U_i will replace the original exemplar *pbest*$_i$ if the fitness value of U_i is less than *pbest*$_i$. This elitism strategy can effectively maintain the evolution of particles during the search process. Moreover,

Algorithm 1: Hybridising PSO with DE (HPSO-DE)

Input: Initialise parameters (i.e. NP, the number of particles; D, dimension, c, acceleration constant; w, inertia weight; F, scale factor; $maxFEs$, the maximal number of function evaluations);
Output: The best solution $gbest$;

Initialise the positions ***X*** and velocities ***V*** of NP particles randomly;
Evaluate $f(X_i)$;
Set ***X*** to be ***pbest*** and find the current ***gbest***;
while $FEs < maxFEs$ **do**
  for i = *1* ***to*** NP **do**
    /* Exemplar update: Mutation operation */
    Construct M_i using Eq. (6);
    Update M_i using Eq. (7);
    for d=*1* ***to*** D **do**
      /* Exemplar update: Crossover operation */
      Construct U_i using Eq. (8);
    end
    /* Exemplar update: Selection operation */
    Evaluate $f(U_i)$;
    Determine EV_i using Eq. (9);
    if *gbest does not change for G generations* **then**
      Update the exemplar EV_i using Eq. (10);
    end
    /* Particle update */
    Update the velocity V_i using Eq. (3);
    Update the position X_i using Eq. (2);
    Evaluate $f(X_i)$;
    Update *pbest*$_i$ and *gbest*;
  end
end
return *gbest*

if *gbest* does not change within a given number of generations (G), PSO may be stuck in a potential local optimum during the search process. In this study, we design a exemplar updating operator based on the spiral-shaped mechanism to update the exemplar EV_i. During the evolutionary process, this mechanism has the potential to change the search direction of particles, thereby increasing the possibility of escaping from the local optima. This mechanism is described as follows:

$$EV_i = D * e^{b*l} * cos(2\pi l) + EV_i \tag{10}$$

where $D = |gbest - EV_i|$ represents the distance of the exemplar EV_i to the current global best position. b is a constant that is defined to tune the shape of the spiral. l is a random number and its range is [0, 1]. More information about the spiral-shaped mechanism can be achieved in [25].

For each particle, the above three operations (i.e. mutation, crossover, and selection) are repeated until a termination criterion (e.g. the maximal number of function evaluations) is met. With the DE operators, a novel hybridising PSO with DE (HPSO-DE) algorithm is designed. The pseudo-code of HPSO-DE is presented in Algorithm 1. It can be seen that HPSO-DE is very easy to implement. Note that the fundamental component of HPSO-DE is PSO, and DE is only adopted as an auxiliary algorithm to construct promising exemplars.

When using HPSO-DE for FS, a particle represents a feature subset, and each position value of the particle within a fixed range (e.g. between 0 and 1) represents that whether the corresponding feature should be reserved or abandoned by using a user-defined threshold value (e.g. 0.6).

B. Fitness function

During the feature selection process, feature evaluation is an important step to evaluate the goodness of selected features. The k-nearest neighbour (KNN) is selected as the classification algorithm due to its effectiveness and simplicity, where K equals to 5. In this study, a fitness function is adopted to assess the selected subset, which combines the feature subset size and the classification accuracy of using the feature subset into one by setting a weight factor. The fitness function is shown as follows:

$$fitness = \alpha * \gamma_R(D) + (1 - \alpha) * \frac{|S|}{|N|} \tag{11}$$

where $\gamma_R(D)$ represents the classification error rate of the classification algorithm, $|S|$ denotes the feature subset size, $|N|$ indicates the number of all features. α is a parameter that is applied to influence the role of the classification error rate and the feature subset size, where α is set to 0.9 to balance these two components [26].

