Multiobjective Optimization of Classifiers by Means of 3D Convex-Hull-Based Evolutionary Algorithms

Jiaqi Zhao[a,*], Vitor Basto Fernandes[b], Licheng Jiao[a], Iryna Yevseyeva[c], Asep Maulana[d], Rui Li[e], Thomas Bäck[d], Ke Tang[f], Michael T. M. Emmerich[d]

[a]*Key Laboratory of Intelligent Perception and Image Understanding of the Ministry of Education, International Research Center for Intelligent Perception and Computation, Joint International Research Laboratory of Intelligent Perception and Computation, Xidian University, Xi'an Shaanxi Province 710071, China*

[b]*School of Technology and Management, Computer Science and Communications Research Centre, Polytechnic Institute of Leiria, 2411-901 Leiria, Portugal*

[c]*Faculty of Technology, De Montfort University, Gateway House 5.33, The Gateway, LE1 9BH Leicester, UK*

[d]*Multicriteria Optimization, Design, and Analytics Group, LIACS, Leiden University, Niels Bohrweg 1, 2333-CA Leiden, The Netherlands*

[e]*Microsoft Research Asia, Beijing 100190, China*

[f]*Nature Inspired Computation and Applications Laboratory (NICAL), the USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI), School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, Anhui 230027, China.*

Abstract

The receiver operating characteristic (ROC) and detection error tradeoff (DET) curves are frequently used in the machine learning community to analyze the performance of binary classifiers. Recently, the convex-hull-based multiobjective genetic programming algorithm was proposed and successfully applied to maximize the convex hull area for binary classification problems by minimizing false positive rate and maximizing true positive rate at the same time using indicator-based evolutionary algorithms. The area under the ROC curve was used for the performance assessment and to guide the search. Here we extend this research and propose two major advancements: Firstly we formulate the algorithm in detection error tradeoff space, minimizing false positives and false negatives, with the advantage that misclassification cost tradeoff can be assessed directly. Secondly, we add complexity as an objective function, which gives rise to a 3D objective space (as opposed to a 2D previous ROC space). A domain specific performance indicator for 3D Pareto front approximations, the volume above DET surface, is introduced, and used to guide the indicator-based evolutionary algorithm to find optimal approximation sets. We assess the performance of the new algorithm on designed theoretical problems with different geometries of Pareto fronts and DET surfaces, and two application-oriented benchmarks: (1) Designing spam filters with low numbers of false rejects, false accepts, and low computational cost using rule ensembles, and (2) finding sparse neural networks for binary classification of test data from the UCI machine learning benchmark. The results show a high performance of the new algorithm as compared to conventional methods for multicriteria optimization.

Keywords: Convex hull, classification, evolutionary multiobjective optimization, parsimony,
ROC analysis, anti-spam filters.

1. Introduction

Classification is one of the most common problems in machine learning. The task of classification is to assign instances in a dataset to target classes based on previously trained classifiers. The ROC (Receiver Operating Characteristic) curve is a technique for visualizing, organizing and selecting binary classifiers based on their performance [24]. ROC curves are typically used to evaluate and compare the performance of classifiers and they also have properties that make them especially useful for domains with skewed class distributions and different classes of problems that assign costs to misclassification. Originating from the field of object classification in radar images, ROC analysis has become increasingly important in many other areas with cost sensitive classification [15] and/or unbalanced data distribution [49], such as medical decision making [50], signal detection [20] and diagnostic systems [52]. As opposed to ROC curves, which show the tradeoff between true positive rate and false positive rate, DET (Detection Error Tradeoff) curves [41] show tradeoffs between false positive and false negative error rates. With DET it is easier to visualize the tradeoff between misclassification cost for binary classifiers than with ROC curves.

More recently, research has drawn attention to ROC convex hull (ROCCH) analysis that covers potentially optimal points for a given set of classifiers [24]. ROCCH makes use of the finding that two hard classifiers can be combined into a classifier that has characteristics in ROC space that correspond to linear combinations of the characteristics of single classifiers and thus, when

[*] Corresponding author.
Email address: jiaqizhao88@126.com (Jiaqi Zhao)

searching for an approximation to the Pareto front, these linear combinations do not have to be explicitly represented in ROC space. A performance indicator for sets of hard binary classifiers that is compliant with the improvement of ROCCH is the area under the convex hull (AUC). And likewise the area above the DET convex hull can serve as an indicator of how well a Pareto front has been approximated. It measures the area attained by the current Pareto front approximation in DET space.

Some evolutionary multiobjective optimization algorithms (EMOAs) [31, 57, 32, 54, 58] have been applied to machine learning [33, 2] and image processing areas [36, 39]. One of the first algorithms where EMOAs were used for ROC optimization was proposed in [35]. Here a niched Pareto multiobjective genetic algorithm was used for classifier optimization by optimizing biobjective ROC curve. The generalization improvement in multiobjective learning was discussed in [27], where the generation of binary neural network classifiers based on the ROC analysis using an evolutionary multiobjective optimization algorithm was presented. It showed that the generalization ability can be more efficiently improved with a multiobjective framework than within a single objective one. ROC for multiclass classification was analyzed in [22], where a multiobjective optimization algorithm was used for classifiers training based on multiclass ROC analysis. The ROC front concept was introduced as an alternative to the ROC curve representation in [13], and the strategy was applied to the selection of classifiers in a pool using a multiobjective optimization algorithm. Moreover, the maximization of the performance of ROC representations with respect to this indicator has been subject to a recent study by Wang et al. [55], who showed that the proposed algorithm, convex-hull-based multiobjective genetic programming algorithm (CH-MOGP), is capable of showing a strong performance for improving ROCCH with respect to AUC as compared to using state-of-the-art EMOAs for the same task, such as NSGA-II (Nondominated Sorting Genetic Algorithm II) [16], GDE3 (the third evolution step of Generalized Differential Evolution) [34], SPEA2 (Strength Pareto Evolutionary Algorithm 2) [63], MOEA/D (Multiobjective Evolutionary Algorithm based on Decomposition) [61], and SMS-EMOA (multiobjective selection based on dominated hypervolume) [7].

So far algorithms that seek to maximize ROCCH performance have only focused on the problem of optimizing binary classifiers with respect to two criteria, i.e., minimization of false positive rate (*fpr*) and maximization of true positive rate (*tpr*). There is however an increasing interest in extending ROCCH performance analysis to more than two criteria. In this research we consider the complexity as an additional objective. The objective here is to find models with maximum simplicity (parsimony) or minimum computational costs. For rule-based systems, it can be described as the number of rules defining a classifier in proportion to the number of all possible rules. As it is easier to see the tradeoff between misclassification costs (i.e., *fpr* and *fnr*) when using DET space than when using ROC space, we use DET curve to describe the performance of binary classifiers.

In the past, convex-hull-based selection operators were employed in EMOA to maintain a well-distributed set or make the non-dominated sorting more effective (cf. [30, 43]). In [14] a multiobjective evolutionary algorithm based on the properties of the convex hulls defined in the ROC space was proposed. It was applied to determine a set of fuzzy rule-based binary classifiers with different tradeoffs between false positive rate (fpr) and true positive rate (tpr). NSGA-II was used to generate an approximation of a Pareto front composed of genetic fuzzy classifiers with different tradeoffs among sensitivity, specificity, and interpretability in [17]. After projecting the overall Pareto front onto the ROC space, ROC convex hull method was used to determine the potentially optimal classifiers on the ROC plane.

In this paper, we add the complexity minimization for parsimony maximization as a third objective function and formulate the problem from the misclassification error optimization point of view by minimizing false positive and false negative error rates objectives. For this we model the problem as a triobjective optimization in augmented DET space, and we propose a 3D convex-hull-based evolutionary multiobjective algorithm (3DCH-EMOA) that takes into account domain specific properties of the 3D augmented DET space. Moreover, we analyze and assess the performance of the algorithm in different studies on, partly new, academic problems and practical applications. To analyze the capability of different algorithms to maximize convex hull volume, in a more fundamental study, a set of test problems named ZEJD (Zhao, Emmerich, Jiao, Deutz) [21] are designed and the capability of 3DCH-EMOA to capture only the convex part of a Pareto front is assessed. Besides, we include a study on spam filter design, in which the number of rules determines the complexity objective in terms of number of used rules. We also apply the proposed algorithm to deal with sparse neural networks, in which not only the classification performance but also the structure of the network optimized.

This paper is organized as follows: the related work is outlined in Section 2, and the background of augmented DET surfaces and the theory of multiobjective optimization are introduced in Section 3. We describe the framework of the 3DCH-EMOA algorithm in Section 4, and experimental results on ZEJD benchmarks test problems are described and discussed in Section 5. The description of the spam filter application and experimental results are shown in Section 6. The experimental results about multiobjective optimization of sparse neural networks are discussed in Section 7. Section 8 provides the conclusion and a discussion on the important aspects and future perspectives of this work. In addition, details of ZEJD test functions are described in Appendix A.

2. Related work

2.1. ROC and DET in classification

Both ROC and DET curves are defined by a two-by-two confusion matrix which describes the relationship between the true labels and predicted labels from a classifier. An example of a confusion matrix is shown in Table 1. There are four possible outcomes with binary classifiers in a confusion matrix. It is a true positive (TP), if a positive instance is classified as positive. We call it false negative (FN or type II error), if a positive instance is classified as negative. If a negative instance is correctly classified we call it a true negative (TN), else we call it a false positive (FP or type I error).

Table 1: A confusion matrix of binary classifiers.

		True class	
		P	N
Predicted class	P	True Positives	False Positives
	N	False Negatives	True Negatives

Let *fpr*= FP/(TN+FP) be the false positive rate, *fnr* = FN/(TP+FN) be the false negative rate and *tpr* = TP/(TP+FN) denote the true positive rate. Since no perfect classifier exists for most real-world classification problems, and *fpr* and *fnr* are conflicting with each other, DET curve is used to depict the tradeoff between them. DET graphs are two-dimensional graphs in which the *fpr* is plotted on the X-axis and *fnr* is plotted on the Y-axis. Similar to DET graphs, in ROC graphs the *tpr* is plotted on the Y-axis and *fpr* is plotted on X-axis. The DET curve can be determined from the ROC curve, as $fnr{+}tpr{=}1$.

The ROC convex hull (ROCCH) covers all the potential optimal classifiers in a given set of classifiers. The potential optimal classifiers also lie on the DET curve, the area under the curve is called DET convex hull (DCH) in this paper. The aim of ROCCH/DCH maximization is to find a group of classifiers that perform well as a set. Despite the fact that ROCCH is an important topic in classification, there is not much work focusing on how to maximize the ROCCH. A reason for this could be that this is a relatively complex task compared to approaches that assess performance of a classifier by means of a single metric. However, the additional gain in information about the tradeoff between different objectives (and the possibilities it offers for online adjustments of classifiers) should justify the development of more mature methods for ROCCH maximization and the closely related DCH maximization. The set of existing methods could be partitioned into two categories: one is ROC geometry-based machine learning methods and the other one is evolutionary multiobjective optimization methods.

ROCCH maximization problems were first described in [48]. One approach to identify portions of the ROCCH is to use iso-performance lines [24] that are translated from operating conditions of classifiers. Suitable classifiers for datasets with different skewed class distribution or misclassification costs can be chosen based on these iso-performance lines. In addition, a rule learning mechanism is described in [23] and in [25]. It combines rule sets to produce instance scores indicating the likelihood that an instance belongs to a given class, which induces decision rules in ROC space. In the above methods a straightforward way was used to analyze the geometrical properties of ROC curves to generate decision rules. However, the procedure is not effective and easily gets trapped in local optima.

In [26], a method for detecting and repairing concavities in ROC curves is studied. In that work, a point in the concavity is mirrored to a better point. In this way the original ROC curve can be transformed into a ROC curve that performs better. The Neyman-Pearson lemma is introduced in the context of ROCCH in [5], which is the theoretical basis for finding the optimal combination of classifiers to maximize the ROCCH. This method not only focuses on repairing the concavity but it also improves the ROC curve, which is different from [26]. For a given set of rules, the method can combine the rules using *and* and *or* to get the optimum rule subset efficiently. But it can not generate new rules in the global rule set. ROCCER (a rule selection algorithm based on ROC analysis) was proposed in [47]. It is reported to be less dependent on the previously induced rules.

Recently, also multiobjective optimization techniques to maximize ROCCH received attention. The ROCCH maximization problem is a special case of a multiobjective optimization problem, because the minimization of false positive rates and maximization of true positive rates can be viewed as conflicting objectives, and the parameters of a classifier can be viewed as decision variables. In [62], non-dominated decision trees were developed, which are used to support the decision on which classifier to choose. A multiobjective genetic programming approach was proposed to envelop the alternative Pareto optimal decision trees. However, it is not a general method for ROCCH maximization because it only pays attention to cost sensitive classification problems.

The Pareto front of multiobjective genetic programming is used to maximize the accuracy of each minority class with unbalanced dataset in [9]. Moreover, in [8], the technique of multiobjective optimization genetic programming is employed to evolve diverse ensembles to maximize the classification performance for unbalanced data.

Other evolutionary multiobjective optimization algorithms (EMOAs) have been combined with genetic programming to maximize ROC performance in [56]. Although they have been used in ROCCH maximization, these techniques do not consider special characteristics of ROCCH. This is done in convex hull multiobjective genetic programming (CH-MOGP), which has been proposed recently in [55]. CH-MOGP is a multiobjective indicator-based genetic programming using the area under the convex hull curve (AUC) as a performance indicator for guiding the search. It has been compared to other state-of-the-art methods and showed the best performance for binary classifiers on the UCI benchmark [37]. However, it is so far limited to biobjective optimization of error rates of binary genetic programming classifiers and it would be desirable to include additional objective functions in the analysis.

The main contributions of this paper are listed in the following. Firstly, the idea of AUC indicator is generalized to evolutionary multiobjective algorithms for classifier optimization. Secondly, we consider one more objective (classifier complexity rate) in augmented DET space for binary classifier optimization with parsimony as a third objective. Thirdly, 3DCH-EMOA is proposed for multiobjective classifier optimization.

3. Augmented DET and Multiobjective Formulation

Finding a set of optimal binary classifiers can be viewed as a biobjective problem, i.e., minimizing *fpr* and *fnr* simultaneously in DET space. Our study aims at looking at optimizing three objectives for parsimony binary classification problem. We define parsimony (to be maximized) or complexity (to be minimized) as a third objective, in addition to *fpr* and *fnr*.

3.1. Augmented DET graphs and multiobjective classifiers

In order to extend the approach to the triobjective case, recent extensions of ROC analysis to deal with multiclass problems will be discussed first. ROC curve is extended to ROC hypersurface for multiclass problem and ROC hypersurface inherits all the desirable properties of ROC curve [51]. It has been shown that a multiclass classifier with good ROC hypersurface can lead to classifiers suitable for various class distributions and misclassification costs [11]. However, due to the increase of the dimensionality of the ROC space, achieving the optimal ROC hypersurface is even more difficult than achieving the optimal ROC curve. A simpler generalization of AUC for multiclass problems, namely multiclass AUC (MAUC) was proposed in [28], and it has been widely used in recent works [38, 53]. As DET space is similar to ROC space, in this paper, we consider a quite different extension of DET curve to a higher dimensional, which is used to deal with parsimony in binary classification problems.

In our study we consider a set of training samples $S_{tr} = \{(s_i, y_i)|s_i \in \mathbb{R}^d, y_i \in \{-1, +1\}, i = 1, 2, \ldots, |S_{tr}|\}$, where y_i is the class label corresponding to a given input s_i, d is the dimensionality of sample features, and $|S_{tr}|$ is the number of instances. Note that in this work we only consider binary classification problems and we set the labels as $\{1, -1\}$, in which 1 represents the positive category and -1 represents the negative category. A classifier C can be trained with samples in S_{tr}. It can be described as an estimate of the unknown function $y = f(x)$, which is denoted by Eq.

