Quadcriteria Optimization of Anti-spam Filtering on MOEA/D

THESIS

submitted in partial fulfillment of the
requirements for the degree of

BACHELOR OF SCIENCE
in
COMPUTER SCIENCE

Author :	Zhiyun Xia
Student ID :	s1731424
Supervisor :	Micheal T. Emmerich
2nd corrector :	Vitor Manuel Basto Fernandes

Leiden, The Netherlands, June 2, 2016

Quadcriteria Optimization of Anti-spam Filtering on MOEA/D

Zhiyun Xia

LIAC, Leiden University

June 2, 2016

Abstract

This paper focuses on the four dimensions optimization of email anti-spam filtering using MOEA/D. The four objectives are error rate (including false positives and false negatives), reliability and complexity.
Anti-spam filtering, which in essence a binary classifier, serves to classify emails to ham or spam. And for instances ambiguous to be assigned to either side, the classifier marks them as suspicious. In this three-way classification, we aim to minimize the false positives and false negatives to get least misclassified emails. The amount of suspicious is meanwhile minimized as the third objective to guarantee a high reliability. The forth objective is complexity, which is interpreted as the number of rules activated in the filtering.
We use MOEA/D evolutionary algorithm to solve this multi-objective problem on different crossover (UniformCrossover and SBXSinglePointCrossover). Moreover, the resulting 4D Pareto Front is assessed using indicator HV, SPREAD, EPLISON and 3D visualization. NSGAII is also performed as a comparison in our study of MOEA/D.

1 Introduction

As nowadays plenty of work and leisure are shifting to internet and being carried out through emails, spam becomes a severe trouble and burden over our mailbox. In this situation, comparing to manual classification, automatic e-mail filtering system seems to be a more effective method here. Currently, most commercial and open source spam filters are basically developed on machine learning or pattern recognition techniques. However, the former usually requires a representative and large training set of labelled instances to learn the difference between spam and ham, let alone the ever-changing disguised tricks on spam. The latter method, though avoids the obstacle of large and changing training set, suffers from the drawback that users have to label sufficient number of emails to establish a user-specific filter.
As one alternative, Apache SpamAssassin[1], a well-known and widely used open source anti-spam platform, provides about nine hundred pre-defined binary valued rules to users. However, as too many rules will create redundancy and slow the calculation, usually we just activate part of them in judgement. Each of these rules aims at detecting a given characteristic of spam or ham, like the presence of a given keyword in the emails text or some kind of email header malformation. A zero is given to the rule if the corresponding characteristic does not present in the email; on the contrary a one is given. In addition to give scores to activated rules, the relative importance (weight) of these rules also needs careful decision. The weights can be positive or negative according to the rule features associated with spam or ham. At last, the weighted scores of all rules of a given email are summed up to obtain a final score. If the final score is smaller than the lower threshold, the email is labelled as ham. And if it is greater than the upper threshold it will be labelled as spam, otherwise labelled as suspicious. In this progress, the activated rules, the weights and the decision thresholds can be modified by user, either automatically or manually. SpamAssassin provides a tool named "mass check" and a training set of labeled emails for automatic modification. Here, as a replacement of mass check, we will explore the best value of these variables using multiobjective optimization algorithm. Works of score tuning accomplished on multi-objective optimization approach including NSGAII, SPEAII and SMS-EMOA has been done in recent works [2]. In this paper, we will expand the study to MOEA/D algorithm.
MOEA/D is an evolutionary multi-objective algorithm based on decomposition, first proposed in paper [3] by Qingfu Zhang and Hui Li in 2007. It combines the decomposition techniques of traditional multi-

objective optimization into evolutionary optimization. The main idea is to decompose a multi-objective optimization problem into a number of scalar optimization subproblems, each of them representing an individual solution in the population. Decomposition approaches like Weighted Sum Approach, Techebycheff Approach and Boundary Intersection Approach are commonly used in this process. Another important notion in MOEA/D is ″neighbor″, which is defined as all the subproblems with the weight vectors from the neighborhood of weight vector of current subproblem. In MOEA/D, each subproblem is optimized by only its neighbors, instead of the whole population since two neighboring subproblems should have close optimal solutions. Study shows that MOEA/D has a lower computational complexity and can outperform or perform similarly to MOGLS and NSGA-II on multiobjective knapsack problems and continuous multiobjective problems. So in this paper we attempt to apply it to solve the anti-spam problem. The paper is organized as follows. In section 2 the way to formulate anti-spam as a multi-objective optimization problem is presented. Section 3 demonstrates some specific algorithm settings and section 4 details the result of experiment, including the analysis of performance and comparison of crossover and NSGAII. Section 5 concludes this paper.

