Pareto Front Feature Selection based on Artificial Bee Colony Optimization

Emrah [1],[3], Bing Xue[3], Mengjie Zhang[3], Dervis Karaboga[2], and Bahriye Akay[2]

[1] *Department of Computer Technology and Information Systems, Mehmet Akif Ersoy University, Burdur 15030, Turkey.*

[2] *Department of Computer Engineering, Erciyes University, Kayseri 38039, Turkey.*

[3] *School of Engineering and Computer Science, Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand. Email: Bing.Xue@ecs.vuw.ac.nz*

Abstract

Feature selection has two major conflicting aims, i.e. to maximize the classification performance and to minimize the number of selected features to overcome the curse of dimensionality. To balance their trade-off, feature selection can be handled as a multi-objective problem. In this paper, a feature selection approach is proposed based on a new multi-objective artificial bee colony algorithm integrated with non-dominated sorting procedure and genetic operators. Two different implementations of the proposed approach are developed: ABC with binary representation and ABC with continuous representation. Their performance are examined on 12 benchmark datasets and the results are compared with those of linear forward selection, greedy stepwise backward selection, two single objective ABC algorithms and three well-known multi-objective evolutionary computation algorithms. The results show that the proposed approach with the binary representation outperformed the other methods in terms of both the dimensionality reduction and the classification accuracy.

Keywords: Feature selection, classification, multi-objective optimization, artificial bee colony.

1. Introduction

Data mining is in the intersection of artificial intelligence, machine learning, statistics and database systems. It is basically the process of extracting valuable knowledge embedded in data and then transforming the knowledge into an understandable format for users through the steps, such as data pre-processing, management, post-processing and visualization [14]. Data mining and machine learning techniques can be mainly divided into unsupervised (e.g. clustering), supervised (e.g. classification) and reinforcement learning [14]. This paper focuses mainly on classification, which aims to learn a model based on a training set of instances and predict the class labels of unseen instances in the test set. Classification has been used in various real-world applications such as medical healthcare, image analysis, marketing and statistical problems [43, 26]. However, the datasets, especially large dimensional ones, may comprise redundant, irrelevant and relevant features. This brings the problems of high complexity and poor learning performance in real-world applications [43].

One of the most common ways to overcome these problems is to apply feature selection [37]. Feature selection aims to select the most relevant/useful features which contribute to the constructed model more efficiently and effectively. Not only for the classification performance, it is also beneficial for simplifying the learned models and shortening the training time. However, finding relevant/useful features is not an easy task due to the huge search space and the complex interactions among features. Feature interaction may occur in two ways, three ways or more than three ways. An individually irrelevant feature may be beneficial for the classification/learning performance while being interacted with other features. An individually relevant feature may become redundant when it is interconnected with other features. Furthermore, there exist 2^n possible feature subsets for a n-dimensional dataset. It is impractical to intimately search all possible solutions for a large n. Accordingly, feature selection is an NP-hard combinatorial problem [37]. Even though a number of search techniques such as sequential forward and backward feature selection (SFS, SBS) [26] have been proposed, they may have premature convergence problems or intensive computational complexity. To alleviate these problems, evolutionary computation (EC) techniques which are population based solvers in the subclass of global optimization and artificial intelligence have been applied due to their global search potential. The mostly commonly applied techniques for feature selection are genetic programming (GP) [36], genetic algorithms (GAs) [32] and particle swarm optimization (PSO) [37, 27, 35]. EC techniques are particularly good at multi-objective optimization because their population based search mechanism can produce multiple trade-off solutions in a single run.

It can be inferred from the two main conflicting objectives of feature selection, i.e. the maximization of the classification accuracy and the minimization of the feature subset size, that feature selection can be treated as a multi-objective problem. Unfortunately, there exist just a few studies concerning multi-objective feature selection in the literature [43], i.e., most of the existing approaches are based on a single objective of maximizing the classification accuracy. One of the recent metaheuristics, artificial bee colony (ABC) [19] is an EC technique with many successful applications to solve different problems, which is a motivation to design ABC for multi-objective feature selection. Furthermore, ABC is easy implement, robust against initialization, and has the ability to explore local solutions with the low risk of local convergence. Our recent study [16] has shown that ABC can be used for multi-objective feature selection, but the method in [16] is for *filter* feature selection and the number of features in the datasets is small. The potential of ABC for multi-objective *wrapper* feature selection, which requires often a different approach from *filters* [21], and with a large number of features, has not been investigated yet.

1.1. Goals

The main goal of this paper is to improve an ABC-based feature selection approach to searching for a set of Pareto optimal solutions yielding a smaller feature subset size and a lower classification error percentage than the case that all features are used. To fulfill this goal, a new multi-objective ABC approach based on non-dominated sorting and genetically inspired search is proposed, and two different implementations of the proposed approach are developed: Bin-MOABC (binary version) and Num-MOABC (continuous version). Bin-MOABC and Num-MOABC are compared with two traditional approaches, two single objective ABC variants and three well-known multi-objective feature selection approaches on 12 benchmark datasets including a variety of features, classes and instances.

Specifically, the following objectives are investigated:

1. the performance of single objective ABC approaches on reducing the feature subset size and increasing the classification performance,
2. the performance of the proposed multi-objective ABC implementations on obtaining Pareto optimal solutions and comparisons with two traditional and two single objective ABC approaches,
3. the performance analysis of the proposed multi-objective ABC implementations versus existing multi-objective approaches, and

4. the effect of considering feature selection in binary domain (Bin-MOABC) and continuous domain (Num-MOABC) on the classification performance.

1.2. Organization

The organization of the rest of the paper is as follows. A general knowledge concerning the standard ABC algorithm and the recent studies on feature selection is provided in Section 2. The proposed feature selection approaches are explained in Section 3. The experimental design is described in Section 4 and the experimental results are presented with discussions in Section 5. Finally, the conclusions and the future trends are introduced in Section 6.

2. Background

In this section, ABC is described, the definition of multi-objective optimization is given, and then the recent research of the feature selection is briefly reviewed.

2.1. Artificial Bee Colony

ABC is a swarm intelligence algorithm that simulates the foraging behavior of a honey bee colony [19]. In the hive, three types of bees are assigned to the foraging task: employed bees, onlooker bees and scout bees. Employed bees are responsible from loading the nectar of discovered sources to the hive and dancing in the hive to share their information about the profitable sources with onlooker bees waiting in the hive. The onlooker bees watch the dances of the employed bees and choose a source to exploit. Scout bees search for undiscovered sources based on internal motivation or an external clue. In other words, employed and onlooker bees are responsible for exploiting food sources and scout bees are responsible for exploring new food sources. From the optimization perspective, each food source corresponds to a solution ($x_i = \{x_{i1}, x_{i2}, ..., x_{iD}\}$) in a D dimensional optimization problem and the nectar amount of a source represents the fitness value of the solution.

In the exploration-exploitation process of food sources, each employed bee searches in the neighborhood of the food source in her memory, while each onlooker bee searches in the neighborhood of the food source according to the information shared by employed bees through waggle dance. The basic steps of ABC are as follows.

1. Food sources are initialized by Eq. (1):

   $$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \tag{1}$$

   where i is the index of a food source in the range of 1 and SN and SN is the population size; j is the position of a food source in the range of 1 and D and D is the dimensionality of the search space; x_j^{min} and x_j^{max} are lower and upper bounds of position j.
2. Each employed bee i evolves its concerning food source by Eq. (2):

   $$\upsilon_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \tag{2}$$

   where x_i is the current food source; x_k is the selected food source for x_i; j is the randomly selected position to be perturbated; υ_i is the evolved food source by evolving jth parameter of x_i; and ϕ_{ij} is a uniformly generated value in the range of -1 and 1.
3. Apply greedy selection between υ_i and x_i. If $f(\upsilon_i) > f(x_i)$, the employed bee leaves x_i and memorizes υ_i as the current source.
4. Each food source is assigned a probability by Eq. (3).

   $$p_i = \frac{fitness_i}{\sum\limits_{i=1}^{SN} fitness_i} \tag{3}$$

   where $fitness_i$ is the fitness value of source x_i and SN is the population size.
5. Each onlooker bee chooses a food source in a probabilistic manner, and then carries out searching as in the employed bee phase.
6. If there exists any exhausted food source which is determined by a ‘limit’ value, the scout bee generates a new food source using Eq. (1) instead of abandoned one.
7. Repeat steps 2 to 6 until the maximum number of cycles is met.

2.2. Multi-Objective Optimization

Many problems involve two or more conflicting objectives, called multi-objective optimization problems. This type of problems are typically with many solutions known as Pareto-optimal solutions.

Let $f(x) = (f_1(x), f_2(x), ..., f_{n_o}(x)) \in O \subseteq \mathbb{R}^{n_0}$ be an objective vector comprising of multiple (n_0) conflicting functions and let $F \subseteq S$ (where S is the search space) represents the feasible space constrained by n_g inequalities and n_h equality constraints;

$$F = \{x : g_m(x) \leq 0, h_l(x) = 0, m = 1, ..., n_g; l = 1, ..., n_h\} \tag{4}$$

where $g_m(x)$ and $h_l(x)$ are constraints. Using this notation, a multi-objective (minimization) problem can be formulated as follows:

$$\text{minimize } f(x) \text{ subject to } x \in F \tag{5}$$

When there are multiple objectives, for two solutions y and z, y dominates z iff y is is not worse than z in all objectives and better than z in at least one objective:

$$\forall k : f_k(y) \leq f_k(z) \wedge \exists k : f_k(y) < f_k(z) \tag{6}$$

A solution $x^* \in F$ is defined as a Pareto optimal (non-dominated) solution if there does not exist a solution $x \neq x^* \in F$ that dominates x^*. The set of all non-dominated solutions form a Pareto-optimal front surface, known as Pareto front.

2.3. Existing Feature Selection Approaches

Feature selection approaches can be categorized into wrapper, filter and embedded approaches [26]. While wrapper approaches use a classification algorithm to select a feature subset according to the classification performance, filter approaches generally use statistical or probabilistic properties of datasets and do not depend on any classifier or learning system. Since filter approaches do not employ any classifier or learning system, they are computationally less intensive and more general than wrappers. However, wrappers are able to get more promising results than filters. On the other hand, embedded approaches try to find an optimal feature subset in the learning process, i.e., they are dependent on the nature of classification model. Although embedded approaches are computationally less intensive than wrappers, they are conceptually more complex, and it is not easy to make a modification in the classification model to get higher performance [26]. Therefore, this paper focuses on wrapper approaches.