1.

$$C : y = f(s;\theta), \;\; (s_i, y_i) \in S_{tr}, \tag{1}$$

where θ is a parameter set of the classifier C which is determined by training. After training, the classifier C should be able to predict the class label y^p for a new input sample s, or a set of class labels for a testing dataset S_{ts}, as described in Eq. 2.

$$y_j^p = f(s_j;\theta), \;\; s_j \in S_{ts}, j = 1, 2, \ldots, |S_{ts}|, \tag{2}$$

where $|S_{ts}|$ is the number of test samples in S_{ts}. In biobjective optimization model, the set of parameters can be obtained by minimizing fpr and fnr in DET space, as it is described in Eq. 3.

$$\min_{\theta\in\Omega} \mathbf{F}(\theta) = \min_{\theta\in\Omega} \mathbf{F}\Big(fpr(\theta), fnr(\theta)\Big), \tag{3}$$

where Ω is the solution space that includes all possible configurations of classifiers.

Besides *fpr* and *fnr*, we define the third objective as complexity of the classifier. In the general case we will use the term classifier complexity ratio (*ccr*) to describe it. We denoted it as Eq. 4, where O represents the complexity of classifiers. The *ccr* can be used to describe the structure of sparse neural networks classifiers [33, 38]. In the case of rule-based classifiers the number of rules from a rule base or used rules rate can be used as a measure of complexity. Usually, overfitting is avoided if the complexity is small. Details of these two cases will be discussed in Section 6 and Section 7.

$$ccr(\theta) \triangleq O. \tag{4}$$

The *ccr* is a normalized complexity, which divides the number of components (rules, neurons) used by the classifier by the maximal possible number of components (size of rule base, maximal number of neurons). We will denote it with *ccr* and it obtains a value between 0 (no rules are used) and 1 (all rules are used). The parsimony (or sparseness) of the model can then be defined as $1-ccr$. The computational cost of a classifier with high *ccr* is considered to be higher than that of a classifier with lower *ccr*. This is why a classifier with lower *ccr* should be preferred, given its other performance criteria (fpr, fnr) are equally good. Besides, classifiers with a lower *ccr* will have a lower tendency of overfitting. It is always possible to construct a classifier whose characteristic is a convex combination of the original classifier by means of randomization. Although it does not make much sense in practice, it is important for theoretical considerations that we can always construct a more complex classifier with the same performance in terms of *fpr* and *fnr*, by simply adding components but not using them. Such solutions will be Pareto dominated, but should be included to measure the volume of the convex hull. We name the DET space with complexity of binary classifiers in the third axis as the augmented DET space.

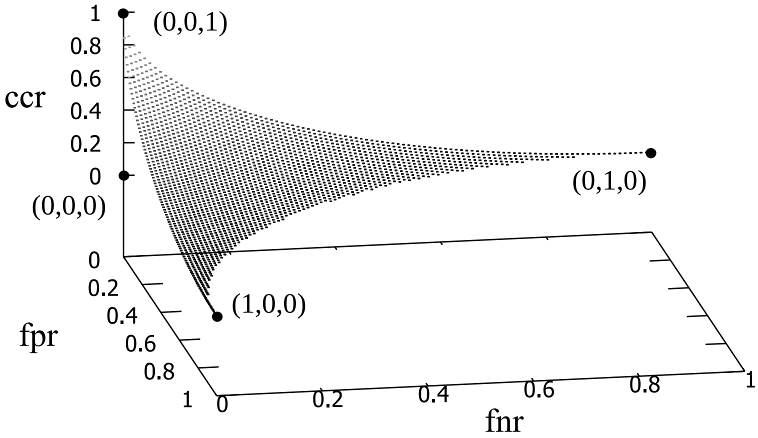


Figure 1: An example of an augmented DET graph with complexity of binary classifiers as a third axis.

In augmented DET space, fpr is plotted on X-axis, fnr is plotted on Y-axis, and *ccr* is plotted on the Z-axis, which is depicted in Fig. 1. Normally, *ccr*, fpr and fnr are conflicting with each other. The newly proposed algorithm aims at finding optimal tradeoffs among the three objectives, as it is denoted in Eq. 5.

$$\min_{\theta\in\Omega} \mathbf{F}(\theta) = \min_{\theta\in\Omega} \mathbf{F}\Big(fpr(\theta), fnr(\theta), ccr(\theta)\Big), \tag{5}$$

where θ represents the parameter of classifiers, such as neural networks [29], support vector machine (SVM) [12], and so on. Ω is the solution space, it includes all possible configurations of classifiers. The performance of a certain classifier can be determined by the parameter θ. We try to find a set of θ that has good performance based on the property of augmented DET convex hull (ADCH).

3.2. ADCH maximization and multiobjective optimization

The convex hull of a set of points is the smallest convex set that contains all those points [46]. The 3D convex hull (CH) of a finite set $A \subseteq \mathbb{R}^3$ is given by Eq. 6.

$$CH(A) \triangleq \left\{ \mathbf{x} : \mathbf{x} = \sum_{i=1}^{|A|} \lambda_i \mathbf{a}_i, \sum \lambda_i = 1, 0 \leq \lambda_i \leq 1, \mathbf{x} \in \mathbb{R}^3 \right\}, \tag{6}$$

where $\mathbf{a}_i \in A$ is a set of initial points. The boundary of the convex hull can be represented with a set of facets, a set of adjacency edges and vertices (V) for each facet [4]. The volume of convex hull (VCH) which is constructed with some points in set A is denoted by Eq. 7.

$$VCH(A) \triangleq Volume\Big(CH(A)\Big). \tag{7}$$

With a set of classifiers, the augmented DET convex hull (ADCH) covers all the potential optimal classifiers. The proposed 3DCH-EMOA aims at maximizing the volume of ADCH with three objectives. We denoted multiobjective optimization of parsimony binary classifiers as ADCH maximization problem.

Several points in augmented DET space are important to note. The point (0, 0, 0) represents the strategy of never issuing a wrong classification and a classifier with a cost of zero. This point represents a perfect classifier and a classifier corresponding to such a point typically does not exist for a non-trivial problem but can be approximated as closely as possible. The points in set $\{(0, 0, ccr) | \, 0 \leq ccr \leq 1\}$ also represent classifiers having perfect performance with respect to the complexity of ccr. The point (1, 0, 0) represents the strategy of issuing all the instances as negative by a classifier whose complexity is zero. The point (0, 1, 0) represents a classifier that predicts all instances as positive without using any rules. In a similar way, predicting all of the instances as negative with all the rules results in the point (1, 0, 1). The point (0, 1, 1) can be obtained by predicting all of the instances as positive with all the rules. For all points in {(1, 0, 0), (0, 1, 0), (1, 0, 1), (0, 1, 1) } a classifier can be constructed, e.g., by randomization. The surface of $fpr+fnr = 1$ represents randomly guessing classifiers, as is shown in Fig. 2. The classifiers which we search for should be in the space of $fpr + fnr < 1$, which have better performance than classifiers obtained by random guessing. In this paper, we treat the points (1,0,0), (0,1,0), (1,0,1) and (0,1,1) as reference points to build convex hull and to calculate the volume above DET surface (VAS) in augmented DET space. For a set of randomly guessing classifiers the VAS will be 0, as all points are in the same surface of $fpr + fnr = 1$. And for a set of perfect classifiers the VAS will be 0.5 (which is the maximum attainable volume), as in augmented DET space the convex hull is constructed with a point set {(1, 0, 0), (0, 1, 0), (1, 0, 1), (0, 1, 1), (0, 0, 0), (0, 0, 1)}.

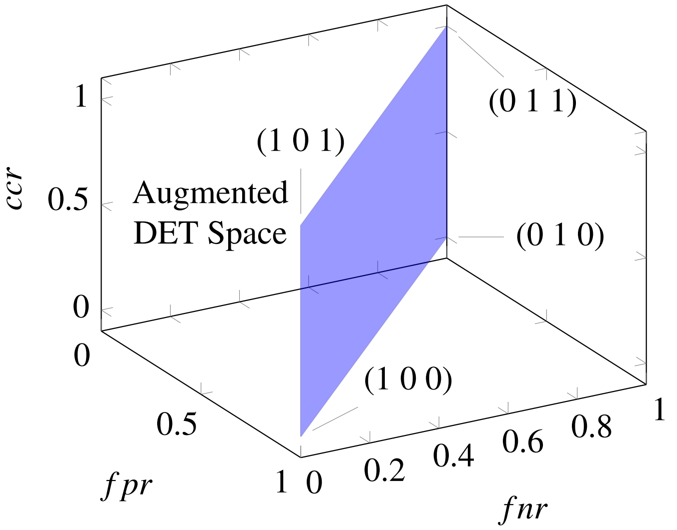


Figure 2: An example of an augmented DET surface for random binary classifiers.

Every binary classifier can be mapped to the augmented DET space. The ADCH is the collection of all attainable classifiers in a given set of classifiers. Furthermore, a classifier is potentially optimal if and only if it lies on the lower boundary of the ADCH. In Fig. 3 the points *a, b, e* are on the augmented DET surface and the point *c, d* are above it. *a, b, e* represent potentially optimal classifiers and *c, d* represent non-optimal ones.

Imprecise distribution information of data defines a range of parameters for iso-performance lines (surfaces) and the range of lines (surfaces) will intersect a segment of ADCH. If the segment defined by a range of lines corresponds to a single point in augmented DET space, then there is no sensitivity to the distribution assumptions, otherwise the ADCH is sensitive to the distribution assumptions. In order to improve the robustness of ADCH not only the VAS should be maximized but also its distribution of points on the convex hull surface should be optimized. Usually, the more uniform the distribution of points in the augmented DET space, the more robust and representative the ADCH is. The Gini coefficient [60] is used to evaluate the uniformity of the distances between solutions of the test functions in this paper, and the nearest neighbor distance of each individual is used to calculate the value of Gini coefficient. Details of Gini coefficient evaluation are discussed later in the paper.

The goal of ADCH maximization is to find a group of classifiers that approximate the perfect point (0,0,0) for binary classifiers. The ADCH maximization problems turn out to be multiobjective optimization problem as it is described in Eq. 8.

$$\min_{\mathbf{x}\in\Omega} \mathbf{F}(\mathbf{x}) = \min_{\mathbf{x}\in\Omega} \mathbf{F}\Big(f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x})\Big), \tag{8}$$

where fpr, fnr and ccr are represented as f_1, f_2 and f_3, respectively, and θ is represented by $\mathbf{x}$. In Eq. 8, $\mathbf{x}$ is the classifiers parameters set, Ω is the solution space, i.e., the set of all possible classifier sets, and $\mathbf{F}(\mathbf{x})$ is a vector function to describe the performance of classifiers in augmented DET space. In the problem of multiobjective optimization, Pareto dominance is an important concept which is defined as: Let $\boldsymbol{\omega} = (\omega_1, \omega_2, \omega_3)$, $\boldsymbol{\nu}$= (ν_1, ν_2, ν_3) be two vectors, $\boldsymbol{\nu}$ is said to dominate $\boldsymbol{\omega}$ if and only if $\nu_i \leq \omega_i$ for all $i = 1, 2, 3$, and $\boldsymbol{\nu}$≠$\boldsymbol{\omega}$, this is denoted as $\boldsymbol{\nu} \prec \boldsymbol{\omega}$. Two distinct points $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$ are incomparable if and only if $\boldsymbol{\nu}$ and $\boldsymbol{\omega}$ do not dominate each other. The Pareto set (PS) is the collection of all the Pareto optimal points in decision space, i.e., of all points $\mathbf{x} \in \Omega$ with no $\mathbf{x}' \in \Omega$ such that $\mathbf{F}(\mathbf{x}) \prec \mathbf{F}(\mathbf{x}')$. The Pareto front (PF) is the set of all PS points in objective space $PF = \{\mathbf{F}(\mathbf{x}) \mid \mathbf{x} \in PS\}$, (see, e.g., [56]).

A special approach based on ROCCH is proposed in [55] to solve the ROC maximization problem for binary classification. Even though the concepts of ROC convex hull and the Pareto

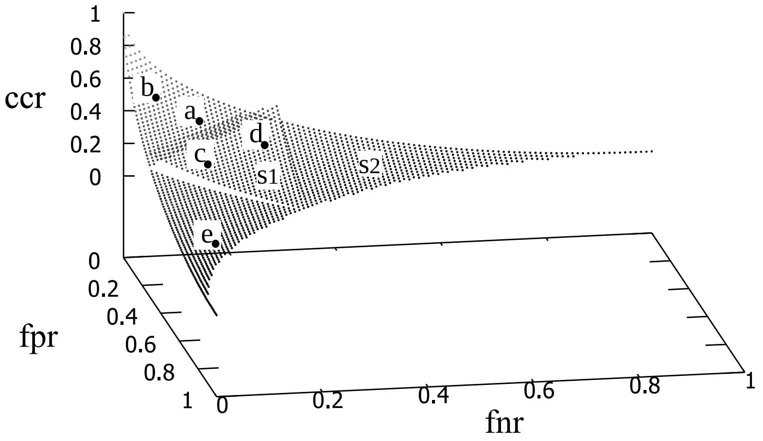


Figure 3: Convex hull and Pareto front in an augmented DET space.

front were reported to be similar, specific and important differences exist. In the example of Fig. 3 points *a, b, c, d, e* are non-dominated points in non-dominated multiobjective optimization algorithms. However, only points *a, b* and *e* are on the convex hull surface. Usually, the points on the higher part of the convex hull surface are non-dominated to each other, but there can be non-dominated points in the Pareto front approximation that are not part of the augmented DET convex hull surface approximation. Such points occur in concave parts of the Pareto front approximation and they are not relevant in the context of augmented DET convex hull optimization. These points are dominated by points that can be obtained by linear, convex combination of classifiers in the approximation set that are not explicitly represented. This is a special characteristic of the ROCCH and ADCH maximization problem and new strategies need to be researched to deal with it.

4. 3D Convex-Hull-based Evolutionary Multiobjective Optimization

In this section, we propose 3D convex-hull-based evolutionary multiobjective algorithm (3DCH-EMOA) for ADCH maximization with three objectives. In this paper, we only consider 3D convex hull, and the solutions of 3DCH-EMOA act as vertices on the convex hull in augmented DET space. The aim of 3DCH-EMOA is to find a set of non-dominated solutions that covers part of the surface of the 3D convex hull, which is constructed with population $Q \subset \mathbb{R}^3$ (the population is described in objective space) and reference points $R \subset \mathbb{R}^3$. The set of frontal solutions (FS), which includes solutions that are located on the surface of the 3D convex hull, of 3DCH-EMOA, can be denoted by Eq. 9.

$$FS(Q) \triangleq \Big\{p : p \in CH(Q \cup R), p \in Q, p \in V\Big\}, \tag{9}$$

where V is the set of vertices of the 3D convex hull as it is described after Eq. 6. The proposed algorithm consists of two key modules, i.e., 3D convex-hull-based sorting and VAS contribution. Details on the proposed algorithm are presented next.

