DMEA-II and its application on spam email detection problems

Long Nguyen
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
Email: longit76@gmail.com

Anh Quang Tran
Faculty of Information Technology
Hanoi University
Hanoi, Vietnam
Email: anhtq@hanu.edu.vn

Lam Thu Bui
Faculty of Information Technology
Le Quy Don Technical University
Hanoi, Vietnam
Email: lam.bui07@gmail.com

***Abstract*—This paper deals with multi-objectitivty in the problem of Vietnamese spam detection. We first analyze the problem taking into account the specific Vietnamese characterises as well as multi-objectivity. With the use of multi-objectivity, we can allow the users more flexibility on selecting the solution. Our proposal is to extende a multi-objective optimization algorithm using directional information, called DMEA-II for finding sets of feasible trade-off solutions for an anti-spam email system (using Apache SpamAssassin). The two objectives for considering are the Spam Detection Rate (SDR) and False Alarm Rate (FAR). The experiments were conducted based on spam data sets through several scenarios with different numbers of SpamAssassin rules. According to the obtained results, the new approach based on DMEA-II not only achieved more efficient results but also created a set of ready-to-use rule scores trading-off between SDR and FAR. It demonstrates the ability to give users more flexibility and efficiency in the Anti-spam email System.**

I. Introduction

Since email spamming has become more fierce and uncontrollable, researchers all around the world have been trying to manage stopping spammers from annoying email users by proposing a wide range of Anti-Spam solutions. For each solution with different approach, the pros and cons are various. There are also a number of factors to evaluate the efficiency of solutions. Among them, the Spam Detection Rate (SDR) and the False Alarm Rate (FAR) seems to be most obvious criteria to measure the effectiveness of a spam detection resolution. The final purpose of any Anti-Spam approach is to maximize SDR and to minimize FAR as much as possible. The key point is that SDR is proportional to FAR: the higher rate of detecting spam an approach brings the higher probability to alarm a ham (non-spam email) as spam it gets and vice versa. An effective spam detection system is not expected to gain an absolute optimum which are 100% for SDR and 0% for FAR, but it is an acceptable trade-off between these criteria. Current approaches achieve the desired SDR (or FAR) by the following procedure:

- A threshold at which an email is considered to be spam is predefined.
- Model is built to train the system.
- SDR (or FAR) is measured to evaluate the effectiveness of Anti-Spam solution at specific thresholds.

In this paper we used DMEA-II for finding the trade-off solutions to help users in anti-spam email system configuration more flexibility and efficiency. We also compare DMEA-II's results against the ones obtained form other MOEA (namely NSGA-II)
The remainder of this paper is organized as follows. A description of Spam Detection System is given in Section II. The common concepts of MOEAs and briefly description about DMEA-II in Section III. Our methodology is given in section IV, we presented the experiments and discussion in Section V. Finally, the last section concluded the paper and talked about the future of our works.

978-1-4799-5431-5/14/$31.00 ©2014 IEEE

II. SpamAssassin

SpamAssassin is a common antispam system develop by the Apache Software Foundation. It examines email and assign a score to indicate the likelihood that the email is spam. SpamAssassin uses a rule-based detection method that compares different parts of email with many pre-defined rules. Each rule adds or removes points from an email's score. An email with a high enough score is considered to be spam. An example of rule in SpamAssassin is follow:

- **Body** DEAR_FRIEND $/\hat{} \setminus s * DearFriend \setminus b/i.$
- **Describe** DEAR_FRIEND Dear Friend? That's not very dear!
- **Score** DEAR_FRIEND 0.542

In this example, the rule's name is $DEAR\_FRIEND$. By applying the rule, SpamAssassin will examine whether if a body part of an email matches the regular expression $/\hat{} \setminus s * DearFriend \setminus b/i$. If yes, then it adds a score of 0.542 to the emails score. An anatomy of a rule was described in details by Schwartz [1].