Since most of these datasets are unbalanced, we adopts a balance accuracy [27] to calculate the first component of the fitness function in this study. Furthermore, in order to avoid feature selection bias, 5-fold cross-validation is adopted to assess the classification accuracy on the training set. Eq. (12) shows the equation of the classification error rate.

$$\gamma_R(D) = 1 - \frac{1}{c} * \sum_{i=1}^{c} TPR_i \tag{12}$$

where c denotes the number of classes in a classification problem, and TPR_i represents the proportion of correctly identified instances in class i. Since there is no bias to each class in the classification problem, and the weight for each class is set to $1/c$.

C. Summary

From an overall perspective, designing an efficient search algorithm with strong global search ability will improve the performance of the wrapper-based feature selection method. The proposed algorithm is expected to improve the search effectiveness by breeding a promising exemplar based on the DE operators. It can increase the population diversity during the feature selection process and provide effective guidance for particles.

IV. Experiment design

To investigate the efficiency and effectiveness of the proposed algorithm, two experiments have been implemented for feature selection problems. In this section, we describe the details of the experiment design including the investigated datasets, the fitness function, and the compared methods.

TABLE I
LIST OF DATASETS USED IN THE EXPERIMENTS

Dataset	#Features	#Instance	#Class	%Smallest class	%Largest class
Breastcancer	9	699	2	34.48	65.52
Glass	9	214	6	4.21	35.51
Zoo	16	101	7	3.96	40.59
Segmentation	19	2310	7	14.29	14.29
WaveformEW	21	5000	3	32.94	33.92
SpectEW	22	267	2	41.20	58.80
WDBC	30	569	2	37.26	62.74
IonosphereEW	34	351	2	35.90	64.10
KrvskpEW	36	3196	2	47.78	52.22
Satellite	36	6435	6	9.73	23.82
Movementlibras	90	360	15	6.67	6.67
Musk1	166	476	2	43.49	56.51
Semeion	256	1593	10	9.73	10.17
Madelon	500	2000	2	50.00	50.00
Isolet	617	1559	26	3.78	3.85

A. Datasets

Fifteen different feature selection problems from the UCI machine learning repository are employed to verify the proposed algorithm [28]. Table I shows the key characteristics (i.e. the number of features, instances, classes) of these datasets. Furthermore, the distribution of these data is unbalanced. In the experiments, each classification dataset is randomly divided into two sets, that is, 70% of the instances as the training set, and the remaining 30% as the test set.

B. Benchmark methods

So as to evaluate the performance of the proposed algorithm, we compare HPSO-DE with three classic PSO variants on all datasets, which are PSO with inertia weight (GPSO) [29], fully informed particle swarm (FIPS) [18], and comprehensive learning PSO (CLPSO) [15]. The parameter settings of these compared approaches are according to its corresponding reference. For the proposed HPSO-DE algorithm, the inertial weight ω, the acceleration constant c, the scale factor F and the crossover probability CR are set to 0.7298, 1.49618, 0.5, and 0.2, respectively. Several different values (i.e. from 2 to 10) for the predefined generation G are conducted to determine the appropriate value. The results of G=5 is better than other values. Therefore, G is set to 5 in HPSO-DE for all datasets. The maximum number of function evaluations (FEs) and the population size are set to 4,000 and 40 for all algorithms, respectively. In addition, we also compare the developed HPSO-DE algorithm with five traditional feature selection approaches, which are Correlation-based FS (CFS) [30], Fast Correlation Based Filter (FCBF) [31], Information Gain (IG) [32], Fisher Score (F-Score) [1], and ReliefF [33]. This is because these six approaches are commonly used and represent typical traditional methods to address feature selection problems. CFS and FCBF can determine the feature subset automatically. According to [34], IG, F-Score, and ReliefF use the top 5 features to perform the classification accuracy.

In this study, each stochastic approach is performed 30 independent times on each dataset. The results are recorded for each approach, which are the best classification accuracy (BeAcc), the average classification accuracies (AvgAcc), the best feature subset size (BeNF), the average feature subset size (AvgNF), and their corresponding standard deviations (Std) based on the results of the 30 independent runs.

