Multi-objective Feature Selection in Classification: A Differential Evolution Approach

Bing Xue[1], Wenlong Fu[2], and Mengjie Zhang[1]

[1] School of Engineering and Computer Science
[2] School of Mathematics, Statistics and Operations Research
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
{Bing.Xue,Mengjie.Zhang}@ecs.vuw.ac.nz
Wenlong.Fu@msor.vuw.ac.nz

Abstract. Feature selection is an important pre-processing step in classification tasks. Feature selection aims to minimise both the classification error rate and the number of features, which are usually two conflicting objectives. This paper develops a differential evolution (DE) based multi-objective feature selection approach. The multi-objective approach is compared with two conventional methods and two DE based single objective methods, where the first algorithm is to minimise the classification error rate only while the second algorithm combines the number of features and the classification error rate into a single fitness function. Their performances are examined on nine different datasets and the results show that the proposed multi-objective algorithm successfully evolved a number of trade-off solutions, which reduce the number of features and keep or reduce the classification error rate. In almost all cases, the proposed multi-objective algorithm achieved better performance than all the other four methods in terms of both the classification accuracy and the number of features.

Keywords: Differential evolution, Feature selection, Multi-objective optimisation, Classification.

1 Introduction

In machine learning and data mining tasks, such as classification, feature selection (FS), also called dimensionality reduction, is a process of selecting a small subset of features from a large set of original features. FS can effectively increase the classification performance, speed up the training process, reduce the dimensionality of the data, and simplify the built classifiers/models [11].

FS has been of interest for many decades [6]. One of the main challenges in FS is the large search space. For a dataset including n features, the size of the search space is 2^n. Therefore, exhaustive search is impractical in most situations because of the long computational time. Although many different search techniques have been applied to FS tasks [6], most of them still have the limitations of high computational cost and being stuck in local optima [6]. Evolutionary computation (EC) includes a group of global search techniques in which differential evolution (DE) [20] is a simple yet powerful algorithm. DE has been successfully used to solve problems in a variety of fields [5], including FS [1].

G. Dick et al. (Eds.): SEAL 2014, LNCS 8886, pp. 516–528, 2014.

FS aims to minimise the number of features and maximise the classification accuracy (minimise the classification error rate). These two objectives are conflicting to each other in most cases, which makes FS a multi-objective problem. However, there are only a limited number of multi-objective FS algorithms and most of them are based on EC techniques [3, 12, 27]. Those EC techniques are population based algorithms, which are particularly suitable for multi-objective optimisation because they can produce multiple solutions in a single run [4]. Multi-objective DE gains more and more attention to solve complex multi-objective problems. Recently, Wang et al. [21] showed that DE can achieve better performance than many other EC algorithms on single objective FS, but there is only one initial work [24] on DE for multi-objective FS. This paper will further investigate this topic by significantly extend the work in [24].

1.1 Goals

The overall goal is to develop a DE based multi-objective FS approach to searching for a set of non-dominated feature subsets, which include a small subset of features and achieve similar or better classification performance than using all features. To achieve this goal, we propose a multi-objective FS approach (DEMOFS) by using a multi-objective DE algorithm to simultaneously minimise the number of features and the classification error rate. DEMOFS is compared with two conventional methods and two DE based single objective algorithms, where the first DE algorithm aims to minimise the classification error rate only and the second DE algorithm combines the classification error rate and the number of features into a single fitness function. Specifically, we will investigate:

- whether the two single objective DE algorithms can successfully reduce the number of features and maintain or even improve the classification performance over using all features,
- whether DEMOFS can achieve a set of non-dominated feature subsets, which can further reduce the number of features and improve the classification performance, and
- whether DEMOFS can outperform the two conventional FS algorithms in terms of the number of features and the classification performance.

