Multiple Reference Points based Decomposition for Multi-objective Feature Selection in Classification: Static and Dynamic Mechanisms

Bach Hoai Nguyen, *Member, IEEE,* Bing Xue, *Member, IEEE,* Peter Andreae, Hisao Ishibuchi, *Fellow, IEEE*, and Mengjie Zhang, *Fellow, IEEE*

***Abstract*—Feature selection is an important task in machine learning that has two main objectives: reducing dimensionality and improving learning performance. Feature selection can be considered a multi-objective problem. However, it has its problematic characteristics, such as a highly discontinuous Pareto front, imbalance preferences and partially conflicting objectives. These characteristics are not easy for existing evolutionary multi-objective optimization algorithms. We propose a new decomposition approach with two mechanisms (static and dynamic) based on multiple reference points under the MOEA/D (Multi-objective Evolutionary Algorithm based on Decomposition) framework to address the above-mentioned difficulties of feature selection. The static mechanism alleviates the dependence of the decomposition on the Pareto front shape and the effect of the discontinuity. The dynamic one is able to detect regions in which the objectives are mostly conflicting, and allocates more computational resources to the detected regions. In comparison with other evolutionary multi-objective optimization algorithms on 12 different classification datasets, the proposed decomposition approach finds more diverse feature subsets with better performance in terms of hypervolume and inverted generational distance. The dynamic mechanism successfully identifies conflicting regions and further improves the approximation quality for the Pareto fronts.**

***Index Terms*—MOEA/D, Feature Selection, Classification, Multi-objective Optimization, Partially Conflicting**

I. Introduction

Rapid advancements of technologies result in high-dimensional datasets that usually suffer from various issues such as noisy feature values, irrelevant features, and redundant features. Such issues can reduce accuracy and increase training time [1]. Feature selection is one of the most popular ways to improve the quality of a feature set. For a classification problem, feature selection aims to extract a small subset of features with a high discriminating ability. By reducing the dimensionality, feature selection not only improves the classification performance but also yields simpler and more general classifiers [2]. However, feature selection is challenging due to complicated interactions among features and a large search space which increases exponentially with respect to the number of features [3]. Feature selection is considered a multi-objective problem since its two main objectives, reducing the number of features and improving the classification performance, are usually conflicting with each other.

As a family of population-based optimization techniques, evolutionary computation (EC) can be naturally applied to evolve a set of trade-off solutions for multi-objective problems, including feature selection. A number of different evolutionary multi-objective optimization (EMO) methods have been proposed. Some methods evaluate candidate solutions by using a Pareto dominance relation together with a crowding distance to maintain the population's diversity, which are called Pareto dominance-based algorithms. Non-dominated Sorting Genetic Algorithm (NSGA-II) [4], Strength Pareto Evolutionary Algorithm (SPEA2) [5], OMOPSO [6], and Multi-objective Differential Evolution (MODE) [7] are well-known representatives of this type of EMO algorithms. Pareto dominance-based algorithms work well on continuous multi-objective problems having two or three objectives but not on combinatorial problems. For example, it is difficult for them to find non-dominated solutions on the edges of the Pareto front for knapsack problems [8]. A similar issue occurs in feature selection where only a few non-dominated solutions around the center of the Pareto front are obtained [9]. This is probably because the crowding distance has only a small effect in comparison with the Pareto dominance in the case of two or three objectives. Thus the population loses its diversity quickly in an early state of evolution [10].

In contrast to Pareto dominance-based algorithms, decomposition-based EMO has a good search ability for combinatorial multi-objective problems [11]. It works by decomposing a multi-objective problem into a number of single-objective sub-problems and recombines the results. Decomposition-based algorithms often achieve better diversity than Pareto dominance-based algorithms, are easier to integrate with local search mechanisms [10], and may cope better with problems having many objectives [12] or complicated Pareto fronts [13].

MOEA/D (Multi-objective Evolutionary Algorithm based on Decomposition) [14] is a representative of decomposition-based EMO algorithms. Standard MOEA/D decomposes a multi-objective problem to a number of scalar sub-problems using a set of weight vectors. Each weight vector defines a scalar sub-problem whose optimal solution will be a Pareto optimal solution to the original problem. A good set of weight vectors can generate a reasonable approximation of the Pareto front. However, defining an appropriate set of weight vectors is a difficult task in MOEA/D since it depends strongly on the shape of the Pareto front. Ishibuchi et al. [15] showed that if the Pareto front and the hyperplane on which the weight vectors are generated have the same or a similar shape, a set of well-distributed solutions on the Pareto front can be obtained. However, when the weight vectors are not well defined (the

corresponding reference line has no intersection with the Pareto front), the performance deteriorates. Many attempts have been made to adjust weight vectors dynamically during the evolutionary process to cope with different complicated Pareto front shapes [16], [17]. However, most approaches depend on the Pareto front consisting of continuous regions (if not being fully continuous). It is not clear how they can be made to work on problems with discrete Pareto fronts. Furthermore, the use of weight vectors depends on the objectives being in conflict and having roughly equal importance so that the weights can represent the trade-off.

Feature selection has a discrete Pareto front, a strong preference for the classification accuracy over the number of features, and the two objectives are *not always* conflicting with each other. Therefore, it is necessary to design a new decomposition strategy for MOEA/D to achieve feature selection. Our preliminary work [18] proposed a new way of applying MOEA/D to feature selection using a different way of decomposing a multi-objective problem based on reference points rather than weight vectors. The reference points were selected in the space of feature set sizes.

Goal: The overall goal of this paper is to develop a new strategy for MOEA/D to decompose a feature selection problem with an expectation of obtaining a diverse set of non-dominated feature subsets, which achieves better classification performance than using all features. In the proposed decomposition strategy, multiple reference points are used instead of multiple weight vectors. Based on the new decomposition, static and dynamic reference points methods are developed to identify conflicting regions, which are then focused on by allocating more resources to achieve better feature subsets. Both static and dynamic multiple reference points algorithms, called MOEA/D-STAT and MOEA/D-DYN, respectively, are compared with a standard MOEA/D algorithm and four Pareto dominance-based algorithms on 12 real-world datasets and five gene expression datasets of varying difficulties. Specifically, we will investigate:

- whether MOEA/D-STAT and MOEA/D-DYN can evolve feature subsets that achieve better performance than using all features,
- whether decomposition with multiple reference points can help MOEA/D to improve the solution's quality and obtain better approximation of the Pareto front than the multiple weight vectors decomposition,
- whether the new decomposition strategy could generate more diverse feature subsets with various numbers of features than four representatives of Pareto dominance-based algorithms, and
- whether the dynamic strategy can recognize the conflicting regions and further improve the quality of evolved feature subsets.

The main contribution of this work is a new dynamic strategy to allocate reference points that identifies and focuses on conflicting regions. To the best of our knowledge, this is the first study able to automatically analyze partially conflicting relationships between objectives. This is also the main difference between this work and our preliminary work [18]. We also propose a repair mechanism for duplicated solutions, which enhances the population diversity. A deep analysis on different real-world problems is conducted to explain why the proposed decomposition results in better solution sets. We also examine the scalability of the proposed decomposition on gene expression datasets containing thousands of features. The results demonstrate high scalability of the proposed algorithms.

II. Background

A. Multi-objective optimization problems

In a multi-objective problem, two or more conflicting objectives are optimized simultaneously. An o-objective minimization problem can be written as follows.

$$\text{Minimize } \mathbf{f(x)} = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_o(\mathbf{x})) \tag{1}$$

$$\text{subject to } g_i(\mathbf{x}) \leq 0, i = 1, 2, ..., k$$
$$h_i(\mathbf{x}) = 0, i = 1, 2, ..., l$$

where **f(x)** is a vector of objectives, $f_i(\mathbf{x})$ is the i^{th} objective, $\mathbf{x}$ is a decision vector, $g_i(\mathbf{x})$ and $h_i(\mathbf{x})$ are the constraint functions of the problem.

The quality of a solution is based on the trade-off between the objectives. A solution $\mathbf{y}$ is better than a solution $\mathbf{z}$ if:

$$\forall i\text{: } f_i(\mathbf{y}) \leq f_i(\mathbf{z}) \text{ and } \exists j\text{: } f_j(\mathbf{y}) < f_j(\mathbf{z}) \tag{2}$$

It can be said that $\mathbf{y}$ dominates $\mathbf{z}$ (assuming the smaller the better). If a solution is not dominated by any other feasible solutions, the solution is called a Pareto optimal solution. The set of all Pareto optimal solutions forms a trade-off surface in the objective space, which is called the Pareto front. The task of an EMO algorithm is to evolve a set of well-distributed non-dominated solutions, which is a good approximation of the Pareto front. Feature selection can be considered a two-objective minimization problem, in which the number of features and the classification error rate need to be minimized.

B. MOEA/D

MOEA/D [14] is an EMO framework that treats a multi-objective problem as a set of sub-problems. In the *standard* MOEA/D framework, each sub-problem is a single-objective problem and has a corresponding weight vector $\mathbf{w}$, which is used to define its own fitness function. Each sub-problem has a candidate solution to find an optimal solution, so the number of decomposed single-objective sub-problems is equal to the population size. The sub-problems can be multi-objective problems but simpler than the original problem, which is the main idea of MOEA/D-M2M [19]. In MOEA/D-M2M, a set of vectors is used to divide the original objective space into a number of smaller sub-objective spaces. A part of the Pareto front in each sub-space is expected to be easier to approximate.

Neighborhood is an essential property of MOEA/D. Each sub-problem has T sub-problems as its neighbors. The distance between the weight vectors defines the neighborhood relation. It is expected that the solutions of neighboring sub-problems should be similar so that each sub-problem can improve its solution by using information from its neighbors.

In the standard MOEA/D framework, given an o-objectives problem, each weight vectors has o elements, $\mathbf{w} = (w_1, w_2, ..., w_o)$, which satisfies the following conditions:

$$\sum_{i=1}^{o} w_i = 1 \text{ and } w_i \in \left\{0, \frac{1}{N}, \frac{2}{N}, ..., \frac{N}{N}\right\} \quad (3)$$

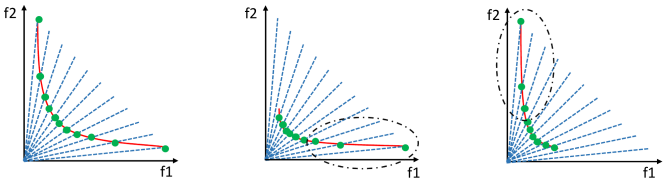
where N is a predefined positive integer. Following this method, the number of weight vectors is C_{N+o-1}^{o-1}, which is a large number when o, the number of objectives, is large [20]. One can apply other strategies to generate weight vectors. For two-objective problems such as feature selection, N is the number of sub-problems, which is also the population size [14]. There are various ways to aggregate multiple objectives into a single scalar function, in which Weighted Sum, Tchebycheff and Penalty-based Boundary Intersection (PBI) are three common approaches [21]. In Tchebycheff and PBI, a reference point is needed to define a reference line for a sub-problem.

C. Related work on EMO for feature selection

EC has been widely applied to multi-objective feature selection. Mukhopadhay et al. [22] utilized NSGA-II with a SVM classification algorithm to identify miRNA markers. Both feature subsets and SVM's parameters were encoded which identified miRNA related to cancers. Leandro et al. [23] applied a multi-objective GA (MOGA) to perform feature selection for face recognition. There were three objectives, which were the aggregation of the classification accuracy and the feature subset size, the number of selected coefficients, and the mutual information between selected features. The solutions found by MOGA selected fewer features and achieved similar accuracies to those found by single-objective GAs. Among EC techniques, GA-based multi-objective algorithms are the most popular. However, those studies simply applied GAs without considering the characteristics of feature selection [3].