2 Problem Formulation

The difficulty of anti-spam filtering system lies on the fact that there's more than one criterion for a good classification. First and foremost, as the top target, spam and ham should be correctly marked at the greatest extent, in other words, the error rate needs to be minimized. So to avoid false classification, some ambiguous instances which swing between spam and ham should be assigned to neither sides, but marked as suspicious as a compromise. This can be an effective mean to improve the accuracy of prediction, however, too many suspicious would meanwhile reduce the coverage of spam/ham classification thus weaken the effect of automatic filtering. Making a larger set of rules participate in the judgement is another feasible method to decrease the error rate, but it will also increase the complexity, which is another objective cannot be neglected. These important but conflicting goals makes the anti-spam filtering a complexed multi-objective optimization problem and it becomes impossible to get one best solution as a result. In this situation, what we can accomplish instead is to classify some possible solutions better than others to form a non-dominated solution set thus strike a balance between all these objectives, and the goals mentioned above can be concluded as follows:

Negative Rate (FNR)	The proportion of spam detected as ham to the total number of spam
False Positive Rate (FPR)	The proportion of emails tagged as spam but actually ham to the total number of ham
Unclassified Rate (UR)	The proportion of suspicious email to the total number of instances
Complexity Rate (CR)	The proportion of active rules to the total number of used rules

In particular, when calculating CR, though there are 2440 rules available in SpamAssassin, only those fitting at least one message in the database are considered in the experiment, so the number of used rules is actually 330.
Having the objectives of optimization, the decision space also needs to be determined. As we have discussed, the variables to modify are activated rules, weights and threshold. The activeness of rule can be represented by binary values while weights and thresholds need real values. Therefore, a mixed-integer decision space is chosen.
A binary vector $b = (b_1, b_2, \dots b_n) \in \{0,1\}^n$ is defined to indicate the activeness of rules. $b_i = 0$ means rule i is inactive and this rule will not be considered in the prediction of spam, while $bi = 1$ means this rule is active and will affect the judgment. The weights of rules is denoted as a real number vector $w = (w_1, w_2, \dots w_n)$, where n is the number of used rules and w_i is the weight of rule i. For each instance, its binary scores of rules would be weighted by this vector to compute a final score. In addition, two real numbers t_1 and t_2 are used to denote lower and upper threshold. To make sure the upper bound is greater than the lower bound, we use $t_1 + t_2$ instead of t_2 as the upper threshold. That is, instances with score lower than t_1 would be marked as ham, while instances with score greater than $t_1 + t_2$ would be marked as spam. And those between t_1 and $t_1 + t_2$ are tagged as suspicious.
Having the decision variables and objective functions, now the anti-spam classifier can be formulated as a multi-objective optimization problem described by normalized Equations 1-4.

$$FNR(w;b;t1;t2) = fn(w;b;t1;t2) = TotalNumberOfSpamMessages \tag{1}$$

$$FPR(w;b;t1;t2) = fp(w;b;t1;t2) = TotalNumberOfHamMessages \tag{2}$$

$$UR(w;b;t1;t2) = unclassified(w;b;t1;t2) = TotalNumberOfMessages \tag{3}$$

$$CR(w;b;t1;t2) = \sum_{i=1}^{n} b_i / NumberOfUsedRules \tag{4}$$

with w being the rule weights, b being the activeness of rules and t_1,t_2 being the thresholds of spam and ham.

3 Algorithm settings

3.1 Parameters

The experiment is performed on jMetal[4] version 4.5, a framework in Java designed for Metaheuristic Algorithms including MOEA/D. The decision variables are denoted by an ArrayRealandBianry solution type for the mixed-integer decision space. An ArrayReal variable and a Binary variable is defined in the solution to represent weight vector and activness vector respectively. The lower and upper thresholds are attached to the end of the weight vector in the ArrayReal variable. The weight of each rule is restricted in the range $[-5,5]$ and the activeness of rule is set to be in $\{0,1\}^n$. As MOEA/D tends to lose variety of population rapidly, $t1$ and $t2$, instead of being $[0,1]$ as the setting of original experiment in [2], expand to a wider range of $[-5,5]$ and $[0,5]$ respectively.
The population size and evaluation time are set to be 300 and 25000 as default. The size of the neighborhood is set to be one-tenth of the population size, which is 30 and the maximal number of solutions replaced by each child solution is set to be 3. Probability that parents are selected from neighborhood is 0.9, meaning that they still have a 0.1 possibility to be selected from the whole population.
In MOEA/D, a set of even spread weight vectors $\{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ is needed for decomposing multi-objective problem into single objective function. The directory with the files containing the weight vectors of two, three and five dimensions used in [5] by Q. Zhang, W. Liu, and H Li is available at [6]. Here a four dimensions weight vector is needed, so we choose to generate weight vectors in a method referring to [7]. In details, a real number generator would create four values in $[0,1]$ randomly. Then a transformation would be done to make sure the sum of the four numbers is exactly 1 to form one weight vector. A large amount of candidate weight vectors are randomly created in this way

and finally those with largest distance to others are selected as a set of even spread weight vectors.