2 Background

2.1 Differential Evolution (DE)

Differential evolution (DE) was first developed by Storn and Price [20] in 1997. Due to its simplicity, robustness and effectiveness, DE has attracted more and more attention of researchers from different fields. In DE, a candidate solution is encoded as an individual in the population. Considering there are P individuals in the population, the individual i ($1 \leq i \leq P$) can be shown by a vector $(x_{i1}, x_{i2}, ...x_{iD})$, where D is the dimensionality of the problem or search space. There is a vector $(x_{max1}, x_{max2}, ...x_{maxD})$ to define the upper bound of the search space and a vector $(x_{min1}, x_{min2}, ...x_{minD})$ to limit the lower bound of the search space. A DE algorithm starts with randomly generated initial

individuals. Then DE employs the mutation operation to produce a mutant candidate solution C_i for each individual x_i, which is also called the parent x_i.

There are different types of mutation strategies [19]. Equation (1) takes *DE/rand/1/bin* as an example to how C_i is generated.

$$C_{id} = \begin{cases} x_{id}^{i,r1} + F * (x_d^{i,r2} - x_{id}^{i,r3}), \text{ if } rand() < CR \\ x_{id}, \ otherwise \end{cases} \tag{1}$$

where $x^{i,r1}$, $x^{i,r2}$, and $x^{i,r3}$ are randomly selected from the population. x_i, $x^{i,r1}$, $x^{i,r2}$, and $x^{i,r3}$ are different from each other. $F \in (0, 1)$ is a scale factor, which controls the rate at which the population evolves. $rand()$ is a random number uniformly distributed in (0,1). CR is the crossover probability. If C_{id} falls out of the lower and upper bounds, a constraint method is usually applied to handle it. A simply way is to replace the value in C_{id} that exceeds the boundary value with the closest boundary value.

2.2 Related Work on Feature Selection

EC algorithms have been applied to FS problems, such as DE [14], genetic algorithms (GAs) [28], genetic programming (GP) [17], and particle swarm optimisation (PSO) [2, 22, 23, 25–27]. Typical EC based FS algorithms are reviewed in this section. Zhu et al. [28] proposed a FS method incorporating GA with local search (i.e. forms a memetic algorithm). Meanwhile, this algorithm combines filter ranking measure into a wrapper framework to take advantage of both filter and wrapper approaches. Fdhila et al. [8] applied multi-swarm PSO to solve FS problems. However, the computational cost of the proposed algorithm is high because it involves parallel evolutionary processes and multiple sub-swarms with a relative large number of particles. He et al. [14] applied a binary differential evolution (BDE) algorithm to filter FS, where mutual information is used to evaluate the goodness of the selected feature subsets. However, the proposed algorithm is not compared with any other algorithm and the datasets used in the experiments include a relatively small number (maximum 56) of features. Al-Ani et al. [1] also proposed a DE based FS method, where features are distributed to a set of wheels and DE is employed to select features from each wheel. This algorithm can significantly reduce the number of features and improve the classification performance.

EC algorithms have been applied to multi-objective FS. Hamdani et al. [12] developed a multi-objective FS algorithm using non-dominated sorting based multi-objective GA II (NSGAII). Neshatian and Zhang [17] proposed a GP based filter model as a multi-objective algorithm for FS in binary classification problems. Ke et al. [3] developed a Pareto-based multi-objective ant colony optimisation (ACO) for FS based on rough set theory. Xue et al. [27] proposed a PSO based multi-objective approach for wrapper FS, which shows that the PSO based algorithm outperforms three other commonly used EC based multi-objective algorithms. Wang et al. [21] showed that DE can achieve better performance than GA, PSO, ACO, and harmony search on single objective FS, but the use of DE for multi-objective FS has not been investigated to date.

