A Decomposition based Multi-objective Evolutionary Algorithm with ReliefF based Local Search and Solution Repair Mechanism for Feature Selection

Kaan Demir, Bach Hoai Nguyen, Bing Xue, and Mengjie Zhang
School of Engineering and Computer Scicene
Victoria University of Wellington, PO Box 600, Wellington 6140, New Zealand
Email: Kaan.Demir@vuw.ac.nz, {Hoai.Bach.Nguyen, Bing.Xue, Mengjie.Zhang}@ecs.vuw.ac.nz

***Abstract*—Feature selection has two main objectives which are to maximise the classification accuracy and to minimise the number of selected features. Unfortunately, the two objectives are usually in conflict, which makes feature selection a multi-objective problem. MOEA/D (multi-objective optimisation evolutionary algorithm based on decomposition) has shown to be effective in solving multi-objective feature selection, which evolves more diverse fronts than other multi-objective algorithms such as SPEA2 or NSGAII. However, sometimes the feature subsets around the middle of the evolved fronts do not have high classification performance. The goal of this work is to propose a local search for MOEA/D with an expectation of maintaining the front diversity while improving the classification performance of the feature subsets in the evolved fronts. The local search is based on three operators: insert, remove, and swap. The insert/remove operators either add/remove a single feature from the current feature subset, while the swap operator exchanges a selected feature with an unselected feature. The selection of added/removed/swapped features is based on Relief, a well-known measure which considers feature interactions. The experimental results show that the proposed local search can maintain or improve the fronts evolved by MOEA/D-DYN, a state-of-the-art MOEA/D algorithm for feature selection.**

***Index Terms*—Feature Interaction, Feature Selection, Classification, Multi-objective Optimization, Local Search**

I. INTRODUCTION

With regard to classification problems, it is common for instances of data to possess irrelevant and/or redundant features which reduce the classification performance. To address the problem, feature selection is proposed with two main goals: to reduce the number of features and to improve the classification performance. However, feature selection is a challenging task due to two issues. The first issue is its large search space. A feature selection problem with n original features can have 2^n possible feature subsets. Thus, the search space size increases exponentially with respect to the number of original features. Traditional feature selection algorithms such as Sequential Forward Selection (SFS) [1], and Sequential Backward Selection (SBS) [2] are capable of finding reduced feature sets by sequentially adding or removing a single feature from the current feature subset, but are limited by their tendency to converge at local optima. The second issue is the conflict between the two main goals of feature selection, which makes feature selection a multi-objective problem [3]. For example, removing too many features may result in worse classification performance since important information for classification is lost. Evolutionary multi-objective optimization (EMO) [4] is capable to deal with the two issues of feature selection. Firstly, EMO is a population-based optimisation family, which results in its potential global search ability on the large search space of feature selection. Secondly, EMO is designed for a multi-objective problem, which aims to evolve a front consisting a number of trade-off solutions. EMO can be divided into three main categories: dominance-based, decomposition-based, and indicator-based, among which the first two categories have been widely applied to feature selection. NSGAII and SPEA2 are two well-known representatives of the dominance-based category, while MOEA/D is the recently proposed algorithm in the decomposition-based category. The main idea of MOEA/D is to decompose a multi-objective problem into many single-objective sub-problems, which makes it easier to control the front's diversity. It has been shown that MOEA/D usually evolves more diverse fronts than NSGAII and SPEA2 [5]. Thus, this work focuses on developing a MOEA/D-based feature selection algorithm.

Despite evolving diverse fronts, MOEA/D-based feature selection algorithms usually generate low-quality feature subsets around the middle of the fronts (which we called the knee-point region) [6], [7]. Particularly, given the same number of selected features, the feature subsets evolved by MOEA/D have lower classification accuracy than those evolved by NSGAII or SPEA2. This work aims to address the problem by proposing a local search to improve the feature subsets around the knee-point region.

ISRPSOFS [8] and CMDPSOFS [9] are representatives of EMO that utilize local search. Local search has been investigated to further improve the quality of individuals/solutions. These algorithms with local search rely on traditional mechanisms such as SFS, SBS. These mechanisms do not account for feature interaction. Feature interaction methods detect important information of a feature's relative importance when

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

in cooperation with other features in a feature subset. This information cannot be detected with traditional mechanisms that only detect the importance of a feature by isolating and assessing features individually. Relief is a simple and computationally inexpensive method of performing feature ranking while still considering feature interactions [10]. Thus, we utilize Relief to develop a local search for MOEA/D-based feature selection. In addition, since MOEA/D segments the population into sub-regions, mutation often causes solutions to violate constraints. Such infeasible solutions need to be repaired. Repair mechanisms for recent MOEA/D-based algorithms [7] simply add/remove features based on the classification accuracy of individual features. This method of repairing does not account for feature interaction, and has been acknowledged to be a potential downside in discarding features useful only in combination with other features [7]. This work also proposes a novel repair mechanism based on Relief, which accounts for feature interaction.