3.2 Indicator

The performance assessment is also a notable step in optimization. Contrary to single-objective optimization, where the algorithm with the best solution is absolutely the optimal, what we obtained from multiobjective optimization is an approximate set to the Pareto front of the problem, which makes it inapplicable to compare them directly. Thus as a substitution, some kinds of indicators have to be defined in order to assess the quality of the obtained solution set. A number of quality indicators have been proposed in past literature. Generational Distance (GD) was first raised in [8], then comes the Inverse Generational Distance (IGD) and Hypervolume (HV) in [9], Epsilon in [10], Spread or Δ in [11]. Though there are various indicators trying to assess the solution set from different perspectives, all these indicators are leaded by mainly two criteria, the convergence or diversity(or both). The mainstream indicators are divided based on which aspects they measure as figure 1 given by paper [12].

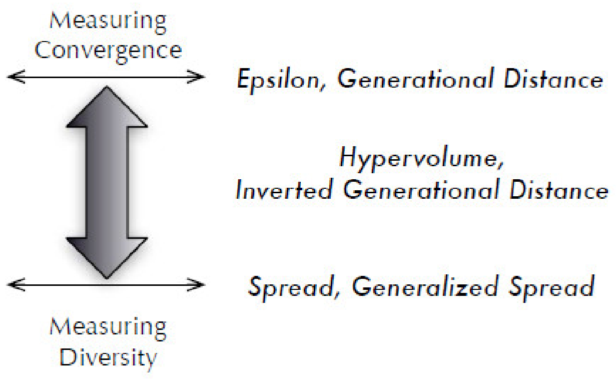

Measuring Convergence
Epsilon, Generational Distance
Hypervolume,
Inverted Generational Distance
Spread, Generalized Spread
Measuring Diversity

Figure 1: *Measures of performance*

In this experiment, to cover both convergence and diversity, indicator Epsilon, HV and Spread are selected to assess the performance. Among them, Epsilon-indicator is defined as:

$$I_{\varepsilon 1}(A) = I_{\varepsilon}(A, P) = \inf_{\varepsilon \in R}\{\forall z^2 \in P \exists z^1 \in A : z^1 \geq_{\varepsilon} z^2\} \tag{5}$$

where P is the optimal Pareto front or any reference set R if P is unknown.

Epsilon Indicator is fast to compute, but a reference set is required. It well measures the convergence of solution set through calculating minimal translation to apply on solution set so that every solution in reference set is dominated by at least one solution of it.
HV represents the size of the space covered or size of dominated space. It is the Lebesgue measure $\bigwedge$ of the union of hypercubes a_i defined by a non-dominated point m_i and a reference point x_{ref} . It can be described as follows:

$$S(M) := \Lambda(\{\bigcup_i a_i | m_i \in M\}) = \Lambda(\bigcup_{m \in M} \{x | m \prec x \prec x_{ref}\}) \tag{6}$$

HV is also affected by reference point but it can perfectly points out how close the solution set located to the true Pareto front. In paper [13][14], the effect of HV is amplified by introduced as a criterion of selection.
The last indicator we choose is Spread. To compute Spread, two solutions of Z^{all} that reach the extrema relatively to the two objectives are filtered out of the set and the spread indicator is given by:

$$\Delta = \frac{d_f + d_l + \sum \left| d_i - \overline{d} \right|}{d_f + d_l + \sum \overline{d}} \tag{7}$$

where d_f and d_l is the distances of the selected solutions to those extreme points, $\overline{d}$ is the mean of distances and d_i is the distance between solutions of the set.
This indicator focuses on the broadness of the solution set spreads in objective space and it can work as a good complement for HV and Epsilon.

