Toward an Interactive Method for DMEA-II and Application to the Spam-Email Detection System

Long Nguyen[1], Lam Thu Bui[1], Anh Quang Tran[2]

[1] *Le Quy Don Technical University, Vietnam*
[2] *Hanoi University, Vietnam*

Abstract

Multi-Objective Evolutionary Algorithms (MOEAs) have shown a great potential in dealing with many real-world optimization problems. There has been a popular trend in getting suitable solutions and increasing the convergence of MOEAs by consideration of Decision Makers (DMs) during the optimization process (in other words interacting with DM). Activities of DM includes checking, analyzing the results and giving the preference. In this paper, we propose an interactive method for DMEA-II and apply it to a spam-email detection system. In DMEA-II, an explicit niching operator is used with a set of rays which divides the space evenly for the selection of non-dominated solutions to fill the solution archive and the population of the next generation. We found that, with DMEA-II solutions will effectively converge to Pareto optimal sets under the guidance of the ray system. By this reason, we propose an interactive method using three Ray based approaches: 1) Rays Replacement: The furthest rays from DM's preferred region are replaced by new rays that generated from set of reference points. 2) Rays Redistribution: Which redistribute the system of rays to be in DM's preferred region. 3) Value Added Niching: Based on the distances from non-dominated solutions in archive to DM's preferred region, the niching values for the solutions is increased to be priority selected. By those approaches for the proposal interactive method, the next generation will be guided toward the DM's preferred region. We carried out a case study on several popular test problems and it obtained good results. We apply the proposed method for a real application in a spam-email detection system. With this system, a set of feasible trade-off solutions will be offered for choosing scores and thresholds of the filter rules.

Manuscript communication: received 01 April 2014, accepted 08 April 2014
Corresponding author: Long Nguyen, longit76@gmail.com

Keywords: Interactive, DMEA-II, Improvement Direction, Spread Direction, Convergence Direction.

1. Introduction

Methods for multi-objective optimization can be classified into several classes including *the Interactive method*. With the interactive method, DM iteratively directs the search process by indicating his/her preference information over the set of solutions until DM satisfies or prefers to stop the process [1]. An interesting feature of interactive methods is that during the optimization process DM is able to learn about the underlying problem as well as his/her own preference. To date, many interactive techniques have been proposed for solving MOPs [2, 3, 4, 5, 6, 7, 8, 9, 10]. It is worthwhile to note that the aim of interactive methods is to find the most suitable solutions in several conflicting objectives regarding the DM's preference. It requires a mechanism to support DM in formulating his/her preferences and identifying preferred solutions in the set of Pareto optimal solutions.

In this paper, we introduce an interactive method for DMEA-II [11], a direction-based multi-objective evolutionary algorithm. With this proposal, we allow DM to specify a set of reference points representing the area of interests. Based on those reference points we propose three approaches to be used in the proposal interactive method. The first approach, the rays are generated from the reference points and paralleled with the central line which starts from the ideal point to the centre of the hyperquadrant containing POFs. In the second approach, the system of rays is redistributed to be in DM's preferred region. At the third approach, based on the distances from non-dominated solutions in archive to DM's preferred region, the niching values for the solutions is increased to be priority selected. By the proposal interactive method, DM has more flexibility to express his/her preference and the population will converge to preferred region. This is implemented via the niching mechanism in DMEA-II. If DM is not satisfied, he/she can specify other reference points. In our experiments, several test cases on well-known benchmark sets were carried out to demonstrate the method.

In applying the proposed method for a real application, we implemented it in a spam-email detection system (we call it as an interactive anti-spam system). With this system, a set of feasible trade-off solutions are offered for choosing scores and thresholds. The two objectives for consideration are the Spam Detection Rate (SDR) and False Alarm Rate (FAR). For this multi-objective problem, DM has interaction with the optimization process in order to control the population converging toward his/her preferred areas.

In the remainder of the paper, section II briefly describes the concepts and related works about multi-objective optimization interactive method using reference points. In section III we have a short description for DMEA-II. Section IV we propose our methodology for an interactive with DMEA-II. Section V presents simulation results on several well-known test problems. The results for applying the proposed method for Spam Email Detection System are shown on section VI. Finally, the conclusion of this paper is outlined in section VII.