A. Goals

The goal is to propose a novel decomposition-based multi-objective algorithm for feature selection, which can generate diverse fronts consisting of feature subsets with various numbers of features and better classification performance than using all features. The goal is achieved by developing a local search technique and solution repair mechanisms based on the Relief score to improve the quality of the evolved features. Particularly, we will investigate:

1) whether the proposed algorithm can generate feature subsets which achieve better classification performance than using all features,
2) whether the proposed algorithm can generate more diverse fronts than NSGAII and SPEA2 which are two representatives of non-dominance based EMO,
3) whether the proposed algorithm can evolve the feature subsets around the knee-point region which have better classification performance than the subsets evolved by MOEA/D-DYN, a state-of-the-art MOEA/D-based feature selection algorithm.

II. Background

A. Multi-objective optimisation

In a multi-objective problem, there are at least two objectives that are being optimised simultaneously. Measuring the quality of a solution in a multi-objective problem is not as simple as in a single-objective problem due to the conflict between the objectives. A way of comparing two solutions is by strictly comparing each one of their corresponding objectives together. For a minimisation task, a solution $\mathbf{y}$ dominates a solution $\mathbf{z}$ if:

$$\forall i : f_i(\mathbf{y}) \leq f_i(\mathbf{z}) \wedge \exists j : f_j(\mathbf{y}) < f_j(\mathbf{z}) \tag{1}$$

where f_i is the i^{th} objective. If a solution is not dominated by any other solutions, it is called a Pareto optimal solution. A set of Pareto optimal solutions form a Pareto front. The task of a multi-objective optimisation algorithm is to closely approximate the Pareto front by a diverse set of non-dominated solutions. Feature selection is a multi-objective problem since its two main objectives are usually in conflict. Thus, the task of a multi-objective feature selection algorithm is to evolve a set of feature subsets with various trade-offs between the number of selected features and the classification performance.

B. Relief

Relief is a well-known feature selection measure quantifying feature interaction, which is presented in the works of Urbanowicz et al. [10]. Relief scores for features are calculated on the basis of a weight vector, with each weight corresponding to a feature. Relief is similar to the KNN algorithm in its use of a distance measure (Manhattan, or Euclidean) to calculate neighbouring instances. Weights are decreased by the average difference of a sampled instance R and k other instances of the same class, and increased by the average difference of k instances not belonging to the same class. Intuitively this allows measuring how useful a feature is in predicting an instance class by simultaneously decreasing its score by how much intra-class feature variation exists and increasing the score by how much inter-class feature variation exists. Ideally, a relevant feature should have low intra-class variation and vice-versa. Feature interaction is accounted for by finding neighbours based on an aggregate distance measure such as Euclidean distance which considers all features together. We utilize a slightly different version of Relief (ReliefF) [11], which has a different update equation capable of handling multi-class misses.

C. Related work on multi-objective feature selection (MOFS)

Xue et al. [9] proposed a PSO based MOFS algorithm utilizing ideas such as crowding, mutation, and dominance (CMD) [12]. CMDPSOFS randomly and equally divides the whole swarm into three groups; no mutation, uniform mutation, and non-uniform mutation. Uniform mutation allows particles to maintain global search ability. Non-uniform mutation variability decreases over time to retain exploitative ability - that is in other words its local search ability.

Nguyen et al. [8] proposed a new PSO based MOFS algorithm. ISRPSOFS introduced a novel way of performing local search on an archive set, by inserting, swapping and removing up to one feature from an imminent archive set solution. By utilizing all three operations in conjunction with high quality solutions in the archive set, local search adds pressure toward higher quality feature subsets present in the archive during the evolutionary process. However, ISRPSOFS is computationally expensive due to the repeated evaluation of feature subsets that have been altered by the local search operators.

Developed by Zhang and Li [6], MOEA/D decomposes a multi-objective problem into N single-objective scalar optimization problems with a corresponding weight vector. Pareto optimal solutions are found by continual reproduction of best solutions by applying genetic operators to neighbouring sub-problems. A significant strength of MOEA/D includes a natural preservation of diversity, as each sub-problem relies on neighbouring sub-problems for reproduction.

Nguyen et al. [7] proposed an alternative decomposition method to standard MOEA/D, with Static (STAT) and Dynamic (DYN) reference points. MOEA/D-STAT (and DYN) do not rely on weight vectors the same way as MOEA/D does. For feature selection tasks, MOEA/D-(STAT/DYN) decomposes a problem along the feature ratio axis $fRatio = \frac{selectedfeatures}{totalfeatures}$. Whether or not a feature is selected is a discrete binary variable, therefore the $fRatio$ axis is on discrete space. The algorithm allocates a predefined number of reference points as boundaries for sub-regions. Allocated reference points represent ideal solutions with 0% error rate, using exactly $fRatio$ of the total features. Reference points can either be static or dynamic. A predefined number of static reference points are allocated at even intervals across an axis. Dynamic points are initially allocated evenly, and are continually adjusted at boundary iterations. MOEA/D-STAT only utilizes static reference points, whilst MOEA/D-DYN utilizes both static and dynamic reference points. The dynamic approach focuses on delivering computation resources to areas of the search space that are conflicting. Conflicting regions contain solutions that possess tradeoffs between objectives, this is typical of solutions that are pareto-optimal. It has also been shown that not all sub-problems require the same computation resources [13]. MOEA/D-DYN has been shown to achieve better results than MOEA/D-STAT [7], hence its use as the backbone of this work. One limitation of MOEA/D-DYN is its low-quality feature subsets around the knee-point region. Such limitation can be addressed by using local search to further improve the classification performance of the feature subsets.