4 Result analysis

4.1 SBX Single Point Crossover vs. Uniform Crossover

There are two operations in MOEA/D, crossover and mutation. We use polynomial bit flip mutation as mutation operation. SBX single point crossover and uniform crossover are both performed in different runs to compare the effect of different crossover. The SBX single point combines SBX crossover, which deals with array real solution type, and single point crossover which deals with binary string. Particularly, SBX single point crossover can only produce offspring from two parents, so the MOEA/D algorithm is slightly adjusted to select two parents from the neighborhood instead of three. And one of the resulting two offspring is abandoned randomly. The uniform crossover, on the other sides, can perfectly handle three parents and array real and binary solution type so there is no need to changing anything. Figure 2 shows the Pareto Front obtained by MOEA/D using SBX single point crossover and uniform crossover.

Though not so remarkable, it can be discerned that the solutions of SBX single point crossover edges out on convergence as the solution set is closer to the left front of the plot and the color is perfectly maintained in blue and green part while the solution set of Uniform crossover is at a worse position and some of points jump to red and orange. This observation is confirmed by the statistic data of indicators as table 1.

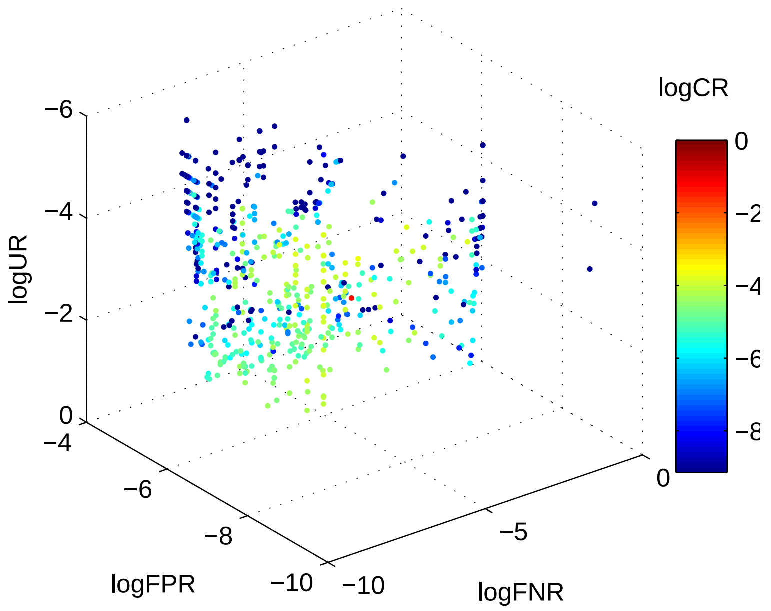
	SBX single point crossover	Uniform crossover
HV mean and standard deviation	$8.12e-01_{1.2e-01}$	$9.92e-01_{1.3e-03}$
SPREAD mean and standard deviation	$1.21e+00_{1.4e-01}$	$1.66e+00_{7.1e-02}$
EPLISON mean and standard deviation	$9.54e-02_{6.1e-02}$	$7.01e-03_{3.8e-03}$

Table 1: *Indicators of SBX single point crossover and Uniform crossover*

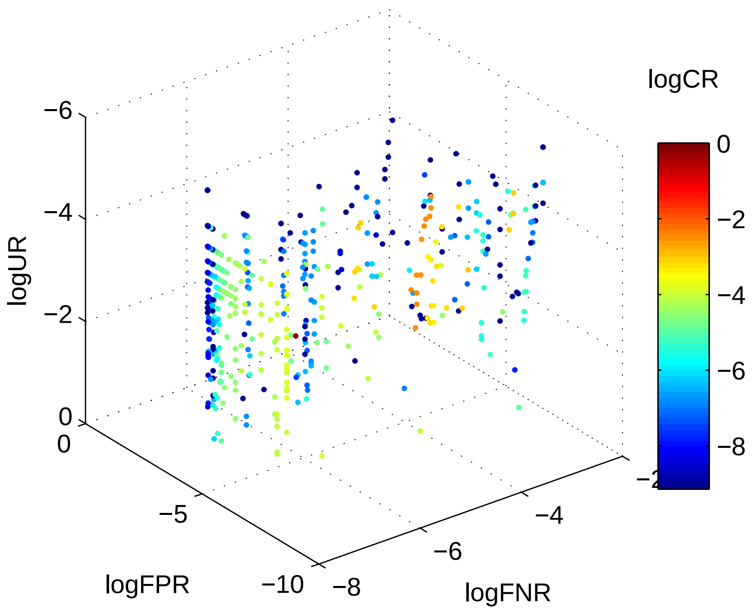
The table shows that SBX single point crossover obtains a solution set of 697 best points in 30 runs and Uniform crossover obtains 615, so the two crossovers play even on the size of Pareto Front. From the table, SBX single point crossover gets a higher grade on Eplison, which indicates that it needs less translation to dominate all solutions of reference front, in other words, it's much closer to optimal front. However, Uniform crossover has advantages on broadness and uniform distribution because it has a higher value of both Spread and Eplison indicator. It can be seen that different crossover would in great extent effect the performance of MOEA/D and it requires careful selection depending on which property we emphasize.