2.3.1. Non-EC Approaches

The most well-known traditional wrapper approaches are sequential forward selection (SFS) [40] and sequential backward selection (SBS) [28]. SFS starts with an empty feature subset and sequentially selects features for this subset until no improvement is received on the classification performance. In contrast to SFS, SBS starts with a feature subset including all available features in the dataset and then sequentially eliminates features from this set until no improvement is received on the classification performance via further elimination. Although both SFS and SBS are simple to implement, they may converge to local minima and are computationally expensive in high-dimensional datasets. Based on SFS and SBS, the sequential forward floating selection (SFFS) and sequential backward floating selection (SFBS) [31] were introduced to sort out the common limitation of SFS and SBS, in which a feature selected or removed in earlier steps cannot be updated later. Unfortunately, these attempts to overcome local-minima were not sufficient.

In traditional filter approaches, FOCUS [12] exhaustively examines all possible feature subsets and then selects the smallest subset through correlation. However, exhaustive search is computationally intensive when concerned with a great number of features. Relief [20] ranks features according to their weights obtained by randomly sampling instances from the data. Each weight reflects the relevance of its associated feature with the class labels. However, it does not address the redundancy among features. In contrast to FOCUS and Relief, information theoretic approaches such as MIFS [5], mRmR [30] and MIFS-U [22] considers both the relevance of each feature with the class labels and the redundancy within the feature subset.

2.3.2. Single Objective EC based Approaches

To address the drawbacks of traditional approaches, researchers have also applied EC techniques, including GAs [29], GP [36], PSO [27] and ABC [34] to feature selection problems.

Raymer et al. [32] introduced a GA based approach performing feature selection and feature extraction processes simultaneously, which achieved better results than the SFFS [31] and linear discriminant analysis (LDA) approaches. Oh et al. [29] hybridized GA (HGA) through embedding local search operations. The results showed that HGA performed better than standard GA.

Liu et al. [27] introduced a multi-swarm PSO based approach using the classification accuracy and the F-score in a weighted manner. Different from the existing studies, it considers the population as sub-populations. However, it is computationally inefficient. Huang and Dun [17] proposed an efficient distributed PSO-SVM approach, which includes two components: 1) binary PSO for feature selection and 2) standard PSO for parameter optimization of SVM. Chuang et al. [7] proposed an improved binary PSO algorithm based on catfish effect. In the proposed algorithm, when the global best particle could not be improved for a predefined number of iterations, 10% of particles with low quality are exchanged with new generated ones. According to the results, catfish based PSO performed better than Oh's HGA [29]. Unler et al. [37] proposed a hybrid filter-wrapper PSO approach to bring the advantages of filters and wrappers together. The effectiveness of the wrapper-filter approach was demonstrated by comparing it with another hybrid filter and wrapper approach based on GA.

ABC has been successfully applied to a wide range of fields [19], such as color quantization, automatic clustering, image analysis, and parameter optimization. Recently, researchers have also tried to address the feature selection problem using ABC in a single objective manner. Uzer et al. [38] introduced a combined ABC-SVM feature selection approach for medical datasets. Subanya and Rajalaxmi [34] proposed a hybridization of ABC and Naive Bayes, and it was tested on the Cleveland Heart disease dataset. However, the proposed approach was not compared with existing studies. Schiezaro and Pedrini [33] proposed a feature selection approach using ABC based on single modification rate (MR). The results indicated that the proposed ABC algorithm outperformed the standard ABC, PSO and GA algorithms. Hancer et al. [15] improved an advanced similarity based discrete ABC wrapper approach. The superiority of the discrete ABC wrapper approach was demonstrated by making comparisons with six well-known binary ABC and PSO variants on 10 benchmark datasets.

2.3.3. Multi-Objective EC Based Approaches

Hamdani et al. [13] proposed a non-dominated sorting GA II (NSGAII) based approach. Waqas et al. [39] also proposed a multi-objective GA based wrapper approach, in which a decision tree was chosen as the classifier. Xue et al. [45] introduced a multi-objective PSO based wrapper approach (CMDPSO) inspired by crowding distance, non-dominated sorting and mutation for feature selection. In this work, the classification error rate and the number of features were chosen as objective functions. The results showed that CMDPSO outperformed NSGAII and strength Pareto evolutionary algorithm 2 (SPEA2). Xue et al. [41, 42] also used multi-objective PSO for filter feature selection with objective functions formed by mutual information and rough set theory.

Despite a number of existing approaches, most of them are single objective approaches considering the classification accuracy as a single objective. It is not possible to find a sufficient number of studies in the literature approaching feature selection as a multi-objective problem, i.e., this issue has just recently come into consideration. Furthermore, our recent study [16] has shown that ABC can be used for multi-objective feature selection, but the method is for *filter* feature selection and the number of features in the datasets is small. The potential of ABC for multi-objective *wrapper* feature selection, which often requires a different approach from *filters* [21], and with a large number of features, has not been investigated yet.

3. Proposed Multi-Objective ABC Approach

As mentioned in previous section, feature selection can be considered as a multi-objective problem through two main conflicting objectives: 1) minimizing the feature subset size and 2) maximizing the classification accuracy. Despite the success of ABC in different fields, there is no multi-objective ABC based wrapper approach in the literature. To cover this issue, an ABC based multi-objective feature selection approach with its two implementations are proposed in this section.

The standard ABC algorithm was proposed for single objective problems, and cannot be used for multi-objective feature selection. So modifications/adaptations are required on probability calculation scheme, solution update scheme and solution generation scheme to deal with multi-objective problems. Inspired by the concept and ideas of NSGAII [9] and non-dominated sorting synchronous ABC (NSSABC) [2], we develop and implement both the binary and continuous versions of the multi-objective ABC approach, named Bin-MOABC and Num-MOABC respectively. For the clarity of the presentation purpose, we first present the structure of Bin-MOABC and Num-MOABC in Algorithm 1 to give an overall idea of the proposed methods, then describe more details of the key components.

A. How to Calculate Probabilities for Onlookers: For a single objective problem, a probability is simply assigned to a food source according to Eq. (3). However, Eq. (3) is not suitable for multi-objective problems since

```
begin
    Generate initial population $X = X_1, X_2, ..., X_n$ by Eq. (1);
    Evaluate initial population $X$ (i.e. error rate and number of features);
    Apply non-dominated sorting to solutions;
    for $cycle \leftarrow 1$ to $MCN$ do
        foreach employed bee $i$ do
            Randomly select a solution $X_k$ for $X_i$;
            Generate solutions by applying BSG (or NSG) between $X_i$ and $X_k$;
            Evaluate the generated solutions and add them to set $S$;
        end
        Rank the union set $X \cup S$ via non-dominated sorting;
        Update $X$ by selecting the best $SN$ solutions through ranking and
          crowding distance scores;
        $S = \emptyset$;
        foreach onlooker bee $i$ do
            Select a solution $X_i$ using thermodynamic principles by Eq. (8);
            Randomly select a solution $X_k$ for $X_i$;
            Generate solutions by applying BSG (or NSG) between $X_i$ and $X_k$;
            Evaluate generated solutions and add them to set $S$;
        end
        Rank the union set $X \cup S$ via non-dominated sorting;
        Update $X$ by selecting the best $SN$ solutions through ranking and
          crowding distance scores;
        if any abandoned solution then
            Generate a new solution instead of abandoned one by Eq. (1);
        end
    end
    Compute the classification accuracy of population $X$ on the test set;
    Rank the population using non-dominated sorting and return the population;
end
```

Algorithm 1: Pseudo code of Bin-MOABC and Num-MOABC.

they have more than one objectives. Therefore, the following probability assignment scheme is employed:

$$p_i = \frac{Newfitness_i}{\sum\limits_{i=1}^{SN} Newfitness_i} \tag{7}$$

where $Newfitness_i$ (calculated by Eq. (8)) is based on Gibbs distribution [23, 48] and Pareto rank value. In statistical physics, the Gibbs distribution designs a framework in thermo-dynamical equilibrium at a given temperature and minimizes the free energy (the principle of the minimal free energy).

Since the key goals of multi-objective optimization (convergence towards the Pareto-optimal set and the maximization of diversity) are analogous to the principle of finding the minimum free energy state in a thermodynamic system, in MOABC, a fitness assignment technique (8) proposed in [48] is used to compute the fitness of an individual.

$$Newfitness_i = \frac{1}{R(i) - T * S(i) - d(i)} \tag{8}$$

where $R(i)$ is the Pareto rank value of the individual i, $T > 0$ is a predefined constant value referred as temperature, $d(i)$ is the crowding distance determined by the crowding distance assignment scheme [9], and

$$S(i) = -p_T(i) \log_{p_T}(i) \tag{9}$$

where

$$p_T(i) = (1/Z) \exp(-R(i)/T),$$

and

$$Z = \sum_{1}^{SN} \exp(-R(i)/T)$$

where $p_T(i)$ is the Gibbs distribution, Z is the partition function and SN is the population size.

This fitness assignment scheme helps to converge to the Pareto-optimal solutions with a high diversity among the solutions, based on the principle of thermodynamics [47].

B. How to Update Individuals: To update individuals, greedy selection is applied between the current and newly generated individuals through mutation and crossover. However, the individuals do not always dominate the other individuals in multi-objective scenario. Therefore, a fast non-dominated sorting scheme instead of greedy selection is applied to select better individuals with lower cost to be retained in the population. The purpose of this scheme is to sort individuals according to the level of non-domination. Each solution is compared with other solutions to determine whether it is dominated. Solutions that are not dominated by any other solution form the first non-dominated Pareto front. To find the solutions in the next front, the solutions appeared in the first front are temporarily discounted and the same procedure is repeated. For each solution p, two entities are calculated: the number of solutions dominating solution p (referred

```
begin
    foreach $p \in P$ do
        foreach $q \in P$ do
            if $p$ dominates $q$ then
                $S_p = S_p + \{q\}$;
            else
                $n_p = n_p + 1$;
            end
        end
        if $n_p = 0$ then
            $F_1 = F_1 \cup \{p\}$;
        end
    end
    i=1;
    while $F_i \neq \emptyset$ do
        $H = \emptyset$;
        foreach $p \in F_i$ do
            foreach $q \in F_i$ do
                $n_q = n_q - 1$;
                if $n_q = 0$ then
                    $H = \cup \{q\}$;
                end
            end
        end
        $i = i + 1$;
        $F_i = H$;
    end
end
```

Algorithm 2: Pseudo code of Fast Non-Dominated Sorting.

as n_p) and the number of solutions dominated by solution p (referred as S_p). In the non-dominated sorting, good solutions are determined by a ranking selection method, and a niche method is applied to keep sub-populations of good points stable. The fast non-dominated sorting for set P is presented in Algorithm 2 [9].

C. How to Generate New Individuals: Due to the large dimensionality and the complex interactions among features, some improvements in the algorithm are also required to overcome the curse of dimensionality and to increase the classification accuracy together. To search the solution space more deeply and to maintain diversity in the population, a deeper search is required. To achieve this, each solution of the population should be evaluated in different perspectives.