Previous work with local search in MOFS such as CMDPSOFS [9] performs local search via mutation. Mutation based local search makes greedy choices on random changes caused by flipping a random subset of bits in the bit vector representation of a solutions features. This method of local search is uninformed, and not intelligent enough to identify strong feature interactions. ISRPSOFS [8] introduced an intelligent way to perform local search by ranking features on individual accuracy. The three local search operators insert, swap, and remove in ISRPSOFS utilized this ranking to determine which of the features get added/swapped or removed. Ranking on single feature classification accuracy is not sensitive to feature interaction. In addition, the single feature classification accuracy method of ranking is implemented as a baseline ranking mechanism for repairing in MOEA/D-DYN, and is acknowledged to potentially disregard feature interaction [7]. MOEA/D-DYN has been demonstrated to possess stronger diversity preservation in comparison to traditional algorithms like NSGAII, MOEA/D, and SPEA2. However, the algorithm struggles to produce better knee-points than SPEA2 on a number of datasets. MOEA/D-DYN provides a strong baseline to test local search, and has significant potential to be improved by both the addition of feature interaction based local search and an enhanced feature interaction based repair mechanism. Due to the page limit, more related work on multi-objective feature selection and evolutionary computation for feature selection can be found in works such as [9], [14], [15].

III. Proposed Algorithm

In this section, we begin by providing a brief overview of representation and the objective functions. We then discuss how we introduce local search into our baseline MOEA/D-DYN algorithm with motivations and key differences between each of the different local search operators. Next we illustrate the role of ReliefF to calculate feature scores, and how we use those scores to perform local search using a roulette wheel selection method. The final step is to evaluate and repair solutions if necessary. We then demonstrate our improved repair mechanism. Finally, an overall view of the MOEA/D-DRLS (Dynamic ReliefF Local Search) algorithm is presented.

A. Representation and objective function

Solutions for the MOEA/D-DRLS algorithm are represented as continuous vectors with values ranging between 0 and 1. Each element of the vector corresponds to a feature of the dataset. If an element is greater than a threshold then its corresponding feature is selected. Otherwise, the corresponding feature is discarded. A threshold value of 0.6 is used as per literature [7].

In a multi-objective feature selection problem, we have two main objectives that we aim to minimise simultaneously, the classification error rate ($eRate$), and the ratio of selected features ($fRatio$). In order to evaluate the quality of a candidate solution S in each sub-problem, we develop a fitness function that evaluates both objectives

$$fitness_S = eRate_S + 100 * max(|S| - n_{ref}, 0) + \alpha * fRatio_S \quad (2)$$

where $|S|$ is the number of selected features, and n_{ref} is our upper boundary for the sub-region search space. Our upper boundary $n_{ref} = refRatio * n$, is the number of features that the reference point at $refRatio$ along the $fRatio$ axis has selected out of n total features.

The main objective to minimise is $eRate_S$, which corresponds to the first term in the fitness function. The second term is the penalty for feature subsets larger than the allocated sub-regions n_{ref}. A max function is used to choose between 0 and $|S| - n_{ref}$. In the event that $|S| - n_{ref}$ is above 0, the solution is in violation of the sub-region's size, and has a penalty value of 100 times each feature above n_{ref}. The last term is the $fRatio_S$ with variable parameter α. When α is set to 1, both $eRate$ and $fRatio$ become equally important in the fitness function. Lower values of α correspond to an increase in importance of minimising the $eRate$ over minimising the $fRatio$.

B. Local search

We propose three local search operators which are insert, remove, and swap operators. The aim of the three operators is to search for neighboring feature subsets that have better classification performance than the current feature subset. Given a feature subset, the three operators work as follows.

Algorithm 1 : Insert operator

```
procedure INSERT(Subset S)
    picks a feature that is not selected in S;
    set the corresponding element of the picked feature to threshold × 1.1;
```

Algorithm 2 : Remove operator

```
procedure REMOVE(Subset S)
    picks a feature that is selected in S;
    set the corresponding element of the picked feature to threshold (i.e. remove the picked feature from S);
```

- Insert operator (Algorithm 1) picks a feature among all unselected features to add to the feature subset. The feature is added by setting the corresponding representation element to (1.1 × threshold).
- Remove operator (Algorithm 2) picks a feature among all selected features to be removed from the feature subset. The feature is removed by setting the corresponding representation element to threshold.
- Swap operator (Algorithm 3) picks an unselected feature which then replaces a selected feature from the feature subset.