V. RESULTS AND DISSCUSSIONS

In this section, two set experiments are conducted to evaluate the effectiveness and efficiency of the proposed algorithm. Best results are shown in bold. In the following results, "+", "-", and "≈" indicate that the proposed algorithm is significantly better than, worse than, and similar with the compared algorithms in Wilcoxon rank-sum test with a significance level of 0.05. This indicates that the more "+" symbols, the better the proposed algorithm is. In addition, Friedman's test is adopted to assess the overall performance of the algorithms on the 15 classification problems. In each Table, "Mean rank" and "Final rank" denote the average of ranks in each column and the order of the algorithms, respectively.

A. Comparisons with other PSO variants

In this subsection, the experimental results of the four algorithms are showed to illustrate the effectiveness of the proposed algorithm. Table II shows the results of the best and average classification accuracy on the test set and its standard deviation on all algorithms. In Table II, "All" denotes that all available features are used for classification. We notice that the proposed HPSO-DE algorithm has the best classification performance, where it achieved the best average classification accuracy on 10 out of the 15 datasets (i.e. Breastcancer, Glass, Zoo, WaveformEW, WDBC, Ionosphere, KrvskpEW, Movementlibras, Musk1, and Isolet). FIPS comes next by obtaining the best average accuracy on two datasets (i.e. SpectEW and Madelon). GPSO is the third most effective algorithm at achieving best average classification accuracy on two datasets (i.e. Satellite and Semeion). CLPSO comes as the last algorithm by achieving the best accuracy on the Segmentation dataset. Compared with the complete set of features, HPSO-DE significantly improves the classification accuracy on all datasets. The highest improvement is on the WaveformEW dataset by 8.23% on the average classification accuracy and 10.40% on the best classification accuracy. In addition, the results of the significance test indicate that HPSO-DE achieves significantly better results than other algorithms (GPSO, FIPS, and CLPSO) in most cases.

Table III compares HPSO-DE with the other three algorithms in terms of the number of selected features. As can be seen from Table III, the proposed HPSO-DE algorithm conducts better than other compared methods, followed by FIPS, GPSO, and CLPSO. For Isolet, which is the highest dimensional classification problems with 617 features, the proposed algorithm obtains the best average classification accuracy of 84.75%, and the subset size is only 215 features. FIPS is the most second effective algorithm for the Isolet dataset, which uses about 228 selected features to achieve