2. Reference-point interactive approaches

2.1. Concepts

In this section we summarize the reference point interactive method, which is the most popular one in the literature. It is suggested in [12]; and this method is known as a classical reference point approach. The idea is to control the search by reference points using *achievement functions*. Here the achievement function is constructed in such a way that if the reference point is dominated, the optimization will advance past the reference point to a non-dominated solution. A reference point z^* is given for an M-objective optimization problem of minimizing $(f_1(x), \dots, f_k(x))$ with $x \in S$. Then solve a single-objective optimization problem as follows:

$$\min max_{i=1}^{M}[w_i(f_i(x) - z_i^*)]$$

subject to $x \in S$. The common step-wise structure of the interactive method as follows:

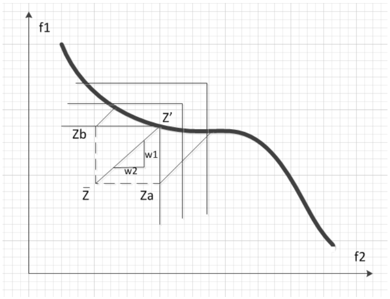

Fig. 1. Altering the reference point, Here Z_A, Z_B are reference points,w is chosen weight vector used for scalarizing the objectives.

- **Step 1**: Present information to the DM. Set h=1.
- **Step 2**: Ask the DM to specify a reference point z_*^h.
- **Step 3**: Minimize achievement function. Present z^h to the DM.
- **Step 4**: Calculate k other solutions with reference points.
  $\overline{z}(i) = \overline{z}^h + d^h e^i$ where $d^h = ||z_*^h - z^h||$ and e_i is the i^{th} unit vector.

- **Step 5**: If the DM can select the final solution, stop. Otherwise, ask DM to specify z_*^{h+1}. Set $h = h + 1$ and go to **Step 3**.

Here h is the number that DM specifies a reference point during process. By the way of using the series of reference points, DM actually tries to evaluate the region of Pareto Optimality, instead of one particular Pareto-optimal point. However DM usually deals with two situations:

1. The reference point is feasible and not a Pareto-optimal solution, DM is interested in knowing solutions which are Pareto-optimal ones and near the reference point.
2. DM finds Pareto-optimal solutions which is near the supplied reference point.

2.2. Related interactive MOEAs

In this section, we summarize several typical works on this area. In [4], authors proposed an interactive MOEA using a concept of the reference point and finding a set of preferred Pareto optimal solutions near the regions of interest to a DM. The authors suggest two approaches: The first is to modify a well-known MOEA called NSGA-II, for effectively solving 10-objective. The other is to use hybrid-MOEA methodology in allowing DM to solve multi-objective optimization problems better and with more confidence.

The authors proposed in [7], a trade-off analysis tool that was used to offer the DM a way to analyze solution candidates. The ideas proposed here are directed to users of both classification and reference point based methods. The motivation here is that DM in certain cases miss additional local trade-off information so that they could get to know how values of objectives are changing, in other words, in which directions to direct the solution process so that they could avoid trial-and-error, that is, specify some preference information so that more preferred solutions will be generated.

In [1], the idea of incorporating preference information into evolutionary multi-objective optimization is discussed and proposed a preference-based evolutionary approach that can be used as an integral part of an interactive algorithm. At each iteration, the DM is asked to give preference information in terms of his/her reference point consisting of desirable aspiration levels for objective functions. The information is used in an evolutionary algorithm to generate a new population by combining the fitness function and an achievement scalarizing function. In multi-objective optimization, achievement scalarizing functions are widely used to project a given reference point into the Pareto optimal set. In the proposal method, the next population is thus more concentrated in the area where more preferred alternatives are assumed to lie and the whole Pareto optimal set does not have to be generated with equal accuracy.

In papers [9] and [10], two reference point interactive methods are proposed to use single or multi reference points with multi-objective optimization based on decomposition-based MOEA (MOEA/D). In this method, a single point or a set of reference points are used in objective space to represent for DM's preferred region. The aggregated point from set of reference points (in case of multi-point) or the reference point is used in optimal process by two ways: replace or combine the current ideal point at the loop.