4.1. 3DCH-based sorting without redundancy

Convex-hull-based sorting without redundancy strategy was first proposed in [55]. It has a good performance to deal with binary classification problems. In this paper, we define redundant solutions that have the same performance in objective space. The convex-hull-based sorting approach is extended to three dimensions in this paper. The strategy works effectively, not only because it can maintain the diversity of the population, but also because it takes into consideration the properties of ADCH. With this strategy, if there are not enough non-redundant solutions to fill the whole population, the redundant solutions which are preserved in the archive should be randomly selected and added to the population. It will be shown in Section 5 that this strategy can preserve the diversity of the population by keeping non-redundant solutions with bad performance and discarding the redundant solutions even with good performance. In addition, the non-redundancy strategy can avoid the solutions at the convex hull being copied many times at the selection step of the algorithm.

The framework of 3DCH-based sorting without redundancy is described in Algorithm 1. At first the solution set Q is divided into two parts, one is the redundant solution set Q_r, the other is the non-redundant solution set Q_{nr}. The redundant solution set Q_r will be assigned to the last level of priority of the solution set and the non-redundant solution set Q_{nr} will be assigned into different priority levels by 3DCH-based ranking method. Before ranking the non-redundant solution set Q_{nr}, a reference point set R should be merged with it and a set of candidate points of convex hull CH is constructed. Four points, i.e., $(1, 0, 0)$, $(0, 1, 0)$, $(1, 0, 1)$, $(0, 1, 1)$, are included in reference point set R (see also Fig. 2). The 3D quickhull algorithm [4] is adopted to build the convex hull

Algorithm 1 3DCH-based sorting without redundancy (Q, R)

```
Require: $Q \neq \varnothing, R \neq \varnothing$
    $Q$ is a solution set.
    $R$ is the set of reference points.
Ensure: ranked solution set $F$
Split $Q$ into two subset $Q_r$ and $Q_{nr}$, where $Q_r$ is the redundant solution set, and $Q_{nr}$ is the non-redundant solution set.
$i \leftarrow 0$
while $Q_{nr} \neq \varnothing$ do
    $T \leftarrow Q_{nr} \cup R$
    $F_i \leftarrow FS(T)$
    $Q_{nr} \leftarrow Q_{nr} \setminus F_i$
    $i \leftarrow i + 1$
end while
$F_i \leftarrow Q_r$  //$F$ is the ranked solution set in different levels.
return the ranked solution set $F = \{F_0, F_1, \dots\}$
```

with the candidate points set, which is widely used in 3D convex hull related applications. The points (solutions) on the convex hull surface are considered as the current Pareto set. The first layer consists of the points on the surface of the convex hull. And the remaining points will be used to build the new convex hull for the next layer of Pareto front. Note that all points can be constructed by means of a convex combination of classifiers in the set. Usually, there are several layers of solutions in the beginning of the algorithm and the number of layers will converge to one in the evolution of the population. The computational complexity of the quickhull algorithm to build a 3D convex hull is $O(n \log n)$ [4] with set of candidate points of size n. In the worst case, there is only one point in each convex hull layer, then the complexity of 3DCH-based sorting without redundancy would be $O\Big(\sum_{i=5}^{N+4} i \log i \Big)$, which tends to $O\Big(N^2 \log N\Big)$.

An example of the result of 3DCH-based sorting without redundancy is given in Fig. 3. The surfaces s1 and s2 represent two layers of solutions of different priority, the solutions on surface s1 are better than those on surface s2 and the solutions on surface s1 have much more opportunity to survive to the next generation than those on s2.

After ranking the individuals into different priority levels other questions arise such as how to analyze the importance of individuals in the same priority layer. As the redundant solutions have no additional information about the population, the algorithm selects some of them to survive to the next generation randomly. If there are too many non-redundant solutions to fill the new population, the contribution to the VAS will be used as metric measure to rank the individuals in the same layer, and only the individuals with high VAS contribution will survive. Details of the contribution of VAS are described in the next part.

4.2. VAS contribution selection scheme

In this section, we describe the VAS contribution indicator to evaluate the importance of individuals within the same priority layer. We hypothesize that the new VAS contribution indicator is a more efficient strategy to maximize the volume under the 3D convex hull when compared to the hypervolume based contribution [7] or crowding distance indicator [16]. In the case of 3DCH-EMOA, VAS is defined as the volume above DET convex hull surface, the VAS of population Q is denoted by Eq. 10:

$$VAS(Q) = VCH\left(Q \cup R\right), \tag{10}$$

where R is a set of reference points. To calculate the contribution of an individual, a new convex hull should be built by subtracting from the total population volume the volume of the population without the individual, as shown in Eq. 11:

$$\Delta VAS_i = VAS\left(Q\right) - VAS\left(Q \setminus \{q_i\}\right), \;\; i = 1, 2, ..., m, \tag{11}$$

where m is the number of solutions in Q on the 3D convex hull. The procedure of calculating the VAS contribution for the non-redundant solution set Q_{nr} is given in Algorithm 2. After calculating the contribution to the VAS of each individual in Q_{nr}, the individuals in the same priority layer can be ranked by the volume of ΔVAS. The larger the volume of the contribution to VAS the more important the individual will be.

To analyze the computational complexity of VAS contribution selection stage, we only consider the worst case scenario. In the worst case, there is only one point beyond the set of reference points to rank population for each layer F_i, then the complexity of Algorithm 2 would be $O\left(\sum_{i=5}^{N+4} i \log i\right)$, which tends to $O\left(N^2 \log N\right)$.

Algorithm 2 $\Delta VAS(Q_{nr}, R)$

```
Require: $Q_{nr} \neq \varnothing$
    $Q_{nr}$ is the non-redundant solution set
    $R$ is a set of reference points
Ensure: $VAS$ contribution of each population
$m \leftarrow \text{sizeof}(Q_{nr})$
$P \leftarrow Q_{nr} \cup R$
$Volume_{all} \leftarrow VAS(P)$ // using algorithm described in [4]
for all $i \leftarrow 1$ to $m$ do
    $q_i \leftarrow Q_{nr}(i)$
    $\Delta VAS_i \leftarrow Volume_{all} - VAS(P \setminus \{q_i\})$
end for
return set of $\Delta VAS$
```

4.3. 3DCH-EMOA

The framework of 3DCH-EMOA is described in Algorithm 3, which is inspired by indicator-based evolutionary algorithms. To optimize the multiple objectives on the convex hull space the initial population Q_0 should be built randomly with a uniform distribution. Due to the high computational complexity of 3D convex hull construction, a steady-state selection scheme is used, which has been successfully used in many EMOAs [7, 45]. The steady-state selection is also denoted as $(N + 1)$, where N represents the population size of EMOA, $(N + 1)$ means that only a new solution is produced in each generation. The advantage of using a steady-state scheme was analyzed theoretically in [7]. Most importantly it will lead to a series of population with increasing size of the convex hull, and, when compared to other subset selection strategies with this property, has a small computational effort. For each iteration there is only one offspring produced by the evolutionary operators and, in order to keep the size of population constant, the least performing individual should be deleted, or in other words, the best performing subset of size N should be kept. The non-descending reduce strategy given in Algorithm 4 is adopted in this method to remove an individual from the population.

In Algorithm 4, the population is firstly divided into non-redundant part Q_{nr} and redundant part Q_r. If the redundant set Q_r is not empty, an individual can be selected randomly to be deleted from the population. If there is no individual in Q_r, all of the solutions are of non-redundant type, then 3DCH-based sorting without redundancy can be used to rank the population into several

Algorithm 3 3DCH-based EMOA (Max, N)

```
Require:  $Max > 0, N > 0$
    $Max$ is the maximum number of evaluations
    $N$ is the population size
Ensure:  a set of $FS$
$Q_0$ randomly generated with a uniform distribution
$t_0 \leftarrow 0$
$m \leftarrow 0$
while $m < Max$ do
    $q_i \leftarrow$ Generate New Offspring ($Q_t$)
    $Q_{t+1} \leftarrow$ Non-Descending Reduce($Q_t, q_i$)
    $t \leftarrow t + 1$
    $m \leftarrow m + 1$
end while
return  $FS(Q_t)$
```

priority layers. If there is only one layer of solutions, it means that all solutions in the population are non-dominated, then the contribution of each solution to VAS should be calculated and the individual with the least contribution will be deleted from the population. If there are several layers of the population, only the contribution to VAS of individuals on the last priority layer should be calculated and the individual with least contribution should be removed from the population. In Algorithm 4, the comparison in the 9th line is added to save time in cases when there is no improvement after adding the new point.

4.4. Computational complexity of 3DCH-EMOA

As described above, 3DCH-EMOA is a general evolutionary algorithm. Its computational complexity can be described by considering one iteration of the entire algorithm. The complexity of variation operation on generating new offspring is $O\left(N\right)$. 3DCH-based sorting without redundancy has complexity of $O\left(N^2 \log N\right)$ and VAS contribution selection has complexity of $O\left(N^2 \log N\right)$. Here, N is the size of population. The overall complexity of the algorithm is $O\left(N^2 \log N\right)$. As we only consider triobjective optimization, the number of objectives is not involved in the asymptotical analysis.

Algorithm 4 Non-Descending Reduce (Q, q)

```
Require:  $Q \neq \emptyset$
     $Q$ is a set of solutions
     $q$ is a solution
Ensure:  a new solution set $Q'$
Split $Q \cup \{q\}$ into two sub-population $Q_r$ and $Q_{nr}$ ($Q_r$ is the collection of redundant individuals
  and $Q_{nr}$ is the collection of non-redundant individuals)
if sizeof($Q_r$) >= 1 then
    $p \leftarrow$Randomly selected individual from $Q_r$
    $Q' \leftarrow Q_{nr} \cup Q_r \setminus \{p\}$
else
    $F_1, ..., F_l \leftarrow$ 3DCH-based sorting without redundancy($Q_{nr}$)
    $Vol_{ori} \leftarrow VAS(Q)$
    $Vol_q \leftarrow VAS(Q \cup \{q\})$
    if $Vol_{ori} < Vol_q$ then
        /*the index of the minimum value in $\Delta VAS(F_l)$*/
        $k \leftarrow \underset{i}{\text{argmin}}\, \Delta VAS(F_l)$
        $d \leftarrow F_l(k)$ /*the $k^{th}$ solution in $F_l$*/
        $Q' \leftarrow Q_{nr} \setminus \{d\}$
    else
        $Q' \leftarrow Q_{nr} \setminus \{q\}$
    end if
end if
return $Q'$
```

5. Experimental Studies on Artificial Test Problems

In this section, ZEJD test functions are adopted to test the performance of 3DCH-EMOA and several other EMOAs, including NSGA-II, GDE3, SMS-EMOA, SPEA2, MOEA/D. In this first benchmark we are interested in the capability of 3DCH-EMOA to cover the relevant part of the convex hull surface with points. To evaluate the performance of these algorithms VAS, Gini coefficient, computational time and Mann-Whitney test [40] are adopted in this section. By comparing the results of all algorithms we can make a conclusion that the new proposed algorithm has a good performance to deal with augmented ADCH maximization problem. Details of the experiments are described next.

5.1. Metrics

Four metrics are chosen to evaluate the performance of the different algorithms in the comparative experiment on the ZEJD problems. VAS metric can evaluate the solution set directly, the better the solution set the larger the value of VAS will be. For ZEJD problems the smallest value of VAS is 0 with random guessing classifiers and the largest value of VAS is 0.5.

The Gini coefficient is commonly used as a measure of statistical dispersion intended to represent the income distribution of a nation's residents [60]. In this work, the Gini coefficient is used to evaluate the uniformity of the solution set by calculating the statistical distribution of the nearest neighbor distance of each solution. The Gini coefficient can describe the spread of neighboring individuals on the achieved Pareto front. The value of Gini coefficient will be zero if distances in the set are distributed uniformly. The definition of Gini coefficient $g(Q)$ is described in Eq. 12.

$$g(Q) = \frac{1}{|Q|}\left[|Q| + 1 - 2\left(\frac{\sum_{i=1}^{|Q|}(|Q| + 1 - i)d_i}{\sum_{i=1}^{|Q|} d_i}\right)\right], \tag{12}$$

where g represents the value of Gini coefficient, $|Q|$ is the number of solutions in the solution set Q, d_i is the nearest neighbor distance for each solution in the objective space.

In addition the computational effort is measured for each algorithm. As the evaluation of the test problems is fast, this measure indicates how much time is needed to perform the steps of the algorithm. Time cost is used to measure the complexity of each algorithm in this section.

Furthermore, the Mann-Whitney test, which is a statistical test, is selected to evaluate whether the differences between 3DCH-EMOA and other methods are significant or not. We denote it as "▲" if 3DCH-EMOA outperforms other method significantly, if 3DCH-EMOA performs as well as other method in statistical testes denote it as "–", and we denote it as "▽" if 3DCH-EMOA performs not as well as other method.

5.2. Parameter setting

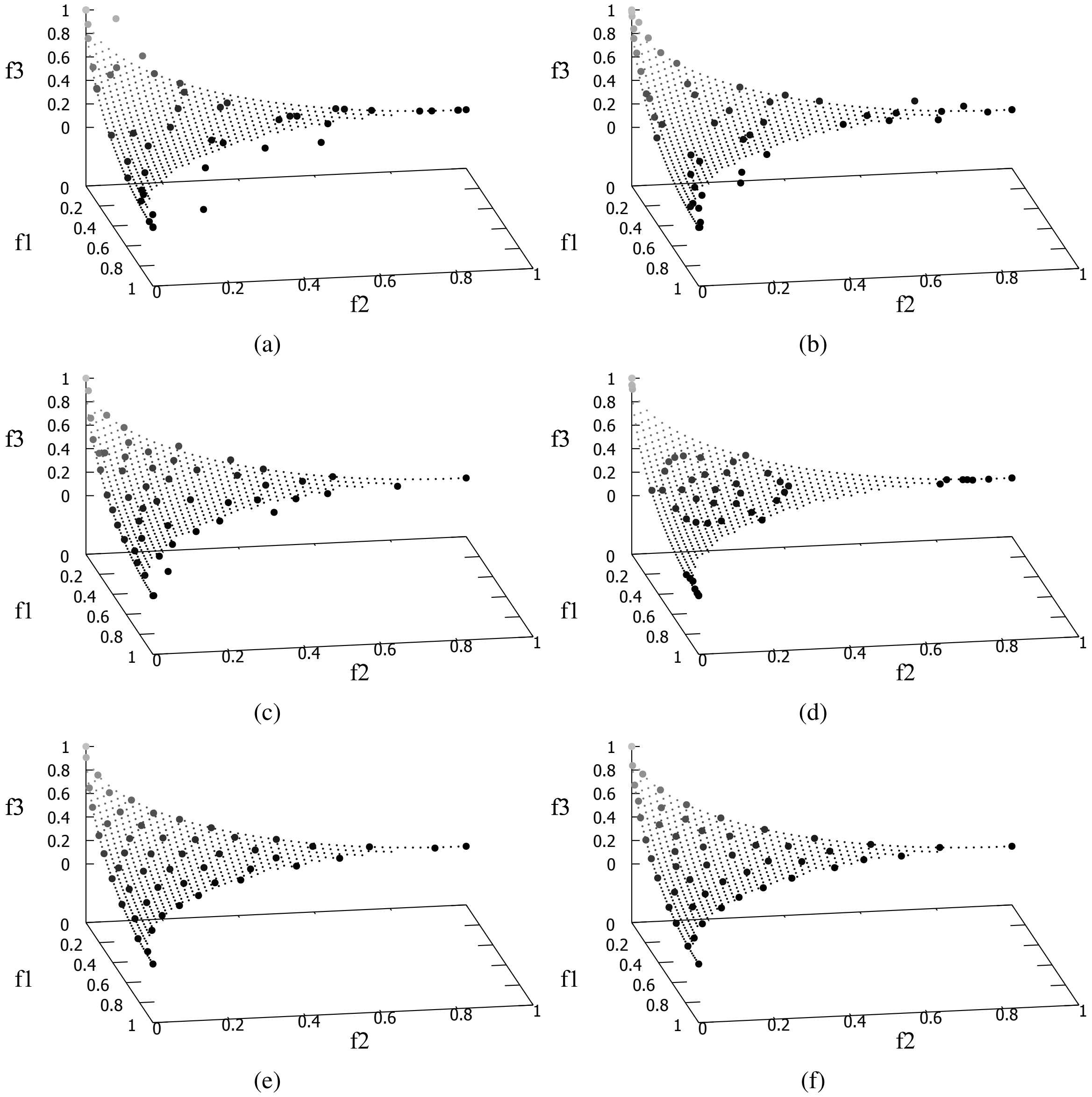
All algorithms are run for 25000 function evaluations. The simulated binary crossover (SBX) operator and the polynomial mutation are applied in all experiments. The crossover probability of $p_c = 0.9$ and a mutation probability of $p_m = 1/n$, where n is the number of decision variables that are used. The population size is set to 50 for ZEJD problems. The size of archive for SPEA2 is equal to the size of the population. All of the experiments are based on jMetal framework [19, 18]. All of the experiments are running on a desktop PC with an i5 3.2 GHz processor and 4GB memory under Ubuntu 14.04 LTS. For each mentioned algorithm, 30 independent trials are conducted on ZEJD test problems.