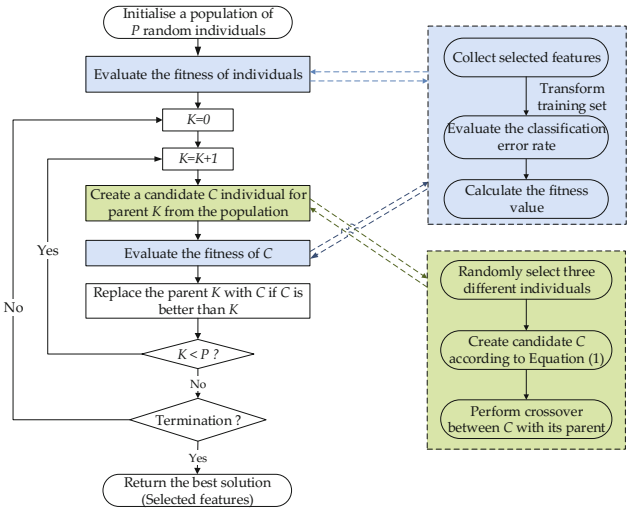

Fig. 1. Training Process of DEFS and DEFS2

3 Proposed Approach

3.1 Single Objective Algorithm 1: DEFS

To investigate the performance of DE for FS, DE is firstly used to optimise the classification performance of the selected features to form the algorithm DEFS. DEFS uses Equation (2) as the fitness function, which is to minimise the classification error rate.

$$Fit_1 = ErrorRate = \frac{FP + FN}{TP + TN + FP + FN} \tag{2}$$

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

Since FS needs to consider both the classification performance and the number of features, the number of features is not considered in the fitness function but during the evolutionary process of DEFS. If there are multiple solutions with the smallest error rate, the one with the smallest size will be reserved and others are discarded. The representation of DE follows the continuous encoding scheme since the original DE algorithm was developed for continuous problems. In DEFS, each individual in DEFS is a vector of real numbers, $x_i = (x_{i1}, x_{i2}, .., x_{id}, .., x_{iD})$, where D is the dimensionality and also the total number of features in the dataset. $0 \leq x_{id} \leq 1$ shows the probability of the dth feature being selected. A threshold θ is used to determine whether this feature is selected. If $\theta \leq x_{id}$, the dth feature is selected. Otherwise, the dth feature is not selected.

Fig. 1 shows the training or search process of DEFS, where the DE scheme *DE/rand/1/bin* [19] is used following [18]. The blue and green dash rectangles show the detailed steps of evaluating the fitness value and creating a new candidate individual, respec-

tively. After this training process, the selected feature subset will be used to transform the test set of the problem and the classification performance of the selected features will be evaluated on the transformed test set.

3.2 Single Objective Algorithm 2: DEFS2

In DEFS, the number of features is not directly included in the fitness function, although it is considered during the search process. To further investigate the use of DE for FS, an integrated fitness function, Equation (3), which combines the two objectives of minimising the classification error rate and the number of features, is used to develop another algorithm named DEFS2. Although DEFS2 considers the two objectives, it is treated as a single objective algorithm since it follows a single objective search process.

$$Fit_2 = \alpha * ErrorRate + (1 - \alpha) * \frac{\#Size}{D} \tag{3}$$

where $\alpha \in (0, 1]$ is a weight parameter showing the relative importance of the classification error rate. $\#Size$ represents the number of selected features and D is the total number of features in the dataset. $\frac{\#Size}{D}$ is used to scale the value to (0,1] to be in the same range of $ErrorRate$. α should be larger than 0.5 to make sure $\alpha > (1 - \alpha)$, i.e. the classification performance is more important than the number of features.

Equation (3) is used to investigate whether directly considering the number of selected features can further reduce the number of features without significantly reducing the classification accuracy. The search or training process of DEFS2 can also be shown by Fig. 1. The main difference between DEFS and DEFS2 is the fitness function.

3.3 Multi-objective Algorithm: DEMOFS

Similar to most of other EC approaches, DE was proposed to solve single objective problems. Based on a popular evolutionary multi-objective algorithm, i.e. non-dominated sorting based genetic algorithm II (NSGAII), Robič and Bogdand [18] developed a multi-objective DE algorithm named DEMO [18] to use DE for multi-objective optimisation. DEMO has shown promising performance on some problems, but it has never been applied to FS problems. In this paper, we develop a multi-objective FS algorithm named DEMOFS based on DEMO to investigate the use of DE for multi-objective FS. DEMOFS aims to minimise the classification error rate and the number of features. The representation of DEMOFS is the same as DEFS and DEFS2.