In paper [13], authors present a multiple reference point approach for multi-objective optimization problems of discrete and combinatorial nature. The reference points can be uniformly distributed within a region that covers the Pareto Optimal Front. An evolutionary algorithm is based on an achievement scalarizing function that does not impose any restrictions with respect to the location of the reference points in the objective space. Authors dealt with the design of a parallelization strategy to efficiently approximate the Pareto Optimal Front. Multiple reference points were used to uniformly divide the objective space into different areas. For each reference point, a set of approximate efficient solutions was found independently, so that the computation was performed in parallel.

3. DMEA-II

In this section, we summarize DMEA-II with the main ideas [11]. In DMEA-II, offsprings are produced by using directions of improvement to perturb randomly-selected parental solutions. Two types of directional information are used to perturb the parental solutions prior to offspring production: convergence and spread (see Fig. 2):

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

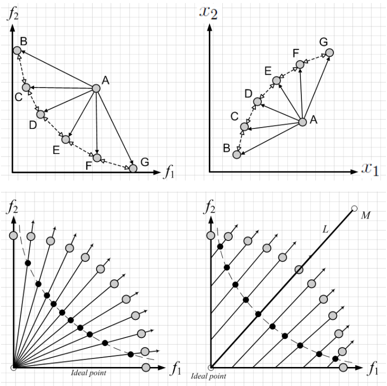

Fig. 2. Illustration of convergence (black arrows in objective space - top left figure) and spread (hollow arrows - top right graph in decision variable space). Two types of ray distribution: parallel and non-parallel (bottom right and left graphs).

3.1. Niching information

A characteristic of solution quality in MOP is the even spread of non-dominated solutions across the POF [14]. In DMEA a bundle of rays are used to emit randomly from the estimated ideal point into the part of objective space that contains the POF estimate, (Fig. 2). The number of rays equals the number of non-dominated solutions wanted by the user. Rays emit into a “hyperquadrant” of objective space, i.e. the sub space that is bounded by the k hyperplanes $f_i = f_{i,\min}$, $i \in \{1, 2, \ldots, k\}$ and described by $f_i \geq f_{i,\min} \forall i \in \{1, 2, \ldots, k\}$ where $f_{i,\min} \approx \min_{\text{all} A_1, A_2, \ldots} f_i$ with $A_1, A_2, \ldots$ being the solutions stored in the current archive. By their construction, the hyperquadrant contains the estimated POF. A niching operator is used to the main population. From the second generation onward, the population is divided into two equal parts: one part for convergence, and one part 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 in the decision space.

3.2. General structure of algorithm

The step-wise structure of the DMEA-II algorithm [11] as follows:

- **Step 1.** Initialize the main population P with size n.
- **Step 2.** Evaluate the population P.
- **Step 3.** Copy non-dominated solutions to the archive A.
- **Step 4.** Generate an interim mixed population (M) of the same size n as P
  - Calculate n_{CD} and n_{SD}
  - Loop {
    - ∗ Select a random parent Par
    - ∗ If (the number of $CD < n_{CD}$)
    - ∗ Generate a CD and then generate a solution S_1 by perturbing Par with CD
    - ∗ Add Add S_1 and to M.
    - ∗ End if
    - ∗ If (the number of $SD < n_{SD}$)
    - ∗ Generate a SD and then generate a solution S_2 by perturbing Par with SD.
    - ∗ Add S_s and to M.
    - ∗ End if

} Until (the mixed population is full).

- **Step 5.** Perform the polynomial mutation operator [14] on the mixed population M with a small rate.
- **Step 6.** Evaluate the mixed population M.
- **Step 7.** Identify the estimated ideal point of the non-dominated solutions in M and determine a system of n rays R (starting from the ideal point and emitting uniformly into the hyperquadrant that contains the non-dominated solutions of M)
- **Step 8.** Combine the interim mixed population M with the current archive A to form a combined population C (i.e. $M+A \rightarrow C$).
- **Step 9:** Create new members of the archive A by copying non-dominated solutions from the combined population C
  - Set counter i=0
  - Loop{
    - ∗ Select a ray $R(i)$.
    - ∗ In C, find the non-dominated solution whose distance to $R(i)$ is minimum.
    - ∗ Select this solution and copy it to the archive.
    - ∗ i = i+1