4.2 MOEA/D vs. NSGAII

In this part, we will take a closer look on the performance of MOEA/D. And in order to get a better understanding, the NSGAII is performed



(a) SBX single point crossover



(b) Uniform crossover

Figure 2: *Scatter plot of two crossover*

in same setting of population size and evaluation time as a comparison. Two algorithms both use polynomial bit flip mutation. As SBX single point crossover and Uniform crossover both have its own superiority, to keep the comparison equal, we choose the same SBX single point crossover for MOEA/D to make it consistent with NSGAII. Binary tournament2 selection is used in NSGAII. Moreover, both algorithms are performed 30 independent runs to make sure a robust performance optimization.
Figure 3 is a scatter matrix of MOEA/D, which provides some 2-dimension information between objectives. From the matrix, we can observe a best point which dominates all the others in most of the subfigures, except for the figure of FNR& CR and UR& CR. In the figure of these two, there is no obvious best solution, which indicating a conflict between objective FNR and CR, UR and CR. It suggests that though it may be possible to optimize other objectives simultaneously, lower false negatives or suspicious would inevitably increase the complexity. So we have to choose between false negative rate and complexity depending on which is more important in the situation, precise prediction or speed of computation.

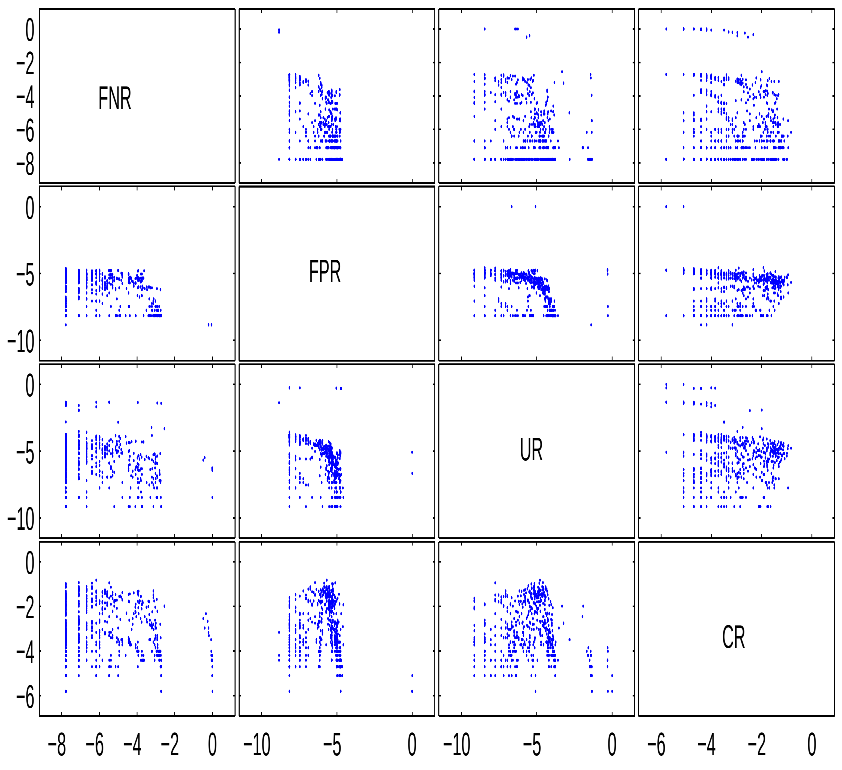


Figure 3: *Scatter matrix of MOEA/D*

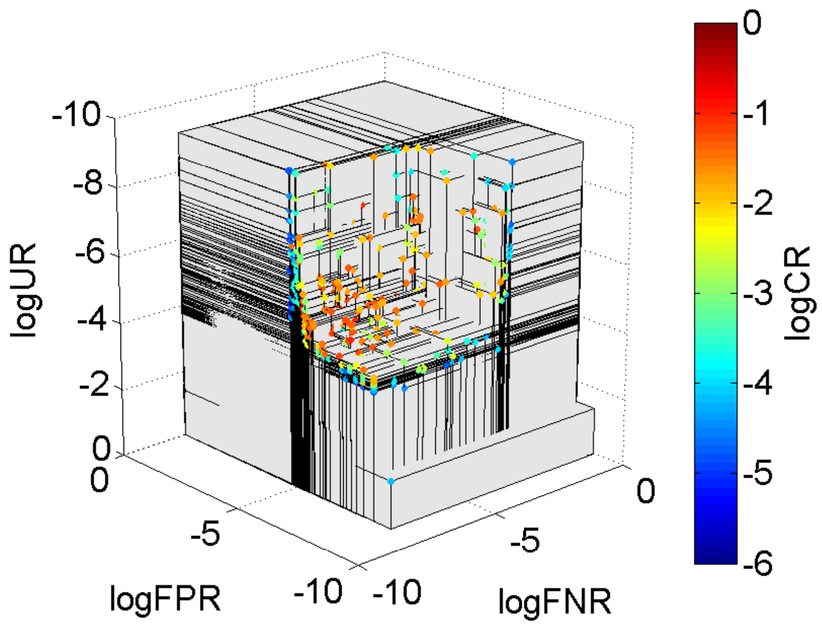
To present the result in a higher dimension, Figure 4 plots the logarithmic form of best solutions obtained in 30 runs of two algorithms in 3D and use color to demonstrate the fourth objective. The log operation makes the result spread to a better scale, while also makes some zero values unable to display