PSO is also widely applied to feature selection. Xue et al. [24] proposed the first multi-objective PSO (MOPSO) algorithm for feature selection, which was superior to NSGA-II, SPEA2 and PAES2 on feature selection. Later, Nguyen et al. [25] improved the archive's solutions in MOPSO by applying three local search operators: Inserting, Removing and Swapping. The algorithm selected a smaller number of features and achieved similar or better classification performance than CMDPSOFS [24]. Recently, a multi-objective Differential Evolution (MODE) based feature selection algorithm was developed by Xue et al. [26]. During the evolutionary process, if the population size exceeded its limit, solutions with lower dominance levels were removed.

Most of the current multi-objective feature selection studies used Pareto dominance-based algorithms, which usually focused on the center of the Pareto front. The MOEA/D framework can address this problem. To the best of our knowledge, Paul et al. [27] proposed the first filter MOEA/D based feature selection algorithm, which considered inter-class and intra-class distance measures as two conflicting objectives. However, the fronts evolved by MOEA/D and NSGA-II were not compared using any performance indicator. In addition, MOEA/D was applied directly to feature selection without considering characteristics of feature selection.



(a) Balance (b) Towards f_1 (c) Towards f_2

Fig. 1: Effect of bias Pareto fronts on MOEA/D.

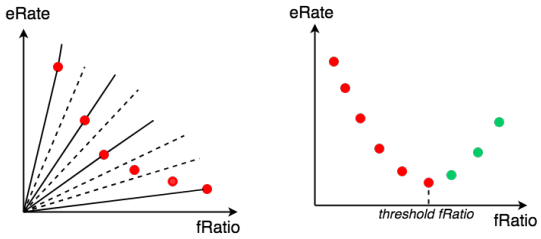
The first characteristic of feature selection is its unknown Pareto front. Therefore, defining weight vectors in feature selection is a challenging task. A simple example is illustrated in Fig. 1, where the task is to minimize both objectives, f_1 and f_2, and the green dots show the best solutions for the weight vectors. When the Pareto front is biased towards f_1 (Fig. 1(b)) or f_2 (Fig. 1(c)), using a set of evenly distributed weight vectors does not work well. In both cases, a number of weight vectors are wasted since they do not contribute any solution. More solutions on the edge of the Pareto front can be obtained if the wasted weight vectors are located near the the edge. Several works attempted to update weight vectors based on the densities of regions to preserve the population diversity [28], [29]. However, they require additional computational cost to adaptively adjust the weight vector set. Although they are tested on numeric problems having irregular Pareto fronts, none of the problems has a front as highly discontinuous as feature selection. Instead of adaptively adjusting weight vectors, this work develops a new decomposition mechanism for feature selection, which reduces the dependency on the Pareto front shape and copes well with the front's discontinuity.

Another characteristic of feature selection is the complicated relationship between its two objectives. Firstly, the objective of reducing the classification error has higher priority than reducing the number of selected features. Secondly, the two objectives are not always in conflict. Therefore, some parts of a Pareto front on the conflicting regions are more difficult to approximate than other parts on the non-conflicting regions. In standard MOEA/D, all sub-problems are treated equally, and they usually receive the same amount of computational resource [21]. However, it is shown that some parts of a Pareto front can be more challenging to approximate than others [30]. It is natural to allocate resources differently to different sub-problems (weight vectors) with respect to their difficulties, which results in better efficiency [31]. A similar question appears in feature selection. Possibly, better Pareto front approximations can be achieved by putting more efforts on the conflicting regions rather than evenly spending resources on both conflicting and non-conflicting regions.

This paper addresses the above limitations by developing a decomposition mechanism that helps MOEA/D to cope with the characteristics of feature selection. It is expected that the proposed algorithm results in a diverse set of non-dominated feature subsets with better classification performance.

III. Proposed Algorithms

This section starts by listing characteristics of feature selection that illustrate difficulties when applying MOEA/D to



(a) Weight vector for FS (b) Not always conflicting

Fig. 2: Characteristics of Feature Selection.

feature selection. It then shows how to use multiple reference points to decompose a feature selection problem.

A. Characteristics of feature selection

The task of feature selection is to reduce the classification error rate ($eRate$) while selecting a small portion of the original feature set ($fRatio$). *eRate* measures the ratio between the number of wrongly classified instances and the total number of instances (m). *eRate* is discrete and the interval between the adjacent values (i.e. the granularity) is $1/m$. Similarly, *fRatio* is the ratio between the number of selected features and the total number of original features n. *fRatio* is also discrete and the interval between its adjacent values is $1/n$. Therefore, the Pareto front of feature selection is discrete. If weight vectors are used to decompose feature selection, these vectors have to be carefully selected, otherwise, there will be vectors which do not correspond to any solution on the Pareto front as shown by the dashed line in Fig. 2(a). Furthermore, although both objectives are in the same range [0,1], they typically have different granularity due to the difference between $1/m$ and $1/n$. It has been shown that solving multi-objective problems where the objectives have different granularity usually results in imbalanced Pareto fronts [32].

The relationship between the two objectives in feature selection makes it an unusually challenging multi-objective problem. In feature selection, the classification performance is usually given a higher priority. For example, if a feature set selects 10% more features than the other feature set but achieves 10% better accuracy, the first set is definitely preferred. Furthermore, the two objectives are not always in conflict. Removing irrelevant or redundant features may improve the classification performance, which means that the two objectives are not conflicting in some regions. However, if all features in a feature set are relevant and complementary, removing any feature degrades the classification performance. Thus, after removing all irrelevant/redundant features, the two objectives become mostly conflicting. In other words, there might be a *threshold feature ratio* beyond which the two objectives are *mostly* harmonious. Fig. 2(b) illustrates the situation, in which each point is the best solution with the corresponding feature ratio. As can be seen, only red points can form the Pareto front while all green points are dominated by the solution at the *threshold feature ratio*. It will be more effective for a multi-objective algorithm to allocate more computational efforts on regions with *fRatio* belows the *threshold*. However, the *threshold* is problem dependent and not easy to identify. In the following subsections, both static and dynamic multiple reference points strategies are introduced to address the above characteristics of feature selection.

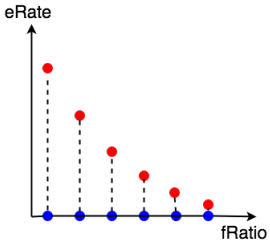


Fig. 3: Multiple reference points in MOEA/D.

B. Decomposition with multiple reference points

In standard MOEA/D, the effectiveness of the weight vector set depends on the shape of the Pareto front which is unknown in feature selection. To alleviate the effect of the Pareto front shape, we use multiple reference points to decompose a multi-objective feature selection problem instead of using multiple weight vectors. Specifically, we allocate a set of *R* reference points on the $fRatio$ axis. A reference point placed at position *refRatio* on the *fRatio* axis represents an idealized solution with an accuracy of 100% (i.e. 0% *eRate*) using exactly ($\lfloor refRatio * n \rfloor$) features where *n* is the total number of the original features. In the MOEA/D search, there will be one individual in the population for each reference point, just as there is one individual for each weight vector when the problem is decomposed using weight vectors. Fig. 3 shows a set of reference points marked by blue dots. Using multiple reference points, the multi-objective feature selection problem is decomposed into a sub-problem for each reference point. The solution of a sub-problem for a reference point at *refRatio* is the feature subset, whose size is at most ($n_{ref} = \lfloor refRatio * n \rfloor$). Such a feature subset will be on the Pareto front. The search space of each sub-problem is smaller than the original one since it is limited by n_{ref}.

In the proposed decomposition, the search space of a sub-problem (S_1) with a smaller n_{ref} (e.g., n_1) is covered by the search space of a sub-problem (S_2) with a larger n_{ref} (e.g., n_2). Therefore, the two sub-problems, S_1 and S_2, may have the same solution. However, this is not a problem but actually beneficial. On the other hand, one could restrict S_2 to consider feature subsets whose sizes are in the range (n_1,n_2]. This would separate the search spaces of S_1 and S_2, so they could not have the same solution. However, with the separated search spaces, S_2 might not contribute any solution to the approximated Pareto front, which can affect the front's diversity [15]. More importantly, the separated search spaces limit the assistance between different sub-problems, which is an essential property of MOEA/D. Our decomposition does not ensure that two sub-problems have distinct solutions, but it satisfies the two above-mentioned desirable properties.

The fitness function of a candidate feature subset S to a sub-problem is designed as follows.

$$fitness_S = eRate_S + 100 * max(|S| - n_{ref}, 0) + \alpha * fRatio_S \quad (4)$$

where $|S|$ is the number of selected features. The main task of the sub-problem is to minimize the classification error $eRate_S$,

which is the first objective of multi-objective feature selection and represented by the first component. The second component is a penalty factor to ensure the condition that the number of selected features in S should not exceed n_{ref}. The last component is related to the objective of reducing the number of selected features. The coefficient α is used to control the priority of the second objective in comparison with the first objective. A large value of α increases the chance of selecting a solution with a smaller number of features but a lower classification accuracy. If α is set to 1, the two objectives have the same priority. Since reducing the classification error is the more important objective, α is usually smaller than 1.

A decomposition using weight vectors in a highly discontinuous space may lead to a sub-problem with no solution, and therefore may result in a very poor approximation of the Pareto front. In contrast, a decomposition using reference points leads to sub-problems that always have a solution from the Pareto front, and therefore should always give a good approximation of the true Pareto front. Because of the choice of the fitness function, this decomposition also handles the strong preference for classification performance in feature selection.

The idea of using multiple reference points in MOEA/D has already been examined in some studies [33], [34]. However, those algorithms update the reference points every generation according to specific mechanisms. Our approach places the reference points on the $fRatio$ axis prior to the evolutionary process. Moreover, there is no weight vector in the proposed algorithm. These two differences make our algorithm simpler than other multiple reference points EMO algorithms.

C. Reference points allocation

The previous subsection shows how multiple reference points can be used to effectively decompose feature selection despite its discrete Pareto front. This subsection describes how the reference points are allocated on the $fRatio$ axis. One way is to fix locations of the reference points at the beginning, which is called $static$ allocation. A more advanced strategy is to *dynamically* modify the locations, which is capable to detect conflicting/non-conflicting regions.

1) Static allocation: In static allocation, the reference points are uniformly placed on the $fRatio$ axis and do not change during the search. Specifically, given R reference points, the position of the i^{th} reference point is $(i/R, 0)$. Notice that there is no reference point at the location $(0, 0)$ since it defines an empty feature subset. For each sub-problem, its neighbors are sub-problems whose reference points are close to this sub-problem's reference point. For example, when the number of neighbors is 3, the neighborhood of $(3/R, 0)$ includes $(2/R, 0)$, $(3/R, 0)$ and $(4/R, 0)$. In general, we expect that the solutions of neighboring sub-problems will be similar, which is an important requirement of MOEA/D.

2) Dynamic allocation: In feature selection, the two objectives are not always in conflict. In a non-conflicting region, there can be at most one solution from the Pareto front. Evenly distributing all reference points on the entire domain of the $fRatio$ axis might limit the performance of MOEA/D since some reference points are wasted in non-conflicting regions. We propose a dynamic mechanism that firstly identifies the conflicting and non-conflicting regions, and then allocates more reference points to the conflicting regions.