For all three operators, the newly generated feature subset replaces the current feature subset if it achieves better classification performance. It can be seen that the three operators focus on improving the classification performance while making little or no change in terms of the subset size. It is expected that the three local searches can further improve the classification performance of the feature subsets in the evolved front, while maintaining the front's diversity. However, since MOEA/D-DYN is a decomposition based algorithm, each local search operator has unique implications.

1) Solutions in regions with lower n_{ref} may not benefit from the insert operator. Even if the insert operator introduces a high quality feature with interaction effects, the solution can be rejected due to the penalty term in the fitness function and will need repair if the resulting subset size is greater than n_{ref}. Solutions in regions with higher n_{ref} should benefit the most from the insert operator. In the event that a solution in region n_{ref_i} (that is identical to a solution in region $n_{ref_{i-1}}$) receives a new feature that would place its $fRatio$ above $n_{ref_{i-1}}$; an improvement of the fitness value and therefore acceptance of the new solution will improve the diversity of solutions in that sub-region.
2) Regions with lower n_{ref} on the other hand may benefit more from the remove operator. If the removal of a feature improves the fitness value, we can introduce solutions with lower $fRatio$ and higher classification accuracy into the smaller sub-regions of the population. However, solutions in the higher sub-regions can continually have their features removed, which might drastically reduce diversity among higher $fRatio$ sub-regions as duplicate solutions begin to emerge.
3) Finally, swap is the most balanced operator. Solutions subsets have the same number of features before and after the operation. The swap operator does not provide any unique benefit to diversity in larger $fRatio$ sub-regions, and does not add pressure toward lower $fRatio$ subsets. The fitness function in this case will only benefit from a reduction in the $eRate$, as the last two terms regarding $fRatio$ are unchanged.

Algorithm 3 : Swap operator

```
procedure SWAP(Subset S)
    picks a feature f_add that is not selected in S;
    picks a feature f_remove that is selected in S;
    set the corresponding element of f_add to threshold × 1.1;
    set the corresponding element of f_remove to threshold;
```

The local search is performed after the mutation and crossover operations in MOEA/D-DYN. With the three local search operators, we introduce three corresponding algorithms.

The question is how to select the feature to insert, remove, or swap. In this work, we propose to use the Relief score since it is efficient and considers feature interaction. The usage of ReliefF is the main difference between our local search and the one proposed in ISRPSO which utilises the accuracy of each individual feature, i.e., feature interaction is ignored. The following subsection shows how to use Relief to select features for the three operators.

C. Selection based on ReliefF scores

ReliefF is utilized as an indicator of a features interaction quality. These scores are precalculated before the main algorithm once, and remain the same during the entire evolutionary process. We have developed a stochastic selection mechanism that behaves similarly to roulette wheel selection. This helps in preventing stagnation at local optima due to repeated identical selection of the same feature indices over many iterations. Selection in each local search operator utilizes the normalized weight vector. Selection favouring features with worse scores utilizes an inverted normalized weight vector. Large ReliefF scores with smaller inverse values will have lower probabilities of selection and vice versa. A range array is constructed of the cumulative sum of the normalized weight scores, where $\sum_{i=1}^{n} w_i = 1$. Each element of this range array denotes boundaries for our roulette wheel. A random real number between 0 and 1 is chosen, denoting the area on the roulette wheel that the selection mechanism chooses. Since the range array is the normalized cumulative sum, all values in that array range from 0 to 1 in ascending order. The roulette wheel is illustrated in Fig. 1, and its corresponding range array is shown in Fig. 2. The index position of the last value in the range array that the random number is equal to or less than is the position of the feature that will be selected for use in either insert, swap, or remove operators.

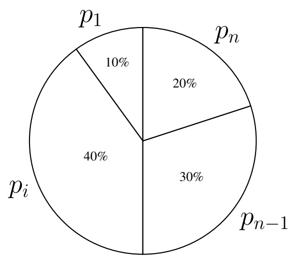

Fig. 1. Range roulette wheel with $n = 4$

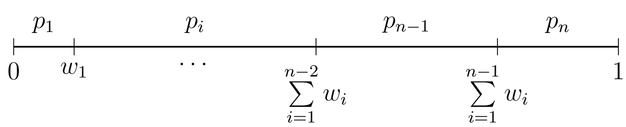

Fig. 2. Normalized weight range array with $n = 4$

D. ReliefF score based repair mechanism

In addition to local search, we propose an enhanced repair mechanism based on the ReliefF ranking. As the population is distributed among all sub-regions, areas with lower $fRatio$ will contain simpler solutions with fewer features, and vice versa. n_{ref} denotes the upper boundary $fRatio$ of that sub-region. Solutions in that sub-region must have an $fRatio$ below the corresponding n_{ref}. However, regions do not have a lower boundary. This means that a sub-region n_{ref_i} can include solutions also in $n_{ref_{i-1}}$. This prevents the unlikely event that higher n_{ref} regions do not contribute any solutions to the final pareto front. If a solution has a larger $fRatio$ than the n_{ref} of its sub-region, repairing will need to be done. Algorithm 4 illustrates the use of interaction score ranking to conduct repair.