SpamAssassin provides a built-in module to score its rules. The scoring module works as a single-objective optimization method. It sets the threshold to a fixed value, then optimizes the scores to decrease the error rate over a given training dataset. SpamAssassin uses the Stochastic Gradient Descent algorithm to of training a single-layer neural network with a transfer function and a logsig activation function. Each node of the neural network represents a rule of SpamAssassin. The input of each node represents whether or not the rule is activated by an email. The weight of each node is respected to the score of that rule. SpamAssassin uses a linear function to map the weights to the score space.

In recent years, there is an increasing trend in dealing with multi-objectivity in optimizing rule scores [2], [3], [4], [5]. Obviously, there will be several objectives for this problem, typically SDR ad FAR. The contribution in this area will be how to designed a MOEA to solve it and how to deal with language-specific email databases.

III. Multi-objective Optimization concepts

A. Common concepts

Practical problems in real-life usually possess the feature of multi-objectivity having multiple competing objectives (or criteria). Their solutions therefore describe alternatives, each of which represents a different compromise between the conflicting objectives. The set of optimal solutions to the problem are called *Pareto optimal* set. Its projection in objective space is known as the *Pareto optimal front* (POF). The *ideal point* of the POF is the vector whose components contain the result of minimizing each objective individually.

Mathematically, in a k-objective unconstrained (bound constrained) minimization problem, a vector function $\vec{f}(\vec{x})$ of k objectives is defined as:

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), ..., f_k(\vec{x})] \qquad (1)$$

in which $\vec{x}$ is a vector of decision variables in v-dimensional $\mathbb{R}^v$. In EC, $\vec{x}$ represents an individual in the population to be evolved. The value $f_j(\vec{x})$, then, describes the performance of individual $\vec{x}$ as evaluated against the jth objective in the MOP.

An individual $\vec{x}_1$ *dominates* $\vec{x}_2$ if $\vec{x}_1$ is not worse than $\vec{x}_2$ on all k objectives and is better than $\vec{x}_2$ on at least one objective. If $\vec{x}_1$ does not dominate $\vec{x}_2$ and $\vec{x}_2$ also does not dominate $\vec{x}_1$, then $\vec{x}_1$ and $\vec{x}_2$ are said to be *non-dominated* with respect to each other. If we use the symbol "$\preceq$" to denote that $\vec{x}_1 \preceq \vec{x}_2$ means $\vec{x}_1$ dominates $\vec{x}_2$, and the symbol "$\ntriangleright$" between two scalars a and b to indicate that $a \ntriangleright b$ means a is not worse than b, then *dominance* can be formally defined as [6]:

Definition 1 (Dominance): $\vec{x}_1 \preceq \vec{x}_2$ *if the following conditions are held*:

1. $f_j(\vec{x}_1) \ntriangleright f_j(\vec{x}_2) \forall j \in \{1, 2, \ldots, k\}$; and,
2. $\exists j \in \{1, 2, \ldots, k\} : f_j(\vec{x}_1) \triangleleft f_j(\vec{x}_2)$.

In general, if an individual is not dominated by any other individual in the population, it is called a non-dominated solution. All non-dominated solutions in a population form the non-dominated set as formally described in the following definition:

Definition 2 (Non-Dominated Set): *A set S is said to be the non-dominated set of a population P if the following conditions are met:*

1. $S \subseteq P$; and,
2. $\forall \vec{s} \in S \nexists \vec{x} \in P : \vec{x} \preceq \vec{s}$.

If P represents the entire search space, then S is referred to as the *global Pareto optimal set*. If P represents only a sub-space, then S is called the *local Pareto optimal set*. While there can be multiple local Pareto optimal sets, there exists only one global one.

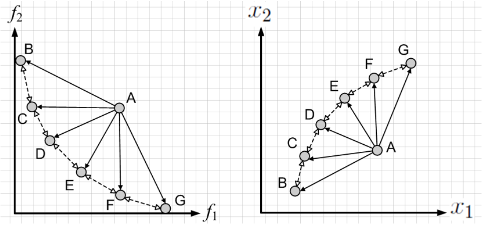

Fig. 1. Illustration of convergence (black arrows) and spread (hollow arrows) directions in objective space (left) and decision variable space (right).