on the graph. The plot clearly indicates that MOEA/D gets less but better solutions than NSGAII according to the position and color of points. This is demonstrated even more obvious in Figure 5 and 6. In Figure 5, the plots of two algorithms are compounded into one, where the dominated space of MOEA/D is drawn in red and the dominated space of NSGAII is drawn in grey, so the finally color can reveal the domination between two algorithms. Figure 6 shows the X-Y, X-Z, Y-Z view of Fig.5, from which an apparent dominance of MOEA/D in terms of false negative rate, unknown rate and complexity rate is demonstrated. Indeed, the average of complexity rate of MOEA/D is 7.4% to get a prediction with error rate less than 5% and an unknown rate less than 1%, while the complexity rate of NSGAII is 27.8% to get a solution set of the same level. Approximately four times of rules are required by NSGAII.

The statistic comparison of two algorithms is presented in table 2, which compares the speed and size and quality of solution set between two algorithms.

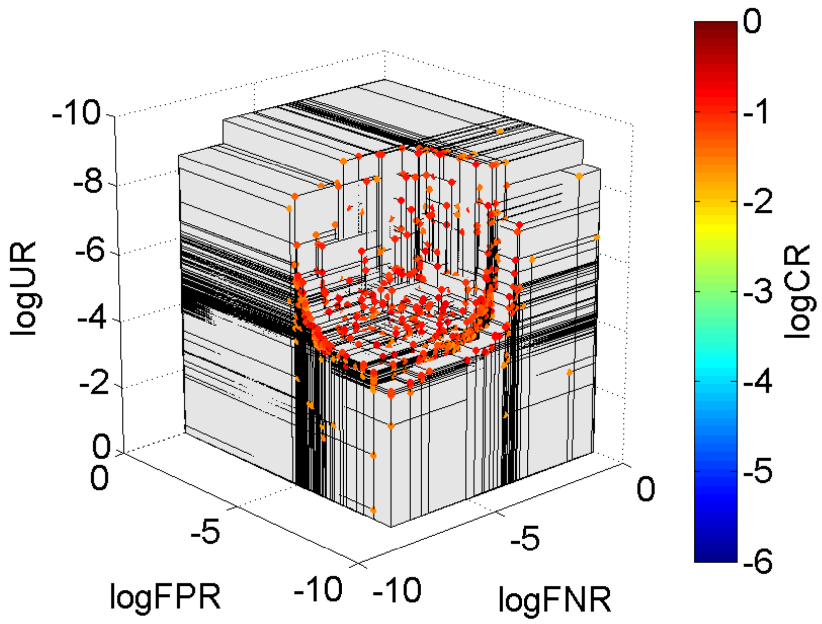
	MOEA/D	NSGAII
CPU time	122584	123700
Number of best solutions in 30 runs	697	1639
Number of dominated solutions	38	780
HV mean and standard deviation	$8.12e-01_{1.2e-01}$	$1.16e-01_{1.3e-02}$
SPREAD mean and standard deviation	$1.21e+00_{1.4e-01}$	$9.19e-01_{8.1e-02}$
EPLISON mean and standard deviation	$9.54e-02_{6.1e-02}$	$4.40e-01_{9.1e-03}$

Table 2: *Performance of MOEA/D and NSGAII*

The table gives the average CPU time used by each algorithm and MOEA/D shows a tiny dominance comparing to NSGAII. This is rational because the computation complexity of MOEA/D is $O(mNT)$ while NSGAII is $O(mN^2)$, where N is the size of population and T is the size of neighborhood. Since T is definitely smaller than N, MOEA/D gets a lower computation complexity than NSGAII. The table also shows