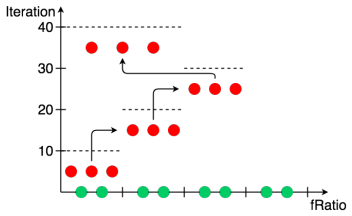


Fig. 4: Dynamic reference points example: $fixed$ points are green, $moving$ points are red, dashed line shows the interval that $moving$ points are located in the corresponding iterations.

To achieve the above aim, the $fRatio$ axis is divided into I intervals all of the same length, $1/I$. We assume that there will be one interval containing the *threshold feature ratio*, beyond which the two objectives are *mostly* not conflicting (Fig. 2b). The R reference points are divided into F $fixed$ points and M $moving$ points ($R = F + M$). The F fixed points are evenly located across the I intervals, shown by the green points on Fig. 4. At the beginning, the M moving points are all located on the first interval, and the locating mechanism spreads the moving points while avoiding overlapping between the two types of reference points as much as possible. After a certain number of iterations defined by the division between the maximum number of iterations and the number of intervals, the moving points are re-allocated on the next interval. For example, in Fig. 4, in the first 10 iterations, the three moving points are located on the first interval. In the next 10 iterations, the moving points are re-allocated to the second interval and so on. The $10^{th}, 20^{th}$... iterations are called *boundary iterations*, since on these iterations the moving points are re-allocated.

The re-allocation process is continued until the algorithm detects that the two objectives are potentially not conflicting any more. As can be seen in Fig. 2(b), most solutions in the potentially non-conflicting region (green) are dominated by a solution in the conflicting region (red). Therefore, to determine whether the two objectives are still conflicting in the i^{th} interval, the solution with the lowest classification error in the interval is compared with all solutions from the previous interval. If the solution from the i^{th} interval is dominated by a solution in the previous interval, the algorithm assumes the two objectives are not conflicting in any interval from the i^{th} one. The moving points are then evenly allocated on all the intervals prior to the i^{th} one and their locations are not changed until the evolutionary process is finished. An example is given in Fig. 4, where after allocating moving points on the third interval, the algorithm finds that the solution with the best accuracy obtained by reference points in the third interval is dominated by one of solutions from the second interval. This is an indication that in the regions from the third interval, the two objectives may not conflict. Thus the algorithm allocates all moving points on the first and second intervals.

In the evolutionary process, the moving points are re-allocated many times. However, Giagkiozis et al. [35] showed that $dynamic$ mechanisms are not always good since they may cause divergence in the population. To avoid the divergence but still preserve the population's diversity, the re-allocation

Algorithm 1 : Reduce size of an infeasible feature subset S

Input: Ranking of features *selected* in S

1: **while** $|S| > n_{ref}$ **do**
2: remove the selected feature with the lowest accuracy from S
3: **end while**

Algorithm 2 : Increase size of a feature subset S

Input: Ranking of features *unselected* in S

1: **while** $|S| < n_{ref}$ **do**
2: add the unselected feature with the highest accuracy to S
3: **end while**

process has to be done carefully. Firstly, the moving points are re-allocated so that there will be the least overlap with the fixed points because a diverse allocation usually leads to diverse solutions on the Pareto front. Secondly, when reallocating a reference point to a new value on the $fRatio$ axis, the algorithm attempts to preserve as much information from the solution found for the sub-problem at the previous location of the reference point. Therefore, the algorithm initializes the reference point with a feature subset as close as possible to the feature subset from the previous solution. Since the new location requires a different number of features, the feature subset from the previous solution must be "repaired", which will be discussed in the following subsection.

It should be noted that the dynamic mechanism does not ensure that the *threshold* interval is found exactly. It just needs to estimate possible regions in which the two objectives are mostly conflicting and puts more effort (reference points) on these regions. There are still some fixed reference points locating in the other regions (i.e., possible non-conflicting regions) just in case the estimation is not good enough. In addition, these fixed points on non-conflicting regions usually have large n_{ref} values, which may allow different features to be introduced into solutions for neighboring reference points with smaller n_{ref} values. This helps to prevent premature convergence of the sub-problems in the conflicting regions.

D. Repair mechanism

All evolutionary algorithms create new candidate solutions from current solutions. If a number of generated candidates are infeasible, the search mechanism may waste a lot of search time on exploring useless parts of the search space. One option is to identify and remove all invalid candidates, but this may lose valuable information contained in the candidates. An alternative option is to "repair" an invalid candidate by transforming it into a close valid solution, which has the advantage of retaining information in the candidate, but may be expensive if the repair mechanism is not efficient.

For sub-problems in the proposed MOEA/D based feature selection algorithm, repair is particularly important because each sub-problem corresponds to a small part of the search space - the sub-space of features subsets whose size is close to but not more than the n_{ref} of the reference point - and it is difficult to ensure that new candidates are always within the subspace. When the search mechanism creates a candidate feature set S that is larger than n_{ref}, the repair mechanism must remove ($|S|-n_{ref}$) features in order to make it valid. The mechanism chooses the ($|S| - n_{ref}$) features with the lowest individual classification accuracies (which are pre-calculated at the start of the algorithm). This process is shown in Algorithm 1. A potential problem is that it may remove important features that are strongly complementary to other features, even though they are individually weak. However, this information about complementary features is usually retained in the neighboring sub-problems with larger n_{ref} values. The search mechanism is able to re-select the removed features using information from the neighboring subproblems.

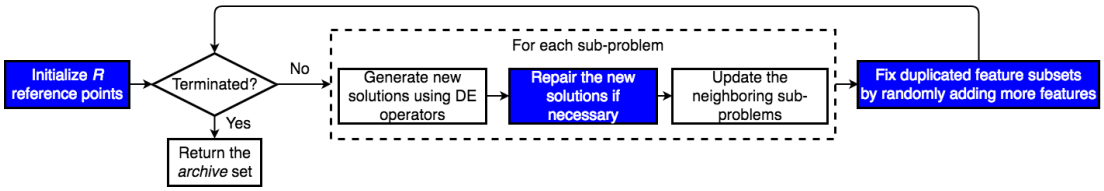
Re-allocating reference points in the dynamic mechanism is even more prone to creating invalid candidates since re-allocating a reference point means changing its n_{ref} value. If a reference point is re-allocated to a smaller n_{ref} value, its current candidate feature subset is likely to be too large for the new n_{ref} value, and features will be removed by the same mechanism described above. If a reference point is re-allocated to a larger n_{ref}, its candidate feature subset will still be valid, but may be much smaller than the new n_{ref} value, which is highly problematic because it is likely to be similar to the candidates of subproblems with smaller reference points and therefore will reduce the population diversity and limit the ability to explore new feature combinations. The repair process ranks all unselected features based on their individual accuracies and sequentially add them to the candidate feature subset until its size reaches n_{ref}. The re-allocating process may require to decrease or increase the size of a feature subset, which are shown in Algorithms 1 and 2, respectively.

E. Fixing duplicated feature subsets

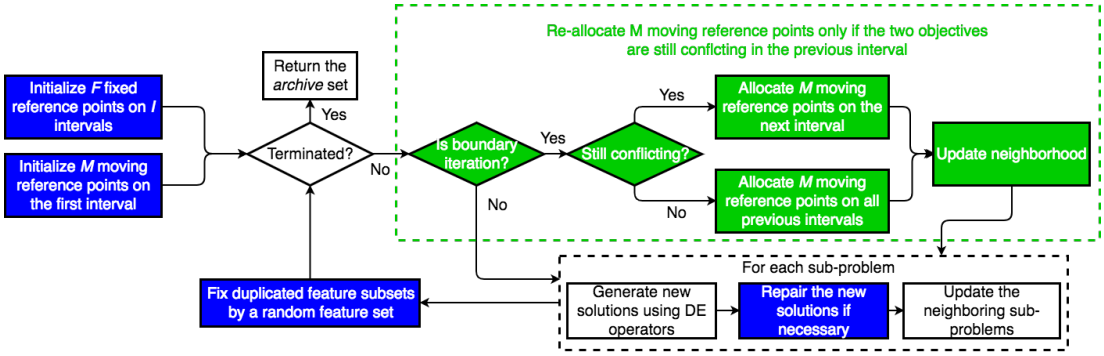
One problem of the proposed decomposition approach is that a feature subset for a reference point with a smaller n_{ref} value can also become the solutions for reference points with larger n_{ref} values. The duplicated feature subsets might cause a low diversity and premature convergence. To avoid such an undesirable situation, all duplicated sets from the larger reference points are repaired. Since the reference points density on an interval is not high in the static strategy, randomly adding unselected features to the duplicated subsets should be sufficient. However, due to the dynamic allocation of moving reference points, the reference points on a particular interval is more dense. Duplicated feature subsets in the dynamic strategy are replaced by randomly generated feature subsets.

F. Overall proposed algorithms

Figs. 5(a) and 5(b) show an overview of the static and dynamic multiple reference points algorithms. In Fig. 5(a), the blue parts are the essential differences in comparison with the standard MOEA/D algorithm. The difference between the dynamic mechanism and the static one is the moving reference points re-allocation, marked by the green color in Fig. 5(b). Note that the re-allocation is only performed when the algorithm has not identified the *threshold* interval yet. Once the *threshold* interval is found, the M moving reference points are allocated to the conflicting intervals and no further re-allocation is needed. Both algorithms use the same differential



(a) Static multiple reference points strategy (MOEA/D-STAT).



(b) Dynamic multiple reference points strategy (MOEA/D-DYN).

Fig. 5: Overall multiple reference points MOEA/D algorithms.

evolution (DE) crossover and mutation operators, which is an efficient approach to preserve the population diversity [14].

Algorithm 3 : Pesudo-code of MOEA/D-STAT

begin
for each feature, calculate its individual accuracy;
initialize R reference points: $refPoint_i = (i * \frac{1}{R}, 0)$ where $i = 1, ..., R$;
find the set of T neighboring reference points of each reference point;
each i^{th} sub-problem's neighboring set is denoted B_i;
randomly initialize the population $P = (p_1, p_2, ..., p_R)$ where p_i is the candidate solution of the i^{th} sub-problem;
while maximum iteration is not reached **do**
  for i=1,...,R **do**
    $Ne = \begin{cases} B(i) \text{ if } rand < \sigma \\ P \text{ otherwise} \end{cases}$
    randomly select two solutions from Ne to generate a new solution y by using DE crossover and mutation operators;
    repair y if it selects more than n_{ref} features;
    update solutions of neighboring sub-problems if y is better than the solutions of sub-problems in terms of the fitness values calculated by Eq. (4);
  end for
  repair duplicated feature subsets;
  update the archive set;
end while
Output the archive set;
end

The pseudo-code of the static multiple reference points MOEA/D for feature selection (MOEA/D-STAT) is shown in Algorithm 3. Each individual is represented by a vector of real numbers. The vector length is equal to the total number of original features. Each entry corresponds to an original feature and its value determines whether or not the corresponding feature is selected. Specifically, the feature is chosen if the entry's value is greater than a threshold θ. σ is the probability that a sub-problem selects its T neighboring sub-problems to create a new solution. The Tchebycheff approach [14] is used as a representative of standard MOEA/D to compare with the proposed decomposition since it usually achieves better results than the Weighted Sum approach [14] and it does not need to specify a penalty factor like the PBI approach. In addition, the Tchebycheff approach has good theoretical properties [36].