B. Multi-objective Evolutionary Algorithms

Multi-objective evolutionary algorithms (MOEAs) are stochastic techniques being used to find Pareto optimal solutions for MOPs. There are two key problems that MOEAs have to deal with [6]. The first one is how to get as close as possible to the POF. This is challenging because of the stochasticity of the convergence process. The second one is how to keep solutions diverse. A diverse set of solutions will provide decision makers, designers, etc with more choice. However, working on a set of solutions instead of only one, makes the measurement of MOEA convergence more difficult because one individual's closeness to the POF does not act as a measure for the entire set. Unsurprisingly, then, convergence and diversity are commonly used performance criteria when optimization algorithms are assessed and compared with each other [7].

To date, many MOEAs have been developed: PAES [8], SPEA2 [9], PDE [10], NSGA-II [11], MOEA/D [12], MODE-LD+SS [13] and DMEA [14]. MOEAs are usually classified into two broad categories: with and without elitism. Elitist approach is a mechanism to preserve the best individuals, once found, during the optimization process. The concept of elitism was established at an early stage of EC (see, for example, [15]); and to date, it has been widely used in EAs.

C. DMEA-II

DMEA-II is an elitism MOEA introduced in [16]. In DMEA-II, two types of directional information are maintained and used to perturb the parental population prior to offspring production: convergence and spread (see Figure 1).

Convergence direction (CD). In general defined as the direction from a solution to a better one, CD in MOP is a normalized vector that points from dominated to non-dominated solutions.

Spread direction (SD). Generally defined as the direction between two equivalent solutions, SD in MOP is an un-normalized vector that points from one non-dominated solution to another.

In DMEA-II a bundle of rays are used either emitting uniformly from the estimated ideal point into the part of objective space that contains the POF estimate, or being parallel as depicted in Figure 2. The number of rays equals the number of non-dominated solutions wanted by the user. Rays emit into a "hyperquadrant" of objective space. During the archival update (inserting non-dominated solutions), the rays are used as reference lines to select particular non-dominated solutions from the combined population. One by one, the rays are scanned and the non-dominated solution closest to a given ray is selected and archived.

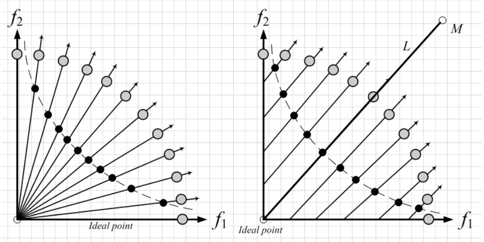

Fig. 2. Illustration of the ray system in a 2-dim MOP. The left graph: origin of the bundle is collocated with the estimated ideal point. The ray bundle contains the POF estimate. The right graph: The rays start from generated points and parallel with the central lines of the top right quadrant.

A niching operator is used for the main population. From the second generation onward, the population is composed from two equal parts: one part for convergence, and the other one for diversity. The first part is filled by non-dominated solutions up to a maximum of $n/2$ solutions from the combined population, where n is the population size. This filling task is based on niching information.

IV. Methodology

In this section, we will describe our method for detecting Vietnamese spams using SpamAssassin which incorporated with DMEA-II. Note that spam detection is a major problem in any anti-spam system, in which it detects spams from a set of emails. Given an email, which is usually in plain text, an anti-spam system needs to answer the question that this email is spam or not. If yes, the anti-spam system will drop it or alert spam to the recipient of the email. Vietnamese spam refers to the spam written in Vietnamese language. Along with the popularity of the Internet services and application, Vietnamese spamming has become one of the major problems in computer network security in the Vietnamese Internet society. Since written in the Vietnamese language, any spam detection method, which uses the content of emails for detecting spam, should be revised so that it can work properly with the Vietnamese language.