Bin-MOABC and Num-MOABC use different representations which are binary domain and continuous domain, respectively, so they use different ways to generate new solutions. In Bin-MOABC, for each solution x_i, a neighborhood solution x_k is selected via random selection in the employed bee phase or via probabilistic selection in the onlooker bee phase. After selection, the two-point crossover and two-way mutation are sequentially applied to generate new offsprings (defined as binary solution generator (BSG)):

1. Two-point crossover: Two positions are randomly determined on binary parents x_i and x_k. Everything between the positions of x_i is copied to x_k to generate the first offspring. Then, everything between the positions of x_k is copied to x_i to generate the latter one. In this way, two offsprings are generated.
2. Two-way mutation: A new mutation scheme is applied in this study. First, a number within the range of 0 and 1 is uniformly generated. If the generated number is greater than 0.5, a position with value 1 is chosen and its position is set to 0. Otherwise, a position with value 0 is chosen and its position is set to 1. In this way, diversity is satisfied in solution generation and two offsprings are generated. An illustrative sample of two-way mutation is presented in Fig. 1.

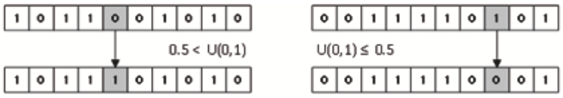

Figure 1: An illustrative representation on how two-way mutation is applied.

In Num-MOABC, the simulated binary crossover (SBX) [1] and polynomial mutation [25] are sequentially applied to the current and neighborhood solutions (defined as numeric solution generator (NSG)):

1. Simulated Binary Crossover (SBX) generates two offsprings in the following way [1]:

$$\begin{aligned}\text{off}_{1,k} &= \frac{1}{2}\left[(1-\beta_k)x_{i,k} + (1+\beta_k)x_{j,k}\right] \\ \text{off}_{2,k} &= \frac{1}{2}\left[(1+\beta_k)x_{i,k} + (1-\beta_k)x_{j,k}\right]\end{aligned} \tag{11}$$

   where $off_{1,k}$ is the offspring with kth dimension, $x_{i,k}$ and $x_{j,k}$ are the ith and jth solutions with kth dimension, and β_k is the uniformly distributed sample.

2. Polynomial Mutation generates offsprings in the following way [25]:

$$\text{off} = x_{i,j} + (x_{i,j}^{max} - x_{i,j}^{min})\delta_j \tag{12}$$

where δ_j is a variation calculated through polynomial distribution:

$$\begin{aligned} &\delta_j = (2U(0,1))^{\frac{1}{\eta_m+1}} - 1, \text{ if } U(0,1) < 0.5 \\ &\delta_j = 1 - [2(1 - U(0,1))]^{\frac{1}{\eta_m+1}} \text{ , otherwise} \end{aligned} \tag{13}$$

where $U(0,1)$ is a uniformly generated number between 0 and 1, and η_m is mutation distribution index.

Therefore, totally four new offsprings are generated for each parent.

Based on the overall structure shown in Algorithm 1 and the above mentioned schemes, one can see that in the proposed algorithms, a solution (referred as neighborhood solution) is randomly chosen for each current solution in the employed bee phase. Between the current solution and its neighborhood solution, the proposed solution generator is applied to form a new solution set S. In this way, four offsprings are generated for each solution. Note that if the applied algorithm is Bin-MOABC, BSG is used; otherwise, NSG is applied. After the employed bee phase is completed, the solutions in the union set of X and S are ranked using non-dominated sorting, and SN number of solutions are selected to update the population set X through rank and crowding distance. Then, the onlooker bee phase is carried out. In the onlooker bee phase, a neighbor is randomly chosen using thermodynamic principles formulated into Eq. (8), and then genetically inspired NSG or BSG generators are applied to generate new solutions as in employed bee phase. After that, the population set X is updated by selecting the SN highest ranked solutions from the union set of X and S.

D. Representation and Fitness Function: Each solution represents the activation code (selected or unselected) of the corresponding feature. While activation codes vary in the range between 0 and 1 in Num-MOABC, they are shown through discrete values 0 and 1 in Bin-MOABC. If the activation code of a position is greater than a user specified threshold value, its corresponding feature is selected; otherwise, it is not selected. In this study, the threshold value is defined as 0.5 as in [45, 42]. The classification error rate of a feature subset is calculated by:

$$ErrorRate = \frac{FP + FN}{FP + FN + TP + TN} \tag{14}$$

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

4. Experiment Design

Twelve datasets comprising of various numbers of features (from 24 to 657), classes (from 2 to 26) and samples (from 351 to 6598) are chosen from UCI machine learning repository [4] and are shown in Table 1, where the Multiple Features and Optic Characters datasets are referred as 'Multi' and 'Optic', respectively. Each dataset is randomly divided into two sets: 70% as the training set and 30% as the test set, where the partition is stratified to make sure the same class distribution in both sets. The classification performance the feature subsets is evaluated using K Nearest Neighbor (KNN) with $K = 5$. During the feature selection process, the training set is further partitioned to 10 folds in a stratified way, and 10-fold cross-validation with 5NN is applied as an inner on the training set to evaluate the classification performance of the selected features, i.e. to be used in the fitness function. The inner loop of 10-fold cross-validation is used to avoid feature selection bias, and a detailed discussion on why and how they should be applied in this way is given in [21]. Note that for the proposed wrapper feature selection methods, any classification algorithm can be used here. We chose KNN because it is simple and relatively cheap, which is particularly important for feature selection problems. Since two main disadvantages of wrapper feature selection are being computationally expensive and less general to other classification methods, using a relatively cheap and simple method can avoid such issues to some extent. Previous research [46] has shown that using a simple and relatively cheap classification algorithm (like KNN) in a wrapper approach can select a good (near-optimal) feature subset for other complex learning/classification algorithms (e.g. SVM), which are computationally expensive but able to achieve better classification performance.

To evaluate the performance of the proposed multi-objective ABC based feature selection methods, two traditional, two single objective and three multi-objective algorithms are employed in the experimental studies. The two traditional approaches are linear forward selection (LFS) [10] and greedy stepwise backward selection (GSBS) [6] based on SFS and SBS, respectively. They are computationally more efficient and can achieve better performance than SFS and SBS. The experiments of LFS and GSBS are performed via

Table 1: Datasets

Datasets	Features	Classes	Samples
Vehicle	18	4	846
German	24	2	1000
Ionosphere	34	2	351
***Optical* Recognition of Handwritten Digits**	64	10	5620
Libras *Movement*	90	15	360
Hill Valley	100	2	606
Musk 1	166	2	476
Musk 2	166	2	6598
Semeion	256	10	1593
Madelon	500	2	2600
Isolet	617	26	1559
***Multiple* Features**	649	10	2000

Waikato Environment for Knowledge Analysis (WEKA) [11] and the feature sets obtained by the approaches are evaluated on the test sets using 5NN.

The single objective feature selection approaches are based on standard ABC (ABC-ER and ABC-Fit_{2C}) using only the classification error rate (Eq. (14)), and the classification error rate and the number of features together (Eq. (15)) in a weighted manner defined by the parameter α. As in Num-MOABC, solutions representing feature subsets are within the range of 0 and 1. If a dimension in a solution is greater than 0.5, the corresponding feature is selected; otherwise, it is not selected.

$$Fit_{2C} = \alpha * \frac{SubsetSize}{AllSetSize} + (1-\alpha) * \frac{ErrorRate}{ER} \tag{15}$$

where α is the predefined value within 0 and 1; $SubsetSize$ is the feature subset size; $AllSetSize$ is the number of all available features in the dataset; $ErrorRate$ is the classification error rate calculated through the selected feature subset; and ER is the error rate calculated through all available features in the dataset.

The employed multi-objective feature selection approaches are as follows: NSGAII [9], NSSABC [2] and multi-objective PSO (MOPSO) [8]. Previous research shows that MOPSO with a continuous representation achieved better performance than with a binary representation [44] and NSGAII with a binary representation achieved worse performance than MOPSO [45]. Therefore, we use a continuous representation in both MOPSO and NSGAII as benchmark methods for comparison. NSGAII uses non-dominated sorting, where the population is sorted based on non-dominance relationship. While doing this, crowding distance measure defining how close individual pairs are to each other is used to satisfy diversity in population. NSSABC [2] is inspired by the non-dominated sorting concept of NSGAII. In NSSABC, non-dominated sorting is applied after mutants are generated, and a mutant solution is generated by Eq. (16).

$$\upsilon_{ij} = \begin{cases} x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \text{ if } U(0,1) < MR \\ x_{ij} \text{ otherwise} \end{cases} \tag{16}$$

where MR is the predefined parameter which controls the number of parameters to be modified.

MOPSO [8] uses an external repository to keep a historical record of the non-dominated vectors detected during the search process. A mutation operator is also integrated to the algorithm to avoid premature convergence. Although there exist various multi-objective PSO algorithms, the reason of selecting Coello's MOPSO [8] for comparisons is that this variant is one of the most well-known multi-objective PSO algorithms.

For the experiments of multi-objective algorithms, the defined parameter values are as follows: the number of individuals (particles, foods and chromosomes) is set to 30; the maximum number of evaluations is empirically defined as 6000; the parameters of MOPSO are selected as in [8] where $c_1 = 1.49$, $c_2 = 1.49$ and inertial weight= 0.72; the parameters of NSGAII are selected according to [18] where crossover rate and mutation rate are set to 0.8 and 0.3, respectively; the limit parameter of all ABC based feature selection approaches is set to 100; the T parameter of multi-objective ABC variants is set to 10000; and the MR parameter of NSSABC is chosen 0.5 as in [2]. Lastly, the α parameter of Eq. (15) is set to 0.2 as in [45].

The results of the feature selection approaches are presented over 30 independent runs in terms of the classification accuracy and feature subset size in Section 5. Note that, LFS and GSBS obtain a unique feature subset on each dataset, and standard ABC obtains a single best result in each of the 30 runs on each dataset, while multi-objective approaches obtain a set of feature subsets in each run. The results obtained by multi-objective approaches are collected into a union set. In the union set, the classification accuracy of the feature subsets including same subset size are averaged. These mean classification accuracy of the same sized feature subsets are called the average Pareto front. In addition to the "average" Pareto front, the non-dominated solutions in the union set are also used for the comparison of different algorithms.

5. Experimental Results

The results are mainly presented in three subsections: 1) Single objective ABC vs. Traditional Approaches, 2) Multi-objective ABC vs. Single objective ABC and Traditional Approaches, and 3) Comparisons of multi-objective approaches. In addition to these subsections, the computational CPU times, and comparisons via the Hypervolume indicator are reported to investigate the effectiveness and search ability of the approaches.

5.1. Single Objective ABC vs. Traditional Approaches

The experimental results of ABC-ER, ABC-Fit_{2C}, LFS and GSBS are presented in Table 2 in terms of the classification accuracy ('CAcc') and the number of features ('NOF'). Furthermore, the results obtained by 5-NN using all features are also presented in Table 2, denoted as 'All'. As LFS and GSBS generate a unique solution, there is no standard deviation value for their results.