    } Until (all rays are scanned)
- **Step 10:** Determine the new population P for the next generation.
  - Determine the number m of non-dominated solutions in C.
    - ∗ If $m < n/2$, select all non-dominated solutions from C and copy to P.
    - ∗ Else,
      - · Determine density-based niching value for all non-dominated solutions in C.
      - · Sort non-dominated solutions in C according to niching values.
      - · Copy the $n/2$ solutions with highest niching value to P.
  - Repeatedly scan all rays copy $\max\{n-m, n/2\}$ solutions to P.
- **Step 11**: Go to Step 4 if stopping criterion is not satisfied.

In DMEA-II, the selection of non-dominated solutions to fill the archive and the next population is assisted by a ray based technique of explicit niching in the objective space by using a system of straight lines or rays starting from the current estimation of the ideal point and dividing the space evenly. Each ray is in charge of locating a non-dominated solution, for that reason, a ray has an important role in the optimization process. By this reason, we propose an interactive method using three Ray based approaches: Rays Replacement, Rays Redistribution and Value Added Niching approach. The details for the approaches will be described in next section. The proposed interactive MOEA bases on the system of ray is called ***the Ray based interactive method*** using DMEA-II. In our experiments, the rays start from generated points and paralleled with the central line of the top right hypequadrant.

4. Methodology

Due to the conflicts among the objectives in MOPs, the total number of Pareto optimal solutions might be very large or even infinite. However, the DM may be only interested in preferred solutions instead of all Pareto optimal solutions. To find the preferred solutions, the preference information is needed to guide the search towards the region of the PF of interest to the DM. Based on the role of the DM in the solution process, In an interactive method, the intermediate search results are presented to the DM to investigate; then the DM can understand the problem better and provide more preference information for guiding the search. In this paper proposed two guiding techniques used in interactive method with MOEAs.

4.1. A ray-based interactive method

This section, an interactive method for DMEA-II [11] is introduced. With this proposal, DMs are allowed to specify a set of reference points. With each reference point, a ray is generated by the similar way to building the system of rays in the original DMEA-II : the rays are generated from control points (which might be the reference points) and paralleled with the central line which starts from the ideal point to centre of the hyperquadrant containing POFs). In this way, DM has more flexibility to express his preference. Among several methods for taking set information, we propose to define reference points by using three ray-based approaches: 1) Generate new rays and use them to replace some existing rays; 2) Redistribute the system of rays towards DM's preferred region and 3) Increasing the niching values for non-dominated solutions based on their distance to DM's preferred region. Those techniques are used to control the population to be convergeed to the DM's preferred region. We hypothesise that by those techniques we have a good way to express DM's preferences. After DM has specified a set of reference points, those techniques are applied and the Pareto optimal solutions are found that best corresponds to preferred region in objective space. If DM is not satisfied, he/she can specify other reference points.

4.1.1. Rays Replacement

The approach for interactive method are described as following steps:

- **Step 1**: Ask DM to input n_p reference points which are their preferred regions in objective space.
- **Step 2**: Generate n_p rays from reference points which paralleled with the central line.
- **Step 3**: Calculate the central point of DM's preferred region P_c.
- **Step 4**: Find n_p rays which are farthest from P_c by n_p new ones are generated from Step 2.
- **Step 5**: Apply a niching to control external population (the archive) and next generation.

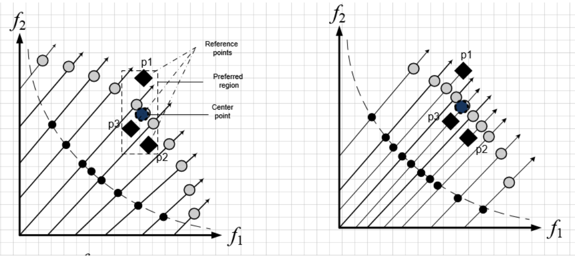

Fig. 3. Illustration of proposed ray based interactive method for DMEA in a 2-dim MOP. Three reference points are given by DM: p1, p2, p3. p_c is the central point of DM's preferred region, there are three new rays (added rays) replace three ones (removed rays).