There is a large set of rules in SpamAssassin including content-based rules. These content-based rules can only work with a specific language. Some research works of making rules for specific language such as Chinese, Thai or Turkey have been carried out [17], [18] and [19]. There have been also some attempts to research of Vietnamese rules for SpamAssassin. M.T. Vu [20] extended the statistical rule-based method proposed in [17] to produce Vietnamese rules as a part of multilingual rules for SpamAssassin. There are two main phases to produce Vietnamese rules for SpamAssassin: *Rule pattern selection* and *rule scoring*. In the first phase, a set of spam-liked patterns is generated from a training dataset including both spam and ham. A set of SpamAssassin rules (without scores) could be produced from that set of spam-liked patterns by following the writing format of rules in SpamAssassin. In the second phase, a score is assigned for each rule so that the rules best classify spam and ham from the training dataset.

The point that makes languages different from one another in producing SpamAssassin rules is in the rule pattern selection phase. That is how to split emails into meaningful words. For the case of the Vietnamese language, a segmentation technique is used to retrieve the meaningful words in the body part of spam and ham in the training dataset. Unlike other languages such as English or French, in which words can be identified easily by the blank spaces, Vietnamese words might consist of more than one single word, so there is no clear boundary such as blank spaces between Vietnamese words. In this paper, a Vietnamese word segmentation program proposed by H.P. Le (2008)[21] was used to split an Vietnamese email into words. In order to select a good set of spam-liked patterns used for SpamAssassin rules, for each pattern *t*, we compute a value V_{ts} and V_{th}, which can best evaluate the connection between *t* and spam *s*, *t* and ham *h*, respectively. Then, top N patterns that have the highest value of ratio $R_t = \frac{V_{ts}}{V_{th}}$ are selected. The values of V_{ts} and V_{th} are computed by using the conditional probability formula [17]:

$$V_{ts} = P(E|H) = \frac{E \cap H}{P(H)} \quad (2)$$

$$V_{th} = P(\overline{E}|H) = \frac{\overline{E} \cap H}{P(H)} \quad (3)$$

Where E is a hypothesis that an email occurs as spam and H is a hypothesis that a email contains pattern.

The scoring process is an optimization problem, in which we optimize the scores of rules to maximize the performances of the rules over a training dataset. The scoring processes used in [17] and [20] are single objective problem. This paper attempts to utilize a multi-objective approach in the rule scoring phase

As mentioned in the introduction, the main concern of the traditional Anti-Spam approach is difficult and time-consuming to find out the optimized trade-off between values of SDR and FAR if the threshold changes. If the set of spam detection rules remains unchanged, there is only one pair of values for SDR and FAR which are considered as the most wanted solution at a specific threshold. When the algorithm runs with different thresholds, the rule's scores (optimized for the predefined threshold) are no longer optimized for the current threshold which would cause the rate of spam detection and false alarm not optimized anymore. The training process must restart from the beginning to meet the email users' demand on various SDR and FAR.

There are two popular measures to evaluate an anti-spam system: SDR and false alarm rate FAR). The SDR refers to the rate that a spam is detected correctly while FAR is the rate that a normal email (also called ham) is detected as spam. In practice, these two measures can be computed by applying the anti-spam system over a testing dataset. Suppose that a testing dataset includes K spams and L hams. If the antispam system is able to detect M spams out of K spams, at the same time, it incorrectly detects P hams among L hams as spam, then we can compute SDR = M / K and FAR = P / L.

Given a set of rules with scores and a threshold T, for each email, SpamAssassin can compute its score and detect whether if it is spam by comparing the score with threshold T. Therefore, we can evaluate the performance of SpamAssassin by computing the two measures SDR and FAR upon a testing dataset.