For GSBS and LFS, it is seen that LFS can reduce at least half of the available features for each dataset, but it obtains poor classification accuracies for 5 out of 12 datasets. On the other hand, GSBS selects a larger number of features, but it can perform much better than LFS in terms of the classification accuracy rate. However, the feature subsets obtained by GSBS may still include irrelevant and redundant features.

Table 2 shows that ABC-ER (only based on the classification error rate) almost always achieves higher classification performance than the case that all features are used, and it can select around half of the available features. Further, it obtains small feature subset size and similar or high classification accuracy in most cases when compared to GSBS.

According to Table 2, ABC-Fit_{2C} can get a feature subset the size of which is around half or less than half of the available features, and performs better than the case that all feature are used. ABC-Fit_{2C} also gets higher classification accuracy than LFS in most cases. Furthermore, ABC-Fit_{2C} performs similar or slightly better than GSBS except for the Madelon dataset in terms of the classification performance, and reduces feature subset size more effectively than GSBS. When compared with the ABC-ER approach, ABC-Fit_{2C} generally obtains similar classification performances, but eliminates irrelevant and redundant features effectively. To improve the classification accuracy and reduce the feature subset size simultaneously, Pareto front multi-objective algorithms are needed.

Table 2: Results of ABC-ER, ABC-Fit_{2C}, LFS and GSBS

Datasets		**ABC-ER**	**ABC-Fit_{2C}**	**LFS**	**GSBS**	**ALL**
Vehicle	CAcc	79.53(1.67)	77.88(1.87)	72.11	75.3	76.10
	NOF	9.86	7.73	9	16	18
German	CAcc	70.17(1.14)	70.1(1.94)	68.33	69.33	68
	NOF	10.76	9.13	5	20	24
Ionosphere	CAcc	92.12(1.80)	91.74(2.02)	90.48	89.52	89.52
	NOF	12	11.53	6	29	34
Optical	CAcc	98.10(0.31)	98.22(0.24)	97.86	98.75	98.87
	NOF	41.13	37.43	32	38	64
Movement	CAcc	77.58(2.21)	77.46(2.59)	71.43	77.14	80.00
	NOF	42.56	40.23	10	79	90
Hill Valley	CAcc	54.13(2.11)	54.92(1.78)	55.49	54.40	52.75
	NOF	47.63	44.96	9	95	100
Musk 1	CAcc	83.11(2.42)	82.32(2.95)	80.71	82.86	80.00
	NOF	83.03	80.56	12	124	166
Musk 2	CAcc	81.52(2.93)	81.54(2.55)	82.87	80.24	79.99
	NOF	82.26	81.26	8	122	166
Semeion	CAcc	87.96(0.84)	86.56(1.09)	77.33	91.04	90.83
	NOF	131.96	132.2	27	237	256
Madelon	CAcc	72.91(1.74)	72.20(2.08)	71.03	74.88	71.79
	NOF	252.46	248.03	7	250	500
Isolet	CAcc	82.52(1.23)	82.71(1.14)	76.28	80.77	80.98
	NOF	312.93	306.80	27	585	617
Multiple	CAcc	96.57(0.28)	96.78(0.28)	82.33	93.20	95.17
	NOF	322.83	315.50	20	472	649

5.2. Multi-objective ABC vs. Single objective ABC

To investigate whether considering feature selection problem in a multi-objective ABC manner can perform better than considering in a single objective ABC manner, the experimental results of Bin-MOABC, Num-MOABC, ABC-ER and ABC-Fit_{2C} are presented through charts in Fig. 2. Each chart concerns with one of the datasets considered in the experimental study. In each chart, the horizontal and vertical axes represent the feature subset size and the classification accuracy, respectively. On top of each chart, the numbers in the brackets correspond to the number of available features and the classification accuracy using all features. On the corner side of each chart, '-A' and -B' represent the "average" Pareto front and the non-dominated solutions, respectively. Single objective approaches may converge to the same solution (feature subset) in different runs in some datasets. Therefore, the appeared points on some charts for single objective approaches may be fewer than 30 points.

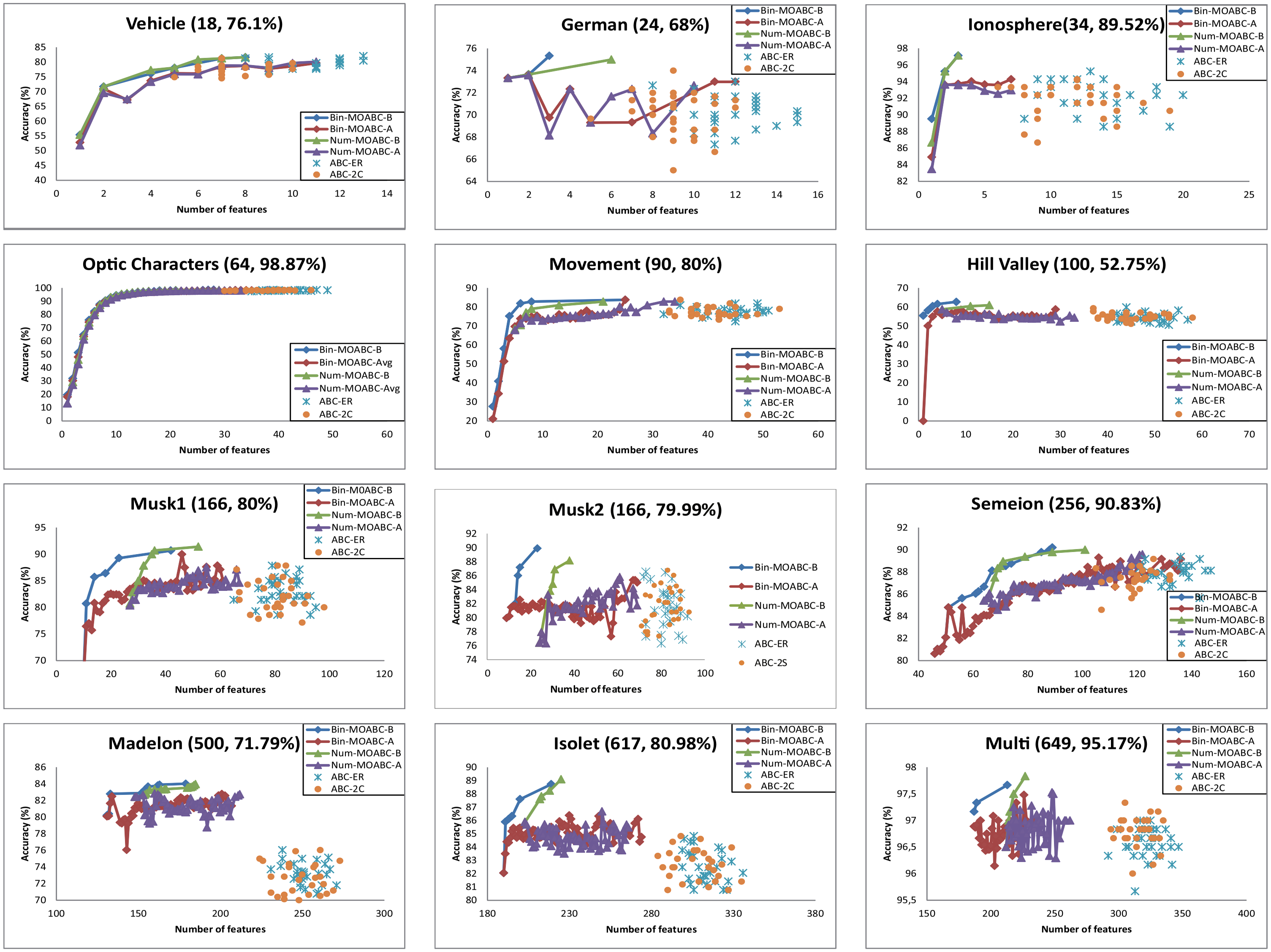

Figure 2: Results of Bin-MOABC, Num-MOABC, ABC-ER and ABC-Fit_{2C} (in color).

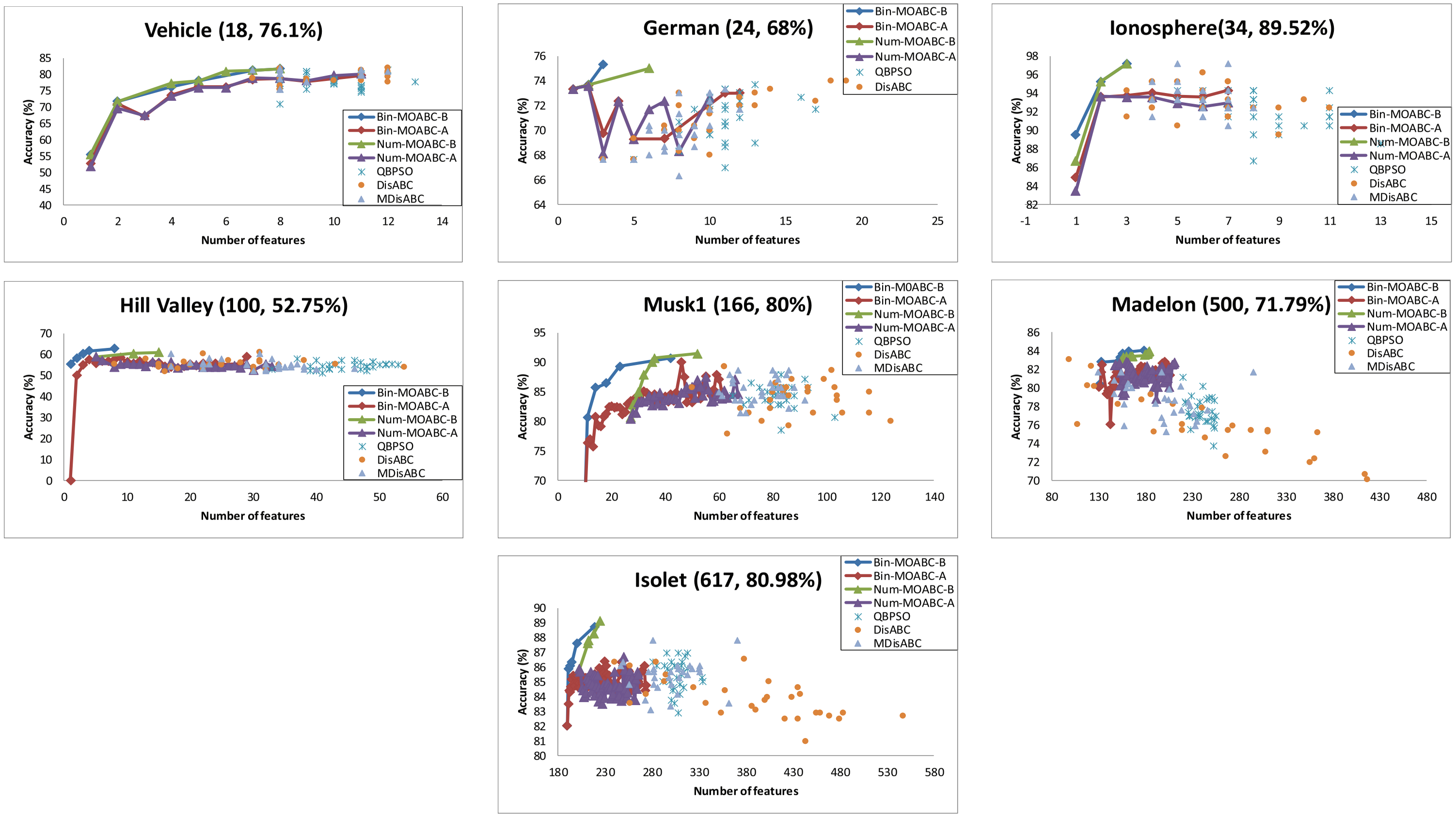

Figure 3: Results of Bin-MOABC, Num-MOABC, QBPSO, DisABC and MDisABC (in color).