TABLE II
COMPARISON BETWEEN THE PROPOSED METHOD BASED ON THE CLASSIFICATION ACCURACY ON THE TEST SET

Dataset	HPSO-DE			GPSO			FIPS			CLPSO			All
	BeAcc	AvgAcc	Std	BeAcc	AvgAcc	Std	BeAcc	AvgAcc	Std	BeAcc	AvgAcc	Std	Acc
Breastcancer	**97.34**	**96.68**	4.34E-14	96.78	95.89	1.14E+00	96.84	96.61	3.46E-01	96.88	96.62	2.06E-01	91.66
Glass	**70.70**	**65.10**	2.80E+00	69.70	64.09	4.74E+00	69.31	64.55	3.10E+00	68.18	64.24	3.46E+00	60.56
Zoo	**92.91**	**86.67**	3.52E+00	90.91	80.81	5.36E+00	90.91	82.42	5.12E+00	91.94	84.14	6.04E+00	78.06
Segmentation	**96.86**	95.44	7.82E-02	95.86	95.36	2.78E-01	95.86	95.46	1.09E-01	96.14	**95.58**	2.49E-01	89.74
WaveformEW	**82.72**	**80.55**	1.15E+00	81.76	80.22	8.77E-01	82.62	80.42	1.24E+00	82.09	79.91	1.27E+00	72.32
SpectEW	**60.02**	52.55	3.15E+00	59.49	52.35	3.95E+00	58.02	**53.21**	3.50E+00	58.02	51.48	3.97E+00	49.78
WDBC	**96.19**	**94.19**	2.89E-14	94.19	93.41	1.58E+00	95.19	93.95	8.85E-01	95.35	93.68	1.33E+00	92.20
IonosphereEW	**92.51**	**90.19**	1.98E+00	91.02	87.52	3.06E+00	90.45	89.91	2.02E+00	90.57	85.13	2.41E+00	82.17
KrvskpEW	**97.15**	**95.90**	8.48E-02	96.15	95.54	1.03E+00	96.35	95.89	1.20E-01	94.69	92.13	1.64E+00	89.53
Satellite	90.73	89.49	5.93E-01	**90.89**	**89.66**	4.70E-01	89.76	89.35	2.93E-01	90.74	89.61	5.80E-01	86.09
Movementlibras	**80.47**	**75.47**	1.89E+00	78.97	75.14	1.86E+00	79.37	75.39	2.23E+00	79.57	74.25	2.45E+00	73.52
Musk1	**91.67**	**86.90**	2.53E+00	90.28	85.76	2.97E+00	90.67	86.67	2.11E+00	88.19	84.88	2.25E+00	81.09
Semeion	91.68	89.10	8.69E-01	90.80	**89.56**	8.66E-01	90.89	89.19	9.75E-01	**92.13**	88.00	1.34E+00	84.51
Madelon	81.69	78.03	1.47E+00	**81.90**	77.37	1.59E+00	81.67	**78.28**	1.37E+00	75.00	72.23	2.34E+00	70.93
Isolet	**87.97**	**84.75**	1.21E+00	87.61	84.19	1.44E+00	87.39	84.47	1.45E+00	84.19	80.93	1.19E+00	79.78
Mean rank	1.53			3.07			2.13			3.27			5.00
Fianl rank	1			3			2			4			5
$+/\approx/-$				7/8/0			4/11/0			10/4/1			15/0/0

TABLE III
COMPARISON BETWEEN THE PROPOSED METHOD BASED ON THE NUMBER OF SELECTED FEATURES

Dataset	HPSO-DE			GPSO			FIPS			CLPSO			All
	BeNF	AvgNF	Std	BeNF	AvgNF	Std	BeNF	AvgNF	Std	BeNF	AvgNF	Std	NF
Breastcancer	**3.00**	**3.00**	0.00E+00	**3.00**	3.10	3.05E-01	**3.00**	3.00	0.00E+00	**3.00**	3.03	1.83E-01	9
Glass	**3.00**	3.53	7.30E-01	**3.00**	3.97	9.64E-01	**3.00**	**3.43**	7.28E-01	**3.00**	3.70	6.51E-01	9
Zoo	**3.00**	5.67	1.12E+00	**3.00**	5.20	1.58E+00	**3.00**	**4.43**	1.14E+00	**3.00**	5.60	1.79E+00	16
Segmentation	**4.00**	**4.00**	0.00E+00	**4.00**	4.23	4.30E-01	**4.00**	4.00	0.00E+00	**4.00**	4.63	6.15E-01	19
WaveformEW	9.00	**9.40**	6.75E-01	**8.00**	9.77	1.01E+00	9.00	9.57	7.74E-01	**8.00**	9.90	1.42E+00	21
SpectEW	6.00	**7.50**	8.61E-01	**5.00**	7.93	1.48E+00	6.00	7.60	1.43E+00	7.00	8.73	1.17E+00	22
WDBC	**2.00**	**2.03**	1.83E-01	**2.00**	2.13	4.34E-01	**2.00**	2.10	4.03E-01	**2.00**	5.47	1.33E+00	30
IonosphereEW	**2.00**	**3.27**	5.83E-01	3.00	5.20	1.81E+00	**2.00**	3.30	7.02E-01	3.00	6.50	1.89E+00	34
KrvskpEW	9.00	9.47	5.07E-01	**8.00**	9.63	7.65E-01	9.00	**9.40**	4.98E-01	11.00	14.73	2.20E+00	36
Satellite	**7.00**	9.33	9.59E-01	8.00	9.97	1.03E+00	**7.00**	**9.23**	9.35E-01	8.00	11.10	1.35E+00	36
Movementlibras	**16.00**	**27.37**	3.74E+00	24.00	30.93	4.10E+00	20.00	27.37	3.39E+00	24.00	32.17	5.21E+00	90
Musk1	**46.00**	**56.53**	4.67E+00	52.00	64.10	5.93E+00	48.00	58.13	5.73E+00	50.00	61.73	6.21E+00	166
Semeion	**89.00**	**101.90**	7.21E+00	92.00	110.53	6.66E+00	95.00	105.67	4.74E+00	90.00	103.50	6.28E+00	256
Madelon	**149.00**	**172.20**	1.13E+01	163.00	194.87	1.25E+01	146.00	175.87	1.29E+01	179.00	200.17	1.19E+01	500
Isolet	**194.00**	**215.43**	1.17E+01	218.00	244.23	1.25E+01	205.00	228.97	1.12E+01	224.00	246.43	1.30E+01	617
Mean rank	1.50			3.2			1.70			3.60			5.00
Final rank	1			3			2			4			5
$+/\approx/-$				9/6/0			4/10/1			10/5/0			15/0/0