Let $S = \{s_1, s_2, ..., s_K\}$ be a set of spam and $H = \{h_1, h_2, ..., h_L\}$ be a set of ham in the testing dataset. Let $R = \{r_1, r_2, ..., r_N\}$ be the set of rules in SpamAssassin. Each rule $r \in R$ might match with some email $e \in S, H$ through a matching function:

$$m\left(r, e\right) = \begin{cases} 1 \; if\_r\_matches\_e \\ 0 \; otherwise \end{cases} \tag{4}$$

Where $r \;\in R, \; e \;\in \{S, H\}$
Let $X = \{x_1, x_2, ..., x_N\}$ be the score set of R, in which x_0 is the threshold ($x_0 = 30$ or $x_0 = 100$ in our experiments)and x_1 is the score of r_1 for $1 \leq i \leq N$. SpamAssassin computes the score of email e as the formula:

$$Score(e) = \sum_{i=1}^{N} m(r_i, e) x_i \tag{5}$$

At threshold T, the formula to detect spam is shown as follow:

$$Detect\left(e\right) = \begin{cases} 1 \; if Score(e) \geq T \\ 0 \; otherwise \end{cases} \tag{6}$$

The two measures SDR and FAR are then computed by the formulas:

$$SDR = \frac{1}{K} \sum_{i=1}^{K} Detect(s_i) \tag{7}$$

$$FAR = \frac{1}{L} \sum_{i=1}^{L} Detect(h_i) \tag{8}$$

Different value of X and T causes different values of SDR and FAR [22], which are the measure of SpamAssassins performance. We always expect a high SDR and low FAR, however we dont always obtain these two objectives at the same time, that is when SDR is high, FAR also tends to be high and vice versa. This paper aims to utilize a multi-objective approach to find a Pareto sets of X and T. We applied DMEA-II to solve problem, in order we do following steps:

- **Step 1:** *Initialize the data input*
  For the problem, the objective is also to find a set of ideal scores called x where $x = (x_1, .., x_N)$, $N = 31$, $x_1 \in [2, 5]$, $x_{2...N} \in [0, 2]$ The set of x will be generated randomly with a random algorithm which is a part of MOEAs. Each value inside the set is considered as a chromosome. The first value is set limitation from 2 to 5 because it is the threshold – the point at what an email is considered as spam. The other values are set from 0 to 2 which are the score of SpamAssassin rules. Experiments were carried out with 30 rules and 1 threshold (N = 31).
- **Step 2:** *Create the objective function*
  The objective function is designed to run on the spam dataset S (Spam data sets) and ham dataset H (Ham data sets).

  $$S = \{s_1, \; s_2, .., s_K\}$$

  $$H = \{h_1, \; h_2, .., h_L\}$$

  The set of N rules is pre-designed based on the framework in [20].

  $$R = \{r_1, \; r_2, .., r_N\}$$

  Each rule might match with some spams or hams through the matching function 4
  The effectiveness of the set of rules with randomly-generated scores (from step 1) is evaluated by SpamAssassin against the dataset S and H. Score sets bringing the best results would be selected as a solution for this multi-objective problem.
  At threshold T (that is x_0), the function to detect spam is implemented as 6.
- **Step 3:** *Compute two objectives*
  The purpose of the objective function is to compute two objectives of the problem. Within the scope of this problem, two objectives SDR and FAR are compute against the formulas 7, 8,.
  However, all objectives are supposed minimized. Therefore, the SDR objective of this specific problem is reformulate as (1 - SDR)
- **Step 4:** *Run DMEA-II* After all data input and required parameters are ready, DMEA-II is called to run and figure out the best population. Based on that population, the final result would be evaluated and compared.

V. Case Studies

In order to have a concrete argument on our proposal, we implemented several case studies on the SpamAssassin system using two Vietnamese email databases with 272 and 426 emails respectively. The experiments were carried out for 30 rules. This means that the system can be tested on both rule and data scales. The results will be analyzed on aspects of numerical values of SDR and FAR as well as the multi-objectivity.

A. Experimental Parameters

For the sake of simplicity, we keep parameters are unchanged DMEA-II. Parameters are shown in Table I.