(a) MOEA/D



(b) NSGAII

Figure 4: *Boxes of non-dominated solutions in 30 runs*

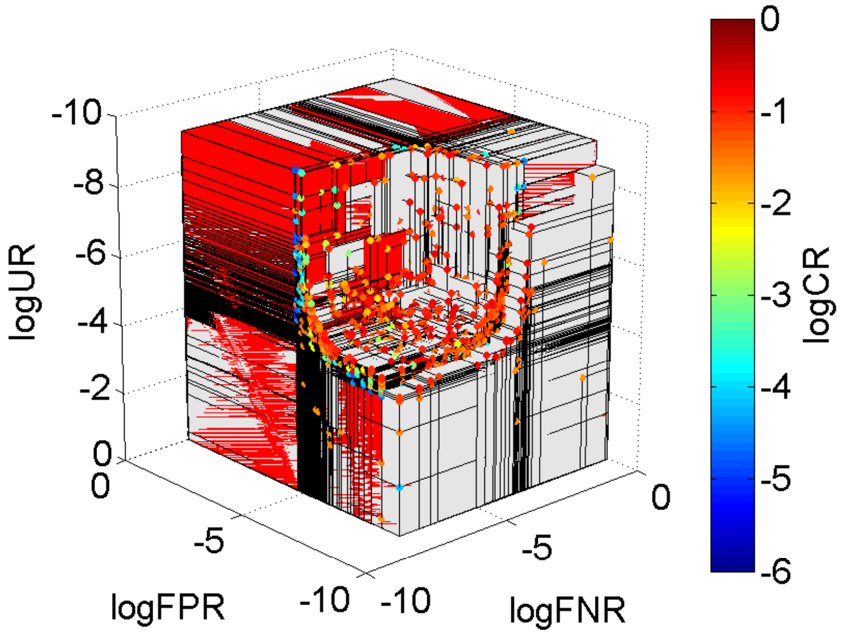
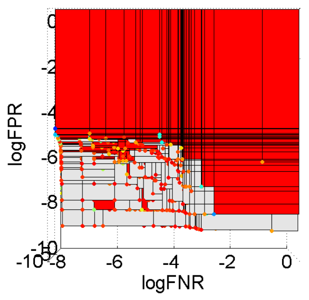
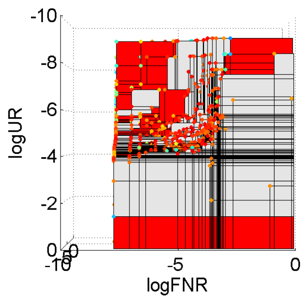


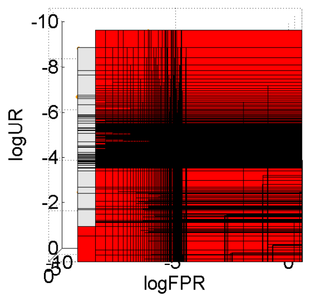
Figure 5: *Domination between MOEA/D and NSGAII*



(a) x-y view



(b) x-z view



(c) y-z view

Figure 6: *X-Y, X-Z, Y-Z view*

that MOEA/D found nearly two times of best solutions less than NSGAII, however, there are 780 best solutions found by NSGAII are dominated by Pareto Front obtained by MOEA/D, while only 38 solutions of MOEA/D are dominated by NSGAII. Therefore, we can conclude that though MOEA/D lags behind NSGAII on quantity, it stands out on the quality of convergence. The three indicators, HV, Spread, Eplison, confirmed this conclusion as MOEA/D has a much better value on indicator HV and Eplison though do worse on Spread. Another thing worth notice is that MOEA/D has a lower standard deviation in terms of all indicators, even on Spread, which means that it has a much more stable behavior than NSGAII.

From the results above, we consider MOEA/D to be a better algorithm if no strict demand on variety is required as it clearly achieve a solution set with higher quality than NSGAII.