84.47% classification accuracy. This indicates that the proposed HPSO-DE algorithm is more effective to eliminate irrelevant and redundant features and does not reduce the classification accuracy. The results of Friedman's test show that HPSO-DE obtains significantly better average feature subset size than other compared algorithms. In terms of the significance test, the proposed algorithm wins 38, draws 21, loses 1 over the 60 comparisons.

Fig. 1 shows the fitness value changing of HPSO-DE and other compared algorithms on the fifteen datasets during the feature selection process. As can be seen from Fig. 1, we can see that the proposed algorithm is the fastest to converge than other algorithms on thirteen out of fifteen datasets (i.e. Breastcancer, Glass, Zoo, Segmentation, WaveformEW, SpectEW, WDBC, Ionosphere, Satellite, Movementlibras, Musk1, Madelon, and Isolet). This is due to using the differential operators to breed exemplars that can effectively help PSO fly out of potential local optima.

B. Comparisons with traditional methods

In this subsection, the proposed HPSO-DE algorithm is compared with five traditional feature selection methods in terms of the classification accuracy. Table IV shows the results of each method. It is obviously that the proposed HPSO-DE algorithm achieves the highest accuracy on 11 out of the 15 datasets (i.e. Glass, Zoo, Segmentation, WaveformEW, WDBC, IonosphereEW, KrvskpEW, Satellite, Musk1, Semeion, and Isolet). This verifies that HPSO-DE can effectively explore the search space to obtain better feature subsets from the