IV. Experiment Design

A. Benchmark techniques

The proposed algorithms, MOEA/D-STAT and MOEA/D-DYN, are compared with five well-known multi-objective algorithms: standard MOEA/D, NSGA-II [4], SPEA2 [5], OMOPSO [6], and MODE [7]. We also compare the proposed algorithms with four classical feature selection algorithms: mRMR (information-based) [37], reliefF (similarity-based) [38], CFS (correlation-based) [39], and RFS (sparse learning-based) [40]. The algorithms are examined on 12 UCI datasets [41] from different real-world areas such as physic/chemistry (Wine, Sonar, Musk1), finance (Australian, German), image analysis (Vehicle), health (WBCD, Arrhythmia), speech recognition (Isolet5), handwritten recognition (Multiple Features). The selected datasets also have different numbers of features (from 13 to 649), classes (from 2 to 16), and instances (from 178 to 7797) with an expectation that they are representative samples of the problems that the proposed algorithms can address. The dataset details can be seen in Table I.

Each algorithm is run 50 independent times. Each dataset is divided into training and test sets with the proportions of 70% and 30%, respectively. During the training process, KNN with 10-fold cross-validation is applied to calculate the classification error rate on the training set to avoid feature

TABLE I: Datasets.

Dataset	#Features	#Classes	#Instances
Wine	13	3	178
Australian	14	2	6650
Vehicle	18	4	946
German	24	2	1000
WBCD	30	2	569
Sonar	60	2	208
Hillvalley	100	2	606
Musk1	166	2	476
Arrhythmia	279	16	452
Madelon	500	11	4400
Isolet5	617	5	7797
MultipleFeatures	649	15	2000

selection bias where the selected features overfit the training data. The evolved feature subsets are then evaluated on the test set to obtain their testing accuracies. These settings are commonly used in feature selection [24], [18].

To examine the performance of the six multi-objective algorithms, the hypervolume indicator [42] and the inverted generational distance (IGD) indicator [43] are used. In each run, an algorithm obtains two Pareto front approximations, which are "training Pareto front" and "testing Pareto front". After 50 executions, each algorithm has two sets of metric values based on the training and test sets, respectively. To calculate the two indicators, it is necessary to know the true Pareto front, but it is not known in feature selection. Therefore, the true Pareto front is approximated by the non-dominated solutions obtained from the union of all solutions generated by 50 independent runs of the six algorithms. Here, the hypervolume value of a front is calculated by its inverted front, which is implemented in the JMetal package [44]. Therefore, the larger the hypervolume value, the better the algorithm. A significance test, Wilcoxon test with its significance level set to 0.05, is used to compare the performance between MOEA/D-STAT, MOEA/D-DYN and the benchmark algorithms.

For each algorithm the attainment surface (i.e., non-dominated surface) corresponds to the median hypervolume value is obtained, which is called a $median$ front. Note that although the $median$ front can give a good visualization, the indicator values are more reliable to compare different algorithms. The reason is that the indicator values are calculated based on all the solution sets generated by 50 independent runs of each algorithm whereas the median fronts in figures only show the median non-dominated solution set obtained from a single run. We visualize the median fronts to provide a visual intuition about the search performance of each algorithm.

B. Parameter settings

In general, choosing a proper parameter setting for an EMO algorithm is a difficult task since it is problem-dependent. On the basis of several trial experiments on the Musk1 dataset using various parameter choices as suggested in [14], the parameter values for the proposed algorithms are set as follows. The number of neighbors T is set to $R/10$, which is much smaller than the population size to preserve diversity. However, the smallest value of T is 4 to ensure the diversity between neighboring sub-problems. The maximum number of solutions replaced by a newly generated solution is set to 1 which is much smaller than T [14]. The proposed algorithms use the DE crossover (crossover rate is 0.6, scaling factor F is 0.7) and polynomial mutation (mutation rate is $1/n$). The probability of selecting parents from the neighboring sub-problems, σ, is 0.85. α in Eq. (4) is set to 0.01, which shows a very weak preference for a smaller feature subset among the feature subsets with the same classification error and different numbers of features. More analysis of α can be seen in Section V.F. The settings of NSGA-II, SPEA2 and OMOPSO follow the recommended setting from their original papers, which are default settings in the JMetal package [44].

The dynamic strategy has two main parameters: M - the number of moving reference points, I - the number of intervals. Based on experiments, M is set to $0.4 * R$, which ensures a significant effect of moving reference points while maintaining enough fixed reference points to explore all intervals. Different values of M are examined and the results show that the proposed algorithm is not sensitive to M. More details can be seen in the supplementary material[1]. When the number of features is less than 20, the number of intervals is set to 9. Otherwise, the number of intervals is set to 4. On datasets having less than 20 features, since the search space corresponding to each interval is not large, it is fine to have 9 intervals, which ensures a fine-grained intervals leading to a more accurate estimation of the non-conflicting region. However, on datasets with large numbers of features, the search space corresponding to each interval is much larger, which requires more efforts (the number of iterations and reference points) to be well explored. Given a smaller number of intervals, the moving points can explore an interval in a larger number of iterations. Therefore, the number of intervals on large datasets is set to 4, a small value.

The number of nearest neighbors in KNN is set to 5 to avoid noisy instances while still maintaining its efficiency. The maximum number of iterations are 200. The population size is set to the number of features due to the exponential increase of the search space size with respect to the number of features. However, the population size is bounded by 200 to avoid high computational costs. The threshold θ is set to 0.6 so that the algorithms start with slightly small numbers of features.

V. RESULTS

The average IGD and hypervolume values of the six algorithms on the training and test sets are shown in Tables II-V. The two signs besides the average values of the five benchmark algorithms show significance test results compared with the two proposed algorithms, MOEA/D-STAT and MOEA/D-DYN, respectively. "↑", "↓", "∘" mean that the corresponding benchmark algorithm is significantly better than, worse than or has no significant difference from MOEA/D-STAT (MOEA/D-DYN), respectively. The single sign in MOEA/D-STAT's column shows the comparison result with MOEA/D-DYN.

The $median$ fronts of the algorithms on the test sets are shown in Fig. 7. In each sub-figure, the two numbers inside the brackets show the number of the original features and the training or testing error when using all features. The horizontal and vertical axes represent $fRatio$ and $eRate$, respectively.

[1]Online Supplementary Material: https://ecs.victoria.ac.nz/foswiki/pub/Groups/ECRG/OnlineSupplimentaryMaterials/MOEAD_FS.pdf

TABLE II: IGD on training sets.

Dataset	NSGA-II	MODE	SPEA2	OMOPSO	MOEAD	MOEA/D-STAT	MOEA/D-DYN
Wine	0.049±0.010 (↓ ↓)	0.051±0.016 (↓ ↓)	0.036±0.011 (↓ ↓)	0.027±0.008 (↓ ↓)	0.036±0.011 (↓ ↓)	**0.018±0.013** (○)	0.023±0.011
Australian	0.029±0.015 (↓ ↓)	0.057±0.019 (↓ ↓)	0.030±0.013 (↓ ↓)	0.010±0.010 (↓ ↓)	0.024±0.016 (↓ ↓)	0.003±0.007 (○)	**0.002±0.005**
Vehicle	0.021±0.011 (↓ ↓)	0.036±0.010 (↓ ↓)	0.026±0.010 (↓ ↓)	0.015±0.010 (↓ ↓)	0.020±0.012 (↓ ↓)	**0.004±0.003** (↑)	0.006±0.004
German	0.056±0.015 (↓ ↓)	0.049±0.011 (↓ ↓)	0.051±0.017 (↓ ↓)	0.045±0.017 (↓ ↓)	0.037±0.021 (↓ ↓)	**0.023±0.018** (○)	0.025±0.021
WBCD	0.012±0.011 (↓ ↓)	0.080±0.028 (↓ ↓)	0.012±0.011 (↓ ↓)	0.007±0.010 (○ ↓)	0.015±0.010 (↓ ↓)	0.009±0.009 (↓)	**0.000±0.002**
Sonar	0.016±0.003 (↓ ↓)	0.072±0.009 (↓ ↓)	0.015±0.003 (↓ ↓)	0.015±0.003 (↓ ↓)	0.014±0.003 (↓ ↓)	0.010±0.002 (↓)	**0.009±0.003**
Hillvalley	0.005±0.002 (↑ ↓)	0.083±0.005 (↓ ↓)	0.006±0.002 (↑ ↓)	0.007±0.003 (○ ↓)	0.006±0.001 (○ ↓)	0.006±0.001 (↓)	**0.005±0.001**
Musk1	0.008±0.001 (↓ ↓)	0.060±0.004 (↓ ↓)	0.007±0.002 (○ ○)	0.010±0.002 (↓ ↓)	0.007±0.001 (○ ○)	**0.007±0.001** (↑)	0.007±0.001
Arrhythmia	0.003±0.001 (↓ ↓)	0.082±0.003 (↓ ↓)	0.002±0.001 (↓ ↓)	0.003±0.001 (↓ ↓)	0.002±0.000 (↓ ↓)	0.002±0.000 (↓)	**0.002±0.000**
Madelon	0.024±0.001 (↓ ↓)	0.094±0.003 (↓ ↓)	0.023±0.001 (↓ ↓)	0.018±0.005 (↓ ↓)	0.013±0.003 (↓ ↓)	0.007±0.001 (↓)	**0.004±0.001**
Isolet5	0.005±0.001 (↓ ↓)	0.050±0.001 (↓ ↓)	0.003±0.001 (↓ ↓)	0.004±0.001 (↓ ↓)	0.001±0.000 (↓ ↓)	0.001±0.000 (↓)	**0.001±0.000**
MultipleFeatures	0.008±0.001 (↓ ↓)	0.069±0.002 (↓ ↓)	0.007±0.001 (↓ ↓)	0.010±0.003 (↓ ↓)	0.004±0.001 (↓ ↓)	0.003±0.000 (↓)	**0.001±0.000**

TABLE III: IGD on test sets.

Dataset	NSGA-II	MODE	SPEA2	OMOPSO	MOEAD	MOEA/D-STAT	MOEA/D-DYN
Wine	0.081±0.020 (↓ ↓)	0.074±0.031 (↓ ↓)	0.014±0.015 (○ ↓)	0.009±0.002 (○ ○)	0.018±0.014 (↓ ↓)	0.009±0.006 (○)	**0.008±0.000**
Australian	0.044±0.031 (↓ ↓)	0.061±0.019 (↓ ↓)	0.049±0.034 (↓ ↓)	0.024±0.015 (↓ ↓)	0.037±0.029 (↓ ↓)	**0.016±0.007** (○)	0.018±0.004
Vehicle	0.020±0.009 (↓ ○)	0.036±0.014 (↓ ↓)	0.024±0.013 (↓ ○)	0.023±0.010 (↓ ○)	0.023±0.013 (↓ ○)	**0.017±0.009** (↑)	0.022±0.009
German	0.068±0.015 (○ ○)	**0.056±0.010** (↑ ○)	0.070±0.017 (○ ○)	0.073±0.018 (○ ↓)	0.067±0.021 (○ ○)	0.064±0.027 (○)	0.063±0.023
WBCD	0.007±0.009 (○ ↓)	0.113±0.036 (↓ ↓)	0.008±0.011 (○ ↓)	0.003±0.003 (↑ ○)	0.008±0.010 (○ ↓)	0.007±0.009 (↓)	**0.003±0.002**
Sonar	0.032±0.009 (○ ○)	0.122±0.014 (↓ ↓)	0.029±0.006 (○ ○)	**0.028±0.007** (↑ ↑)	0.029±0.007 (○ ○)	0.031±0.005 (○)	0.031±0.005
Hillvalley	0.011±0.004 (↑ ○)	0.143±0.009 (↓ ↓)	**0.011±0.002** (↑ ↑)	0.013±0.003 (↑ ○)	0.013±0.004 (↑ ○)	0.015±0.004 (↓)	0.013±0.004
Musk1	0.021±0.003 (↓ ↓)	0.078±0.005 (↓ ↓)	0.020±0.003 (↓ ↓)	0.022±0.004 (↓ ↓)	0.018±0.003 (↓ ↓)	**0.015±0.004** (○)	0.015±0.004
Arrhythmia	0.005±0.001 (↓ ↓)	0.127±0.004 (↓ ↓)	0.005±0.001 (○ ↓)	0.006±0.002 (↓ ↓)	0.004±0.001 (○ ○)	0.005±0.001 (↓)	**0.004±0.001**
Madelon	0.048±0.001 (↓ ↓)	0.134±0.003 (↓ ↓)	0.047±0.003 (↓ ↓)	0.036±0.011 (↓ ↓)	0.019±0.006 (↓ ↓)	0.014±0.002 (↓)	**0.008±0.001**
Isolet5	0.009±0.001 (↓ ↓)	0.068±0.002 (↓ ↓)	0.006±0.001 (↓ ↓)	0.008±0.002 (↓ ↓)	0.002±0.001 (↓ ↓)	0.001±0.000 (↓)	**0.001±0.000**
MultipleFeatures	0.011±0.001 (↓ ↓)	0.085±0.002 (↓ ↓)	0.010±0.001 (↓ ↓)	0.014±0.003 (↓ ↓)	0.005±0.001 (↓ ↓)	0.005±0.001 (↓)	**0.001±0.000**

TABLE IV: Hypervolume on training sets.