5.3. Experimental results and discussion

The comparison of simulation experiments with NSGA-II, GDE3, SPEA2, MOEA/D, SMS-EMOA, and 3DCH-EMOA on ZEJD problems is discussed in this section. The results of experiments are given as follows: the results not only include the plots of solution set in the objective space but also include statistical analysis on the metrics of these results. The illustrations of solution set in the $f_1 - f_2 - f_3$ objectives space are plotted for the ZEJD problems. The results of ZEJD1 are shown in Fig. 4, of ZEJD2 are shown in Fig. 5 and of ZEJD3 are shown in Fig. 6. The solutions obtained are depicted in dark dots with large size and the true Pareto fronts are in gray dots with small size.

By comparing the Pareto fronts and results of ZEJD1 we can make some conclusions. NSGA-II, GDE3 and SPEA2 show the worst convergence. MOEA/D can converge to the true Pareto front, however it does not give good results on diversity and distribution uniformity. The result of MOEA/D has no solutions on the edges of the solution space. SMS-EMOA and 3DCH-EMOA have good performance on convergence, diversity and distribution uniformity. SMS-EMOA and 3DCH-EMOA work well with the ZEJD1 test problem.

By comparing the results of ZEJD2 and ZEJD3 we can see that SMS-EMOA and 3DCH-EMOA have better performance on convergence, diversity and distribution uniformity than other algorithms. However, the results of SMS-EMOA have several points on the dent areas in the results of ZEJD2 and ZEJD3. Only 3DCH-EMOA omits the dent areas of ZEJD2 and ZEJD3, which allows it to add more points in parts that are relevant for maximizing VAS. As there are some points on the Pareto front but not on the 3D convex hull surface, which contribute to hypervolume metric but do not contribute to VAS, SMS-EMOA preserves these solutions and 3DCH-EMOA ignores them. In summary, 3DCH-EMOA can always achieve better results than other algorithms, not only on convergence and distribution uniformity, but it also does not waste resources by approximating concave parts of the Pareto front.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 4: Experimental results of ZEJD1 are shown in $f_1 - f_2 - f_3$ space. (a) Result of NSGA-II. (b) Result of GDE3. (c) Result of SPEA2. (d) Result of MOEA/D. (e) Result of SMS-EMOA. (f) Result of 3DCH-EMOA.

In the experiments, all algorithms have been running for 30 times independently on ZEJD test problems to evaluate and compare the robustness of these algorithms. The performance characteristics of each algorithm can be seen from the statistical analysis of the experimental results. The statistical results (mean and standard deviation) of the VAS are shown in Table 2. The detailed

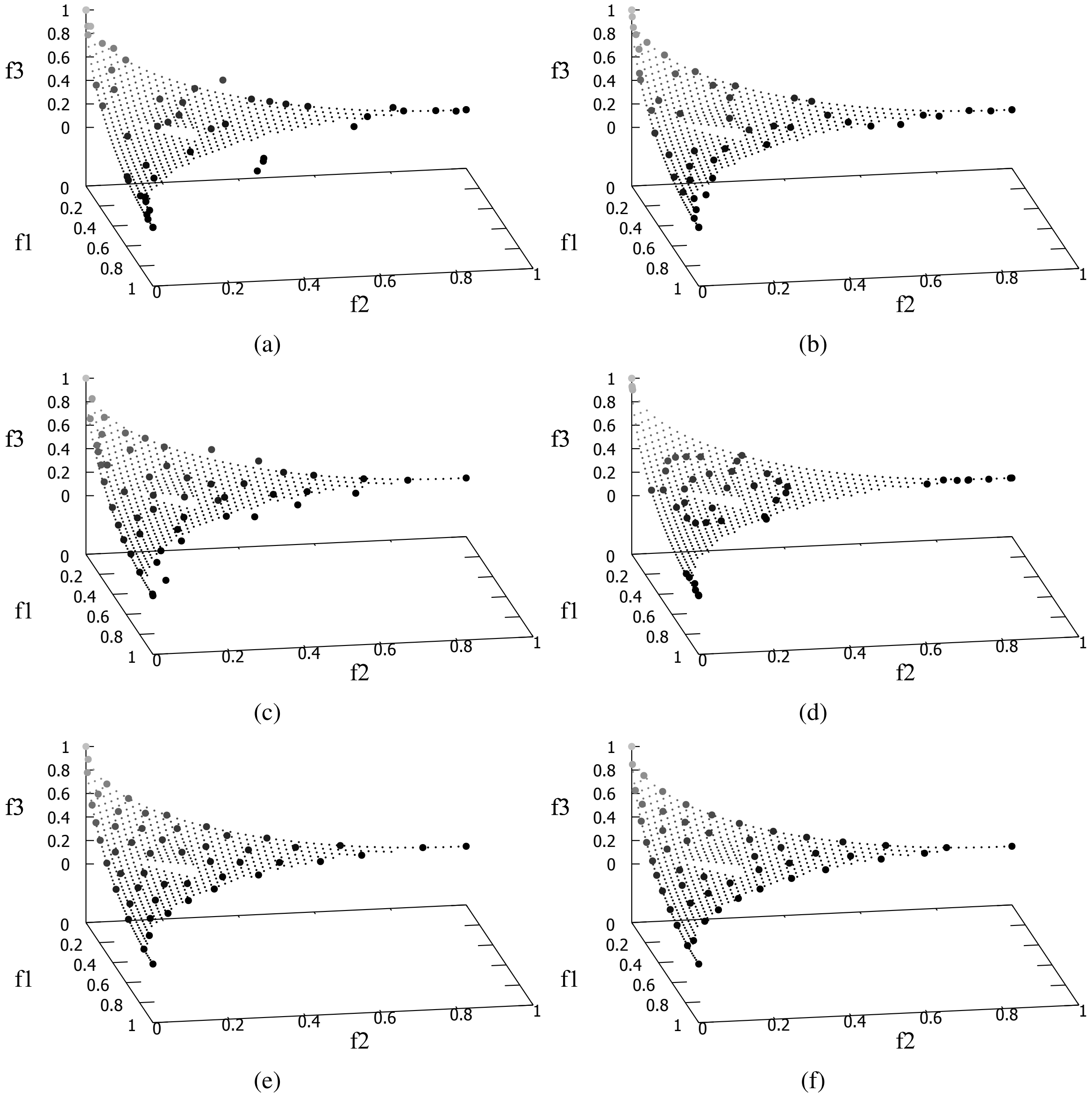
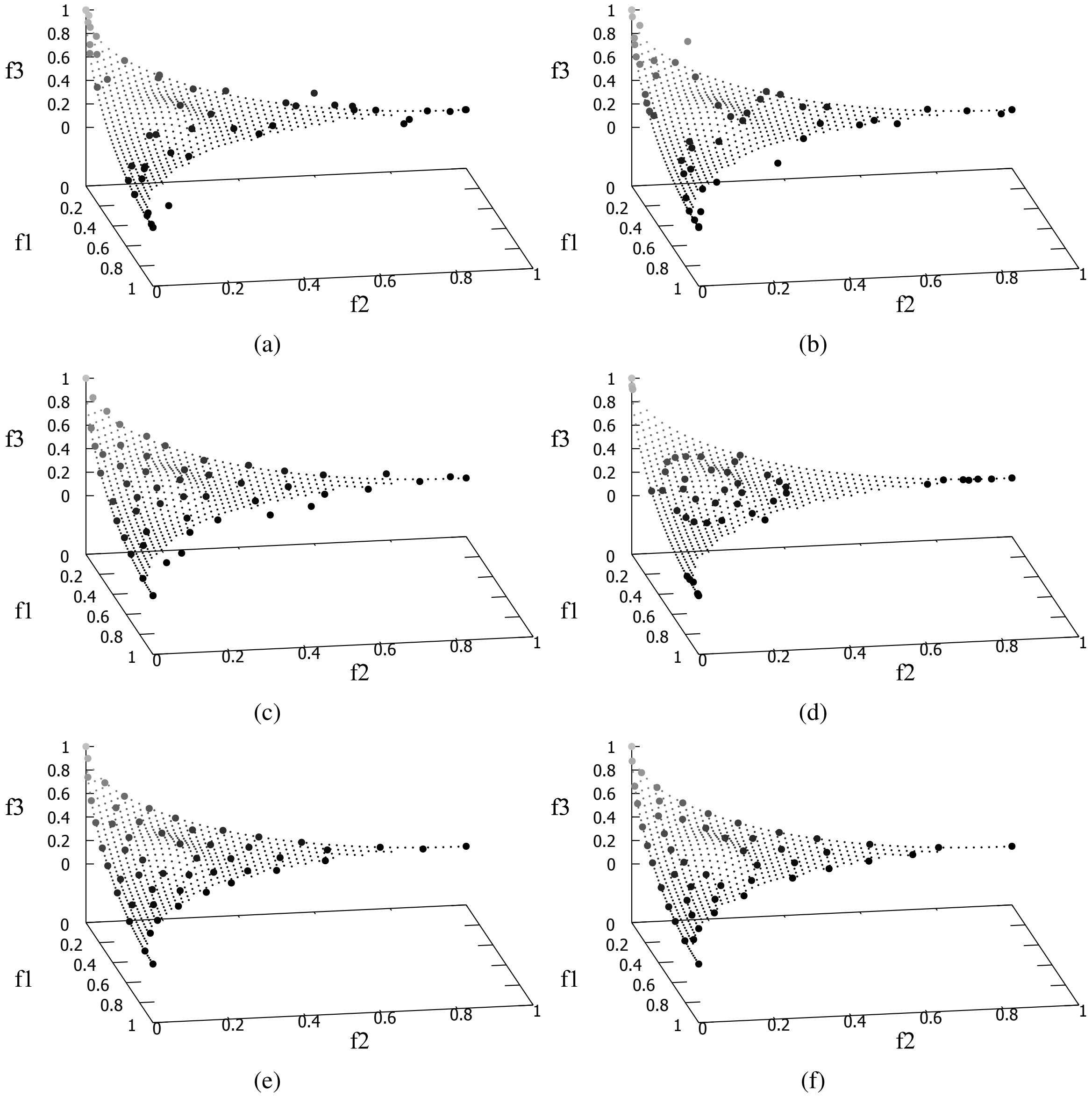


Figure 5: Experimental results of ZEJD2 are shown in $f_1 - f_2 - f_3$ space. (a) Result of NSGA-II. (b) Result of GDE3. (c) Result of SPEA2. (d) Result of MOEA/D. (e) Result of SMS-EMOA. (f) Result of 3DCH-EMOA.

discussion follows next.

While dealing with the ZEJD problems and considering the metric of VAS, 3DCH-EMOA gets the largest value of mean and the smallest value of standard deviation, which shows that 3DCH-EMOA has a good performance not only in convergence but also in stability of these results. GDE3 obtains the second best result with these test functions. As 3DCH-EMOA uses VAS metric to guide its evolution of population, it can obtain solutions with higher value of VAS than others.



(a)

(b)

(c)

(d)

(e)

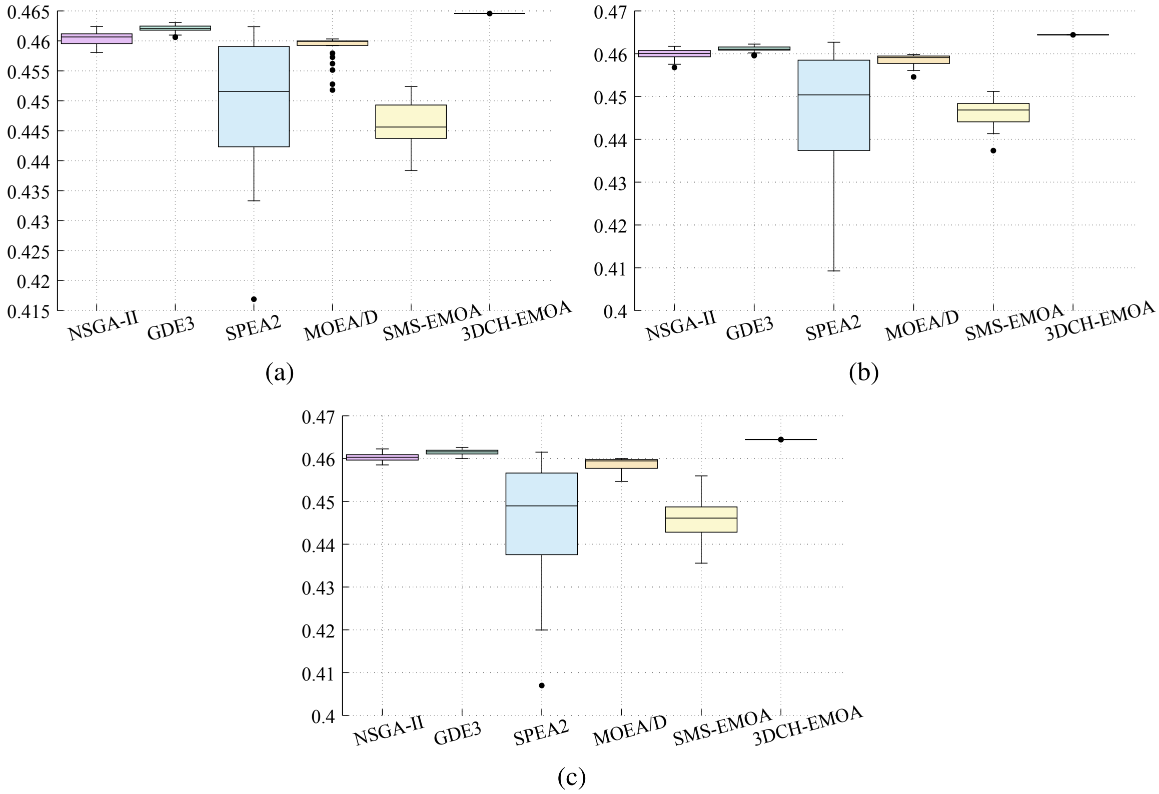
(f)

Figure 6: Experimental results of ZEJD3 are shown in $f_1 - f_2 - f_3$ space. (a) Result of NSGA-II. (b) Result of GDE3. (c) Result of SPEA2. (d) Result of MOEA/D. (e) Result of SMS-EMOA. (f) Result of 3DCH-EMOA.

Table 2: Mean and standard deviation of VAS on ZEJD test problems.