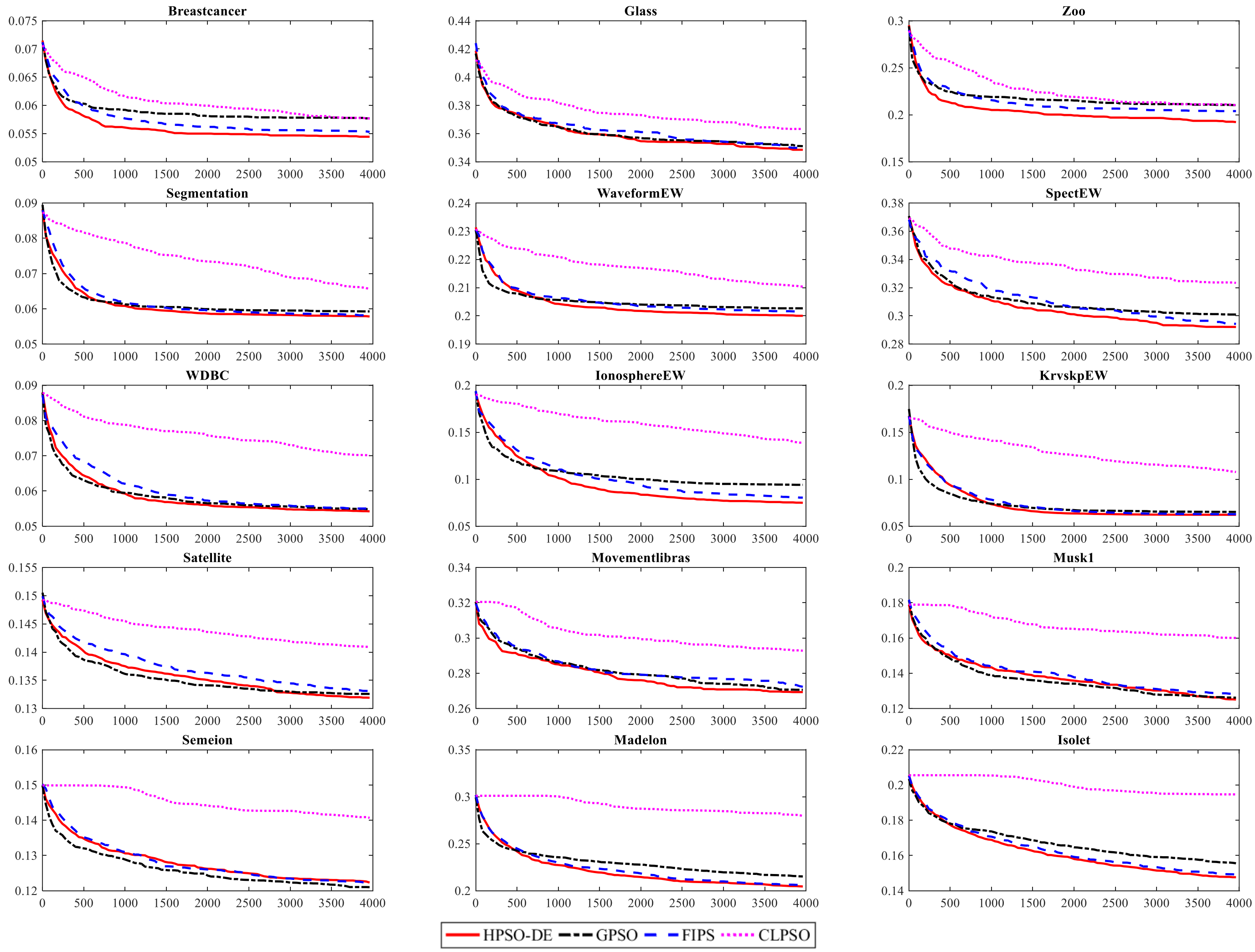

Fig. 1. Convergence curve of HPSO-DE and other three algorithms for fifteen datasets (X-axis represents the number of fitness evaluations and Y-axis denotes the fitness value).

original features than other five traditional feature selection methods. Furthermore, according to the results of Friedman's test, the proposed algorithm obtains the highest classification performance, followed by ReliefF, FCBF, F-Score, IG, and CFS.

VI. Conclusions

The goal of this article was to design an effective feature selection method in classification problems. The goal has been successfully implemented by developing a novel hybrid PSO variant based feature selection method that can effectively choose a feature subset by constructing promising exemplars based on DE operators during the search process.

The results showed that the proposed HPSO-DE algorithm can obtain a higher classification accuracy with a small feature subset than the compared methods in most of the examined datasets. This is because these exemplars constructed by DE operators can provide the ability for particles to do an efficient search. In addition, the proposed exemplar updating operator based on the spiral-shaped mechanism can effectively change the search direction of particles to help HPSO-DE fly out of potential local optima and explore more fruitful regions. In general, the proposed algorithm can successfully enhance the performance of canonical PSO, and achieve a good feature subset automatically for feature selection in classification.