From Fig. 2, it can be observed that Bin-MOABC and Num-MOABC can reduce the feature subset size, and can perform better than using all features in all cases. In almost all datasets, the number of features obtained by Bin-MOABC and Num-MOABC is smaller than 50% of all available features. For instance, on the Musk1 dataset, Bin-MOABC and Num-MOABC reduce the dimensionality from 166 to 40, but increase the classification accuracy from 80% to 90% and 91%, respectively. Accordingly, it can be inferred from the results that considering feature selection in the multi-objective ABC framework is useful and successful versus using all features.

When comparing Bin-MOABC and Num-MOABC with single objective ABC approaches (ABC-ER and ABC-Fit_{2C}), it is seen that the lines obtained by Bin-MOABC and Num-MOABC mostly dominate the points representing the results of single objective approaches, which means that ABC-ER and ABC-Fit_{2C} mostly cannot remove irrelevant or redundant features as well as Bin-MOABC and Num-MOABC. Although ABC-ER and ABC-Fit_{2C} reach similar feature subset size with Bin-MOABC and Num-MOABC in some cases, they cannot obtain higher classification performance than Bin-MOABC and Num-MOABC. For instance, on the Madelon dataset, Bin-MOABC and Num-MOABC obtain 83.26% and 82.71% accuracies using 154 features, but ABC-ER and ABC-Fit_{2C} cannot reduce feature subset size and improve the classification performance as well as Bin-MOABC and Num-MOABC. Therefore, the comparisons suggest that the proposed Bin-MOABC and Num-MOABC approaches can explore the search space more effectively than the single objective ABC approaches to detecting better feature subsets. In addition, the weight (α) between the classification error rate and the feature subset size shown in Eq. (15) does not need to be fine tuned in multi-objective approaches as in single objective approaches.

Comparisons With Recent Single Objective Approaches: To clarify the performance of Bin-MOABC and Num-MOABC versus single-objective approaches, we futher compare them with quantum binary inspired PSO (QBPSO), discrete binary ABC (DisABC) and advanced similarity based discrete binary ABC (MDisABC) feature selection approaches. The experimental results of Bin-MOABC, Num-MOABC, QBPSO, DisABC and MDisABC are presented over 7 common datasets in Fig. 3.

From Fig. 3, it can be seen that Bin-MOABC and Num-MOABC also perform better than QBPSO, DisABC and MDisABC in terms of reducing the feature subset size and increasing the classification accuracy in most cases. For the cases where recent single-objective ABC and PSO variants achieve similar classification results, multi-objective feature selection approaches successfully eliminates the irrelevant and redundant features compared to ABC and PSO variants. For instance, on the Ionosphere datasets, Bin-MOABC and Num-MOABC obtain 97.14% accuracy using 3 features, MDisABC obtains the same accuracy using 5 and 7 features. Only on the Madelon dataset, DisABC obtains smaller feature subsets. It therefore suggests that considering feature selection in the multi-objective framework is successful and useful versus considering in the single objective framework.

5.3. Comparisons Between Multi-Objective Approaches

To test the performance of Bin-MOABC and Num-MOABC with multi-objective approaches, NSGAII, NSSABC and MOPSO are employed. We first present the overall results of using the Hypervolume indicator to give an overall idea of the performance difference. Then the results of non-dominated and average Pareto fronts are presented on the test sets in Figs. 4 and 5, and on the training sets in Figs. 6 and 7.

Comparisons via Hypervolume Indicator: In order to measure the quality of the obtained Pareto fronts, hypervolume indicator [3] is employed to further compare the approaches. Hypervolume metric defined by Eq. (17) gives the volume of hypercube covered by the members of Pareto-solutions.

$$HV = volume\left(\bigcup_{i=1}^{|P|} v_i\right) \tag{17}$$

In each run, each approach obtains two Pareto fronts: training Pareto front based on the training classification accuracy and feature subset size, and testing Pareto front based on the testing classification accuracy and feature subset size. For each approach, 30 hypervolume values are calculated based on the training results, and 30 hypervolume values are calculated based on the test results. The obtained hypervolume values are normalized into the range of 0 and 1 and then Wilcoxon Rank Sum test (in which the confidence level is 95%) is applied to measure the differences between the proposed and existing approaches. To determine whether there exists any difference between approaches, the following markers are used in the tables:

- "+" indicates that Bin-MOABC (Num-MOABC) is significantly better than another corresponding approach, while "-" indicates that corresponding approach is better than Bin-MOABC (Num-MOABC).

- "=" indicates that the results of Bin-MOABC (Num-MOABC) are similar to the results of corresponding approach.
- the empty cells indicate that Bin-MOABC (Num-MOABC) is "non-applicable" with itself.

Table 3: Wilcoxon Rank Sum Test of Hypervolume Ratios on Training

	Vehicle		**German**		**Ionosphere**		**Optical**		**Movement**		**Hill Valley**	
	Bin	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**
BinMOABC		=		=		-		-		-		-
NumMOABC	=		=		+		+		+		+	
NSSABC	+	+	=	+	+	+	+	+	+	+	+	+
NSGAII	=	=	=	=	+	+	+	-	+	-	+	+
MOPSO	=	=	+	+	=	=	+	+	+	=	+	+

	Musk 1		**Musk2**		**Semeion**		**Madelon**		**Isolet**		**Multiple**	
	Bin	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**
BinMOABC		-		-		-		-		-		=
NumMOABC	+		+		+		+		+		=	
NSSABC	+	=	+	+	+	=	+	+	+	+	+	+
NSGAII	+	-	+	-	=	-	+	=	+	=	=	=
MOPSO	+	=	+	=	+	=	-	-	=	-	+	+

Table 3 shows the results of Wilcoxon Rank Sum test on the hypervolume ratio in the training process, in which 'Bin' and 'Num' refer to Bin-MOABC and Num-MOABC, respectively. Note that the comparisons are processed from the top side (Num and Bin) to the left side. The results indicate that Bin-MOABC is superior to the other approaches in most cases. Only for 1 out of 60 cases (5 algorithms × 12 datasets), Bin-MOABC gets significantly worse results than MOPSO. However, the same case cannot be suggested for Num-MOABC. For instance, Num-MOABC obtains worse results than NSGAII in most cases.

Table 4 presents the results of Wilcoxon Rank Sum test on the hypervolume ratio in the testing process. According to Table 4, for low dimensional datasets, Vehicle, German and Ionosphere, Bin-MOABC achieves similar results with Num-MOABC, NSGAII and MOPSO, but gets significantly better results than NSSABC. For the high dimensional datasets, Bin-MOABC achieves significantly better results than the other approaches in all cases. Num-MOABC generally obtains significantly better results than NSSABC, and similar or worse results than NSGAII and MOPSO.

Detailed Comparisons: According to Fig. 4, for the datasets such as Vehicle, German and Ionosphere, there is no significant difference between the non-dominated results of all algorithms in most cases. Except for these

Table 4: Wilcoxon Rank Sum Test of Hypervolume Ratios on Testing

	Vehicle		**German**		**Ionosphere**		**Optical**		**Movement**		**Hill Valley**	
	Bin	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**
Bin-MOABC		=		=		=		-		-		-
Num-MOABC	=		=		=		+		+		+	
NSSABC	=	=	+	+	+	+	+	+	+	+	+	+
NSGAII	=	=	=	=	=	=	+	-	+	-	+	=
MOPSO	=	=	-	-	=	=	+	+	+	=	+	=

	Musk 1		**Musk2**		**Semeion**		**Madelon**		**Isolet**		**Multiple**	
	Bin	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**	**Bin**	**Num**
Bin-MOABC		-		-		-		-		-		=
Num-MOABC	+		+		+		+		+		=	
NSSABC	+	=	+	+	+	= +	+	+	+	+	+	
NSGAII	+	-	+	=	+	-	+	=	+	=	+	+
MOPSO	+	=	+	=	+	=	+	-	=	-	=	=

low dimensional datasets, the differences between the algorithms can be easily illustrated such that the proposed Bin-MOABC outperforms the others in almost all cases in terms of the classification performance and the number of features. For instance, on the Madelon dataset, Bin-MOABC reduces the feature subset size from 500 to 148 and obtains 82.56% classification accuracy. However, the other approaches cannot remove features and increase the classification accuracy as in Bin-MOABC. It can be extracted from Fig. 4 that NSGAII, Num-MOABC, NSSABC and MOPSO are ranked as second, third, fourth and the last order, respectively. It is seen that NSSABC and especially MOPSO are not good at eliminating irrelevant and redundant features. When taking a look at the overall distribution of solutions (average Pareto fronts) in Fig. 5, the results indicate that the success of Bin-MOABC and Num-MOABC carry on in most cases, especially in Movement, Hill Valley, Musk1, Musk2, Madelon and Isolet.

Not only on the test sets, but also on the training sets Bin-MOABC outperforms the others in terms of both non-dominated solutions and average Pareto fronts, as shown in Figs. 6 and 7. Most of the lines representing the results of other approaches are appeared under the lines of Bin-MOABC, which reflects that Bin-MOABC also has the potential to significantly minimize the number of features and increase the training classification accuracy together. As in the test sets, the performances of the NSGAII, NSSABC and MOPSO approaches can be ranked as second, third and last positions in the training sets.

5.4. Further Comparisons using the Quadratic Classifier

In order to see whether the proposed approaches can carry on their successful performances through different classifiers against other approaches, we use quadratic discriminant analysis [24] which is a more general version of linear discriminant analysis. Quadratic discriminant analysis first computes the sample mean of each class. Then, it evaluates the sample covariances by first subtracting the sample mean of each class from the observations of that class, and taking the empirical covariance matrix of each class. The results of multi-objective approaches over quadratic discriminant analysis are presented in Figs. 8 and 9 on the test sets. In each chart, the horizontal axes represent the number of features and vertical axes represent the classification accuracy. On top of each chart, the numbers in the brackets correspond to the number of available features and the classification accuracy obtained by quadratic discriminant analysis using all features. Note that it could not be applied to the other 5 datasets since the computed covariance matrix of each group must be positive.