Algorithm 1 shows the pseudo-code of DEMOFS, where the two key steps are the decision of the newly constructed individual and the update of the population. In Algorithm 1, Line 8 to Line 16 show the decision on the newly constructed individual. After this procedure, the population exceeds the pre-defined maximum number of individuals. To update the population, a truncation step is needed, which is shown in Line 19. This is similar to that in NSGAII [7], which involves the use of the non-dominated sorting and crowding distance metric. Specifically, the non-dominated solutions in the population are called the first non-dominated front, which are excluded from the population. Then the non-dominated solutions in the new population are called the second non-dominated

Algorithm 1. Pseudo-Code of DEMOFS

```
1  begin
2      randomly initialise individuals;
3      while Stopping Criterion is not met do
4          evaluate the number of features selected by each individual and its Training
           error rate;
5          for i=1 to Number of individuals do
6              create candidate C from parent i ;      /* details as shown in
               Fig. 1 */
7              evaluate the two objective values of C;
8              if C dominates i then
9                  use C to replace i;
10             end
11             else if i dominates C then
12                 C is discarded;
13             end
14             else if i and C are non-dominated to each other then
15                 C is added to the population;
16             end
17         end
18         if the population size exceeds the maximum value then
19             truncate the population according to non-dominated sorting
20         end
21         randomly enumerate the individuals in the population;
22     end
23 end
24 calculate the testing classification error rate of the non-dominated solutions;
25 return the non-dominated solutions and their training and testing error rates.
```

front. The following levels of non-dominated fronts are identified by repeating this procedure. For the next generation, solutions (individuals) are selected from the top levels of the non-dominated fronts to form a new/updated population, 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 population for the next generation. Otherwise, the solutions in the current non-dominated front are ranked according to the crowding distance and the highest ranked (least crowded) solutions are added into the next generation.

4 Design of Experiments

4.1 Benchmark Techniques

The three DE based FS algorithms (i.e. DEFS, DEFS2 and DEMOFS) are examined and compared with each other on nine datasets shown in Table 1, which were selected from the UCI machine learning repository [9]. For each dataset, the instances are randomly

Table 1. Datasets

Dataset	NO. of Features	NO. of Classes	NO. of Instances
Wine	13	3	178
Australian	14	2	690
Zoo	17	7	101
Vehicle	18	4	846
German	24	2	1000
Lung Cancer	56	3	32
Sonar	60	2	208
Hillvalley	100	2	606
Musk Version 1 (Musk1)	166	2	476

divided into two sets: 70% as the training set and 30% as the test set. The nine datasets are chosen to have different numbers of features, classes and instances to be used as representatives of problems that the proposed algorithms can address.

All the algorithms are wrapper approaches, i.e. requiring a classification algorithm to evaluate the classification error rate of the selected features. A commonly used classification algorithm, K-nearest neighbour (KNN), is used here and K=5. During the training process, KNN with 10-fold cross-validation is employed to evaluate the classification error rate of the selected feature subset on the training set, and then the selected features are evaluated on the test set to obtain the testing classification error rate [15].

Two traditional wrapper FS methods are used to compare with that of the DE based algorithms, which are linear forward selection (LFS) [10] and greedy stepwise backward selection (GSBS), which were derived from two typical greedy search based FS, i.e. SFS and SBS, respectively. Details about LFS and GSBS can be seen from [10] and [16]. Weka [13] is used to run the experiments of LFS and GSBS. All the settings in LFS and GSBS are kept to the defaults. The parameters of the three DE based algorithms are set as follows. The population size is 80 and the maximum number of generations is 100. The crossover rate is set as 0.3. All the three DE based algorithms share the same representation. The threshold θ is set as 0.6 [27]. The parameter α in DEFS2 is set as 0.95, which means that the classification performance is much more important than the number of features. LFS and GSBS are deterministic methods, which produce a unique solution. The DE based algorithms are stochastic methods and each of them has been performed for 30 independent runs on each dataset.

5 Results and Discussions

In this section, we first compare the performance of DE based stochastic algorithms, DEFS, DEFS2 and DEMOFS, then compare their performance with that of the two traditional (deterministic) methods, LFS and GSBS.