As future work, we would like to apply the proposed HPSO-DE algorithm to other search and optimisation problems such as travelling salesman problem and knapsack problem.

VII. Acknowledgment

This work is supported in part by The National Key R & D Program of China (Grant No. 2017YFB1302400), National Natural Science Foundation of China (Grant No. 61773242, 61803227 and 61876169), Major Agricultural Applied Technological Innovation Projects of Shandong Province (Grant No. SD2019NJ014), Shandong Natural Science Foundation (Grant No. ZR2019MF064), Intelligent Robot and System Innovation Center Foundation (Grant No. 2019IRS19), China

TABLE IV
THE CLASSIFICATION ACCURACY OF THESE SIX METHODS

Dataset	HPSO-DE	CFS	FCBF	IG	F-Score	ReliefF
Breastcancer	96.68	95.71	**98.57**	95.74	96.24	96.55
Glass	**65.10**	63.11	64.69	62.57	62.27	64.37
Zoo	**86.67**	80.11	83.29	85.00	65.63	80.41
Segmentation	**95.44**	90.78	91.36	92.84	93.26	94.57
WaveformEW	**80.55**	62.00	71.12	66.20	66.29	74.19
SpectEW	52.55	50.34	51.24	51.89	51.99	**54.93**
WDBC	**94.19**	91.51	93.24	92.06	90.54	93.69
IonosphereEW	**90.19**	85.73	85.71	80.01	72.86	84.30
KrvskpEW	**95.90**	76.84	93.43	93.21	95.55	95.43
Satellite	**89.49**	81.90	87.18	81.74	80.89	82.45
Movementlibras	75.47	69.33	73.54	70.34	72.89	**75.93**
Musk1	**86.90**	85.23	82.53	81.94	83.79	84.08
Semeion	**89.10**	80.77	83.65	81.07	86.47	87.93
Madelon	78.03	**79.32**	78.15	77.75	72.59	76.59
Isolet	**84.75**	81.64	84.19	82.47	83.24	83.99
Mean rank	1.33	4.73	3.13	4.60	4.47	2.73
Final rank	1	6	3	5	4	2
+/≈/–		13/1/1	10/4/1	12/3/0	12/3/0	8/6/1

Scholarship Council (Grant No. 201806220187), Marsden Fund of New Zealand Government (Grant No. VUW1509 and VUW1615), the Science for Technological Innovation Challenge (SfTI) fund (Grant No. E3603/2903), the University Research Fund at Victoria University of Wellington (Grant No. 216378/3764 and 223805/3986), and MBIE Data Science SSIF Fund (Grant No. RTVU1914).

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[2] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: A multi-objective approach," *IEEE transactions on cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2012.

[3] F. Zhang, Y. Mei, and M. Zhang, "A two-stage genetic programming hyper-heuristic approach with feature selection for dynamic flexible job shop scheduling," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2019, pp. 347–355.

[4] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2015.

[5] K. Javed, H. A. Babri, and M. Saeed, "Feature selection based on class-dependent densities for high-dimensional binary data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 3, pp. 465–477, 2010.

[6] B. Xue, M. Zhang, and W. N. Browne, "A comprehensive comparison on evolutionary feature selection approaches to classification," *International Journal of Computational Intelligence and Applications*, vol. 14, no. 02, p. 1550008, 2015.

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

[8] J. Tang, S. Alelyani, and H. Liu, "Feature selection for classification: A review," *Data classification: Algorithms and applications*, p. 37, 2014.

[9] G. I. Sayed, A. E. Hassanien, and A. T. Azar, "Feature selection via a novel chaotic crow search algorithm," *Neural Computing and Applications*, vol. 31, no. 1, pp. 171–188, 2019.

[10] R. Eberhart and J. Kennedy, "Particle swarm optimization," in *Proceedings of the IEEE international conference on neural networks*, vol. 4, 1995, pp. 1942–1948.