	NSGA-II	GDE3	SPEA2	MOEA/D	SMS-EMOA	3DCH-EMOA
ZEJD1	$4.60e-01_{1.3e-03}$	$4.62e-01_{6.3e-04}$	$4.49e-01_{1.1e-02}$	$4.59e-01_{2.1e-03}$	$4.46e-01_{3.6e-03}$	$4.65e-01_{5.0e-06}$
ZEJD2	$4.60e-01_{1.2e-03}$	$4.61e-01_{6.4e-04}$	$4.48e-01_{1.3e-02}$	$4.59e-01_{1.3e-03}$	$4.46e-01_{3.1e-03}$	$4.64e-01_{4.5e-06}$
ZEJD3	$4.60e-01_{8.8e-04}$	$4.61e-01_{6.9e-04}$	$4.46e-01_{1.2e-02}$	$4.59e-01_{1.5e-03}$	$4.46e-01_{4.1e-03}$	$4.64e-01_{3.9e-06}$

Fig. 7 uses box-plots to show statistical results of VAS with different EMOAs. By comparing the three box-plots of ZEJD test problems, we can see that 3DCH-EMOA not only can obtain solutions with the largest value of VAS, but it can also obtain solutions with the best stability with the metric of VAS.



(a)

(b)

(c)

Figure 7: Box-plots of VAS for three ZEJD test problems, each box-plot is generated by running 30 independent trials. (a) Box-plot of VAS for ZEJD1 test problem. (b) Box-plot of VAS for ZEJD2 test problem. (c) Box-plot of VAS for ZEJD3 test problem.

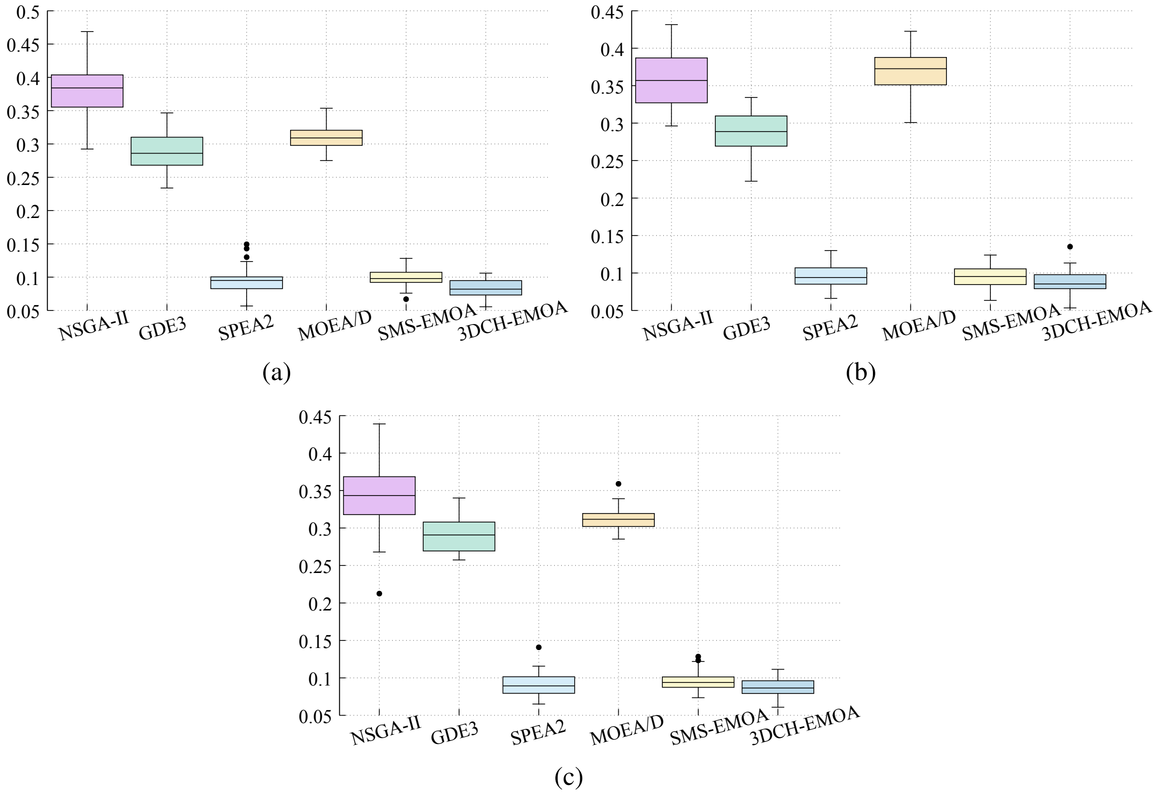
The statistical results of the Gini coefficient are shown in Table 3. By comparing the results in the table we can see that 3DCH-EMOA gets the smallest value of mean, which shows that 3DCH-EMOA has a good uniformity and diversity of the population performance. SMS-EMOA shows the best stability as it gets the smallest value of standard deviation. SPEA2 obtains the second best result, however it did not have good convergence performance.

Fig. 8 uses box-plots to show statistical results of VAS with different EMOAs. By comparing the three box-plots of ZEJD test problems, we can see that 3DCH-EMOA can obtain solutions with the smallest value of Gini coefficient, i.e., 3DCH-EMOA can obtain solutions with the best

Table 3: Mean and standard deviation of Gini coefficient on ZEJD test problems.

	NSGA-II	GDE3	SPEA2	MOEA/D	SMS-EMOA	3DCH-EMOA
ZEJD1	$3.83e-01_{4.0e-02}$	$2.90e-01_{2.9e-02}$	$9.60e-02_{2.0e-02}$	$3.09e-01_{1.7e-02}$	$9.88e-02_{1.3e-02}$	$8.18e-02_{1.4e-02}$
ZEJD2	$3.58e-01_{3.9e-02}$	$2.87e-01_{2.7e-02}$	$9.49e-02_{1.5e-02}$	$3.69e-01_{2.6e-02}$	$9.54e-02_{1.4e-02}$	$8.74e-02_{1.6e-02}$
ZEJD3	$3.45e-01_{4.5e-02}$	$2.92e-01_{2.3e-02}$	$9.10e-02_{1.6e-02}$	$3.12e-01_{1.6e-02}$	$9.62e-02_{1.4e-02}$	$8.67e-02_{1.2e-02}$

uniformity.



(a)

(b)

(c)

Figure 8: Box-plots of Gini coefficient for three ZEJD test problems, each box-plot is generated by running 30 independent trials. (a) Box-plot of Gini coefficient for ZEJD1 test problem. (b) Box-plot of Gini coefficient for ZEJD2 test problem. (c) Box-plot of Gini coefficient for ZEJD3 test problem.

The statistical results of optimization time cost are shown in Table 4. MOEA/D always costs the least time and NSGA-II performs better than others. SMS-EMOA is the most time consuming algorithm and 3DCH-EMOA performs only better than SMS-EMOA.

The Mann-Whitney test is adopted to verify whether the differences observed in Table 2, 3 and 4 are significant or not. The results of Mann-Whitney test are listed in Table 5. By comparing the

Table 4: Mean and standard deviation of optimization time cost/ms on ZEJD test problems.

	NSGA-II	GDE3	SPEA2	MOEA/D	SMS-EMOA	3DCH-EMOA
ZEJD1	$1.29e+02_{9.6e+01}$	$2.91e+03_{2.2e+01}$	$2.05e+03_{6.9e+01}$	$8.49e+01_{6.5e+01}$	$7.03e+04_{2.5e+03}$	$5.30e+04_{1.4e+03}$
ZEJD2	$1.32e+02_{1.0e+02}$	$2.91e+03_{3.8e+01}$	$2.04e+03_{6.5e+01}$	$8.82e+01_{6.6e+01}$	$6.80e+04_{2.3e+03}$	$4.41e+04_{1.0e+03}$
ZEJD3	$1.25e+02_{7.8e+01}$	$2.61e+03_{2.8e+02}$	$2.07e+03_{1.9e+02}$	$8.08e+01_{3.8e+01}$	$7.33e+04_{3.6e+03}$	$4.91e+04_{1.3e+03}$

Table 5: The results of Mann-Whitney test on ZEJD test problems.

3DCH-EMOA	vs	NSGA-II	GDE3	SPEA2	MOEA/D	SMS-EMOA
VAS	ZEJD1	▲	▲	▲	▲	▲
	ZEJD2	▲	▲	▲	▲	▲
	ZEJD3	▲	▲	▲	▲	▲
Gini coefficient	ZEJD1	▲	▲	▲	▲	▲
	ZEJD2	▲	▲	–	▲	▲
	ZEJD3	▲	▲	–	▲	▲
Time cost	ZEJD1	▽	▽	▽	▽	▲
	ZEJD2	▽	▽	▽	▽	▲
	ZEJD3	▽	▽	▽	▽	▲

results in Table 5 we can see that: 1) 3DCH-EMOA outperforms other EMOAs significantly on *VAS* metric; 2) 3DCH-EMOA outperforms most of other EMOAs significantly on Gini coefficient metric except for SPEA2 on ZEJD2 and ZEJD3 problems; 3) 3DCH-EMOA performs not as good as most of other EMOAs on time cost metric except for SMS-EMOA.

In the case of machine learning problems, such as feature selection and parameters optimization of classifiers, the evaluation takes much more time than optimization process, which is different from test functions. Considering problems in machine learning, the optimization time is not a key obstacle, especially for offline learning. Details of 3DCH-EMOA dealing with spam problem are discussed in the next section.

6. Spam problem

From a technical point of view, an email anti-spam system consists of a set of boolean filtering rules (denoted as $Ru = \{r_1, r_2, \ldots, r_{|Ru|}\}$), that jointly allows for spam messages detection. Discovering the relative importance of these rules and assigning the corresponding scores (weights) of each rule, is a complex setup process. The need of frequent scores reassignment for existing rules and setting scores for new rules, to keep the anti-spam filter updated and running, requires the adoption of machine learning and optimization techniques. Every time an email is received for evaluation, SpamAssassin [1], probably the most commonly used open source anti-spam filtering system, finds all the rules matching the target message and computes the sum of their scores. This cumulative value is then compared with a configurable threshold (required score) to finally classify the new incoming message as spam or ham (legitimate). An email total score (ets) is computed as shown in Eq. 13:

$$ets = \sum_{i=1}^{|Ru|} w_i \times r_i, \tag{13}$$

where w_i is the weight of r_i, Ru is a set of spam classification rules, whose cardinality is $|Ru|$.

6.1. Multiobjective spam filtering problem formulation

Spam filtering problem optimization has been addressed by the techniques surveyed in [6, 59]. The formulation of the scores setting optimization problem is naturally biobjective. A typical user would wish to minimize both the number of spam messages not identified by anti-spam filtering techniques, called false negative rate (fnr), and the number of legitimate messages classified as spam by mistake, called false positive rate (fpr). A business email is one of extreme cases of anti-spam systems setup with such objectives, where the fpr should be tuned to have lowest possible rate of legitimate messages lost, usually at the expenses of higher *fnr*. On the other extreme is content management systems (CMSs) devoted to entertainment, where dismissing some legitimate messages keeps or improves the interest on their usage, while the acceptance of any spam message is not allowed. The cases between these two extremes are also of high interest for a variety of areas where this problem is studied.

In previous work on anti-spam filter optimization [59], it was observed that many rules were not participating in the classification process and some (with very small weights) only marginally influenced the classification results. This observation suggests that in addition to optimizing fpr and fnr, the complexity of the anti-spam filter or its parsimony can be optimized.

The trend of increasing the number of rules for the system operation creates an empirically known potential inefficiency phenomenon, which is addressed under the so called principle of parsimony. This principle states, in one of its simplified formulations, that unnecessary assumptions for a conclusion should not be considered if they have no effect on the conclusion [3]. Parsimony is measured in the context of the current anti-spam study as the minimum number of rules with score different from zero that support a specific classification quality. The complexity classifier rate can described by Eq. 14.

$$ccr = \frac{\sum_{i=1}^{|Ru|} \mathbf{1}\left\{|w_i| \neq 0\right\}}{|Ru|}, \tag{14}$$

where $\mathbf{1}\{\cdot\}$ is the indicator function, so that $\mathbf{1}$\{a true statement\}=1, and $\mathbf{1}$\{a false statement\}=0.

In our study we also follow a triobjective problem formulation, minimizing all three objectives: fpr, fnr and ccr (number of anti-spam filter rules rate) to be used in the classification process.

6.2. SpamAssassin corpus

For the multiobjective anti-spam problem formulation experiments, we adopted the SpamAssassin system [1]. SpamAssassin was selected due to its popularity and wide adoption by the open source community, the research community on anti-spam systems, wide commercial usage, and available email corpora. The SpamAssassin corpus used in our experiments is composed of 9349 email messages, 2398 of which are spam and 6951 legitimate messages [42]. SpamAssassin became a reference in the anti-spam filtering domain, not only due to its public availability to research and development, but also because of its performance (classification quality). Individual binary classifiers (filtering rules) learning process, such as Naive Bayes, is based on SpamAssassin public corpus with cross-validation training and testing procedures.

6.3. Algorithms involved

Five reference multiobjective algorithms (NSGA-II, SPEA2, MOEA/D, SMS-EMOA and 3DCH-EMOA) were tested for spam classification quality assessment of the three objectives anti-spam filtering problem formulation, using the SpamAssassin corpus [42]. Experiments were performed with jMetal [18], an optimization framework for the development of multiobjective metaheuristics in Java.

6.4. Parameter setting

Default parameters for problem formulation, experiments and algorithms settings were adopted for the experiments.

Encoding: We employed a jMetal RealBinary encoding scheme where the chromosome is constituted by an array of real values in the interval $[-5, 5]$ and a bit string of equal length. The length of the chromosome is determined by the number of anti-spam filtering rules. In this study the number of rules available in the SpamAssassin software public distributions that effectively match SpamAssassin email messages corpus is 330. Each rule is associated with a real value score in the $[-5, 5]$ interval and a one bit in the chromosome. If the ith bit is 0 the ith rule is ignored, and otherwise the rule is considered by the spam classifier with the ith corresponding real value score (weight). Messages are classified as spam when the sum of the active rules that match the message is equal or greater than the threshold value of 5.