According to Fig. 8, for the Vehicle and German datasets which are low-dimensional problems, the non-dominated results obtained by the multi-objective approaches are mostly similar to each other. On the other hand, as for the Musk1, Hill Valley and Madelon datasets, the non-dominated results obtained by Bin-MOABC are strongly better than other approaches in almost all cases in terms of the classification performance and the number of features. For instance, on the Hill Valley dataset, Bin-MOABC achieves 89.28% accuracy for 19 features, while NSGAII obtains 84.28% accuracy for the same number of features. According to Fig. 9, Bin-MOABC performs better than others also in average Pareto fronts. Although NSGAII generally performs better than Num-MOABC in terms of the non-dominated results, Num-MOABC mostly achieves more successful results than NSGAII in terms of average Pareto fronts. Therefore, it can be inferred that the success of the proposed approaches also carries on using quadratic discriminant analysis.

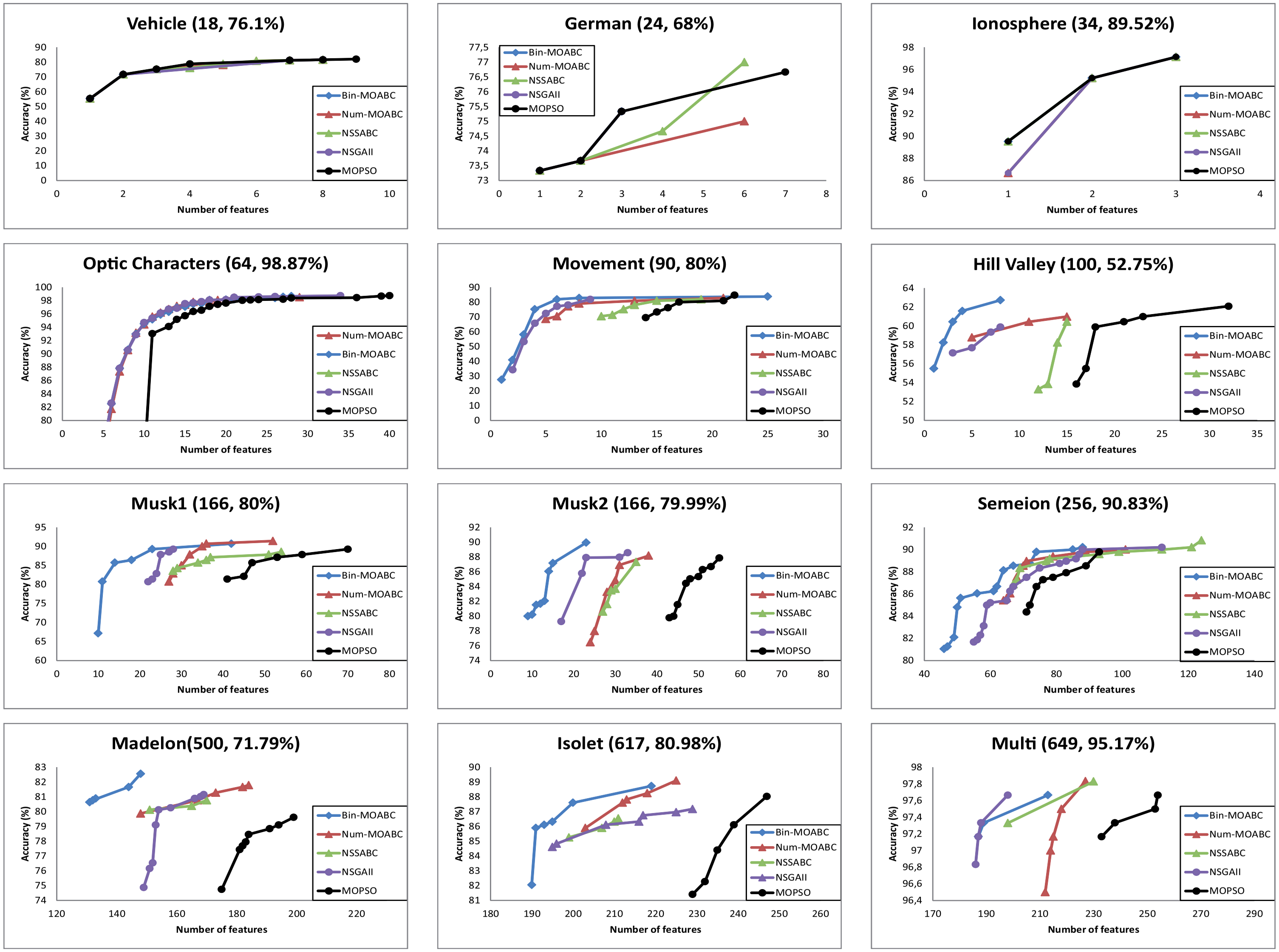

Figure 4: Non-dominated results of multi-objective approaches on the test sets (in color).

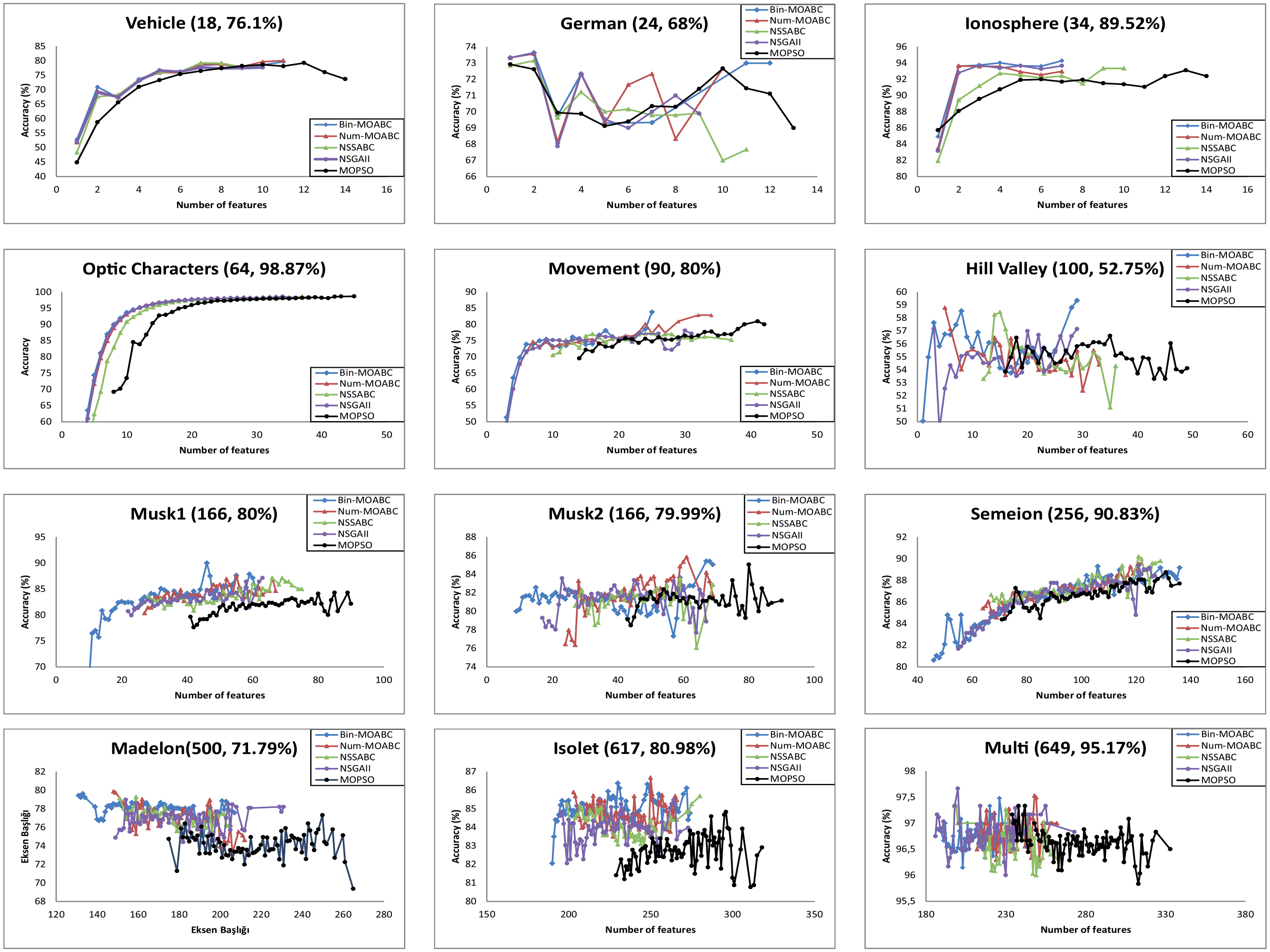

Figure 5: Average Pareto front results of multi-objective approaches on the test sets (in color).

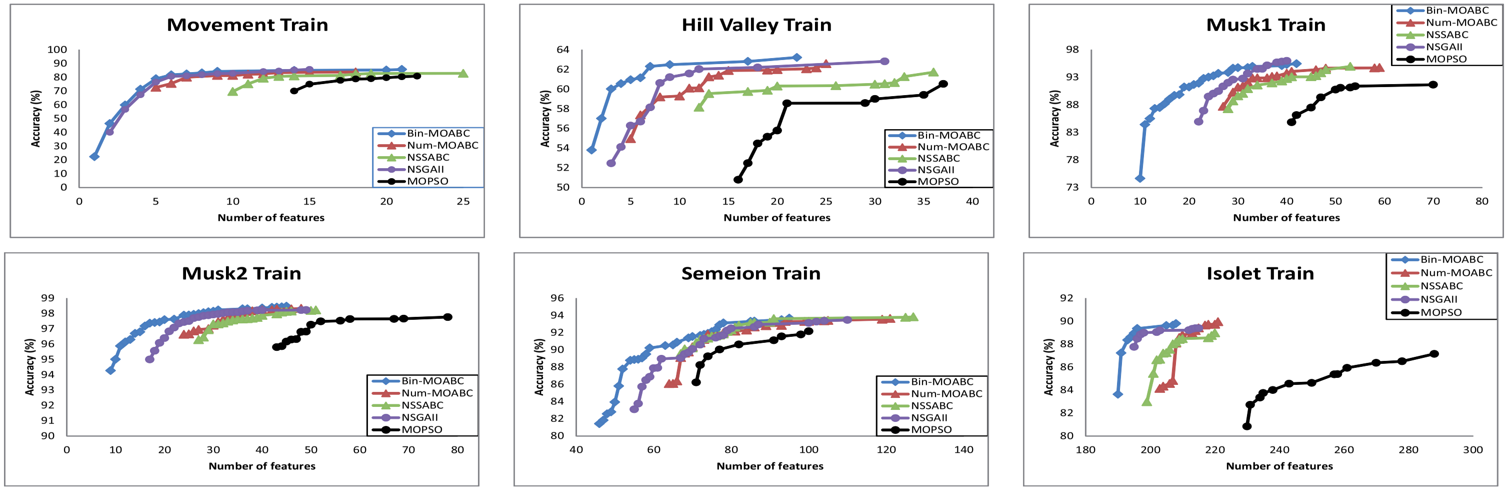

Figure 6: Non-dominated results of multi-objective approaches on the training sets (in color).