5 Conclusion

In the experiment, a four dimension objective of anti-spam classifier, including false positive rate, false negative rate, unknown rate and complexity rate was minimized. A new algorithm MOEA/D was performed to solve this problem and it was proved that a good solution set can be achieved. The FNR, FPR and UR can be minimized to less than 1% using just 1% of 330 rules and that's a great progress comparing to the reality that solutions with the same level of error rate and unknown rate require 16% to 35% rules based on NSGAII, SPEA2 and SMS-EMOA.
The next step of this project is to take what we have done as a basic and try to make further improvement from aspects mentioned below. On one hand, though MOEA/D can achieve an outstanding performance in convergence, the variety of solution set still needs improvement. Changing parameters, like the size of neighborhood or the maximal number of solutions replaced by each child solution may make a progress. Also, different crossover, mutation or decomposition approach may leads to enhanced performance. Instead of Tchebycheff, the boundary intersection methods is becoming popular [15][16][17] as a new approach of decomposition. Moreover, a new version of MOEA/D called MOEA/D-M2M [18] is proved to be able to significantly outperform MOEA/D and NSGA-II on a set of test instances meanwhile maintaining better population diversity. Some newly proposed multiobjective and many-objectives optimization algorithms also

worth a study, for example U-NSGA-III [19], CH-EMOA[20] [21] or mixed integer optimization [12]. On the other hand, the experiment can be improved by consider some progressive visualization techniques to present high dimensional data in a more intuitional way.

Bibliography

[1] Mason J. Filtering spam with spamassassin[C]//HEANet Annual Conference. 2002: 103.

[2] Basto-Fernandes V., Yevseyeva I., Ruano-Ordas D., Zhao J., Fernandez-Riverola F., Mendez J.R., Emmerich M.T.M. Quadcriteria Optimization of Binary Classifiers: Error Rates, Coverage, and Complexity. EVOLVE 2015, June 18-24 (2015)

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

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

[5] Zhang Q, Liu W, Li H. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances[C]//IEEE congress on evolutionary computation. 2009, 1: 203-208.

[6] Directory with the files containing the weight vectors, http://dces.essex.ac.uk/staff/qzhang/MOEAcompetition/CEC-09final/code/ZhangMOEADcode/moead0305.rar

[7] MOEA Framework, a Java library for multiobjective evolutionary developing and experimenting with multiobjective evolutionary algorithms (MOEAs) and other general-purpose multiobjective optimization algorithms, http://moeaframework.org/

[8] D. A. Van Veldhuizen, G. B. Lamont, Multiobjective evolutionary algorithm research: A history and analysis, Tech. Rep. TR-98-

03, Dept.Elec. Comput. Eng., Graduate School of Eng., Air Force Inst.Technol.,Wright-Patterson, AFB, OH (1998).

[9] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Transactions on Evolutionary Computation 3 (4) (1999) 257–271.

[10] J. Knowles, L. Thiele, E. Zitzler, A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers, Tech. Rep. 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich (2006).

[11] K. Deb, Multi-objective optimization using evolutionary algorithms, John Wiley & Sons, 2001.

[12] Durillo J J, Nebro A J, Alba E. The jmetal framework for multi-objective optimization: Design and architecture[C]//Evolutionary Computation (CEC), 2010 IEEE Congress on. IEEE, 2010: 1-8.

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

[14] Emmerich M, Beume N, Naujoks B. An EMO algorithm using the hypervolume measure as selection criterion[C]//Evolutionary Multi-Criterion Optimization. Springer Berlin Heidelberg, 2005: 62-76.

[15] I. Das and J. E. Dennis, "Normal-bounday intersection: A new method for generating Pareto optimal points in multicriteria optimization problems," SIAM J. Optim., vol. 8, no. 3, pp. 631–657, Aug. 1998.

[16] A. Messac, A. Ismail-Yahaya, and C. Mattson, "The normalized normal constraint method for generating the Pareto frontier," Struct Multidisc. Optim., vol. 25, pp. 86–98, 2003.

[17] C. A. Mattson, A. A. Mullur, and A. Messac, "Smart Pareto filter: Obtaining a minimal representation of multiobjective design space," Eng. Optim., vol. 36, no. 6, pp. 721–740, 2004.

[18] Liu H L, Gu F, Zhang Q. Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems[J]. Evolutionary Computation, IEEE Transactions on, 2014, 18(3): 450-455.

[19] Seada H, Deb K. U-nsga-iii: A unified evolutionary optimization procedure for single, multiple, and many objectives: Proof-of-principle results[C]//Evolutionary Multi-Criterion Optimization. Springer International Publishing, 2015: 34-49.

[20] 9. Zhao, J., Basto-Fernandes, V., Jiao, L., Yevseyeva, L., Maulana, A., Li, R., Back,T., Emmerich, M. T. M.: Multiobjective optimization of classiers by means of 3-d convex hull based evolutionary algorithm, ARXIV Computer Science abs/1412.5710, http://arxiv.org/abs/1412.5710 (2014)

[21] Li, R., Emmerich, M. T., Eggermont, J., Back, T., Schutz, M., Dijkstra, J., and Reiber, J. H. (2013). Mixed integer evolution strategies for parameter optimization. Evolutionary computation, 21(1), 29-64.