Algorithm 4 Repair based on ReliefF

1: **procedure** REDUCESIZE(Subset S, n_{ref})
2: *while* $|S| > n_{ref}$:
3: pick a feature in S based on ReliefF scores;
4: set the corresponding element of the picked feature to threshold;

E. Overall algorithm

Fig. 3 illustrates the baseline MOEA/D-DYN algorithm with the MOEA/D-DRLS extensions in orange. As this is an extension of MOEA/D-DYN, the local search operation is performed just after the standard GA operators for each sub-problem. ReliefF ranking is performed before the evolutionary process. The repair function has been changed to utilize the ReliefF ranking. In addition, insert, swap, remove, and repair utilizes the probability based roulette wheel selection when adding and removing features.

IV. Experiment Design

A. Benchmark techniques

We examine our proposed algorithm on 12 UCI datasets, and 5 gene-expression datasets [7] with various numbers of features, classes, and instances. The datasets are selected with an expectation to be good representatives for real-world problems. Details of the 17 datasets are shown in Table I.

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
SRBCT	2308	4	83
Leukemia1	5327	3	72
DLBCL	5469	2	77
Brain1	5920	5	90
Leukemia	7129	2	72

In total, the 17 benchmark datasets provide a good indication of overall performance for small, medium and large-scale feature selection tasks. There are multiple classes for many of these algorithms, hence the requirement to utilize the multiclass version of Relief (ReliefF). The proposed algorithm MOEA/D-DRLS will be compared with three benchmark algorithms: SPEA2, NSGAII, and MOEA/D-DYN. Each of the three benchmark algorithms MOEA/D-DYN, SPEA2 and NSGAII will have a total of 30 independent runs. MOEA/D-DRLS will also have a total of 30 independant runs for each of the three local search operators. All datasets are divided into a training and testing set, with a 70-30 split. Features are selected based only on the training set to avoid feature selection bias [16].

During training we utilize KNN with 10-fold cross validation on the training set to evaluate feature subsets. These settings largely remain the same with previous studies [7], and are common among similar feature selection tasks [17], [5]. To conduct comparison, and calculate the two indicators, we require an approximate true Pareto front. We obtain this front by taking the union of all non-dominated solutions across all runs for all algorithms. We apply a Friedman test and an adhoc Nemenyi multitest in order to determine significant differences between benchmark algorithms and the proposed methods at significance level of 0.05.

B. Parameter settings

The settings of NSGAII and SPEA2 follow their original papers. Parameter settings for MOEA/D-DYN as both a benchmark and base for MOEA/D-DRLS are identical to the