Parameters	**Values**
Population size	100
Number of generations	1000
Number of objectives	2
Number of real variables	N+1
Lower limit of real variable 1	2
Upper limit of real variable 1	5
...	...
Lower limit of real variable N+1	0
Upper limit of real variable N+1	2
Probability of crossover	0.9
Probability of mutation	1/Number of real variables

TABLE I: Parameter settings for the experiments

In order to validate the results of DMEA-II, we also implemented other MOEA, namely NSGA-II. Both NSGA-II and DMEA-II were tested 30 times with different random seeds.

We did our experiments on a Dell PowerEdge R710 Server L5520 2x2.26GHz Quad Core 24GB RAM, 2x146GB storage running Ubuntu OS v13.04 (at the Software Technology Lab of Le Quy Don Technical University).

B. Results and Discussion

At the end of experiments for each set of rules, the results were recorded for analyzing. Further results gained from MOEAs were compared to that from the experiments using the single objective optimization algorithm (SOOA) [20]

1) Experiments on the database of 272 emails: The purpose of using this database is to test the ability of the method to deal with the problem when the database is quite small. Statistical results are visualized in Figure 3 from the experiments with problem size of 30 rules (meaning the chromosome's size is 31), they are solutions found by the algorithms. Obviously, MOEAs found better solutions than SOOA did. They found not only solutions with zero FAR as did by SOOA, but the ones with higher SDR values than that of SDR. In term of minimizing FAR (at 0%), the best solution recorded for SDR was around 62% for SDR (with both NSGA-II and DMEA-II) while that result for SOOA (Table II) is only 40.3%. Among solutions which FAR values are around 10%, SDR values of MOEAs are also much better than SOOA's. They are {(74.03%, 7.79%); (74.46%, 8.66%); (72.29%, 6.93%)} in comparison to the best point {(67.1%, 9.6%)} of SOOA. Note that in Table II, we reported all solutions found by SOOA with manually using different thresholds. When the threshold increased, less solutions are classified as spam. It might cause spams not being recognized and hence FAR values reduced. This makes it difficult for the users to decide the solution. MOEAs will provides an automatic way to find the solutions.

In terms of multi-objectivity, the trade-off solutions found by MOEAs were widely spread in the objective space as oppose to that of SOOA: its skewness towards SDR. This means a strong multi-objectivity in this problem and the need to address by a multi-objective approach. With this set of trade-off solutions, the users will have more good choices for the system. Regarding the performance of two MOEAs, it seems that DMEA-II provided better the set of solutions; the direction guided approach works better in guiding the search.

Dataset	Threshold	SDR	FAR	Dataset	Threshold	SDR	FAR
272	0.5	67.1%	9.6%	426	0.5	23.8%	0.8%
	1	67.1%	9.6%		1	2.8%	0.2%
	1.5	55.8%	0.8%		1.5	0.6%	0%
	2	55.8%	0.8%		2	0.4%	0%
	2.5	40.3%	0.0%		2.5	0%	0%
	3	39.8%	0.0%		3	0%	0%
	3.5	8.7%	0.0%		3.5	0%	0%
	4	6.9%	0.0%		4	0%	0%
	4.5	2.6%	0.0%		4.5	0%	0%

TABLE II: The result of experiments using a single objective optimization algorithm (SOOA) with the database of 272 and 426 emails)

2) Experiments on the database of 426 emails.: We extended our experiments with a larger set of emails (size of 426). With this large set of emails, we expect that the system will have more information for evaluating solutions. The obtained solutions are visually shown in Figure 4. We also reported the results found by SOOA in Table II.