Dataset	NSGA-II	MODE	SPEA2	OMOPSO	MOEAD	MOEA/D-STAT	MOEA/D-DYN
Wine	0.751±0.056 (↓ ↓)	0.752±0.077 (↓ ↓)	0.870±0.020 (↓ ↓)	0.876±0.001 (↓ ↓)	0.872±0.005 (↓ ↓)	**0.877±0.001** (○)	0.877±0.001
Australian	0.778±0.015 (↓ ↓)	0.662±0.065 (↓ ↓)	0.782±0.008 (↓ ↓)	0.794±0.003 (○ ↓)	0.783±0.019 (↓ ↓)	0.794±0.002 (↓)	**0.795±0.000**
Vehicle	0.795±0.009 (↓ ↓)	0.663±0.059 (↓ ↓)	0.794±0.010 (↓ ↓)	0.801±0.002 (○ ↓)	0.796±0.006 (↓ ↓)	0.801±0.001 (↓)	**0.802±0.001**
German	0.709±0.013 (↓ ↓)	0.556±0.046 (↓ ↓)	0.707±0.016 (↓ ↓)	0.717±0.004 (○ ↓)	0.713±0.006 (↓ ↓)	0.718±0.004 (↓)	**0.719±0.003**
WBCD	0.916±0.009 (↓ ↓)	0.755±0.054 (↓ ↓)	0.917±0.006 (↓ ↓)	0.919±0.001 (○ ↓)	0.918±0.002 (↓ ↓)	0.920±0.001 (↓)	**0.920±0.000**
Sonar	0.871±0.014 (↓ ↓)	0.553±0.031 (↓ ↓)	0.867±0.013 (↓ ↓)	0.867±0.012 (↓ ↓)	0.869±0.012 (↓ ↓)	0.887±0.007 (○)	**0.889±0.008**
Hillvalley	0.617±0.007 (↓ ↓)	0.372±0.011 (↓ ↓)	0.616±0.004 (↓ ↓)	0.611±0.007 (↓ ↓)	0.614±0.007 (↓ ↓)	0.620±0.004 (↓)	**0.625±0.003**
Musk1	0.919±0.010 (↓ ↓)	0.587±0.017 (↓ ↓)	0.924±0.007 (↓ ↓)	0.898±0.014 (↓ ↓)	0.929±0.005 (↓ ↓)	**0.933±0.004** (○)	0.932±0.004
Arrhythmia	0.940±0.006 (↓ ↓)	0.580±0.012 (↓ ↓)	0.949±0.005 (↓ ↓)	0.940±0.012 (↓ ↓)	0.955±0.002 (↓ ↓)	0.957±0.001 (○)	**0.957±0.001**
Madelon	0.874±0.011 (↓ ↓)	0.461±0.010 (↓ ↓)	0.883±0.009 (↓ ↓)	0.863±0.018 (↓ ↓)	0.849±0.011 (↓ ↓)	0.891±0.004 (↓)	**0.896±0.003**
Isolet5	0.922±0.010 (↓ ↓)	0.575±0.009 (↓ ↓)	0.944±0.010 (↓ ↓)	0.927±0.012 (↓ ↓)	0.973±0.004 (↓ ↓)	0.988±0.000 (↓)	**0.991±0.000**
MultipleFeatures	0.951±0.007 (↓ ↓)	0.651±0.008 (↓ ↓)	0.960±0.008 (↓ ↓)	0.933±0.016 (↓ ↓)	0.974±0.006 (↓ ↓)	0.991±0.000 (↓)	**0.994±0.000**

TABLE V: Hypervolume on test sets.

Dataset	NSGA-II	MODE	SPEA2	OMOPSO	MOEAD	MOEA/D-STAT	MOEA/D-DYN
Wine	0.754±0.058 (↓ ↓)	0.757±0.085 (↓ ↓)	0.894±0.029 (↓ ↓)	0.904±0.003 (○ ○)	0.890±0.019 (↓ ↓)	0.903±0.006 (○)	**0.904±0.000**
Australian	0.747±0.061 (↓ ↓)	0.663±0.068 (↓ ↓)	0.739±0.065 (↓ ↓)	0.781±0.022 (↓ ↓)	0.760±0.055 (↓ ↓)	**0.791±0.006** (○)	0.790±0.004
Vehicle	0.791±0.011 (↓ ↓)	0.669±0.061 (↓ ↓)	0.788±0.012 (↓ ↓)	0.797±0.004 (↑ ○)	0.793±0.009 (○ ↓)	0.795±0.004 (↓)	**0.798±0.003**
German	0.669±0.022 (↓ ↓)	0.531±0.046 (↓ ↓)	0.671±0.018 (↓ ↓)	0.678±0.010 (○ ○)	0.673±0.014 (↓ ↓)	0.680±0.007 (○)	**0.680±0.006**
WBCD	0.909±0.012 (○ ↓)	0.745±0.054 (↓ ↓)	0.908±0.014 (○ ↓)	0.914±0.001 (↑ ○)	0.908±0.012 (○ ↓)	0.912±0.005 (↓)	**0.914±0.000**
Sonar	0.774±0.031 (↓ ↓)	0.552±0.036 (↓ ↓)	0.782±0.022 (↓ ↓)	0.790±0.027 (○ ○)	0.790±0.027 (○ ○)	**0.798±0.021** (○)	0.793±0.022
Hillvalley	0.595±0.013 (↑ ○)	0.381±0.013 (↓ ↓)	0.598±0.010 (↑ ○)	0.589±0.012 (○ ↓)	0.593±0.012 (○ ○)	0.590±0.010 (↓)	**0.598±0.011**
Musk1	0.846±0.019 (↓ ↓)	0.576±0.017 (↓ ↓)	0.857±0.015 (↓ ↓)	0.834±0.025 (↓ ↓)	0.860±0.013 (↓ ↓)	0.868±0.010 (○)	**0.872±0.010**
Arrhythmia	0.934±0.007 (↓ ↓)	0.582±0.012 (↓ ↓)	0.943±0.005 (↓ ↓)	0.935±0.012 (↓ ↓)	0.951±0.002 (↓ ↓)	0.952±0.002 (○)	**0.952±0.002**
Madelon	0.860±0.011 (↓ ↓)	0.466±0.011 (↓ ↓)	0.869±0.009 (↓ ↓)	0.857±0.016 (↓ ↓)	0.849±0.011 (↓ ↓)	0.883±0.004 (↓)	**0.886±0.004**
Isolet5	0.919±0.011 (↓ ↓)	0.574±0.009 (↓ ↓)	0.941±0.010 (↓ ↓)	0.924±0.013 (↓ ↓)	0.971±0.004 (↓ ↓)	0.985±0.001 (↓)	**0.989±0.001**
MultipleFeatures	0.947±0.007 (↓ ↓)	0.648±0.008 (↓ ↓)	0.956±0.008 (↓ ↓)	0.929±0.016 (↓ ↓)	0.971±0.006 (↓ ↓)	0.987±0.001 (↓)	**0.990±0.001**

Four datasets are selected as representatives of small (Vehicle), medium (Musk1) and large (Madelon, Isolet5) datasets due to page limit. The patterns are similar on the other datasets. Note that in the figure, MODE is not shown since its obtained median fronts contain feature subsets with large numbers of features and high classification errors, which makes the differences between other algorithms difficult to be visible.

A. Comparison with the case of using all features

As can be seen from Figs. 6 and 7, on all datasets, the two proposed algorithms evolve feature subsets containing at most 60% of the original features. There are at least three feature subsets that are better than using all features. Especially, on Madelon, most subsets selected by the two algorithms achieve better classification performance than using all features while selecting less than 5% of the original features.

The results suggest that on all datasets, applying multiple reference points to MOEA/D based feature selection can select a small number of features while still achieve better performance than using all features.

B. MOEA/D-STAT vs other EMO methods

On the training set, as shown in Tables II and IV, MOEA/D-STAT achieves significantly better indicator values on most datasets. Among the benchmark algorithms, SPEA2 achieves the best hypervolume values on the medium and large datasets while OMOPSO has the best performance

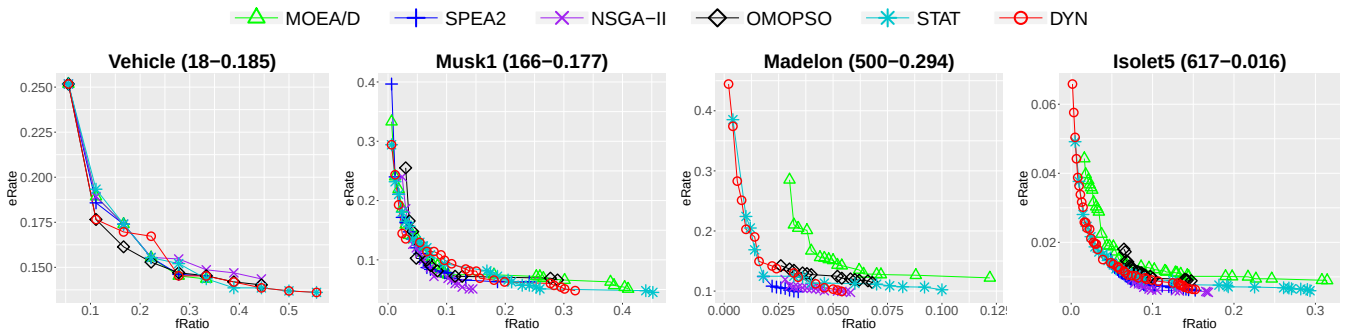


Fig. 6: Median fronts on training sets.