When DM interactive into the optimal process, we replace Step 7 in DMEA-II (see Section 3) with an interactive function is shown in Algorithm 1.

4.1.2. Rays Redistribution

This approach, the system of rays is offset by new DM's referred region (see Fig: 4). The approach for interactive method as following steps:

- **Step 1**: Ask DM to input n_p reference points which are their preferred regions in objective space.
- **Step 2**: Calculate the boundary of DM's preferred region DM_{bd}.
- **Step 3**: Offset the control points (that generate the system of rays) by DM_{bd}.
- **Step 4**: Generate a new system of rays by new list of control points.
- **Step 4**: Apply a niching to control external population (the archive) and next generation.

When DM interactive into the optimal process, the Step 7 in DMEA-II (see Section 3) with an interactive function is shown in Algorithm 2.

Algorithm 1: Rays Replacement Function.

```
Input: Number of reference points $n_p$
Output: New system of rays
for $i \leftarrow 0$ to $n_p$ do
    • (1) Generate a ray $r_i$ from reference point $p$
      ( $r_i$ through $p_i$ and paralleled with the
      central line (see Fig. 2).
• (2) Make a boundary of reference points
  (DM's preferred region) and find the central
  point $p_c$.
for $j \leftarrow 0$ to n (The number of rays) do
    • (3) Calculate the Euclid distance
      from ray(j) to $p_c$.
• (4) Sort the index of rays in decrease of
  Euclid distance values in (3) (Using the
  QuickSort).
• (5) Replace top $n_p$ rays in the Sorted ray
  indexes with $n_p$ ray from (1).
return n rays.;
```

4.1.3. Value Added Niching

In DMEA-II, the archive is used to store non-dominated solutions during evolutionary process, those solutions are calculated the distance to DM's preferred region. These values are kept and add to niching values after calculation of niching values at Step 10 (see Section 3). The approach for interactive method as following steps:

- **Step 1**: Ask DM to input n_p reference points which are their preferred regions in objective space.
- **Step 2**: Calculate the central point of DM's preferred region P_c.
- **Step 3**: Calculate the distance of each solution in archive to P_c and store these values to a list l_v.

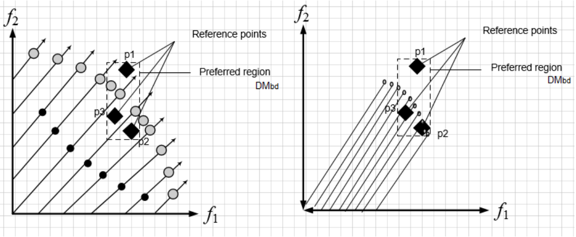

Fig. 4. Illustration of proposed ray based interactive method for DMEA in a 2-dim MOP. Three reference points are given by DM: p1, p2, p3. The system of rays is offset by DM's preferred region DM_{bd}.

Algorithm 2: Rays Redistribution Function.

```
Input: Number of reference points $n_p$
Output: New system of rays
• (1) Make a boundary of reference points
  (DM's preferred region) $DM_{bd}$.
• (2) Calculate the ratio between $DM_{bd}$ and
  current boundary of the hyperquadrant
  which contains the POF $r$.
for $j \leftarrow 0$ to n (The number of control points)
do
    • (3) Offset current control point with ratio $r$.
• (4) Generate a new system of rays by the
  new list of control points.
return n rays.;
```

- **Step 4**: Normalize the values of l_v to be in [0,0.5].
- **Step 5**: Adding values in l_v after calculate the niching values in Step 10.
- **Step 6**: Apply a niching (with additional values) to control external population (the archive) and next generation.

When DM interactive into the optimal process, we replace Step 7 in DMEA-II (see Section 3) with an interactive function is shown in Algorithm 3. Then the list is created above is used to add to niching values in Step 10 during generations.