[11] Z.-H. Zhan, J. Zhang, Y. Li, and Y.-H. Shi, "Orthogonal learning particle swarm optimization," *IEEE transactions on evolutionary computation*, vol. 15, no. 6, pp. 832–847, 2010.

[12] Y. Chen, L. Li, H. Peng, J. Xiao, and Q. Wu, "Dynamic multi-swarm differential learning particle swarm optimizer," *Swarm and evolutionary computation*, vol. 39, pp. 209–221, 2018.

[13] K. Chen, F. Zhou, L. Yin, S. Wang, Y. Wang, and F. Wan, "A hybrid particle swarm optimizer with sine cosine acceleration coefficients," *Information Sciences*, vol. 422, pp. 218–241, 2018.

[14] Y.-J. Gong, J.-J. Li, Y. Zhou, Y. Li, H. S.-H. Chung, Y.-H. Shi, and J. Zhang, "Genetic learning particle swarm optimization," *IEEE transactions on cybernetics*, vol. 46, no. 10, pp. 2277–2290, 2015.

[15] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE transactions on evolutionary computation*, vol. 10, no. 3, pp. 281–295, 2006.

[16] K. V. Price, "Differential evolution," in *Handbook of Optimization*. Springer, 2013, pp. 187–214.

[17] K. Chen, B. Xue, M. Zhang, and F. Zhou, "Novel chaotic grouping particle swarm optimization with a dynamic regrouping strategy for solving numerical optimization tasks," *Knowledge-Based Systems*, p. 105568, 2020.

[18] R. Mendes, J. Kennedy, and J. Neves, "The fully informed particle swarm: simpler, maybe better," *IEEE transactions on evolutionary computation*, vol. 8, no. 3, pp. 204–210, 2004.

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

[20] C.-S. Yang, L.-Y. Chuang, C.-H. Ke, and C.-H. Yang, "Boolean binary particle swarm optimization for feature selection," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2008, pp. 2093–2098.

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

[22] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary pso with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems*, vol. 64, pp. 22–31, 2014.

[23] B. Tran, B. Xue, and M. Zhang, "Adaptive multi-subswarm optimisation for feature selection on high-dimensional classification," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2019, pp. 481–489.

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

[25] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in engineering software*, vol. 95, pp. 51–67, 2016.

[26] K. Chen, F. Zhou, and B. Xue, "Particle swarm optimization for feature selection with adaptive mechanism and new updating strategy," in *Proceedings of the Australasian Joint Conference on Artificial Intelligence*. Springer, 2018, pp. 419–431.

[27] G. Patterson and M. Zhang, "Fitness functions in genetic programming for classification with unbalanced data," in *Australasian Joint Conference on Artificial Intelligence*. Springer, 2007, pp. 769–775.

[28] M. Lichman, "Uci machine learning repository," 2013.

[29] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE international conference on evolutionary computation*. IEEE, 1998, pp. 69–73.

[30] M. A. Hall and L. A. Smith, "Feature subset selection: a correlation based filter approach," pp. 855–858, 1997.

[31] L. Yu and H. Liu, "Feature selection for high-dimensional data: A fast correlation-based filter solution," in *Proceedings of the international conference on machine learning*, 2003, pp. 856–863.

[32] B. Azhagusundari and A. S. Thanamani, "Feature selection based on information gain," *International Journal of Innovative Technology and Exploring Engineering*, vol. 2, no. 2, pp. 18–21, 2013.

[33] M. Robnik-Šikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Machine learning*, vol. 53, no. 1-2, pp. 23–69, 2003.

[34] M. Mafarja, I. Aljarah, H. Faris, A. I. Hammouri, A.-Z. AlaM, and S. Mirjalili, "Binary grasshopper optimisation algorithm approaches for feature selection problems," *Expert Systems with Applications*, vol. 117, pp. 267–286, 2019.