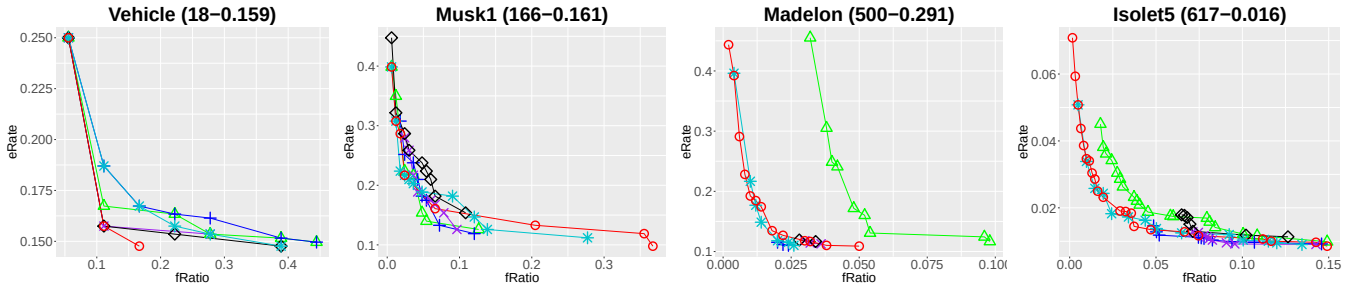


Fig. 7: Median fronts on test sets.

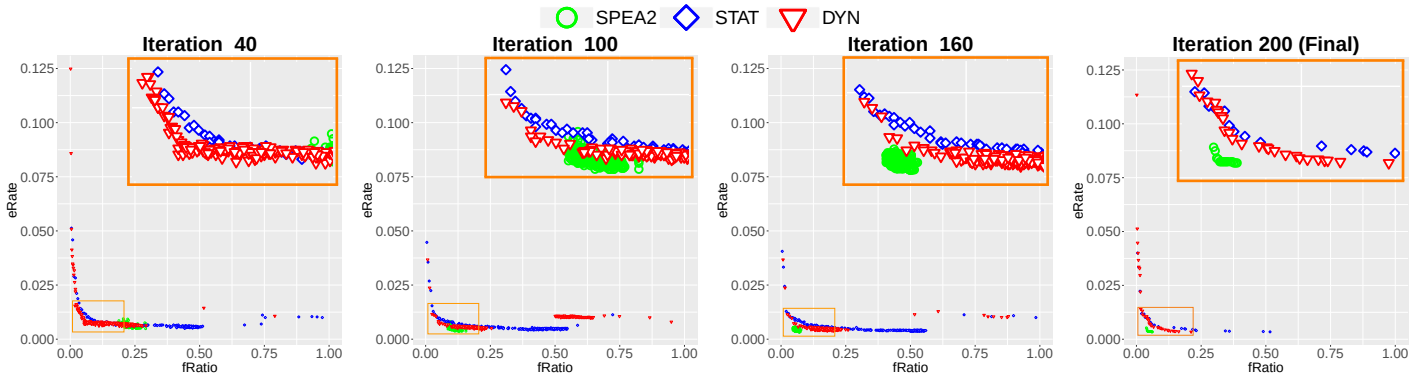


Fig. 8: Evolutionary processes of the 1^{st} run on the MultipleFeatures dataset.

on the small datasets (similar performance with MOEA/D-STAT on four small datasets). When the number of features is large, MOEA/D-STAT significantly outperforms all benchmark algorithms. The possible reasons can be seen in Fig. 6. On the small datasets, MOEA/D-STAT evolves the similar shapes with the four benchmark algorithms, but for the same number of features, MOEA/D-STAT tends to achieve lower classification errors. It is mainly because the decomposition along with feature ratios makes the search space of each sub-problem much smaller than the original search space. Although the upper bound feature ratio limits the number of features, MOEA/D-STAT still allows to replace worse features in the current subset by better features through communications with its neighboring sub-problems. On the larger datasets such as Madelon and Isolet5, the fronts become significantly different. Dominance-based algorithms quickly lose their diversities and their populations focus mostly on the middle of the Pareto front. OMOPSO is the worst dominance-based algorithm while SPEA2 and NSGA-II show similar search behaviors. Meanwhile, decomposition-based algorithms achieve much more diverse front. Even standard MOEA/D's solutions have many more different feature ratios than NSGAII and SPEAII. MOEA/D-STAT also evolves as diverse feature subsets as standard MOEA/D. However, with the same number of features, MOEA/D-STAT always achieves a better classification error. The reason is the fitness function in the proposed decomposition strategy gives higher priority to the classification error. Therefore, MOEA/D-STAT focuses more on the classification error than standard MOEA/D.

On the test sets, as shown in Tables III and V, MOEA/D-STAT is significantly better than the benchmark algorithms on at least eight out of the 12 datasets, especially on the five largest datasets. Fig. 7 shows that the median fronts evolved by MOEA/D-STAT are usually more diverse than the ones by the dominance-based algorithms. On Madelon, the median front by MOEA/D-STAT contains 13 solutions with feature ratios ranging from 0.002 to 0.05. Meanwhile, the four dominance-based algorithms have only two or three solutions on their median fronts and most classification errors achieved by the dominance-based algorithms are attained by MOEA/D-STAT.

The results show that using multiple reference points generates better feature subsets than using multiple weight vectors.

Since the fitness function in the proposed decomposition focuses more on reducing the classification error, its classification performance is significantly better than standard MOEA/D. The new decomposition not only preserves the higher diversity over dominance-based algorithms but also improves the diversity over using multiple weight vectors since it ensures that each sub-problem, defined by a reference point, corresponds to a solution on the Pareto front.

C. MOEA/D-DYN vs others

In Tables II - V, the second sign in the brackets shows the significance test results, which compares each of the five benchmark algorithms with MOEA/D-DYN. On the training set, in terms of the hypervolume indicator, NSGA-II, MODE, SPEA2, OMOPSO and MOEA/D are significantly worse than MOEA/D-DYN on *all* datasets. MOEA/D-STAT achieves similar hypervolume as MOEA/D-DYN on four datasets while being significantly worse on all other eight datasets. In terms of IGD, on most cases the other algorithms are significantly worse than MOEA/D-DYN. Similarly, on the test sets, MOEA/D-DYN is worse than the other algorithms on at most one dataset. The superiority of MOEA/D-DYN to MOEA/D-STAT shows that its dynamic mechanism does not affect the algorithm's convergence and still preserves the high performance of the new decomposition.

Now we will focus more on analyzing the effect of the dynamic mechanism. As shown in Tables II-V, MOEA/D-DYN achieves significantly better IGD/hypervolume values than MOEA/D-STAT. The significant improvement is a result of improvement in both classification performance and diversity of feature subsets, which can be seen in Fig. 6. On the small datasets such as Vehicle, MOEA/D-DYN's fronts have the same length as MOEA/D-STAT. However, with the same feature ratio, MOEA/D-DYN's classification error is always lower. On the medium datasets, MOEA/D-DYN's fronts become shorter because MOEA/D-DYN selects fewer features than MOEA/D-STAT to achieve the same classification performance. This pattern is clearly shown on Musk1 in Fig. 6. On the two large datasets, Madelon and Isolet5, the fronts evolved by MOEA/D-DYN is even much shorter than MOEA/D-STAT's ones. However, shorter fronts do not mean MOEA/D-DYN's solution sets are less diverse than the solutions found by MOEA/D-STAT. Let take the median fronts on Madelon as an example. MOEA/D-DYN's median front contains 16 feature subsets, which have feature ratios varying in the range [0.002, 0.054]. Although MOEA/D-STAT's feature ratios have a longer range, [0.004, 0.1], its median front has only 14 feature subsets. Despite selecting two times more features than MOEA/D-DYN, MOEA/D-STAT's best feature subset in terms of classification error is still worse than that of MOEA/D-DYN. Since the dynamic mechanism does not waste resources on non-conflicting regions, it puts more effort on the conflicting regions, which results in more diverse feature subsets with better classification performance. MOEA/D-DYN also achieves better performance than state-of-the-art many-objective EMO algorithms on feature selection, such as NSGA-III [45], MOEA/DD [46], and θ-DEA [47]. More details can be seen in the supplementary material.

D. Further analysis on the evolutionary processes

In this section, the search behaviors of different algorithms are examined through their evolutionary processes. Besides MOEA/D-DYN and MOEA/D-STAT, SPEA2 is selected as a representative of dominance-based algorithms since it achieves the best performance among the dominance-based algorithms. The largest dataset, MultipleFeatures, is selected to show the differences between the three algorithms clearly. The three evolutionary processes are shown in Fig. 8. There are four sub-figures corresponding to the populations at the $40^{th}, 100^{th}, 160^{th}$ and 200^{th}, the final iteration, respectively. All algorithms start from the same initialization, but it is not shown to save space. Since there is too much overlapping in the center of Pareto fronts, we zoom these parts and put the zoomed figure on the top right of each sub-figure.

As can be seen from the figure, at the 40^{th} iteration, SPEA2 quickly loses its diversity due to the dominance ranking and it mainly searches on a very small area of the objective space. In the following iterations, its diversity becomes gradually worse which finally results in a low-diversity front. On the other hand, MOEA/D-STAT and MOEA/D-DYN maintain their diversities through the whole process. However, their search behaviors are quire different. It can be seen that since MOEA/D-STAT evenly distributed its reference points on the *fRatio* axis, its population spreads on the whole axis. However, the solutions within the small *fRatio* (less than 0.5) is more dense since some sub-problems with a large n_{ref} values can take subsets with low *fRatio* as their solutions. MOEA/D-DYN starts with focusing on the first interval which has the feature ratio ranging in [0, 0.25]. Therefore, in the first figure, most solutions are in this range. In the following boundary iterations, MOEA/D-DYN shifts its focus to the next interval. As shown in the second figure, at the 100^{th} iteration, MOEA/D-DYN starts focusing on the third interval by allocating more reference points there. It seems that after a number of iterations, it finds that the third interval is a $threshold$ interval, from which the two objectives are possibly no longer conflicting. Therefore, MOEA/D-DYN allocates all moving reference points on the first and second intervals as shown in the fourth figure, without further reallocation. Note that MOEA/D-DYN still leaves some reference points on the other intervals in case the $threshold\ interval$ detection is not accurate. The final figure shows that the subsets evolved by MOEA/D-DYN are more diverse with better classification performance than the subsets evolved by MOEA/D-STAT.

E. MOEA/D-DYN vs classical feature selection approaches

In this section, MOEA/D-DYN is compared with four classical feature selection algorithms including mRMR [37], ReliefF [38], RFS [40], and CFS [39]. Since mRMR and CFS can select features without specifying the number of selected features, they result in exactly one feature subset for each dataset. RFS and ReliefF output feature scores and they require to pre-define the number of selected features. To have a relatively fair comparison, we compute an average front of MOEA/D-DYN on each dataset. 50 fronts obtained by MOEA/D-DYN in the 50 independent runs are combined into a union set. The classification errors of all feature subsets

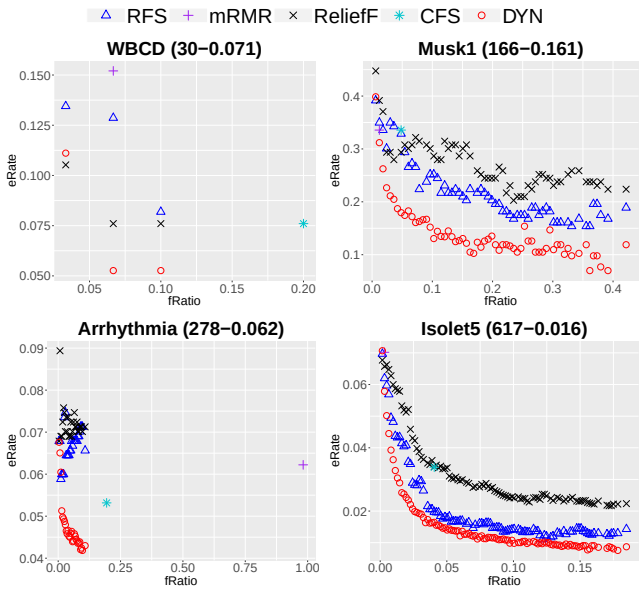


Fig. 9: Comparison with standard feature selection methods.