DEFS and DEFS2 are single objective methods producing 30 solutions for each dataset from the 30 independent runs. DEMOFS is a multi-objective algorithm producing 30 sets of solutions for each dataset from the 30 independent runs. To compare their results, the 30 sets of solutions are combined together to extract two sets of solutions, which are the "best" set and the "average" set. The "best" set means the non-dominated solutions achieved by DEMOFS across the 30 independent runs. The average set contains the solutions with different numbers of features. For a certain number (e.g. m), its

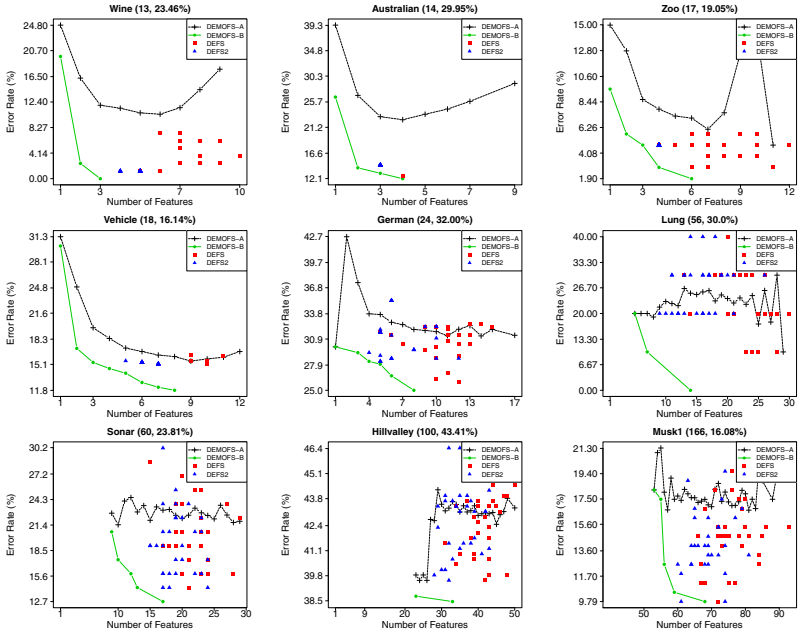

Fig. 2. Experimental Results of DEFS, DEFS2 and DEMOFS

classification error rate is the average error rate of all the available feature subsets that include m features. Fig. 2 shows the results of DEFS, DEFS2 and DEMOFS, where "DEMOFS-A" shows the "average" set and "DEMOFS-B" shows the "best" set. The nine charts correspond to the nine datasets used in the experiments. In each chart, the numbers in the bracket are the total number of features and the classification error rate achieved by using all features. For DEFS and DEFS2, there might be fewer than 30 distinct dots shown in a chart. The reason is that many different solutions may have the same number of features and the same classification error rate and they are plotted in the same dot.

5.1 Results of DEFS and DEFS2

From Fig. 2, it can be seen that on all the nine datasets, DEFS reduced around half of the features and reduced or achieved similar classification error rate to using all the available features. The results show that DEFS uses DE as the search technique can effectively search the solution space to reduce the number of features and maintain or even improve the classification performance.

Fig. 2 shows that DEFS2 selected around one third of the available features and achieved a similar or even lower classification error rate than using all features. For example, on the Australian dataset, DEFS2 selected only three features from the 14 available features and reduced the classification error rate from 29.95% to 14.5%. Comparing DEFS2 with DEFS, DEFS2 which directly considers the number of features in