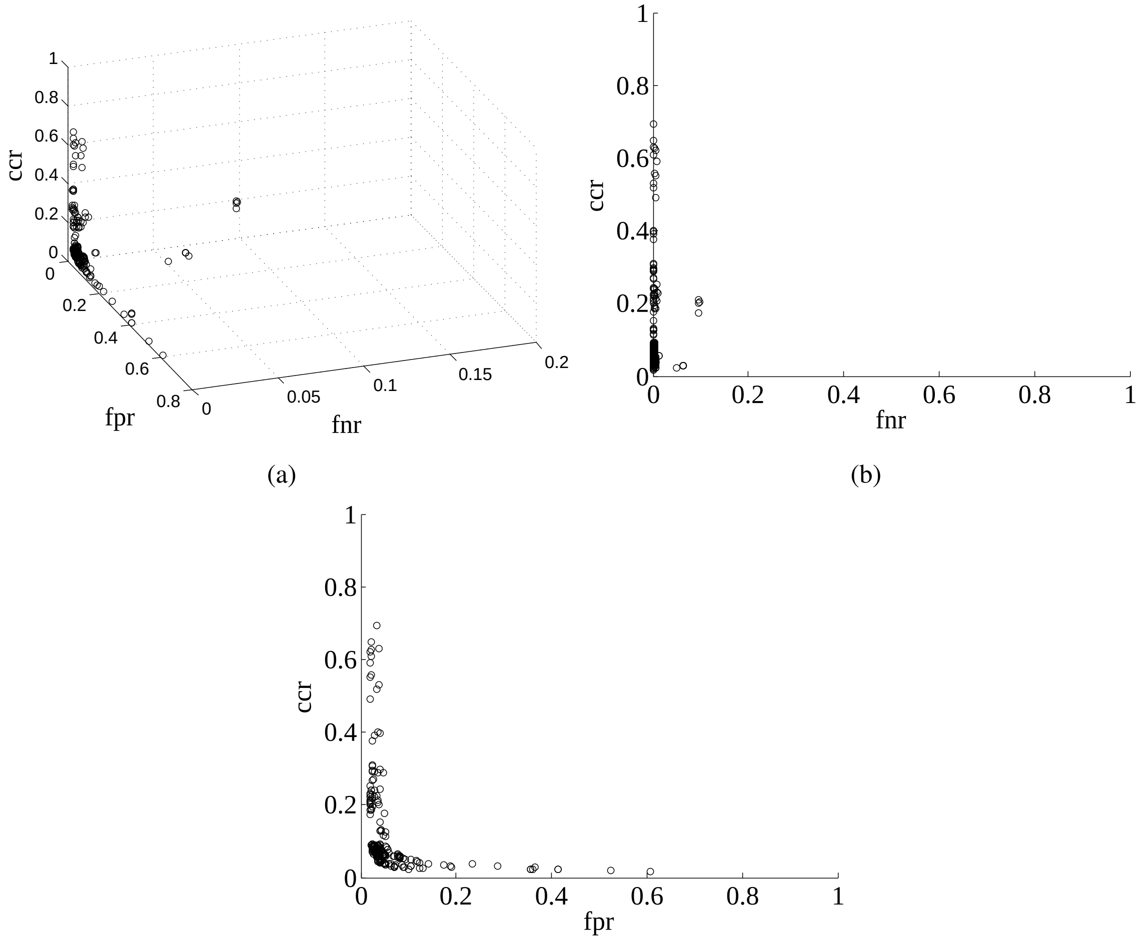
Configuration: The five algorithms (NSGA-II, SPEA2, SMS-EMOA, MOEA/D, 3DCH-EMOA) are set with a maximum of 25000 function evaluations as the experiment stopping criteria. The simulated binary crossover (SBX) single point crossover and polynomial bit flip mutation operators are applied in the experiments. The crossover probability of $p_c = 0.9$ and a mutation probability of $p_m = 1/n$, where n is the number of anti-spam filtering rules, are used. The population size is set to 100 for all algorithms, archive size of 100 is set for SPEA2 and offset size of 100 is set for SMS-EMOA and 3DCH-EMOA. All of the algorithms are run 30 times independently.

6.5. Experimental results and discussion

The comparison of NSGA-II, MOEA/D, SPEA2, SMS-EMOA and 3DCH-EMOA algorithms for the three objectives spam problem formulation is done with respect to the reference Pareto front, which is taken as a close approximation of the true Pareto front. The reference Pareto front is calculated as the best set of solutions of all algorithms achieved in all experimental runs.

We will first interpret this reference Pareto front shown in Fig. 9(a), Fig. 9(b) and Fig. 9(c), corresponding to the three axis projections, $ccr \times fpr$ (classifier complexity ratio × false positive rate) and $ccr \times fnr$ (classifier complexity ratio × false negative rate), respectively (all objectives are to be minimized).

The plots show the boundary between the dominated and non-dominated space (attainment curve). All values are percentages relative to the number of anti-spam filtering rules (330) and total number of email messages (9349).



(a)

(b)

(c)

Figure 9: Reference Pareto front for three objectives spam problem formulation. (a) Reference Pareto front for three objectives spam problem formulation (three axis projection). (b) Reference Pareto front for three objectives spam problem formulation ($ccr \times fpr$ projection). (c) Reference Pareto front for three objectives spam problem formulation ($ccr \times fnr$ projection).

From the plots we conclude that: 1) Even for a classifier using the maximal number of rules, the fnr could not be reduced to zero, but it got very close to it; 2) The fpr is almost exactly zero for spam filters that use only ca. 15% of the rules; 3) Using about 20% of the rules, the knee point solution is found. From then on, only marginal improvements are possible by adding more rules.

In summary, the addition of a third objective is particularly valuable because it can help to reduce the computational effort for the classification to ca. 20% of the effort when all rules are used, losing almost no performance. The second question is how close different algorithms get to the true Pareto front, here represented by the reference Pareto front. For this, one might look at

Pareto fronts of each algorithm that have an average performance in VAS. Also, we can look at summary statistics on performance metrics, first and foremost on the VAS performance.

The performance statistics of VAS, Gini coefficient and time cost are listed in Table 6. Fig. 10(a) and Fig. 10(b) indicate that the 3DCH-EMOA has clearly the best performance in the VAS metric and also it achieves relatively good Gini index values, but has bad performance in the time cost metric. Because the VAS metric is the most relevant to 3D ROC optimization, it is recommended to use 3DCH-EMOA for finding frontal solutions approximations for the 3D anti-spam filtering problem.

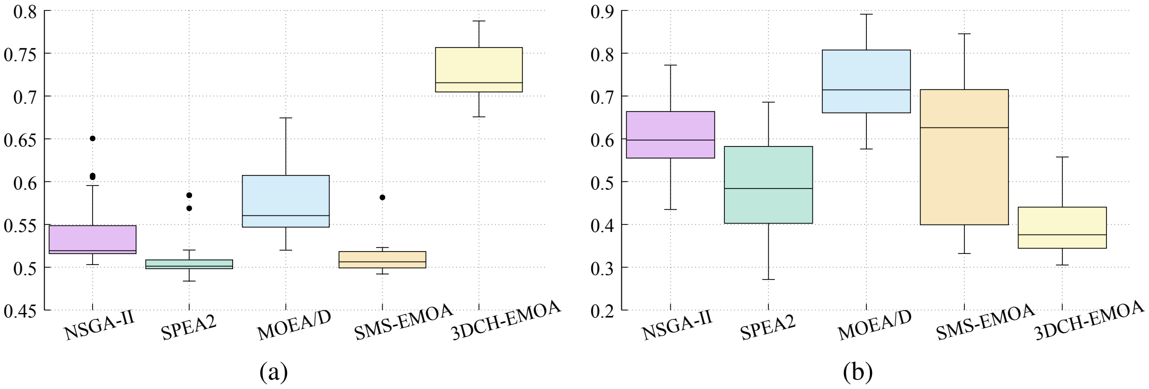


Figure 10: Box-plot of VAS and Gini for triobjective spam problem formulation. (a) Box-plot of VAS for triobjective spam problem formulation. (b) Box-plot of Gini for triobjective spam problem formulation.

Table 6: Mean and standard deviation of VAS, Gini coefficient and time cost/ms.

	NSGA-II	MOEA/D	SPEA2	SMS-EMOA	3DCH-EMOA
VAS	$3.41e-01_{5.8e-03}$	$3.48e-01_{1.3e-02}$	$3.31e-01_{7.8e-03}$	$3.26e-01_{7.7e-03}$	$4.08e-01_{4.9e-03}$
Gini	$5.96e-01_{8.8e-02}$	$4.82e-01_{1.2e-01}$	$7.30e-01_{8.9e-02}$	$5.84e-01_{1.6e-01}$	$2.70e-01_{7.3e-02}$
Time cost	$3.15e+05_{2.0e+04}$	$3.22e+05_{2.0e+04}$	$3.31e+05_{2.5e+04}$	$4.03e+05_{8.0e+04}$	$1.38e+07_{2.7e+06}$

The results of Mann-Whitney test are listed in Table 7. By comparing the results in Table 7 we can see that: 1) 3DCH-EMOA outperforms other EMOAs significantly on VAS and Gini coefficient metrics; 2) 3DCH-EMOA performs not as well as other EMOAs on time cost metric.

Table 7: The results of Mann-Whitney test of SPAM problem.

3DCH-EMOA vs	NSGA-II	SPEA2	MOEA/D	SMS-EMOA
VAS	▲	▲	▲	▲
Gini coefficient	▲	▲	▲	▲
Time cost	▽	▽	▽	▽

7. Multiobjective optimization of sparse neural networks

In this section, the proposed algorithm is applied to optimize multiobjective formulation of sparse neural networks to avoid overfitting by seeking parsimonious neural network models and hence to provide better predictions in augmented DET space. The idea of sparse neural network was proposed in [44], in which a fully connected feedforward neural network was pruned through optimization using single objective differential evolution algorithm to produce a sparse network that has good performance on accuracy.

7.1. Multiobjective formulation of sparse neural networks

In this paper, we propose a multiobjective formulation of sparse neural network, in which the performance of neural networks is evaluated in DET space and the sparsity is defined as the complexity objective to be optimized. Besides fpr and fnr, we define ccr by Eq. 15.

$$ccr = \frac{\sum_{i=1}^{M} \mathbf{1}\left\{|w_i| \neq 0\right\}}{M}, \tag{15}$$

where $w_i, i = 1, 2, \ldots, M$ is a weight in the neural network model, and M is the number of weights in total, $\mathbf{1}\{\cdot\}$ is an indicator function.

7.2. UCI dataset

In this section, a total of 19 two-class datasets from the UCI repository [37] are used to evaluate the performance of two EMOAs for sparse neural networks optimization. As we only optimize binary classifiers, while dealing with dataset which contains multiple classes, we split them into several smaller datasets, each of them including a single pair of classes. Both balanced and unbalanced benchmark datasets are included, details are described in Table 8.

Table 8: 19 balanced and unbalanced UCI datasets.

No.	Data Set	No. features	Class Distribution	No.	Data Set	No. features	Class Distribution
1	Australian	14	383:307	11	Vehicle23	18	217:218
2	Breast	9	458:241	12	Vehicle24	18	217:212
3	Glass12	9	51:163	13	Vehicle34	18	218:212
4	Heart	13	139:164	14	Vote	16	267:168
5	Ionosphere	34	126:225	15	Wdbc	30	212:357
6	Parkinsons	22	147:48	16	Wine12	13	59:71
7	Sonar	60	97:111	17	Wine13	13	59:48
8	Spectf	44	95:254	18	Wine23	13	71:48
9	Vehicle12	18	199:217	19	Wpbc	33	46:148
10	Vehicle13	18	199:218				

7.3. Algorithms involved

Two reference evolutionary multiobjective algorithms (NSGA-II, 3DCH-EMOA) and a single objective algorithm SGD (Stochastic Gradient Descend) algorithm [10] are tested. Experiments are performed with Matlab code running on a desktop PC with an i5 3.2GHz processor and 4GB memory under Ubuntu14.04 LTS.

7.4. Parameter setting

The experiment stopping criteria of the two EMOAs are set with a maximum of 20000 function evaluations. The simulated binary crossover (SBX) and polynomial bit flip mutation operators are applied in the experiments with crossover probability of $p_c = 0.9$ and mutation probability of $p_m = 0.1$. The population size is set to 50 for both of EMOAs.

All algorithms mentioned above are used to optimize a multilayer feedforward network with an input layer with size of the number of features of each dataset, two hidden layers with 10 neuron units and an output layer with 2 neuron units. The sigmoid function is selected as activation function in this neural network. For each dataset 50% of samples are randomly selected for model training and the remaining 50% of samples are selected for testing. For each mentioned algorithm, 30 independent trials are conducted on all of the UCI datasets.

7.5. Experimental results and discussion

To evaluate the performance of these algorithms, we compare the statistical results of VAS, Gini coefficient, time cost and classification accuracy in this section. Classification accuracy is an important metric to evaluate the performance of classifiers. It is defined as the partition of the correctly classified samples to all samples in test dataset. In the case of binary classification problems it is denoted by Eq. 16.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}. \tag{16}$$

Table 9 shows the mean and standard deviation of VAS of NSGA-II and 3DCH-EMOA for UCI datasets. In the table VAS is calculated based on the test datasets. By comparing the results we can make a conclusion that the proposed algorithm 3DCH-EMOA outperforms NSGA-II for all the UCI datasets. The results of Mann-Whitney test are listed in Table 10. In the table we can see that 3DCH-EMOA outperforms NSGA-II significantly over most of these datasets, and 3DCH-EMOA performs as well as NSGA-II on five datasets.

Table 9: Mean and standard deviation of VAS of UCI datasets.

Data Set	NSGA-II	3DCH-EMOA	Data Set	NSGA-II	3DCH-EMOA
Australian	$1.47e-01_{1.41e-02}$	$1.54e-01_{1.50e-02}$	Vehicle23	$1.43e-01_{3.76e-02}$	$1.69e-01_{3.77e-02}$
Breast	$2.84e-01_{9.25e-03}$	$2.98e-01_{7.94e-03}$	Vehicle24	$3.89e-02_{2.00e-02}$	$4.34e-02_{1.64e-02}$
Glass12	$1.71e-01_{1.28e-01}$	$1.79e-01_{1.29e-01}$	Vehicle34	$1.36e-01_{3.01e-02}$	$1.69e-01_{3.34e-02}$
Heart	$2.19e-01_{2.62e-02}$	$2.40e-01_{1.73e-02}$	Vote	$3.27e-01_{8.74e-03}$	$3.57e-01_{8.44e-03}$
Ionosphere	$2.45e-01_{3.03e-02}$	$2.71e-01_{1.93e-02}$	Wdbc	$2.91e-01_{5.73e-02}$	$2.95e-01_{5.77e-02}$
Parkinsons	$5.76e-02_{5.29e-02}$	$1.07e-01_{3.37e-02}$	Wine12	$2.05e-01_{1.31e-01}$	$2.99e-01_{5.97e-02}$
Sonar	$1.17e-01_{3.48e-02}$	$1.57e-01_{2.54e-02}$	Wine13	$2.20e-01_{1.22e-01}$	$3.02e-01_{6.16e-02}$
Spectf	$1.09e-01_{7.59e-02}$	$2.02e-01_{3.19e-02}$	Wine23	$7.59e-02_{6.88e-02}$	$1.51e-01_{7.12e-02}$
Vehicle12	$2.38e-01_{6.68e-02}$	$2.86e-01_{1.40e-02}$	Wpbc	$3.56e-02_{5.84e-02}$	$3.71e-02_{5.46e-02}$
Vehicle13	$2.23e-01_{2.19e-02}$	$2.48e-01_{2.43e-02}$			

Table 10: The results of Mann-Whitney test of VAS of UCI datasets.

Data Set	3DCH-EMOA vs NSGA-II	Data Set	3DCH-EMOA vs NSGA-II
Australian	–	Vehicle23	▲
Breast	▲	Vehicle24	–
Glass12	–	Vehicle34	▲
Heart	▲	Vote	▲
Ionosphere	▲	Wdbc	–
Parkinsons	▲	Wine12	▲
Sonar	▲	Wine13	▲
Spectf	▲	Wine23	▲
Vehicle12	▲	Wpbc	–
Vehicle13	▲		

Table 11 shows the mean and standard deviation of Gini coefficient of NSGA-II and 3DCH-EMOA for UCI datasets. In the table Gini coefficient is calculated based on the test datasets. The results of Mann-Whitney test are listed in Table 12. By comparing the results we can make a conclusion that NSGA-II outperforms 3DCH-EMOA for most of the UCI datasets, but NSGA-II does not outperforms 3DCH-EMOA significantly, as it is shown in Table 12. The proposed method does not work well on the metric of Gini coefficient, since the distribution of solutions of these UCI datasets is not uniform. The proposed method can obtain results with good performance of VAS, but can not obtain good results with respect to Gini coefficient. While dealing with real-world classification problems, VAS is more suitable to evaluate the performance of EMOAs.

Table 11: Mean and standard deviation of Gini coefficient of UCI datasets.