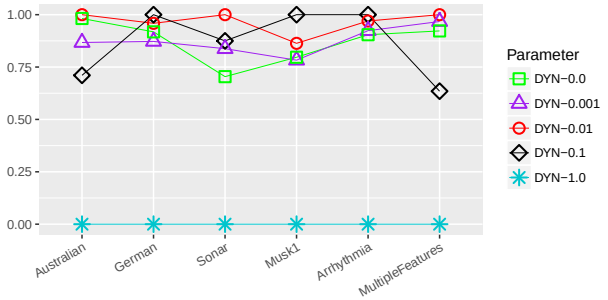


Fig. 10: Hypervolumes of different α values.

with the same number of features (n features) are "averaged" to obtain the average classification of n features. The "average front" is obtained by combining all possible numbers of features and their corresponding average classification errors. The numbers of features appeared in the "average front" are used as pre-defined numbers of features for RFS and ReliefF. Therefore, RFS and ReliefF result in a set of feature subsets.

Fig. 9 show the solutions of the four standard feature selection approaches and the average fronts of MOEA/D-DYN on WBCD, Musk1, Arrhythmia, and Isolet5. Similar patterns are observed in other datasets. Given the same number of features, MOEA/D-DYN achieves lower classification errors than the other four standard approaches. mRMR and CFS are prone to be trapped in local optima since they used greedy searches such as sequential forward search and best first search to select features. ReliefF updates the score of each feature individually, which results in missing feature interactions and selecting redundant features. Among the four classical approaches, RFS achieves the most promising results. The main reason is that RFS updates weights of features to minimize the difference between the predicted output using all features and the desired output, so RFS considers more feature interactions than the other three standard approaches. However, RFS assumes the linear relationship between features and the class labels, which may not be true in real-world applications.

TABLE VI: Gene expression datasets.

Dataset	#Features	#Classes	#Instances
SRBCT	2308	4	83
Leukemia1	5327	3	72
DLBCL	5469	2	77
Brain1	5920	5	90
Leukemia	7129	2	72

F. Further analysis on the effect of α

In order to examine the effect of α in Eq. (4), we examine five different values of α: 1.0, 0.1, 0.01, 0.001, and 0.0 on the 12 datasets. The value of 1.0 shows the equal importance between two objectives while the value of 0.0 means the classification accuracy is absolutely more important. A smaller α value puts more pressure towards the classification accuracy.

For each value, MOEA/D-DYN has been run for 50 independent times on each dataset. The results of the five different values on six datasets are shown in Fig. 10. Similar results are obtained on the other datasets. On each dataset, the five hypervolumes by the five values are normalized for more explicit comparisons.

Among the five values, the value of 1.0 results in the worst performance on all datasets. The main reason is that reducing the number of features is easier than reducing the classification error. If the two objectives have the same importance ($\alpha = 1$), the algorithm focuses more on the number of features. The smaller values of α can select the same number of features with better classification performance, which results in their better performance. Although the value of 0.0 has better performance than the value of 1.0, it is not as good as the other three values due to its absolute priority for the classification performance. The strict requirement of having lower classification errors to be a better solution makes the population less diverse. Significance test results show that the three values 0.1, 0.01, and 0.001 are not significantly different. The value of 0.01 usually achieves the best or second best results on all datasets, so it is the recommended setting for α but the results are clearly not sensitive to this choice.

G. Further discussion-results on high-dimensional datasets

In the above subsections, the proposed algorithms are examined on datasets with up to 650 features. We also examine the scalability of the proposed algorithms on gene expression datasets [48] (Table VI) that contain thousands of features.

The IGD results of MOEA/D-STAT, MOEA/D-DYN and the five benchmark algorithms are shown in Table VII. Similar patterns are obtained on the hypervolume indicator. On all datasets, MOEA/D-STAT significantly outperforms the five benchmark algorithms, except for the test sets of SRBCT where MOEA/D-STAT have the same performance as NSGA-II, OMOPSO and standard MOEA/D. MOEA/D-DYN significantly outperforms all the five benchmark algorithms on all datasets. In comparison with MOEA/D-STAT, MOEA/D-DYN achieves significantly better IGD values on all datasets. The results show that the proposed decomposition mechanism scales well with the number of features.

H. Computational times

The computational times of the proposed algorithms and the five benchmark algorithms are shown in Table VIII. Note that

TABLE VII: IGD on test sets (Gene expression datasets).

Dataset	NSGA-II	MODE	SPEA2	OMOPSO	MOEA/D	MOEA/D-STAT	MOEA/D-DYN
SRBCT	0.076±0.005 (∘ ↓)	0.201±0.004 (↓ ↓)	0.078±0.006 (↓ ↓)	0.072±0.007 (∘ ↓)	0.075±0.007 (∘ ↓)	0.074±0.004 (↓)	**0.008±0.002**
Leukemia1	0.090±0.008 (↓ ↓)	0.233±0.005 (↓ ↓)	0.095±0.008 (↓ ↓)	0.051±0.009 (↓ ↓)	0.087±0.004 (↓ ↓)	0.032±0.009 (↓)	**0.015±0.002**
DLBCL	0.096±0.006 (↓ ↓)	0.216±0.004 (↓ ↓)	0.098±0.010 (↓ ↓)	0.074±0.009 (↓ ↓)	0.095±0.010 (↓ ↓)	0.048±0.011 (↓)	**0.015±0.007**
Brain1	0.069±0.005 (↓ ↓)	0.208±0.004 (↓ ↓)	0.072±0.007 (↓ ↓)	0.037±0.008 (↓ ↓)	0.070±0.004 (↓ ↓)	0.017±0.003 (↓)	**0.010±0.004**
Leukemia	0.120±0.013 (↓ ↓)	0.302±0.004 (↓ ↓)	0.117±0.007 (↓ ↓)	0.072±0.028 (↓ ↓)	0.116±0.008 (↓ ↓)	0.032±0.015 (↓)	**0.016±0.012**

TABLE VIII: CPU times (minutes).

Dataset	NSGA-II	MODE	SPEA2	OMOPSO	MOEA/D	STAT	DYN
Wine	**0.0**	0.1	0.1	0.1	0.12	0.12	0.13
Australian	1.5	1.78	**1.48**	1.59	1.65	2.04	2.05
Vehicle	3.46	3.54	**3.44**	3.46	3.46	3.88	4.06
German	**6.24**	6.76	6.43	6.26	6.25	7.46	7.45
WBCD	2.42	2.62	2.4	**2.38**	2.43	2.8	3.06
Ionosphere	1.03	1.13	**0.99**	1.03	1.14	1.25	1.25
Sonar	0.64	0.71	**0.63**	0.64	0.73	0.75	0.76
Hillvalley	42.58	48.62	**39.53**	42.93	44.49	47.06	46.03
Musk1	11.01	13.55	**10.82**	11.1	11.44	13.2	13.34
Arrhythmia	12.89	18.42	12.83	**12.75**	13.63	14.07	13.69
Madelon	514.48	894.7	529.87	**497.72**	619.81	629.07	541.64
Isolet5	223.26	339.47	216.08	**203.25**	207.74	258.42	256.58
MFs	**343.84**	536.77	361.41	344.31	351.93	496.95	394.56
SRBCT	1.59	2.64	1.95	**1.14**	1.76	1.98	1.86
Leukemia1	3.52	4.95	4.22	**1.93**	3.52	3.52	2.97
DLBCL	3.91	5.95	4.76	**2.09**	3.92	4.02	3.38
Brain1	5.61	8.52	6.08	**3.09**	5.7	6.04	5.55
Leukemia	4.9	6.97	5.39	**2.64**	4.63	5.23	4.97

in a wrapper-based feature selection approach, the most time-consuming step is evaluation due to the involvement of the classification process. In general, MOEA/D-based feature selection algorithms are not as efficient as the Pareto dominance-based algorithms. The main reason is that MOEA/D-based algorithms usually evolve more diverse fronts which may contain feature subsets with large numbers of features. Since the computational cost of a classification process increases when the number of features is increased, MOEA/D-based algorithms have longer computational times. Among the three MOEA/D algorithms, the standard MOEA/D algorithm is more efficient since it does not repair duplicated solutions as in MOEA/D-STAT and MOEA/D-DYN.

In comparison with MOEA/D-STAT, MOEA/D-DYN is a little bit slower on the small and medium datasets due to its reference points re-allocation. However, the differences between these two algorithms are small. On the large datasets such as Arrhythmia and Madelon, MOEA/D-DYN is more efficient than MOEA/D-STAT. A possible reason is that MOEA/D-DYN finds a threshold interval, which allows MOEA/D-DYN to focus on regions corresponding to small numbers of features. On the contrary, MOEA/D-STAT evenly distributes reference points, so it has to consider regions corresponding to large numbers of features. In comparison with dominance-based algorithms, MOEA/D-DYN is at most 10% slower on the large datasets, but it can evolve much more diverse Pareto fronts. In general, among the seven algorithms, MOEA/D-DYN has the best trade-off between effectiveness and efficiency thanks to its mechanism to focus only on conflicting regions that usually have small numbers of features.

VI. Conclusions and future work

In this paper, first a new decomposition for MOEA/D is proposed to solve feature selection problems. Instead of using multiple weight vectors, feature selection is decomposed by a set of reference points allocated along the feature ratio axis. The new decomposition is designed to deal with the discrete Pareto front of feature selection. Then, a dynamic reference point strategy is proposed to detect and allocate more resource to the conflicting regions. The experimental results show that the two multiple reference points algorithms can evolve more diverse feature subsets than the five benchmark multi-objective algorithms, including NSGA-II, MODE, SPEA2, OMOPSO and standard MOEA/D, and the four classical feature selection algorithms including mRMR, ReliefF, RFS, and CFS. The multiple reference points decomposition also assists MOEA/D to achieve better classification accuracy than using weight vectors since there is more search pressure on improving the classification performance. The dynamic mechanism allows MOEA/D to focus more on the conflicting regions, which results in more diverse Pareto fronts with lower classification errors than the static mechanism.

The proposed multiple reference points decomposition allows MOEA/D to work with multi-objective problems which have discrete Pareto fronts like feature selection. The proposed decomposition also alleviates the dependency on the Pareto front shape since it guarantees each sub-problem with a specific reference point corresponding to a solution on the Pareto front. Furthermore, the dynamic multiple reference points mechanism is useful for problems which have their objectives partially conflicting. By investigating different sub-regions of the objective spaces, the dynamic mechanism can estimate in which regions the objectives are mostly conflicting and accordingly allocate more resources on these regions.

A limitation of this work is that the multiple reference points algorithms spend computational time on repairing duplicated feature subsets, which requires to re-evaluate the repaired solutions. In the future, we will investigate a more sophisticated evolutionary mechanism to avoid producing duplicated solution leading to better efficiency. Additionally, although the multiple reference points decomposition achieves more diverse fronts than dominance-based algorithms such as SPEA2, sometimes solutions evolved by SPEA2 have higher classification performances. If more search pressure is putting on regions of those solutions, the feature subsets evolved by MOEA/D can be further improved. However, these regions depend on datasets and it is not easy to identify them. Recently, a number of measures are proposed for feature selection, e.g., information-based measures, and they achieve promising results. It would be interesting to analyze the relationship between these measures and the two main objectives of feature selection to improve the feature selection performance.