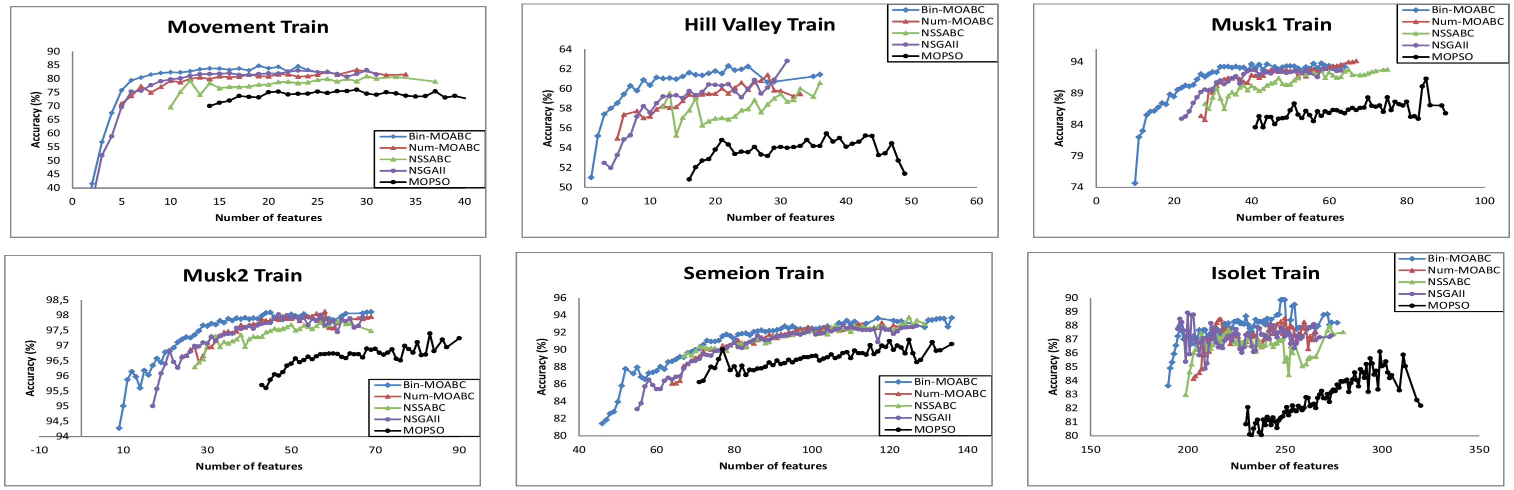

Figure 7: Average Pareto fronts of multi-objective approaches on the training sets (in color).

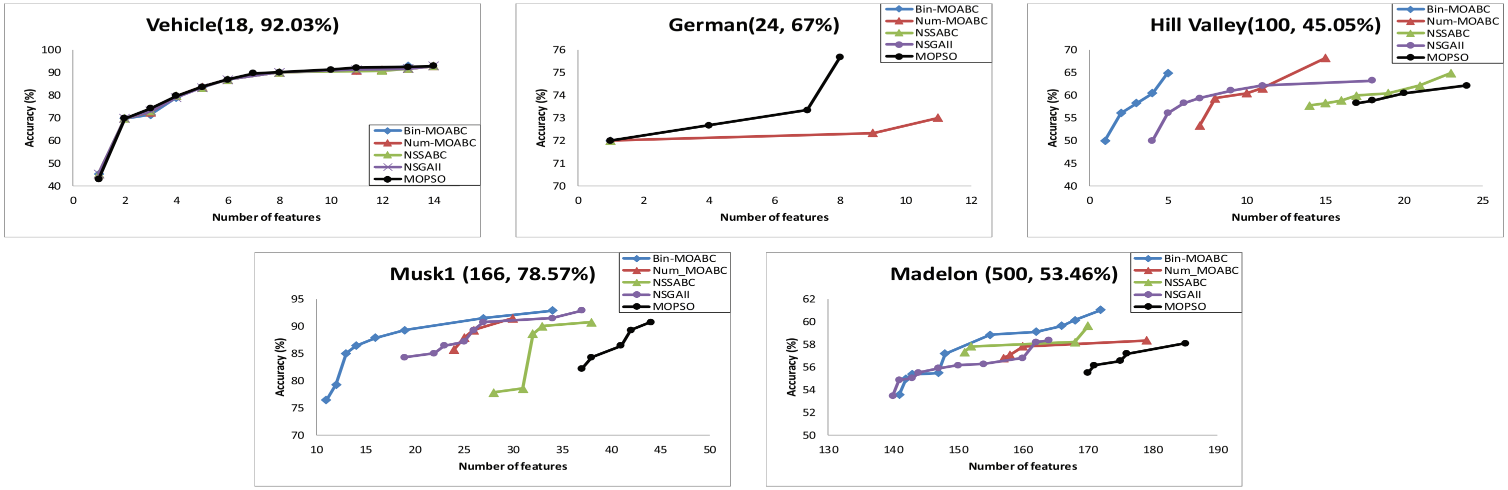

Figure 8: Non-dominated results of multi-objective approaches over quadratic discriminant analysis on the test sets (in color).

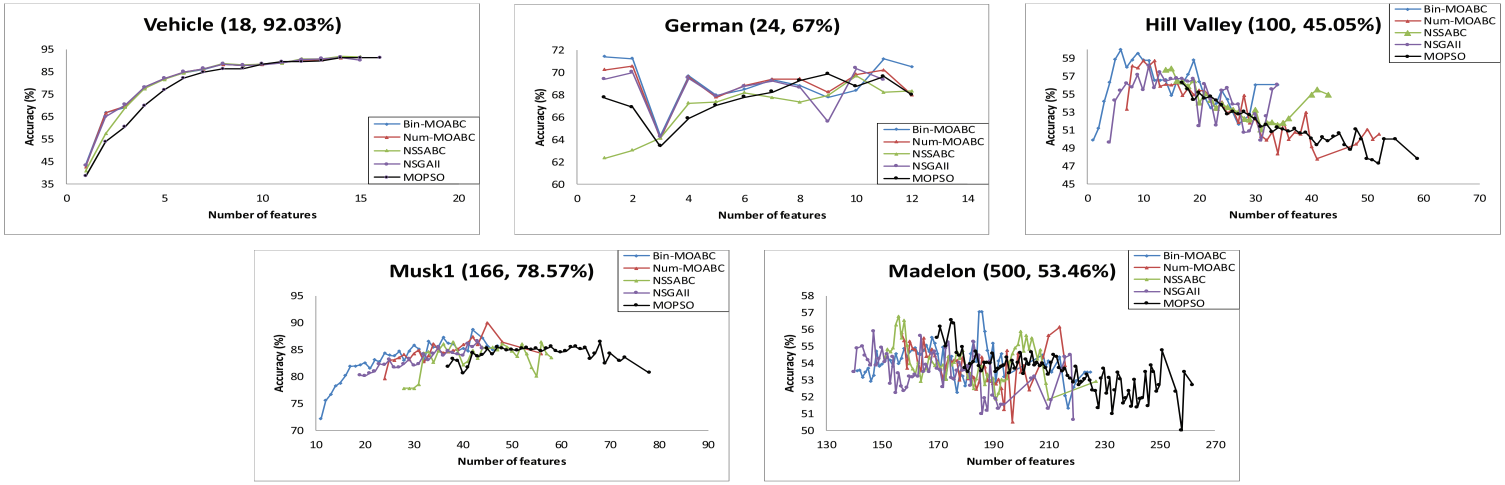

Figure 9: Average Pareto fronts of multi-objective approaches over quadratic discriminant analysis on the test sets (in color).

Table 5: Results of CPU Computational Time (Seconds)

	Bin-MOABC	Num-MOABC	NSSABC	NSGAII	MOPSO
German	118.24	119.89	122.38	131.67	128.36
Vehicle	142.21	141.63	141.85	154.01	149.38
Ionosphere	92.58	93.97	94.40	105.39	99.99
Optic	1353.63	1502.28	1492.78	1686.24	1417.86
Movement	121.06	118.79	109.88	126.67	114.39
Hill Valley	98.58	100.11	99.75	109.90	106.10
Musk	100.77	104.05	104.05	112.63	106.10
Musk2	4232.25	4603.55	4724.53	4257.81	5271.43
Semeion	584.91	642.22	586.88	595.13	657.39
Madelon	2810.69	2876.96	2763.87	2879.01	3128.63
Isolet	1457.02	1482.41	1461.25	1478.76	1599.38
Multiple	2253.72	2381.33	2313.76	2303.15	2538.33

5.5. Computational Time Analysis

The experiments are implemented in MATLAB 2013a and are executed on a computer with an Intel Core i7-4700HQ 2.40 GHz CPU and 8 GB RAM, and the computational time is presented in terms of mean values over the 30 runs in Table 5. According to Table 5, the computational time is increased proportional to the dimensionality and sample size. For example, it takes only a few minutes for the datasets which have a small number of features or samples such as Vehicle, German and Ionosphere. Bin-MOABC is more efficient than the other approaches in terms of the CPU computational time in most cases, i.e, it can complete the training process in shorter time than other approaches.

Considering the other approaches, it is seen that MOPSO consumes more time in high dimensional datasets perhaps due to its external archive mechanism. NSGAII, NSSABC and Num-MOABC perform similar or slightly worse than Bin-MOABC in terms of the CPU time. The reason why Bin-MOABC generally consumes less time than the other approaches may be that Bin-MOABC depends on simple binary crossover and mutation exchanging techniques, i.e., it does not depend on numerical crossover and mutation techniques which requires more calculations. The other reason is that Bin-MOABC tends to choose smaller feature subsets than the other approaches during the training process. Hence, the objective function evaluation overhead is less. Therefore, it can be concluded that not only in the classification rate and feature subset size, but also in the CPU computational time the proposed Bin-MOABC approach performs well.

5.6. Further Discussion

As can be seen from the results, Bin-MOABC outperforms the other approaches in terms of the classification rate, feature subset size and computational time. The factors of Bin-MOABC resulting better performance than the others are as follows. First, searching in binary domain is more suitable than searching in continuous domain for feature selection which is a binary NP-hard problem. However, this may not be individually sufficient to achieve convincing results. In other words, the suitability of search operators on the problem structure is also very crucial to get high classification performance and small feature subset size. For instance, although binary PSO (BPSO) searches in binary domain, it generally cannot achieve better results than standard PSO in feature selection problems [45]. In Bin-MOABC, binary search operators such as two-way mutation and generation strategy are designed for the effective and efficient search in feature selection problems.