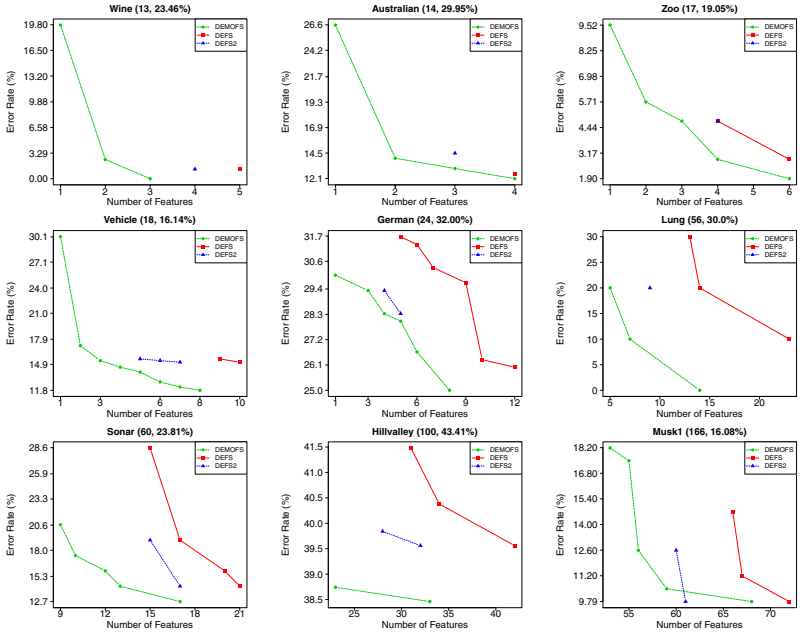

Fig. 3. Further Comparisons Between DEFS, DEFS2 and DEMOFS: Non-Dominated Solutions

the fitness function can further reduce the number of features without increasing the classification error or evening reducing it over DEFS. The main reason is that the number of features is considered but with a very small weight in the fitness function, which on one hand results in the reduction of the number of features. On the other hand, slightly compromising the classification performance with the number of features can also help avoid the over-fitting problem, which may achieve better performance on the test set.

5.2 Results of DEMOFS

From Fig. 2, it can be observed that for all cases, at least one feature subset in the "average" set included a smaller number of features and achieved similar or lower classification error rate than using all features. Note that DEMOFS reports a set of non-dominated solutions in each run, but when combining the solutions from multiple runs, some solutions may be dominated by others. Therefore, some of the solutions in the "average" set dominate others. In all datasets, the "best" set included a significantly smaller number of features and increased the classification accuracy over all features.

Fig. 2 suggests that by employing a multi-objective search mechanism, DEMOFS can effectively explore the search space to obtain a number of non-dominated feature subsets, which significantly reduced the number of features and improve the classification performance over using all features.

Table 2. Results of LFS ang GSBS

	Wine		Australian		Zoo		Vehicle		German	
	# Features	Error (%)	# Features	Error (%)	# Features	Error (%)	# Features	Error (%)	# Features	Error (%)
LFS	7	25.93	4	29.95	8	20.95	9	16.93	3	31.33
GSBS	8	14.81	12	30.43	7	20.0	16	24.21	18	35.67
	Lung		Sonar		Hillvalley		Musk1			
	# Features	Error (%)	# Features	Error (%)	# Features	Error (%)	# Features	Error (%)		
LFS	6	10.0	3	22.22	8	42.31	10	14.69		
GSBS	33	10.0	48	31.75	90	50.55	122	23.78		

5.3 Comparisons between DEFS, DEFS2 and DEMOFS

Fig. 2 also shows that the classification performance of the "average" set is often slightly worse or similar to that of DEFS and DEFS2, but the solutions in the "best" set is always better than DEFS and DEFS2. This is not surprised because DEFS, DEFS2 and DEMOFS share the same (total) number of evaluations, but DEFS and DEFS2 focus on the optimisation of the classification accuracy and return only one single solution from each run. DEMOFS returns a set of feature subsets with trade-off between the accuracy and the number of features. Therefore, when the error rates of feature subsets from different runs are averaged, it may be slightly worse the DEFS and DEFS2.

The "average" set gives an overall idea of the solutions achieved by DEMOFS, especially for the number of features, but it has a potential limitation because the solutions in the "average" set are not meaningful solutions. The reason is that in FS problems, the solutions themselves cannot be averaged because the each solution involves a number of features. Such individual features cannot be averaged to get an "average" solution, although the numbers of features and their classification error rates can be averaged to show the general performance. This is not a problem for the "best" set involving the original non-dominated solutions only. Therefore, in order to further compare the performance of DEFS, DEFS2 and DEMOFS, the non-dominated solutions obtained by DEFS and DEFS2 over the 30 independent runs are also collected and compared with that of DEMOFS, where the results are shown in Fig. 3.