Acknowledgement

This work was supported in part by the Marsden Fund of New Zealand Government under Contracts VUW1509 and VUW1615, Huawei Industry Fund E2880/3663, and the University Research Fund at Victoria University of Wellington grant number 216378/3764.

REFERENCES

[1] J. Fan, F. Han, and H. Liu, "Challenges of big data analysis," *National science review*, vol. 1, no. 2, pp. 293–314, 2014.

[2] Q. Chen, M. Zhang, and B. Xue, "Feature selection to improve generalisation of genetic programming for high-dimensional symbolic regression," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 5, pp. 792–806, 2017.

[3] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 4, pp. 606–626, 2016.

[4] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[5] E. Zitzler, M. Laumanns, L. Thiele *et al.*, "SPEA2: Improving the strength pareto evolutionary algorithm," in *Eurogen*, vol. 3242, no. 103, 2001, pp. 95–100.

[6] M. R. Sierra and C. C. Coello, "Improving PSO-based multi-objective optimization using crowding, mutation and e-dominance," in *Evolutionary multi-criterion optimization*, vol. 3410. Springer, 2005, pp. 505–519.

[7] T. Robič and B. Filipič, "Demo: Differential evolution for multiobjective optimization," in *Evolutionary Multi-Criterion Optimization*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 520–533.

[8] A. Jaszkiewicz, "On the computational efficiency of multiple objective metaheuristics. the knapsack problem case study," *European Journal of Operational Research*, vol. 158, no. 2, pp. 418–433, 2004.

[9] B. Xue, M. Zhang, and W. N. Browne, "Particle swarm optimization for feature selection in classification: a multi-objective approach," *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1656–1671, 2013.

[10] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," in *Evolutionary multi-criterion optimization*. Springer, 2009, pp. 438–452.

[11] V. A. Shim, K. C. Tan, and C. Y. Cheong, "A hybrid estimation of distribution algorithm with decomposition for solving the multiobjective multiple traveling salesman problem," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 5, pp. 682–691, 2012.

[12] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[13] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.

[14] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[15] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 2, pp. 169–190, 2017.

[16] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary Computation*, vol. 22, no. 2, pp. 231–264, 2014.

[17] C. Zhang, K. C. Tan, L. H. Lee, and L. Gao, "Adjust weight vectors in MOEA/D for bi-objective optimization problems with discontinuous pareto fronts," *Soft Computing*, pp. 1–16, 2017.

[18] H. B. Nguyen, B. Xue, H. Ishibuchi, P. Andreae, and M. Zhang, "Multiple reference points MOEA/D for feature selection," in *Proceedings of the Annual Conference on Genetic and Evolutionary Computation (Companion)*, 2017, pp. 157–158.

[19] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 450–455, 2014.

[20] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on evolutionary computation*, vol. 11, no. 6, pp. 712–731, 2007.

[21] A. Trivedi, D. Srinivasan, K. Sanyal, and A. Ghosh, "A survey of multiobjective evolutionary algorithms based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 3, pp. 440–462, 2017.

[22] A. Mukhopadhyay and U. Maulik, "An SVM-wrapped multiobjective evolutionary feature selection approach for identifying cancer-microrna markers," *IEEE Transactions on Nanobioscience*, vol. 12, no. 4, pp. 275–281, 2013.

[23] L. D. Vignolo, D. H. Milone, and J. Scharcanski, "Feature selection for face recognition based on multi-objective evolutionary wrappers," *Expert Systems with Applications*, vol. 40, no. 13, pp. 5077 – 5084, 2013.

[24] B. Xue, M. Zhang, and W. N. Browne, "Multi-objective particle swarm optimisation (PSO) for feature selection," in *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, 2012, pp. 81–88.

[25] H. B. Nguyen, B. Xue, I. Liu, P. Andreae, and M. Zhang, "New mechanism for archive maintenance in pso-based multi-objective feature selection," *Soft Computing*, vol. 20, no. 10, pp. 3927–3946, 2016.

[26] B. Xue, W. Fu, and M. Zhang, "Multi-objective feature selection in classification: a differential evolution approach," in *Proceedings of the 10th International Conference on Simulated Evolution and Learning*. Springer, 2014, pp. 516–528.

[27] S. Paul and S. Das, "Simultaneous feature selection and weighting–an evolutionary multi-objective optimization approach," *Pattern Recognition Letters*, vol. 65, pp. 51–59, 2015.

[28] F. Gu and Y.-M. Cheung, "Self-organizing map-based weight design for decomposition-based many-objective evolutionary algorithm," *IEEE Transactions on Evolutionary Computation*, 2017, DOI: 10.1109/TEVC.2017.2695579.

[29] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

[30] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *IEEE Congress on Evolutionary Computation*, 2009, pp. 203–208.

[31] A. Zhou and Q. Zhang, "Are all the subproblems equally important? resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 52–64, 2016.

[32] H. Ishibuchi, M. Yamane, and Y. Nojima, "Difficulty in evolutionary multiobjective optimization of discrete objective functions with different granularities," in *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 230–245.

[33] Y. Yuan, H. Xu, and B. Wang, "An improved NSGA-III procedure for evolutionary many-objective optimization," in *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, 2014, pp. 661–668.

[34] Q. Zhang, H. Li, D. Maringer, and E. Tsang, "MOEA/D with NBI-style tchebycheff approach for portfolio management," in *IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.

[35] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Towards understanding the cost of adaptation in decomposition-based optimization algorithms," in *IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 615–620.

[36] K. Miettinen, *Nonlinear multiobjective optimization*. Springer Science and Business Media, 2012, vol. 12.

[37] H. Peng, F. Long, and 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*, vol. 27, no. 8, pp. 1226–1238, 2005.

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

[39] M. A. Hall and L. A. Smith, "Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper." in *FLAIRS conference*, vol. 1999, 1999, pp. 235–239.

[40] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint l2, 1-norms minimization," in *Advances in neural information processing systems*, 2010, pp. 1813–1821.

[41] M. Lichman, "UCI machine learning repository Irvine, CA: University of California, School of Information and Computer Sciences," 2013. [Online]. Available: http://archive.ics.uci.edu/ml

[42] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the hypervolume indicator: Optimal-distributions and the choice of the reference point," in *Proceedings of the ACM SIGEVO Workshop on Foundations of Genetic Algorithms*, 2009, pp. 87–102.

[43] J. D. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," *TIK-Report*, vol. 214, 2006.

[44] A. J. Nebro, J. J. Durillo, and M. Vergne, "Redesigning the jMetal multi-objective optimization framework," in *Proceedings of the Annual Conference on Genetic and Evolutionary Computation*, 2015, pp. 1093–1100.

[45] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[46] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[47] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16–37, 2016.

[48] Z. Zhu, Y.-S. Ong, and M. Dash, "Markov blanket-embedded genetic algorithm for gene selection," *Pattern Recognition*, vol. 40, no. 11, pp. 3236–3248, 2007.

Bach Hoai Nguyen (S'14) received his B.Sc. with First Class Honours and the Ph.D. in computer science at Victoria University of Wellington, New Zealand in 2015 and 2018, respectively. He is currently a Postdoctoral Research Fellow in the School of Engineering and Computer Science at Victoria University of Wellington. His research interests are in evolutionary computation, feature selection, feature construction, and transfer learning.

Dr. Nguyen is a member of the IEEE Computational Intelligence Society (CIS). He has been serving as a reviewer for over 10 international journals and conferences in the field such as IEEE TEVC, IEEE TCYB, Swarm and Evolutionary Computation, IEEE CEC, GECCO, SEAL, AAAI.

Bing Xue (M'10) received the B.Sc. degree from the Henan University of Economics and Law, Zhengzhou, China, in 2007, the M.Sc. degree in management from Shenzhen University, Shenzhen, China, in 2010, and the Ph.D. degree in computer science in 2014 at Victoria University of Wellington, New Zealand. She is currently an Associate Professor in School of Engineering and Computer Science at Victoria University of Wellington. She has over 100 papers published in fully refereed international journals and conferencesand her research focuses mainly on evolutionary computation, feature selection, feature construction, image analysis, and transfer learning.

Dr. Xue is currently Chair of the IEEE Computational Intelligence Society (CIS) Data Mining and Big Data Analytics Technical Committee, Chair ofthe IEEE Task Force on Evolutionary Feature Selection and Construction, Vice-Chair of IEEE CIS Task Force on Transfer Learning & Transfer Optimization, and Vice-Chair of IEEE CIS Task Force on Evolutionary Deep Learning and Applications.

Peter Andreae received the B.E. (Honours) degree in electrical engineering from the University of Canterbury, Christchurch, New Zealand, in 1977, and the Ph.D. degree in artificial intelligence from the Massachusetts Institute of Technology, Cambridge, USA in 1985. Since 1985, he has been teaching computer science with the School of Engineering and Computer Science, Victoria University of Wellington, New Zealand. He is currently an Associate Professor of Computer Science, Associate Dean (Students) and Associate Dean (Academic Development) of the Faculty of Engineering. His research interests include making agents that can learn behavior from experience, but he has also worked on a wide range of topics ranging from reconstructing vasculature from X-rays, clustering algorithms, analysis of microarray data, programming by demonstration, and software reuse.

Hisao Ishibuchi (M'93-SM'10-F'14) received the B.S. and M.S. degrees in precision mechanics from Kyoto University, Kyoto, Japan, in 1985 and 1987, respectively, and the Ph.D. degree in computer science from Osaka Prefecture University, Sakai, Osaka, Japan, in 1992. He was with Osaka Prefecture University in 1987-2017. Since 2017, he is a Chair Professor at Southern University of Science and Technology, China. His research interests include fuzzy rule-based classifiers, evolutionary multi-objective and many-objective optimization, memetic algorithms, and evolutionary games.

Dr. Ishibuchi was the IEEE CIS Vice-President for Technical Activities in 2010-2013, an AdCom member of the IEEE CIS in 2014-2019, and the Editor-in-Chief of the IEEE Computational Intelligence Magazine in 2014-2019.

Mengjie Zhang (M'04-SM'10-F'19) received the B.E. and M.E. degrees from Artificial Intelligence Research Center, Agricultural University of Hebei, Hebei, China, and the Ph.D. degree in computer science from RMIT University, Melbourne, VIC, Australia, in 1989, 1992, and 2000, respectively. He is currently Professor of Computer Science, Head of the Evolutionary Computation Research Group, and the Associate Dean (Research and Innovation) in the Faculty of Engineering. His current research interests include evolutionary computation with application areas of image analysis, multi-objective optimization, feature selection and reduction, job shop scheduling, and transfer learning. He has published over 500 research papers in refereed international journals and conferences. Prof. Zhang is a Fellow of Royal Society of New Zealand and has been a Panel member of the Marsden Fund (New Zealand Government Funding), a Fellow of IEEE, and a member of ACM.

He was the chair of the IEEE CIS Intelligent Systems and Applications Technical Committee, and chair for the IEEE CIS Emergent Technologies Technical Committee and the Evolutionary Computation Technical Committee, and a member of the IEEE CIS Award Committee. He is a vice-chair of the IEEE CIS Task Force on Evolutionary Feature Selection and Construction, a vice-chair of the Task Force on Evolutionary Computer Vision and Image Processing, and the founding chair of the IEEE Computational Intelligence Chapter in New Zealand. He is also a committee member of the IEEE NZ Central Section.