Another factor is the positive feedback in the phase of onlooker bees that increases the possibility of selecting high quality food sources for the exploration-exploitation process. Although high quality food sources have more chance than others to be processed, other food sources can also be selected in a probabilistic manner. Accordingly, diversity among sources is tried to be preserved. The last supporting factor is that there exists a balance between exploration and exploitation processes through the 'limit' parameter in ABC. If any food source is exhausted, it is left and a new food source is generated. This property brings innovation and diversity to the population and counterbalance the saturation in the population due to positive feedback.

6. Conclusions

The general goal of this paper was to demonstrate an effective and efficient multi-objective feature selection approach for classification. This goal was fulfilled by introducing two multi-objective ABC frameworks (Bin-MOABC and Num-MOABC). The performance analysis of the proposed algorithms was conducted by making comparisons with the single objective ABC algorithms (ABC-ER and ABC-Fit_{2C}), traditional algorithms (LFS and GSBS) and multi-objective algorithms (NSGAII, NSSABC and MOPSO) on 12 benchmark datasets, most of which are high dimensional. The experimental results show that Bin-MOABC and Num-MOABC outperform ABC-ER, ABC-Fit_{2C}, LFS and GSBS in terms of the classification performance and feature subset size almost in all cases. Therefore, the proposed multi-objective algorithms can be used for feature selection instead of single objective and traditional algorithms. The results also indicate that Bin-MOABC outperformed Num-MOABC, NSGAII, NSSABC and MOPSO in both test set and training set. Furthermore, Bin-MOABC completes the feature selection process more efficiently than the other multi-objective algorithms. The Num-MOABC approach generally cannot obtain as good results as Bin-MOABC and NSGAII, although it employs the similar mechanism with Num-MOABC and NSGAII.

This paper represents an early work on ABC-based multi-objective approach to feature selection. Despite the good performance, there are also some drawbacks with the proposed algorithms, for example it is computationally expensive, and their scalability to datasets with thousands of features is still unknown. In the future, we will carry on developing multi-objective ABC based approaches for feature selection, which can better search the Pareto front of non-dominated solutions in possible solution space. We also would like to investigate multi-objective feature selection methods on large-scale datasets with thousands or even tens of thousands of features, which may requires a very different design of the algorithm.

7. References

[1] R. B. Agrawal, K. Deb, K. Deb, R. B. Agrawal, Simulated binary crossover for continuous search space, Tech. rep. (1994).

[2] B. Akay, Synchronous and asynchronous pareto-based multi-objective artificial bee colony algorithms, Journal of Global Optimization 57 (2) (2013) 415–445.

[3] A. Auger, J. Bader, D. Brockhoff, E. Zitzler, Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point, in: Proceedings of the 10th ACM SIGEVO Workshop on Foundations of Genetic Algorithms, FOGA '09, ACM, 2009.

[4] K. Bache, M. Lichman, UCI machine learning repository (2013).
URL http://archive.ics.uci.edu/ml

[5] R. Battiti, Using mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks 5 (4) (1994) 537–550.

[6] R. Caruana, D. Freitag, Greedy attribute selection, in: Proceedings of the Eleventh International Conference on Machine Learning, Morgan Kaufmann, 1994.

[7] L.-Y. Chuang, S.-W. Tsai, C.-H. Yang, Improved binary particle swarm optimization using catfish effect for feature selection, Expert Systems with Applications 38 (10) (2011) 12699 – 12707.

[8] C. Coello, G. Pulido, M. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Transactions on Evolutionary Computation 8 (3) (2004) 256–279.

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

[10] M. Gutlein, E. Frank, M. Hall, A. Karwath, Large-scale attribute selection using wrappers, in: IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09), 2009.

[11] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I. H. Witten, The weka data mining software: An update, SIGKDD Explor. Newsl. 11 (1) (2009) 10–18.

[12] M. A. Hall, Correlation-based feature selection for machine learning, Ph.D. thesis, The University of Waikato (1999).

[13] T. Hamdani, J.-M. Won, A. Alimi, F. Karray, Multi-objective feature selection with nsga ii, in: Adaptive and Natural Computing Algorithms, vol. 4431 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2007, pp. 240–247.

[14] J. Han, M. Kamber, J. Pei, Data mining: concepts and techniques, Elsevier, 2011.

[15] E. Hancer, B. Xue, D. Karaboga, M. Zhang, A binary ABC algorithm based on advanced similarity scheme for feature selection, Applied Soft Computing 36 (2015) 334 – 348.

[16] E. Hancer, B. Xue, M. Zhang, D. Karaboga, B. Akay, A multi-objective artificial bee colony approach to feature selection using fuzzy mutual

information, in: IEEE Congress on Evolutionary Computation (CEC), 2015.

[17] C.-L. Huang, J.-F. Dun, A distributed pso-svm hybrid system with feature selection and parameter optimization, Applied Soft Computing 8 (4) (2008) 1381 – 1391.

[18] S. M. Kalami, H. Khaloozadeh, Analysis of the optimal treatment methods of aids using non-dominated sorting genetic algorithm II (nsga-II), in: International Conference of Control, Instrumentation and Automation, 2010.

[19] D. Karaboga, B. Gorkemli, C. Ozturk, N. Karaboga, A comprehensive survey: artificial bee colony (ABC) algorithm and applications, Artificial Intelligence Review 42 (1) (2014) 21–57.

[20] K. Kira, L. A. Rendell, A practical approach to feature selection, in: Proceedings of the Ninth International Workshop on Machine Learning, ML92, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

[21] R. Kohavi, G. H. John, Wrappers for feature subset selection, Artificial Intelligence 97 (1997) 273–324.

[22] N. Kwak, C.-H. Choi, Input feature selection for classification problems, IEEE Transactions on Neural Networks 13 (1) (2002) 143–159.

[23] L. D. Landau, E. M. Lifshitz, Statistical Physics. Course of Theoretical Physics 5, 3rd ed., Oxford: Pergamon Press, 1980.

[24] O. Ledoit, M. H. Wolf, Shrunk the sample covariance matrix, The Journal of Portfolio Management 30 (4) (2004) 110–119.

[25] K. Liagkouras, K. Metaxiotis, An elitist polynomial mutation operator for improved performance of moeas in computer networks, in: 22nd International Conference on Computer Communications and Networks (ICCCN'2013), 2013.

[26] H. Liu, L. Yu, Toward integrating feature selection algorithms for classification and clustering, IEEE Transactions on Knowledge and Data Engineering 17 (4) (2005) 491–502.

[27] Y. Liu, G. Wang, H. Chen, H. Dong, X. Zhu, S. Wang, An improved particle swarm optimization for feature selection, Journal of Bionic Engineering 8 (2) (2011) 191–200.

[28] T. Marill, D. Green, On the effectiveness of receptors in recognition systems, IEEE Transactions on Information Theory 9 (1) (2006) 11–17.

[29] I.-S. Oh, J.-S. Lee, B.-R. Moon, Hybrid genetic algorithms for feature selection, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1424–1437.

[30] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1226–1238.

[31] P. Pudil, J. Novovicova, J. Kittler, Floating search methods in feature selection, Pattern Recognition Letters 15 (11) (1994) 1119 – 1125.

[32] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, A. K. Jain, Dimensionality reduction using genetic algorithms, IEEE Transactions on Evolutionary Computation 4 (2) (2000) 164–171.

[33] M. Schiezaro, H. Pedrini, Data feature selection based on artificial bee colony algorithm, EURASIP Journal on Image and Video Processing 2013 (1) (2013) 1–8.

[34] B. Subanya, R. Rajalaxmi, Artificial bee colony based feature selection for effective cardiovascular disease diagnosis, International Journal of Scientific & Engineering Research 5 (5) (2014) 606–612.

[35] B. Tran, B. Xue, M. Zhang, Bare-Bone Particle Swarm Optimisation forSimultaneously Discretising and Selecting Features for High-Dimensional Classification, vol. 9597, chap. 19th European Conference on Applications of Evolutionary Computation, EvoApplications 2016 Part I, 2016, pp. 701–718.

[36] B. Tran, B. Xue, M. Zhang, Genetic programming for feature construction and selection in classification on high-dimensional data, Memetic Computing 8 (1) (2016) 3–15.

[37] A. Unler, A. Murat, R. B. Chinnam, mr2pso: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification, Information Sciences 181 (20) (2011) 4625 – 4641.

[38] M. S. Uzer, Y. Nihat, O. Inan, Feature selection method based on artificial bee colony algorithm and support vector machines for medical datasets classification, The Scientific World Journal 2013 (2013) 1–10.

[39] K. Waqas, R. Baig, S. Ali, Feature subset selection using multi-objective genetic algorithms, in: 13th IEEE International Multitopic Conference (INMIC'2009), 2009.

[40] A. W. Whitney, A direct method of nonparametric measurement selection, IEEE Transactions on Computers C-20 (9) (1971) 1100–1103.

[41] B. Xue, L. Cervante, L. Shang, W. Browne, M. Zhang, A multi-objective particle swarm optimisation for filter-based feature selection in classification problems, Connection Science 24 (2-3) (2012) 91–116.

[42] B. Xue, L. Cervante, L. Shang, W. N. Browne, M. Zhang, Binary PSO and rough set theory for feature selection: A multi-objective filter based approach, International Journal of Computational Intelligence and Applications 13 (02) (2014) 1450009.

[43] B. Xue, M. Zhang, W. Browne, X. Yao, A survey on evolutionary computation approaches to feature selection, IEEE Transactions on Evolutionary Computation 20 (4) (2016) 606–626.

[44] B. Xue, M. Zhang, W. N. Browne, Multi-objective particle swarm optimisation (pso) for feature selection, in: Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation, GECCO, ACM, New York, NY, USA, 2012.

[45] B. Xue, M. Zhang, W. N. Browne, Particle swarm optimization for feature selection in classification: A multi-objective approach, IEEE Transactions on Cybernetics 43 (6) (2013) 1656–1671.

[46] B. Xue, M. Zhang, W. N. Browne, A comprehensive comparison on evolutionary feature selection approaches to classification, International

Journal of Computational Intelligence and Applications 14 (02) (2015) 1550008.

[47] X. Zou, Y. Chen, M. Liu, L. Kang, A new evolutionary algorithm for solving many-objective optimization problems, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 38 (5) (2008) 1402–1412.

[48] X. Zou, M. Liu, L. Kang, J. He, A high performance multi-objective evolutionary algorithm based on the principles of thermodynamics, in: Parallel Problem Solving from Nature - PPSN VIII, vol. 3242 of Lecture Notes in Computer Science, 2004, pp. 922–931.