Data Set	NSGA-II	3DCH-EMOA	Data Set	NSGA-II	3DCH-EMOA
Australian	$2.78e-01_{2.78e-01}$	$5.97e-01_{5.97e-01}$	Vehicle23	$3.04e-01_{3.04e-01}$	$6.13e-01_{6.13e-01}$
Breast	$3.26e-01_{3.26e-01}$	$5.00e-01_{5.00e-01}$	Vehicle24	$2.75e-01_{2.75e-01}$	$7.20e-01_{7.20e-01}$
Glass12	$4.63e-01_{4.63e-01}$	$4.59e-01_{4.59e-01}$	Vehicle34	$3.12e-01_{3.12e-01}$	$6.22e-01_{6.22e-01}$
Heart	$3.11e-01_{3.11e-01}$	$4.60e-01_{4.60e-01}$	Vote	$4.64e-01_{4.64e-01}$	$5.71e-01_{5.71e-01}$
Ionosphere	$3.40e-01_{3.40e-01}$	$5.97e-01_{5.97e-01}$	Wdbc	$4.86e-01_{4.86e-01}$	$4.87e-01_{4.87e-01}$
Parkinsons	$4.58e-01_{4.58e-01}$	$7.02e-01_{7.02e-01}$	Wine12	$4.02e-01_{4.02e-01}$	$5.24e-01_{5.24e-01}$
Sonar	$2.34e-01_{2.34e-01}$	$6.72e-01_{6.72e-01}$	Wine13	$4.10e-01_{4.10e-01}$	$5.06e-01_{5.06e-01}$
Spectf	$2.07e-01_{2.07e-01}$	$6.26e-01_{6.26e-01}$	Wine23	$2.18e-01_{2.18e-01}$	$6.52e-01_{6.52e-01}$
Vehicle12	$4.34e-01_{4.34e-01}$	$5.52e-01_{5.52e-01}$	Wpbc	$4.08e-01_{4.08e-01}$	$3.22e-01_{3.22e-01}$
Vehicle13	$3.98e-01_{3.98e-01}$	$5.08e-01_{5.08e-01}$			

Table 12: The results of Mann-Whitney test of Gini coefficient of UCI datasets.

Data Set	3DCH-EMOA vs NSGA-II	Data Set	3DCH-EMOA vs NSGA-II
Australian	–	Vehicle23	–
Breast	–	Vehicle24	–
Glass12	–	Vehicle34	–
Heart	–	Vote	–
Ionosphere	–	Wdbc	–
Parkinsons	–	Wine12	–
Sonar	–	Wine13	–
Spectf	–	Wine23	–
Vehicle12	–	Wpbc	–
Vehicle13	–		

Table 13 shows the mean of time cost of NSGA-II, 3DCH-EMOA and SGD for UCI datasets. In the table time cost is computed for training procedure only. The results of Mann-Whitney test of time cost are listed in Table 14. By comparing the results we can make a conclusion that SGD is fast to obtain results, and EMOAs are slow to find weighting vectors. 3DCH-EMOA is much more time consuming when compared to NSGA-II. In the future, more strategies can be adopted to speed up the implementation of 3DCH-EMOA.

Table 13: Mean of time cost/ms of compared algorithms of UCI datasets.

Data Set	NSGA-II	3DCH-EMOA	SGD	Data Set	NSGA-II	3DCH-EMOA	SGD
Australian	9.78e + 04	2.29e + 06	2.15e + 03	Vehicle23	3.40e + 04	2.47e + 06	2.97e + 03
Breast	9.02e + 04	1.88e + 06	1.74e + 03	Vehicle24	3.00e + 04	1.88e + 06	7.85e + 02
Glass12	9.44e + 04	1.79e + 06	5.08e + 02	Vehicle34	2.89e + 04	1.95e + 06	1.25e + 03
Heart	4.56e + 04	1.91e + 06	5.43e + 02	Vote	2.91e + 04	1.97e + 06	4.38e + 02
Ionosphere	5.05e + 04	2.43e + 06	1.38e + 03	Wdbc	2.94e + 04	2.33e + 06	3.22e + 03
Parkinsons	4.86e + 04	1.86e + 06	5.29e + 02	Wine12	2.93e + 04	1.72e + 06	7.75e + 02
Sonar	4.73e + 04	2.58e + 06	2.09e + 03	Wine13	2.97e + 04	1.71e + 06	7.22e + 02
Spectf	4.39e + 04	2.55e + 06	2.09e + 03	Wine23	3.25e + 04	1.78e + 06	6.75e + 02
Vehicle12	3.14e + 04	1.91e + 06	1.18e + 03	Wpbc	3.25e + 04	3.54e + 06	5.25e + 01
Vehicle13	3.10e + 04	2.02e + 06	1.20e + 03				

Table 14: The results of Mann-Whitney test of time cost of UCI datasets.

3DCH-EMOA vs	NSGA-II	SGD	3DCH-EMOA vs	NSGA-II	SGD
Australian	▽	▽	Vehicle23	▽	▽
Breast	▽	▽	Vehicle24	▽	▽
Glass12	▽	▽	Vehicle34	▽	▽
Heart	▽	▽	Vote	▽	▽
Ionosphere	▽	▽	Wdbc	▽	▽
Parkinsons	▽	▽	Wine12	▽	▽
Sonar	▽	▽	Wine13	▽	▽
Spectf	▽	▽	Wine23	▽	▽
Vehicle12	▽	▽	Wpbc	▽	▽
Vehicle13	▽	▽			

Moreover, classification accuracy is compared in this part. Table 15 shows the mean and standard deviation of accuracy obtained by NSGA-II, 3DCH-EMOA and SGD for UCI datasets. In this part only the best result in the population of EMOAs is listed in the table. From the table we can see that 3DCH-EMOA outperforms other algorithms for most of the datasets. To compare the results, the accumulation of accuracy across these UCI datasets is shown in Fig. 11. From the figure, we can make some conclusions: 1) EMOAs can obtain better accuracy results than SGD; 2) 3DCH-EMOA outperforms NSGA-II for these UCI datasets. The results of Mann-Whitney test are listed in Table 16. By comparing the results we can see that 3DCH-EMOA outperforms NSGA-II significantly over most of these datasets, and 3DCH-EMOA performs as good as SGD over all these datasets.

Table 15: Mean and standard deviation of classification accuracy on UCI datasets.

Data Set	NSGA-II	3DCH-EMOA	SGD	Data Set	NSGA-II	3DCH-EMOA	SGD
Australian	$0.69_{0.02}$	$0.70_{0.02}$	$0.68_{0.02}$	Vehicle23	$0.69_{0.05}$	$0.71_{0.05}$	$0.51_{0.07}$
Breast	$0.88_{0.01}$	$0.89_{0.01}$	$0.88_{0.02}$	Vehicle24	$0.56_{0.03}$	$0.57_{0.03}$	$0.48_{0.02}$
Glass12	$0.85_{0.08}$	$0.85_{0.08}$	$0.76_{0.03}$	Vehicle34	$0.68_{0.04}$	$0.71_{0.05}$	$0.52_{0.08}$
Heart	$0.79_{0.04}$	$0.81_{0.02}$	$0.80_{0.03}$	Vote	$0.95_{0.01}$	$0.96_{0.01}$	$0.96_{0.01}$
Ionosphere	$0.85_{0.04}$	$0.87_{0.03}$	$0.89_{0.03}$	Wdbc	$0.90_{0.05}$	$0.90_{0.06}$	$0.86_{0.08}$
Parkinsons	$0.77_{0.03}$	$0.77_{0.10}$	$0.75_{0.03}$	Wine12	$0.80_{0.17}$	$0.90_{0.09}$	$0.50_{0.06}$
Sonar	$0.66_{0.05}$	$0.71_{0.03}$	$0.74_{0.03}$	Wine13	$0.82_{0.14}$	$0.90_{0.08}$	$0.56_{0.14}$
Spectf	$0.76_{0.04}$	$0.78_{0.02}$	$0.79_{0.03}$	Wine23	$0.63_{0.06}$	$0.70_{0.11}$	$0.59_{0.05}$
Vehicle12	$0.82_{0.09}$	$0.87_{0.03}$	$0.80_{0.03}$	Wpbc	$0.76_{0.04}$	$0.76_{0.03}$	$0.76_{0.04}$
Vehicle13	$0.78_{0.03}$	$0.82_{0.05}$	$0.76_{0.04}$				

Table 16: The results of Mann-Whitney test of accuracy of UCI datasets.

3DCH-EMOA vs	NSGA-II	SGD	3DCH-EMOA vs	NSGA-II	SGD
Australian	–	–	Vehicle23	–	–
Breast	▲	–	Vehicle24	–	–
Glass12	–	–	Vehicle34	▲	–
Heart	▲	–	Vote	▲	–
Ionosphere	▲	–	Wdbc	–	–
Parkinsons	▲	–	Wine12	▲	–
Sonar	▲	–	Wine13	▲	–
Spectf	▲	–	Wine23	▲	–
Vehicle12	▲	–	Wpbc	–	–
Vehicle13	▲	–			

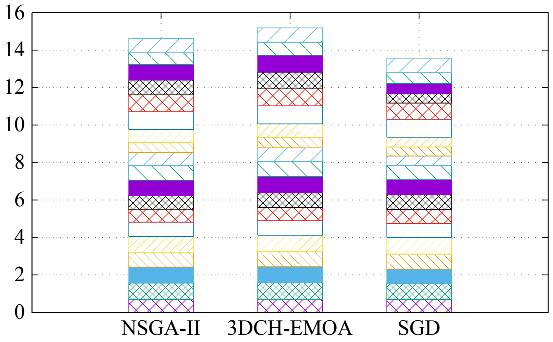


Figure 11: The accumulation of classification accuracy of 19 UCI datasets for NSGA-II, 3DCH-EMOA and SGD. Boxes from bottom to top for each method represent the average accuracy for datasets in Table 8.

8. Conclusions and Future Work

In this paper, we analyzed the properties of augmented DET convex hull (ADCH) maximization problem. 3DCH-EMOA is proposed to optimize the performance of augmented DET for classification. In order to evaluate the performance of several EMOAs a set of test problems ZEJD is designed. 3DCH-EMOA is compared with other EMOAs, such as NSGA-II, GDE3, SPEA2, MOEA/D and SMS-EMOA on ZEJD test problems. 3DCH-EMOA always obtains the best results not only for convergence but also for diversity metrics. By avoiding concave regions, 3DCH-

EMOA is able to focus on relevant parts of the Pareto front, that is, parts that contribute to a high value of VAS. We also applied this algorithm to the real-world applications of spam filtering and multiobjective optimization of sparse neural networks. Testing performance of the newly proposed method and comparing it to state-of-the-art approaches on a number of experimental studies indicate that the proposed algorithm is promising and effective.

However, the new proposed method is time consuming, because it needs to compute the VAS contribution of every point in the first priority layer solutions. This is a drawback of the proposed 3DCH-EMOA approach. It is, however, less important if the evaluations of classifiers are relatively expensive. In the future, more effective strategies will be adopted to reduce the computational complexity.

Acknowledgment

This work was partially supported by the National Basic Research Program (973 Program) of China (No. 2013CB329402), the National Natural Science Foundation of China (No. 61473215, 61371201, 61373111, 61303032, 61271301, 61272279, 61203303 and 61571342), the Excellent Young Science Foundation of China (No. 61522311), the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048), the Major Research Plan of the National Natural Science Foundation of China (No. 91438201 and 91438103), the Program for Cheung Kong Scholars and Innovative Research Team in University (No. IRT_15R53), and the European Union Seventh Framework Programme under Grant agreements (No. 247619) on Nature Inspired Computation and its Applications (NICaiA). Asep Maulana thanks for financial support by the Indonesian Endowment Grant.

Appendix A. ZEJD Problem

Three ZEJD (Zhao, Emmerich, Jiao, Deutz) problems are designed to evaluate the performance of several kinds of EMOAs on ADCH maximization problems and the general principle of their construction is derived in [21]. These test problems are simulation of augmented DET distribution of complexity classifiers, which has several important properties. Firstly, the points (1,0,0), (0,1,0) and (0,0,1) included in the Pareto front are the extremal points of the Pareto front. Note that the point (0,0,1) would correspond to a perfect classfier which uses all the rules. Secondly, the Pareto front should be above the augmented DET surface of random guessing classifiers which is described in Fig. 2. Thirdly, all of the solutions are in the space of the unit cube. The objective of this set of test problems is to find the maximum value of the volume under the convex hull surface. The range of variation of each object is in [0,1], the problem of ZEJD1 is defined in Eq. A.1.

$$\begin{cases} f_1 = 1 - \sqrt{2} cos(x_1 * \pi/2)(1 - x_3) \\ f_2 = 1 - \sqrt{2} sin(x_1 * \pi/2) cos(x_2 * \pi/2)(1 - x_3) \\ f_3 = 1 - \sqrt{2} sin(x_1 * \pi/2) sin(x_2 * \pi/2)(1 - x_3) \end{cases} \tag{A.1}$$

where x_1, x_2, x_3 are all in $[0, 1]$ and f_1, f_2, f_3 are all in $[0, 1]$.

The Pareto front of ZEJD1 is shown in Fig. A.12(a), which is a convex surface. The solutions of EMOAs with good performance can cover the Pareto front uniformly. Both ZEJD2 and ZEJD3 problems are versions of ZEJD1 modified by additional dent on the surface, in which some parts of Pareto Front are not on the convex hull, the Pareto front of ZEJD2 is discontinuous and the Pareto front of ZEJD3 is continuous. These two test problems are designed to test whether the algorithms can avoid the dent areas, i.e., finding solutions only on the convex part of the Pareto front. ZEJD2 is defined by Eq. A.2. A dent is made in the area satisfied $f_1 < a, f_2 < a, g < a$, by making the function decrease slowly. In our experiments we set $a = 0.3, \lambda = 0.5$. The Pareto front of ZEJD2 is shown in Fig. A.12(b). ZEJD3 is defined by Eq. A.3. A dent is made by adding a surface $d(x, y)$. In order to keep the points (1,0,0), (0,1,0) and (0,0,1) in the Pareto front, $d(0, 0)$ is subtracted to obtain f_3. In this paper, we set $A = 0.15, \gamma = 400$. The Pareto front of ZEJD3 is shown in Fig. A.12(c). The objectives of both ZEJD2 and ZEJD3 are f_1, f_2 and f_3, $f_1 \in [0, 1]$, $f_2 \in [0, 1]$, $f_3 \in [0, 1]$.

$$\begin{cases} f_1 = 1 - \sqrt{2}cos(x_1 * \pi/2)(1 - x_3) \\ f_2 = 1 - \sqrt{2}sin(x_1 * \pi/2)cos(x_2 * \pi/2)(1 - x_3) \\ f_3 = \begin{cases} a + \lambda(g - a) & \text{if } f_1 < a, f_2 < a, g < a \\ g & \text{else} \end{cases} \\ g = 1 - \sqrt{2}sin(x_1 * \pi/2)sin(x_2 * \pi/2)(1 - x_3) \end{cases} \tag{A.2}$$

$$\begin{cases} f_1 = 1 - \sqrt{2}cos(x_1 * \pi/2)(1 - x_3) \\ f_2 = 1 - \sqrt{2}sin(x_1 * \pi/2)cos(x_2 * \pi/2)(1 - x_3) \\ f_3 = \begin{cases} k(f_1, f_2) & \text{if } k(f_1, f_2) > 0 \\ 0 & \text{else} \end{cases} \\ g = 1 - \sqrt{2}sin(x_1 * \pi/2)sin(x_2 * \pi/2)(1 - x_3) \\ d(x, y) = A * e^{-\gamma\{(x-0.173)^2+(y-0.173)^2\}} \\ k(f_1, f_2) = g + d(f_1, f_2) - d(0, 0) \end{cases} \tag{A.3}$$

References

[1] The Apache SpamAssassin Project (2011). `http://spamassassin.apache.org`.