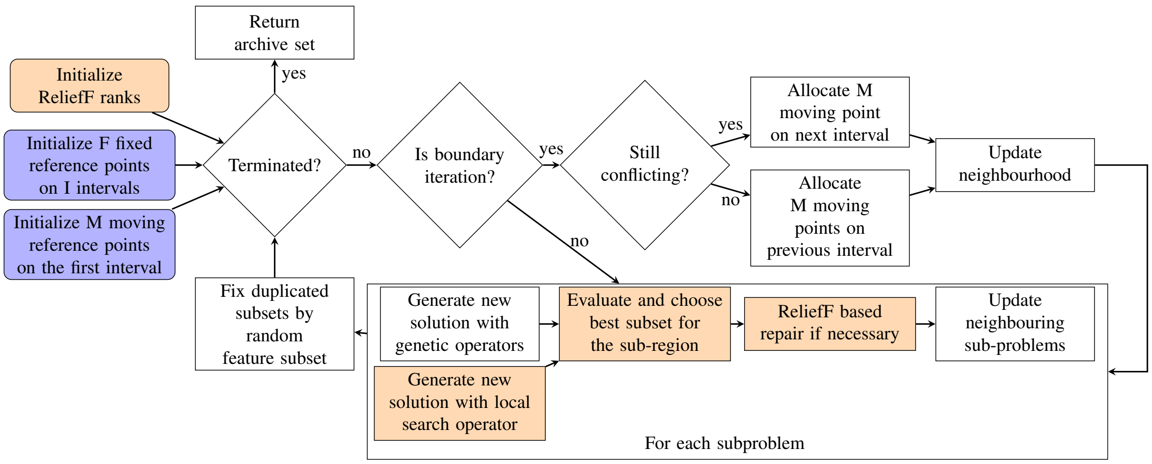

Fig. 3. Overall algorithm for MOEA/D-DRLS

TABLE II
HYPERVOLUME VALUES FOR **TEST** FRONTS

Dataset	NSGAII	SPEA2	Insert	Swap	Remove
Wine	0.765±0.052 (↓,↓,↓)	0.927±0.031 (↓,↓,↓)	**0.952±0.000**	0.951±0.002	**0.952±0.000**
Australian	0.771±0.070 (↓,↓,↓)	0.771±0.069 (↓,↓,↓)	**0.828±0.003**	**0.828±0.006**	0.827±0.005
Vehicle	0.815±0.019 (↓,↓,↓)	0.809±0.025 (↓,↓,↓)	**0.831±0.002**	**0.831±0.003**	0.830±0.002
German	0.681±0.022 (○,↓,↓)	0.680±0.021 (↓,↓,↓)	**0.694±0.006**	**0.694±0.007**	0.693±0.006
WBCD	0.923±0.019 (↓,↓,↓)	0.924±0.017 (↓,↓,↓)	**0.934±0.000**	**0.934±0.000**	**0.934±0.000**
Sonar	0.771±0.030 (↓,↓,↓)	0.786±0.024 (↓,↓,↓)	0.804±0.023	0.807±0.017	**0.808±0.021**
Hillvalley	0.596±0.015 (○,↓,○)	0.602±0.010 (○,↓,↓)	0.603±0.009	**0.608±0.010**	0.605±0.007
Musk1	0.842±0.018 (↓,↓,↓)	0.856±0.014 (↓,↓,↓)	**0.874±0.012**	0.868±0.016	0.873±0.012
Arrhythmia	0.933±0.007 (↓,↓,↓)	0.944±0.004 (↓,↓,↓)	**0.957±0.002**	0.956±0.002	0.956±0.002
Madelon	0.861±0.010 (↓,↓,↓)	0.869±0.010 (↓,↓,↓)	**0.891±0.005**	0.890±0.004	**0.891±0.003**
Isolet5	0.919±0.007 (↓,↓,↓)	0.939±0.009 (↓,↓,↓)	**0.990±0.000**	0.989±0.001	**0.990±0.001**
Multiplefeatures	0.946±0.008 (↓,↓,↓)	0.956±0.006 (↓,↓,↓)	**0.991±0.001**	**0.991±0.001**	**0.991±0.001**
SRBCT	0.927±0.014 (↓,↓,↓)	0.927±0.016 (↓,↓,↓)	**0.982±0.013**	0.981±0.013	0.981±0.014
Leukemia1	0.847±0.022 (↓,↓,↓)	0.844±0.020 (↓,↓,↓)	**0.975±0.020**	0.967±0.020	0.965±0.026
DLBCL	0.782±0.048 (↓,↓,↓)	0.844±0.025 (↓,↓,↓)	0.929±0.041	**0.939±0.044**	0.910±0.064
Brain1	0.789±0.011 (↓,↓,↓)	0.789±0.018 (↓,↓,↓)	0.932±0.010	**0.933±0.011**	0.929±0.012
Leukemia	0.787±0.041 (↓,↓,↓)	0.822±0.028 (↓,↓,↓)	0.966±0.028	0.965±0.030	**0.974±0.023**

study performed by [7], and are suggested by experiments conducted in [18]. The settings for MOEA/D-DYN involve the number of neighbors T, which is set to $R/10$. Differntial evolution's crossover is used with rate 0.6 and scaling factor F 0.7. Polynomial mutation $1/n$ is also used. The probability of selecting parents from neighboring sub-problems σ is 0.85. Since we use the dynamic strategy for MOEA/D, the number of moving reference points M is set to $0.4 * R$. The number of intervals I is set to 9 if the number of features is less than 20, otherwise 4 as in MOEA/D-DYN [7]. α in Fig. 2 is set to 0.001 to allow preference toward minimising classification error rate. The population size is equal to the number of features if there are less than 200 features, otherwise equal to 200. The maximum iterations is 200. These values have been examined and suggested in [7]. A threshold value of 0.6 is again used to determine the cutoff point for feature selection/deselection [7]. The number of nearest neighbors in KNN is set to 5, this value for KNN helps avoid noisy instances while maintaining its efficiency

The two user defined parameters for ReliefF are n and k. The original Relief analysis [19] suggests that an insufficient number of training instances (n) can fool the Relief algorithm. Therefore, we have eliminated this parameter by enabling the algorithm to cycle through all training instances. Empirical analysis on ReliefF also suggests setting k to 10 [10].

V. Results and Discussion

A. Comparison with using all features

Fig. 4 and Fig. 5 show the median fronts obtained by MOEA/D-DYN and MOEA/D-DRLS with three local search operators. Median fronts for each dataset are obtained using the median hypervolume values across all 30 runs. The fronts are graphed using $fRatio$ on the x-axis, and $eRate$ on the y-axis. The "Full feature set" in the figure legend shows the classification error rate when using all features. Due to the space limit, 4 fronts on 4 datasets: German, Musk1, Madelon, and DLBCL are shown as representatives for different numbers of features. As can be seen from the figure most feature subsets evolved by MOEA/D-DRLS have lower classification error rate than using all features. Except for the German dataset with a small number of features, subsets evolved by MOEA/D-DRLS select at most 50% number of features. The experimental results show that the proposed local search can help MOEA/D to evolve feature subsets with better classification performance than using all features.