From Fig. 3, it can be observed that DEFS2 generally achieved similar or better performance than DEFS. This is generally consistent with the results in Fig. 2, but shows a clearer pattern. On all the nine datasets, the solutions of DEMOFS dominate that of DEFS. On eight of the nine datasets, the solutions of DEMOFS dominate that of DEFS2. The only exception is the Musk1 dataset, where one of the solutions from DEFS2 achieved a similar classification performance to that of DEMOFS, but selected a smaller number of features. The results from Fig. 3 further show that DEMOFS has the potential to obtain better feature subsets than DEFS and DEFS2, which included a smaller number of features and a lower classification error rate.

5.4 Comparisons with Traditional Methods

Table 2 shows the results of the two traditional FS algorithms. Both LFS and GSBS are deterministic algorithms that produce a unique solution.

Comparing the results in Table 2 to that in Fig. 2 and 3, it can be seen that DEFS, DEFS2 and DEMOFS were able to outperform both LFS and GSBS in terms of both the classification performance and the number of features on eight of the nine datasets. Only

on the Lung dataset, LFS outperformed DEFS and DEFS2, but DEMOFS achieved better performance than LFS. The results show that DEFS, DEFS2 and DEMOFS employ DE as the search technique can better explore the solution space to obtain better results than LFS and GSBS.

6 Conclusions and Future Work

This paper investigated the use of DE for multi-objective FS in classification. The algorithm DEMOFS was proposed to simultaneously minimise the number of features and the classification error rate. The experiments on the nine different datasets show that DEMOFS successfully evolved a set of trade-off solutions to reduce both the number of features and the classification error rate. The results show that DEMOFS outperformed two commonly used conventional FS methods (LFS and GSBS) in terms of both the classification performance and the number of features. DEMOFS was also compared with two DE based single objective FS algorithms (DEFS and DEFS2), where DEFS aimed to minimise the classification error rates and DEFS2 combined both the classification error rate and the number of features into a single fitness function. DEMOFS outperformed both DEFS and DEFS2 by employing a multi-objective search mechanism. All the three DE based algorithms achieved better classification accuracy than the two traditional algorithms.

This work discovers that DE can be successfully used for multi-objective FS. It also provides motivations for further investigating EC particularly DE methods for multi-objective FS. There are many future research directions, which can be seen as follows:

1. The performance of DEMOFS needs to compare with other EC based multi-objective FS algorithms, such as PSO and GAs, which was not conducted in this paper due to the page limit;
2. A new multi-objective DE algorithm needs to be developed to further improve the performance of DE for multi-objective FS;
3. DE was originally for continuous problems, but FS is a binary task. Therefore, a binary DE is demanded to better solve the problem;
4. This paper focuses on wrapper based algorithms and the investigation of DE for filter based multi-objective FS is still an open issue;
5. Classification on datasets with over a thousand or a few thousands of features is still a challenge. Investigating effective and efficient multi-objective FS approaches on such large-scale problems can help address this challenge; and
6. To investigate the trade-off between the number of features and the classification performance and decide how to select a single solution from a set of non-dominated solutions in multi-objective FS.

Acknowledgment. This work is supported in part by the National Science Foundation of China (NSFC No. 61170180), the Marsden Funds of New Zealand (VUW1209 and VUW0806) and the University Research Funds of Victoria University of Wellington (203936/3337).