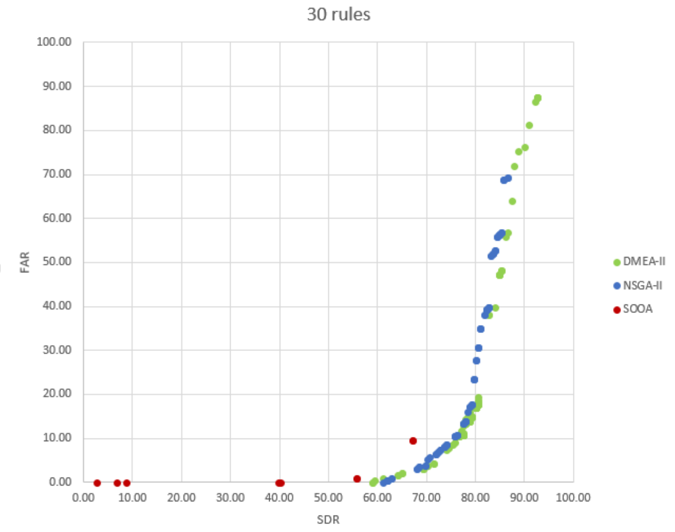

Fig. 3. The trade-off solutions found by NSGA-II, DMEA-II and SOOA with the database of 272 emails.

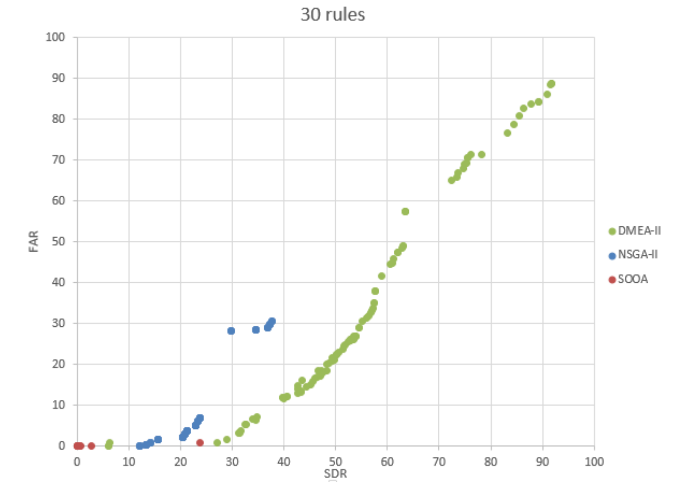

Fig. 4. The trade-off solutions found by NSGA-II, DMEA-II and SOOA with the database of 426 emails

Again, the result in this case once more confirm our previous findings in the case of 426 emails:

- It is clear that the application of multi-objective optimization algorithm to spam detection is reasonable and promising. It can simultaneously offer the users a set of solutions trading-off on SDR ad FAR. With this approach, the users do not need to worry about the threshold, but issuing how is the importance of either SDR or FAR to them?
- The illustration also pointed out that the more set of rules the algorithms working on, the better results it achieved.
- The set of obtained solutions by DMEA-II was uniformly distributed along the POF and got closer to the POF than set of solutions by NSGA-II. It means,

on this real problem, DMEA-II is quite good in keeping balance of convergence and diversity of the population, an important feature of a MOEA.

VI. Conclusions

In this paper, we proposed to apply DMEA-II for solving the problem of an anti-spam email system (using SpamAssassin). In fact, traditional anti-spam approaches have optimized the spam detection rate and the false alarm rate for years and gained specific results. However, the achievement has been optimized for the single objective only. With the-multi objective optimization approach, not only one pair of SDR and FAR for each threshold has been worked out but a set of solutions with different tradeoff levels are computed. They all are feasible depending on specific email users' demands. More important, the score set of selected solutions are always ready to use without any training needed.

Despite of being a promising approach, the proposed framework remains some issues which need more efforts to resolve in the future. It is the problem of runtime. Currently, there is no measurement about the runtime of the system. Because conducted experiments were carried out against quite small dataset, it is not a big issue. However, when the dataset expands in the future, this concern should be analyzed seriously.

Acknowledgments

We acknowledge the financial support from Vietnam's National Foundation for Science and Technology, Grant No 102.01-2010.09.