B. Comparison with dominance-based algorithms

Table II show the comparison between three MOEA/D-DRLS algorithms with two dominance-based algorithms (NSGAII and SPEA2) on the test set. The following symbols $\uparrow$, $\downarrow$, $\circ$ are used to indicate whether the dominance-based algorithms are significantly better than, worse than or have no significant difference to each of the proposed local search operators insert, swap, and remove. As can be seen from the tables, on most datasets, all the three local search operators achieve significantly better hypervolume values than NSGAII and SPEA2. The main reason is that MOEA/D-DRLS can evolve much more diverse fronts than that of NSGAII and SPEA2. Thus, the local search operators do not reduce the population diversity, which is the main advantage of MOEA/D over dominance-based algorithms.

C. Comparison with MOEA/D-DYN

Fig. 4 and 5 show the effect of the proposed local search operators by comparing the fronts evolved by MOEA/D-DYN and the fronts evolved by MOEA/D-DRLS with three local search operators. As can be seen from the figures, except for DLBCL, the local search operators can generate feature subsets with lower classification error rates than that of MOEA/D-DYN. The most significant improvement is at the knee-point region. Only on DLBCL, MOEA/D-DYN is able to achieve better feature subsets than MOEA/D-DRLS. The possible reason is that DLBCL has a much larger number of features than the other three datasets, so its search space is also much larger. Applying the local search on the DLBCL dataset can significantly reduce the global search ability of MOEA/D. Thus, the search space is not well explored, which results in a poor front.

In comparison between the three local search operators, the remove operator seems to have the most significant effect while the insert operators seem to have the least significant effect. The main reason is that the insert operator increases the feature subset size, which can make the feature subset infeasible since its size is larger than its corresponding n_{ref}. Thus, it does not matter how good the feature subset is, the feature subset will be discarded due to its constraint violation. In contrast, the remove operator reduces the number of features, so it will generate a feasible feature subset, not only for its corresponding n_{ref} but also for the smaller n_{ref}. Therefore, it is more likely that the subset generated by the remove operator is accepted.

D. Further analysis of the evolutionary process

Fig. 6 displays the evolutionary process on the Multiple-features dataset. All non-duplicated solutions at each generation during the training process have their $fRatio$ and 10-fold $eRate$ recorded. The evolutionary process graphs these solutions with $fRatio$ on the x-axis and $eRatio$ on the y-axis. For this run, the remove operator achieved a lower knee-point than MOEA/D-DYN, whilst maintaining diversity of the front. We can observe a decrease in diversity for MOEA/D-DYN at iteration 60. Remove regains diversity at iteration 100, and maintains it until the final generation.

VI. Conclusions and Future Work

In this paper, we present a new feature interaction based local search for MOEA/D-DYN to form a novel multi-objective feature selection approach called MOEA/D-DRLS. MOEA/D-DRLS performs local search by calculating a weight vector of feature scores using the ReliefF algorithm. This weight vector is then used to assess a features quality for selection in three local search operators: insert, swap, and remove, as well as in repair mechanisms. The experimental results show that the proposed algorithm achieves better hypervolume values than two dominance-based algorithms, NSGAII and SPEA2. Furthermore, the local searches can improve the classification performance of the feature subsets around the knee-point region, which is the main limitation of MOEA/D-DYN.

A limitation of this study is that the algorithm used to rank features based on interaction, ReliefF, is known to struggle with ranking comparatively redundant features on high dimensional datasets. In addition, local search during the evolutionary process causes solutions to get stuck at local optima when the search space is extremely large (as on the DLBCL dataset). Further work can look at regularization techniques for local search, such as probability based local search.

References

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

[2] T. Marill and D. Green, "On the effectiveness of receptors in recognition systems," *IEEE Transactions on Information Theory*, vol. 9, no. 1, pp. 11–17, January 1963.

[3] C. Emmanouilidis, A. Hunter, and J. MacIntyre, "A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator," in *Proceedings of the 2000 Congress on Evolutionary Computation.*, vol. 1. IEEE, 2000, pp. 309–316.

[4] C. C. Coello, "Evolutionary multi-objective optimization: a historical view of the field," *IEEE Computational Intelligence Magazine*, vol. 1, no. 1, pp. 28–36, 2006.

[5] H. B. Nguyen, B. Xue, H. Ishibuchi, P. Andreae, and M. Zhang, "Multiple reference points moea/d for feature selection," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 157–158.

[6] 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, Dec 2007.

[7] B. H. Nguyen, B. Xue, P. Andreae, H. Ishibuchi, and M. Zhang, "Multiple reference points based decomposition for multi-objective feature selection in classification: Static and dynamic mechanisms," *IEEE Transactions on Evolutionary Computation*, pp. 1–1, 2019.

[8] 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, 03 2016.

[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, Dec 2013.