References

1. Al-Ani, A., Alsukker, A., Khushaba, R.N.: Feature subset selection using differential evolution and a wheel based search strategy. Swarm and Evolutionary Computation 9, 15–26 (2013)
2. Cervante, L., Xue, B., Zhang, M., Shang, L.: Binary particle swarm optimisation for feature selection: A filter based approach. In: IEEE Congress on Evolutionary Computation (CEC 2012), pp. 881–888 (2012)
3. Chen, Y., Miao, D., Wang, R.: A rough set approach to feature selection based on ant colony optimization. Pattern Recognition Letters 31(3), 226–233 (2010)
4. Coello Coello, C., Veldhuizen, L.A.G., Evolutionary Algorithms, D.: Evolutionary Algorithms for Solving Multi-Objective Problems. Genetic and Evolutionary Computation Series. Springer (2007)
5. Das, S., Suganthan, P.: Differential evolution: A survey of the state-of-the-art. IEEE Transactions on Evolutionary Computation 15(1), 4–31 (2011)
6. Dash, M., Liu, H.: Feature selection for classification. Intelligent Data Analysis 1(4), 131–156 (1997)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
8. Fdhila, R., Hamdani, T., Alimi, A.: Distributed mopso with a new population subdivision technique for the feature selection. In: International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII 2011), pp. 81–86 (2011)
9. Frank, A., Asuncion, A.: UCI machine learning repository (2010)
10. Gutlein, M., Frank, E., Hall, M., Karwath, A.: Large-scale attribute selection using wrappers. In: IEEE Symposium on Computational Intelligence and Data Mining, pp. 332–339 (2009)
11. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182 (2003)
12. Hamdani, T.M., Won, J.-M., Alimi, A.M., Karray, F.: Multi-objective feature selection with NSGA II. In: Beliczynski, B., Dzielinski, A., Iwanowski, M., Ribeiro, B. (eds.) ICANNGA 2007. LNCS, vol. 4431, pp. 240–247. Springer, Heidelberg (2007)
13. Hastie, T., Tibshirani, R., Friedman, J., Franklin, J.: The elements of statistical learning: data mining, inference and prediction. The Mathematical Intelligencer 27, 83–85 (2005)
14. He, X., Zhang, Q., Sun, N., Dong, Y.: Feature selection with discrete binary differential evolution. In: International Conference on Artificial Intelligence and Computational Intelligence (AICI 2009), vol. 4, pp. 327–330 (2009)
15. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97, 273–324 (1997)
16. Marill, T., Green, D.: On the effectiveness of receptors in recognition systems. IEEE Transactions on Information Theory 9(1), 11–17 (1963)
17. Neshatian, K., Zhang, M.: Pareto front feature selection: using genetic programming to explore feature space. In: The 11th Annual Conference on Genetic and Evolutionary Computation (GECCO 2009), pp. 1027–1034 (2009)
18. Robič, T., Filipič, B.: DEMO: Differential evolution for multiobjective optimization. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 520–533. Springer, Heidelberg (2005)
19. Storn, R.: On the usage of differential evolution for function optimization. In: 1996 Biennial Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 519–523 (1996)
20. Storn, R., Price, K.: Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4), 341–359 (1997)

21. Wang, L., Ni, H., Yang, R., Pappu, V., Fenn, M.B., Pardalos, P.M.: Feature selection based on meta-heuristics for biomedicine. Optimization Methods and Software, 1–17 (2013)
22. Xue, B., Cervante, L., Shang, L., Browne, W.N., Zhang, M.: A multi-objective particle swarm optimisation for filter based feature selection in classification problems. Connection Science 24(2-3), 91–116 (2012)
23. Xue, B., Cervante, L., Shang, L., Browne, W.N., Zhang, M.: Binary PSO and rough set theory for feature selection: A multi-objective filter based approach. International Journal of Computational Intelligence and Applications 13(02), 1450009 (2014)
24. Xue, B., Fu, W., Zhang, M.: Differential evolution (DE) for multi-objective feature selection in classification. In: Proceedings of the 2014 Conference Companion on Genetic and Evolutionary Computation Companion, GECCO Comp 2014, pp. 83–84 (2014)
25. Xue, B., Zhang, M., Browne, W.N.: Multi-objective particle swarm optimisation (PSO) for feature selection. In: Genetic and Evolutionary Computation Conference, pp. 81–88 (2012)
26. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. Applied Soft Computing (2013)
27. Xue, B., Zhang, M., Browne, W.N.: Particle swarm optimization for feature selection in classification: A multi-objective approach. IEEE Transactions on Cybernetics 43(6), 1656–1671 (2013)
28. Zhu, Z.X., Ong, Y.S., Dash, M.: Wrapper-filter feature selection algorithm using a memetic framework. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 37(1), 70–76 (2007)