References

[1] S. A., *SpamAssassin*. O'Reilly, 2004. 1

[2] I. Yevseyeva, V. Basto-Fernandes, and J. R. Méndez, "Survey on anti-spam single and multi-objective optimization," in *ENTERprise Information Systems*. Springer, 2011, pp. 120–129. 2

[3] A. López-Herrera, E. Herrera-Viedma, and F. Herrera, "A multiobjective evolutionary algorithm for spam e-mail filtering," in *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, vol. 1. IEEE, 2008, pp. 366–371. 2

[4] I. Yevseyeva, V. Basto-Fernandes, D. Ruano-Ordás, and J. R. Méndez, "Optimising anti-spam filters with evolutionary algorithms," *Expert Systems with Applications*, 2013. 2

[5] V. Basto-Fernandes, I. Yevseyeva, and J. R. Méndez, "Optimization of anti-spam systems with multiobjective evolutionary algorithms," *Inf. Resour. Manage. J.*, vol. 26, no. 1, pp. 54–67, Jan. 2000. 2

[6] K. Deb, *Multiobjective Optimization using Evolutionary Algorithms*. New York: John Wiley and Son Ltd, 2001. 2

[7] E. Zitzler, L. Thiele, and K. Deb, "Comparision of multiobjective evolutionary algorithms: Emprical results," *Evolutionary Computation*, vol. 8, no. 1, pp. 173–195, 2000. 2

[8] J. Knowles and D. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary Computation*, vol. 8, no. 2, pp. 149–172, 2000. 2

[9] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, K. C. Giannakoglou, D. T. Tsahalis, J. Periaux, K. D. Papailiou, and T. Fogarty, Eds. Int. Center for Numerical Methods in Engineering (Cmine), 2001, pp. 95–100. 2

[10] H. Abbass, R. Sarker, and C. Newton, "PDE: A pareto-frontier differential evolution approach for multi-objective optimization problems," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC2001)*, vol. 2. Piscataway, NJ: IEEE Press, 2001, pp. 971–978. 2

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

[12] Q. F. Zhang and H. Li., "Moea/d: A multi-objective evolutionary algorithm based on decomposition," 2007. 2

[13] C. C. C. M.-M. E. Arias Montano, A., "Mode-ld+ss: A novel differential evolution algorithm incorporating local dominance and scalar selection mechanisms for multi-objective optimization," *In: 2010 IEEE Congress on Evol Comp (CEC2010)*, 2010. 2

[14] L. T. Bui, J. Liu, A. Bender, M. Barlow, S. Wesolkowski, and H. A. Abbass, "Dmea: a direction-based multiobjective evolutionary algorithm," *Memetic Computing*, pp. 271–285, 2011. 2

[15] K. A. DeJong, "An analysis of the behavior of a class of genetic adaptive systems." Ph.D. dissertation, University of Michigan, Ann Arbor, 1975. 2

[16] L. N. L. T. Bui and H. Abbass, "Dmea-ii: the direction-based multi-objective evolutionary algorithm - ii," *Soft Computing*, Accepted to appear 2013, Ref.: Ms. No. SOCO-D-13-00365R1. 2

[17] X. L. Q.A. Tran, H. Duan, "Real-time statistical rules for spam detection," *Proceedings of the International Journal of Computer Science and Network Security*, pp. 178–184, 2006. 3

[18] K. P. C. Na Songkhla, "Statistical rules for thai spam detection," *Proceeding of: Future Networks*, pp. 178–184, 2010. 3

[19] L. Özgür, T. Güngör, and F. S. Gürgen, "Adaptive anti-spam filtering for agglutinative languages: a special case for turkish," *Pattern Recognition Letters*, vol. 25, no. 16, pp. 1819–1831, 2004. 3

[20] M. T. Vu and F. J. V. T. Tran, Quang Anh, "Multilingual rules for spam detection," *Proceedings of the 7th International Conference on Broadband and Biomedical Communications (IB2COM 2012)*, pp. 106–110, 2012. 3, 4, 5

[21] H. N. Phuong L.H, A. Roussanaly, and H. T. Vinh, "A hybrid approach to word segmentation of vietnamese texts," vol. 5196, pp. 240–249, 2008. 3

[22] M. T. Vu, Q. A. Tran, Q. M. Ha, and L. T. Bui, "A multi-objective approach for vietnamese spam detection," *In Processding: Knowledge and Systems Engineering 2013*, pp. 211–221, 2013. 4