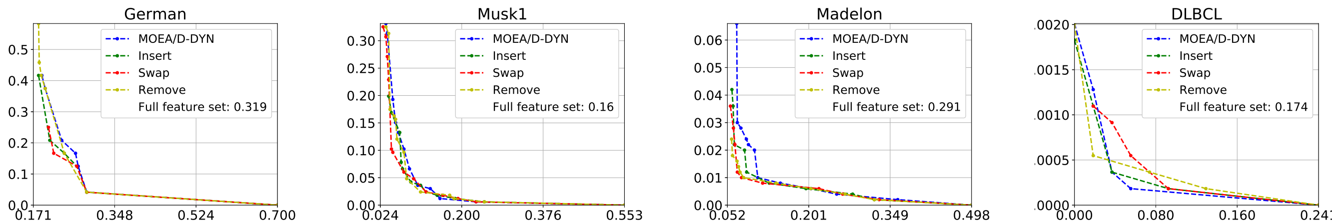

Fig. 4. Median fronts on the training sets (the horizontal and vertical axes represent the selected feature ratio and the classification rate, respectively).

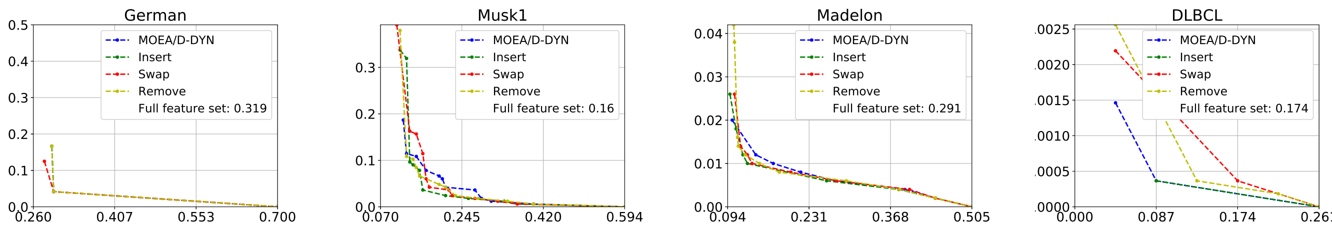

Fig. 5. Median fronts on the test sets.

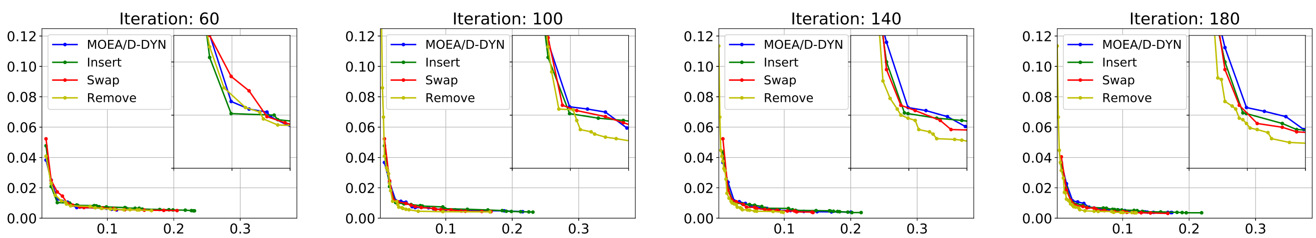

Fig. 6. Evolutionary process on the Multiplefeatures dataset.

[10] R. J. Urbanowicz, M. Meeker, W. L. Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *Journal of Biomedical Informatics*, vol. 85, pp. 189 – 203, 2018.

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

[12] M. R. Sierra and C. A. C. Coello, "Improving pso-based multi-objective optimization using crowding, mutation and epsilon-dominance," in *EMO*, 2005.

[13] Q. Zhang and W. Liu, "The performance of a new version of moea/d on cec09 unconstrained mop test instances," 06 2009, pp. 203 – 208.

[14] B. Xue, M. Zhang, W. Browne, and X. Yao, "A survey on evolutionary computation approaches to feature selection," *IEEE Transactions on Evolutionary Computation*, vol. 20, pp. 1–1, 01 2015.

[15] X. Li, "A non-dominated sorting particle swarm optimizer for multi-objective optimization," in *Genetic and Evolutionary Computation — GECCO 2003*, E. Cantú-Paz, J. A. Foster, K. Deb, L. D. Davis, R. Roy, U.-M. O'Reilly, H.-G. Beyer, R. Standish, G. Kendall, S. Wilson, M. Harman, J. Wegener, D. Dasgupta, M. A. Potter, A. C. Schultz, K. A. Dowsland, N. Jonoska, and J. Miller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 37–48.

[16] B. Tran, B. Xue, M. Zhang, and S. Nguyen, "Investigation on particle swarm optimisation for feature selection on high-dimensional data: local search and selection bias," *Connection Science*, pp. 1–25, 05 2016.

[17] B. Xue, M. Zhang, and W. N. Browne, "Multi-objective particle swarm optimisation (pso) for feature selection," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 81–88.

[18] A. J. Nebro, J. J. Durillo, and M. Vergne, "Redesigning the jmetal multi-objective optimization framework," in *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO Companion '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 1093–1100.

[19] K. Kira and L. A. Rendell, "The feature selection problem: Traditional methods and a new algorithm," in *Proceedings of the Tenth National Conference on Artificial Intelligence*, ser. AAAI'92. AAAI Press, 1992, pp. 129–134.