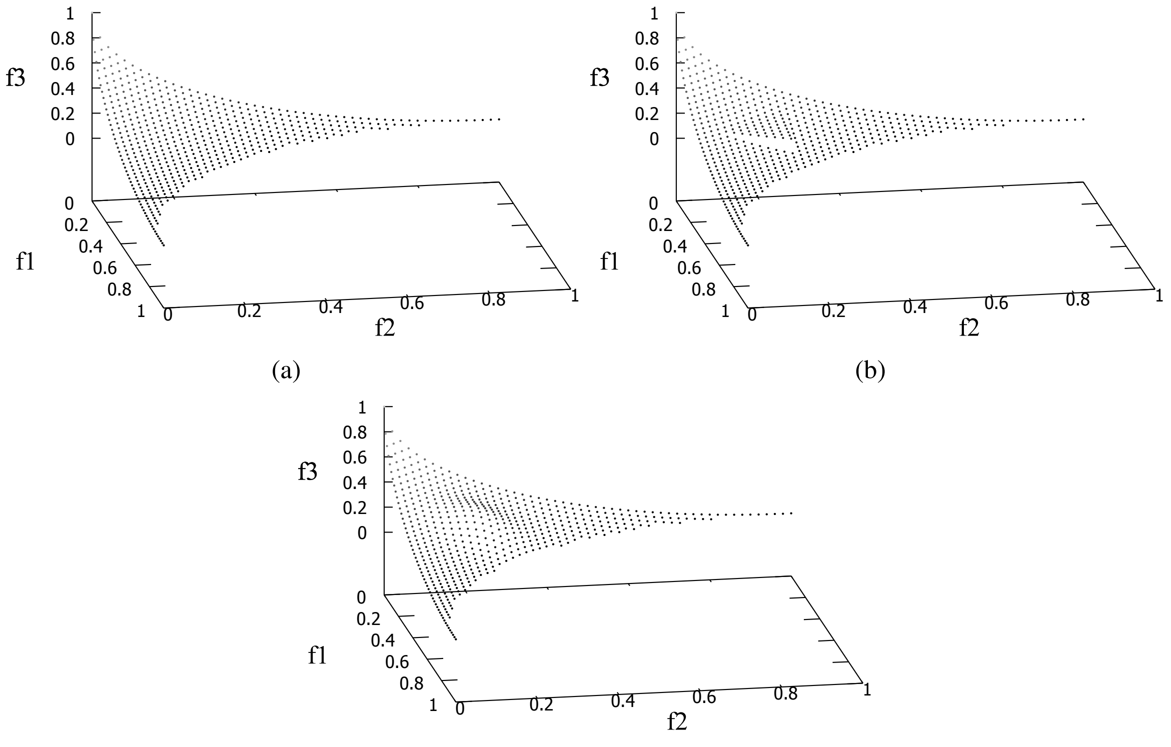
[2] W. A. Albukhanajer, J. A. Briffa, Y. Jin, Evolutionary multi-objective image feature extraction in the presence of noise, IEEE Transactions on Cybernetics 45 (2014) 1757–1768.

[3] R. Ariew, Ockham's razor: A historical and philosophical analysis of Ockham's principle of parsimony, Ph.D. thesis, Champaign-Urbana (1976).

[4] C. B. Barber, D. P. Dobkin, H. Huhdanpaa, The quickhull algorithm for convex hulls, ACM Transactions on Mathematical Software (TOMS) 22 (4) (1996) 469–483.

[5] M. Barreno, A. A. Cárdenas, J. D. Tygar, Optimal ROC curve for a combination of classifiers, in: Proceedings of the 21st Annual Conference on Neural Information Processing Systems: Advances in Neural Information Processing Systems 20 (NIPS 2007, Vancouver, Canada, December 3–5, 2007), MIT Press, 2008, pp. 57–64.

[6] V. Basto-Fernandes, I. Yevseyeva, J. R.Méndez, Anti-spam multiobjective genetic algorithms optimization analysis, International Resource Management Journal 26 (2012) 54–67.



(a)

(b)

(c)

Figure A.12: Pareto fronts of three ZEJD test problems. (a) The Pareto front of ZEJD1 test problem. (b) The Pareto front of ZEJD2 test problem. (c) The Pareto front of ZEJD3 test problem.

[7] N. Beume, B. Naujoks, M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, European Journal of Operational Research 181 (3) (2007) 1653–1669.

[8] U. Bhowan, M. Johnston, M. Zhang, X. Yao, Evolving diverse ensembles using genetic programming for classification with unbalanced data, IEEE Transactions on Evolutionary Computation 17 (3) (2013) 368–386.

[9] U. Bhowan, M. Zhang, M. Johnston, Multi-objective genetic programming for classification with unbalanced data, in: Proceedings of the 22nd Australasian Joint Conference: Advances in Artificial Intelligence, (AI 2009, Melbourne, Australia, December 1–4, 2009), Springer, 2009, pp. 370–380.

[10] L. Bottou, Stochastic gradient learning in neural networks, in: Proceedings of the 4th International Conference on Neural Networks and Their Applications (Neuro-Nîmes 1991, Nîmes, France), 1991, pp. 687–706.

[11] C. Bourke, K. Deng, S. D. Scott, R. E. Schapire, N. V. Vinodchandran, On reoptimizing multi-class classifiers, Machine Learning 71 (2-3) (2008) 219–242.

[12] C. C. Chang, C. J. Lin, LIBSVM: A library for support vector machines, ACM Transactions on Intelligent Systems & Technology 2 (3) (2011) 1–27.

[13] C. Chatelain, S. Adam, Y. Lecourtier, L. Heutte, T. Paquet, A multi-model selection framework for unknown and/or evolutive misclassification cost problems, Pattern Recognition 43 (3) (2010) 815–823.

[14] M. Cococcioni, P. Ducange, B. Lazzerini, F. Marcelloni, A new multi-objective evolutionary algorithm based on convex hull for binary classifier optimization, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2007, Singapore, September 25–28, 2007), IEEE Press, 2007, pp. 3150–3156.

[15] Y.-L. Chen, C.-C. Wu, K. Tang, Time-constrained cost-sensitive decision tree induction, Information Sciences 354 (2016) 140–152.

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

[17] P. Ducange, B. Lazzerini, F. Marcelloni, Multi-objective genetic fuzzy classifiers for imbalanced and cost-sensitive datasets, Soft Computing 14 (7) (2010) 713–728.

[18] J. J. Durillo, A. J. Nebro, jMetal: A java framework for multi-objective optimization, Advances in Engineering Software 42 (10) (2011) 760–771.

[19] J. J. Durillo, A. J. Nebro, E. Alba, The jMetal framework for multi-objective optimization: Design and architecture, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2010, Barcelona, Spain, 18-23 July 2010), IEEE Press, 2010, pp. 1–8.

[20] J. P. Egan, Signal detection theory and ROC analysis, Academic Press, New York, USA, 1975.

[21] M. T. Emmerich, A. H. Deutz, A family of test problems with Pareto-fronts of variable curvature based on super-spheres, in: Proceedings of the 18th International Conference on Multicriteria Decision Making (MCDM 2006, Chania, Crete, Greece, June 19–23, 2006), 2006.

[22] R. M. Everson, J. E. Fieldsend, Multi-class ROC analysis from a multi-objective optimisation perspective, Pattern Recognition Letters 27 (8) (2006) 918–927.

[23] T. Fawcett, Using rule sets to maximize ROC performance, in: Proceedings of the IEEE International Conference on Data Mining, (ICDM 2001, San Jose, California, USA, 29 November – 2 December, 2001), IEEE Press, 2001, pp. 131–138.

[24] T. Fawcett, An introduction to ROC analysis, Pattern recognition letters 27 (8) (2006) 861–874.

[25] T. Fawcett, PRIE: A system for generating rule lists to maximize ROC performance, Data Mining and Knowledge Discovery 17 (2) (2008) 207–224.

[26] P. A. Flach, S. Wu, Repairing concavities in ROC curves, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI'05, Edinburgh, Scotland, UK, 30 July – 5 August, 2005), Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2005, pp. 702–707.

[27] L. Gräning, Y. Jin, B. Sendhoff, Generalization improvement in multi-objective learning, in: Proceedings of the International Joint Conference on Neural Networks (IJCNN 06, Vancouver, Canada, July 16-21, 2006), IEEE Press, 2006, pp. 4839–4846.

[28] D. J. Hand, R. J. Till, A simple generalisation of the area under the ROC curve for multiple class classification problems, Machine Learning 45 (2) (2001) 171–186.

[29] C. Igel, M. Kreutz, Operator adaptation in evolutionary computation and its application to structure optimization of neural networks, Neurocomputing 55 (1-2) (2003) 347–361.

[30] S.-F. Ji, W.-X. Sheng, Z.-W. Jing, The multi-objective differential evolution algorithm based on quick convex hull algorithms, in: Proceedings of the 5th International Conference on Natural Computation (ICNC'09, Tianjin, China, August 14-16, 2009), Vol. 4, IEEE Press, 2009, pp. 469–473.

[31] L. Jiao, L. Li, R. Shang, F. Liu, R. Stolkin, A novel selection evolutionary strategy for constrained optimization, Information Sciences 239 (1) (2013) 122–141.

[32] L. Jiao, J. Luo, R. Shang, F. Liu, A modified objective function method with feasible-guiding strategy to solve constrained multi-objective optimization problems, Applied Soft Computing 14 (1) (2014) 363–380.

[33] Y. Jin, B. Sendhoff, Pareto-based multiobjective machine learning: An overview and case studies, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38 (3) (2008) 397–415.

[34] S. Kukkonen, J. Lampinen, GDE3: The third evolution step of generalized differential evolution, in: Proceedings of the IEEE Congress on Evolutionary Computation (CEC 2005, Edinburgh, UK, 2-4 September 2005), Vol. 1, IEEE Press, 2005, pp. 443–450.

[35] M. A. Kupinski, M. A. Anastasio, Multiobjective genetic optimization of diagnostic classifiers with implications for generating receiver operating characteristic curves, IEEE Transactions on Medical Imaging 18 (8) (1999) 675–685.

[36] L. Li, X. Yao, R. Stolkin, M. Gong, S. He, An evolutionary multiobjective approach to sparse reconstruction, IEEE Transactions on Evolutionary Computation 18 (6) (2014) 827–845.

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

[38] X. Lu, K. Tang, X. Yao, Evolving neural networks with maximum AUC for imbalanced data classification, in: Proceedings of the 5th International Conference on Hybrid Artificial Intelligence Systems, Part I (HAIS 2010, San Sebastián, Spain, June 23-25, 2010), Springer, 2010, pp. 335–342.

[39] J. Luo, L. Jiao, L. A. Lozano, A sparse spectral clustering framework via multi-objective evolutionary algorithm, IEEE Transactions on Evolutionary Computation (2015), http://dx.doi.org/10.1109/TEVC.2015.2476359 doi:10.1109/TEVC.2015.2476359.

[40] H. B. Mann, D. R. Whitney, On a test of whether one of two random variables is stochastically larger than the other, Annals of Mathematical Statistics 18 (1) (1947) 50–60.

[41] A. F. Martin, G. R. Doddington, T. Kamm, M. Ordowski, M. A. Przybocki, The DET curve in assessment of detection task performance, in: G. Kokkinakis, N. Fakotakis, E. Dermatas (Eds.), Proceeding of the Fifth European Conference on Speech Communication and Technology (EUROSPEECH 1997, Rhodes, Greece, September 22-25, 1997), ISCA, 1997, pp. 1895–1898.

[42] J. Mason, The Apache SpamAssassin public corpus (2005). http://spamassassin.apache.org/publiccorpus.

[43] M. D. Monfared, A. Mohades, J. Rezaei, Convex hull ranking algorithm for multi-objective evolutionary algorithms, Scientia Iranica 18 (6) (2011) 1435–1442.

[44] P. H. Morgan, Differential evolution and sparse neural networks, Expert Systems 25 (4) (2008) 394–413.

[45] A. J. Nebro, J. J. Durillo, On the effect of applying a steady-state selection scheme in the multi-objective genetic algorithm NSGA-II, in: R. Chiong (Ed.), Nature-Inspired Algorithms for Optimisation, 2009, pp. 435–456.

[46] J. O'Rourke, A. J. Mallinckrodt, Computational Geometry in C, Cambridge University Press, 1998.

[47] R. C. Prati, P. A. Flach, ROCCER: An algorithm for rule learning based on ROC analysis, in: Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI-05, Edinburgh, Scotland, UK, July 30-August 5, 2005), Vol. 26, 2005, pp. 823–828.

[48] F. Provost, T. Fawcett, Robust classification for imprecise environments, Machine Learning 42 (3) (2001) 203–231.

[49] S. D. Río, V. López, J. M. Benítez, F. Herrera, On the use of mapreduce for imbalanced big data using random forest, Information Sciences 285 (2014) 112–137.

[50] H. C. Sox, M. C. Higgins, D. K. Owens, Medical decision making, 2nd Edition, Wiley-Blackwell, 2013.

[51] A. Srinivasan, Note on the location of optimal classifiers in N-dimensional ROC space, Tech. Rep. PRG-TR-2-99, Oxford University Computing Laboratory, Oxford, UK (November 1999).

[52] J. A. Swets, Measuring the accuracy of diagnostic systems, Science 240 (4857) (1988) 1285–1293.

[53] K. Tang, R. Wang, T. Chen, Towards maximizing the area under the ROC curve for multi-class classification problems, in: Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI 2011, San Francisco, California, USA, August 7-11, 2011), AAAI Press, 2011, pp. 483–488.

[54] H. Wang, L. Jiao, X. Yao, Two_Arch2: an improved two-archive algorithm for many-objective optimization, IEEE Transactions on Evolutionary Computation 19 (4) (2015) 524–541.

[55] P. Wang, M. Emmerich, R. Li, K. Tang, T. Bäck, X. Yao, Convex hull-based multi-objective genetic programming for maximizing receiver operator characteristic performance, IEEE Transactions on Evolutionary Computation 19 (2) (2015) 188–200.

[56] P. Wang, K. Tang, T. Weise, E. Tsang, X. Yao, Multiobjective genetic programming for maximizing ROC performance, Neurocomputing 125 (2014) 102–118.

[57] Z. Wang, Q. Zhang, A. Zhou, M. Gong, L. Jiao, Adaptive replacement strategies for MOEA/D, IEEE Transactions on Cybernetics 46 (2) (2016) 474–486. http://dx.doi.org/10.1109/TCYB.2015.2403849 doi:10.1109/TCYB.2015.2403849.

[58] G. Wu, W. Pedrycz, P. N. Suganthan, R. Mallipeddi, A variable reduction strategy for evolutionary algorithms handling equality constraints, Applied Soft Computing 37 (C) (2015) 774–786.

[59] I. Yevseyeva, V. Basto-Fernandes, D. Ruano-Ordás, J. R. Méndez, Optimising anti-spam filters with evolutionary algorithms, Expert Systems with Applications 40 (10) (2013) 4010–4021.

[60] S. Yitzhaki, Relative deprivation and the Gini coefficient, Quarterly Journal of Economics 93 (2) (1979) 321–324.

[61] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Transactions on Evolutionary Computation 11 (6) (2007) 712–731.

[62] H. Zhao, A multi-objective genetic programming approach to developing Pareto optimal decision trees, Decision Support Systems 43 (3) (2007) 809–826.

[63] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the strength Pareto evolutionary algorithm, TIK Report 103, Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Zurich, Switzerland (2001).