Algorithm 3: Value Added Niching Function.

```
Input: Number of reference points $n_p$
Output: A list of values in [0,0.5]
    • (1) Make a boundary of reference points
      (DM's preferred region) and find the central
      point $p_c$.
for $j \leftarrow 0$ to popsize (The archive's size) do
    • (2) Calculate the Euclid distance
      from solution(j) to $p_c$.
    • (3) Normalize the distances
      to be in [0,0.5] and store in list $l_v$.
return $l_v$;
```

5. Experiment studies

5.1. Test functions

In our experiments, we use 10 2-dim test problems in well-known benchmark sets: ZDTs [15] and UFs [16]. Those test problems are described as below:

ZDT1: It has a convex Pareto-optimal front:

$$f_1(\vec{x}) = x_1,$$
$$f_2(\vec{x}, g) = g(\vec{x}).(1 - \sqrt{\frac{f_1(\vec{x})}{g(\vec{x})}}),$$
$$g(\vec{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i.$$

where n = 30, and $x_i \in [0, 1]$. The true Pareto front is formed with $g(\vec{x}) = 1$.

ZDT2: It has a concave Pareto-optimal front:

$$f_1(\vec{x}) = x_1,$$
$$f_2(\vec{x}, g) = g(\vec{x}).(1 - (\frac{f_1(\vec{x})}{g(\vec{x})})^2),$$
$$g(\vec{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i.$$

where n = 30, and $x_i \in [0, 1]$. The true Pareto front is formed with $g(\vec{x}) = 1$.

ZDT3: It has a Pareto-optimal front disconnected and convex:

$$f_1(\vec{x}) = x_1,$$
$$f_2(\vec{x}, g) = g(\vec{x}).(1 - \sqrt{\frac{f_1(\vec{x})}{g(\vec{x})}} - \frac{f_1(\vec{x})}{g(\vec{x})}.\sin(10\pi f_1(\vec{x}))),$$
$$g(\vec{x}) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i.$$

where $n = 30$, and $x_i \in [0, 1]$. The true Pareto front is formed with $g(\vec{x}) = 1$. The introduction of the sine function causes discontinuities in the Pareto optimal front. However, there is no discontinuity in the parameter space.

ZDT4: It contains 21^9 local Pareto fronts and, therefore, tests for the MOEAs ability to deal with multi-modality:

$$f_1(\vec{x}) = x_1,$$
$$f_2(\vec{x}, g) = g(\vec{x}).(1 - \sqrt{\frac{f_1(\vec{x})}{g(\vec{x})}}),$$
$$g(\vec{x}) = 1 + 10.(n-1) + \sum_{i=2}^{n}(x_i^2 - 10\cos(4\pi x_i)).$$

where $n = 10$, $x_1 \in [0, 1]$ and $x_2, ..., x_n \in [-5, 5]$. The true Pareto front is formed with $g(\vec{x}) = 1$. The best local Pareto front is formed with $g(\vec{x}) = 1.25$.

ZDT6: It includes two difficulties caused by the non-uniformity of the search space: rst, the Pareto optimal set is non-uniformly distributed along the Pareto front (the front is biased for solutions for which $f_1(\vec{x})$ is near to one); and second, the density of the solutions is lowest close to the Pareto front and highest away from the front.

$$f_1(\vec{x}) = 1 - exp(-4x_1).\sin^6(6\pi x_1),$$
$$f_2(\vec{x}, g) = g(\vec{x}).(1 - (\frac{f_1(\vec{x})}{g(\vec{x})})^2),$$
$$g(\vec{x}) = 1 + 9(\frac{1}{9}.\sum_{i=2}^{n}(x_i)).$$

where $n = 10$, $x_i \in [0, 1]$. The true Pareto front is formed with $g(\vec{x}) = 1$ and is non-convex.

UF1: The two objectives to be minimized:

$$f_1(\overrightarrow{x}) = x_1 + \frac{2}{|}J_1|\sum_{j\in J_1}[x_j\sin(6\pi x_1 + \frac{j\pi}{n})]^2,$$

$$f_2(\overrightarrow{x}) = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}[x_j\sin(6\pi x_1 + \frac{j\pi}{n}]^2$$

where $J_1 = \{j|j$ is odd and $(2 \leq j \leq n\}$ and
$J_2 = \{j|j$ is even and $2 \leq j \leq n\}$.
The search space is $[0, 1] \times [-1, 1]^{n-1}$.

UF2: The two objectives to be minimized:

$$f_1(\overrightarrow{x}) = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1} y_j^2,$$

$$f_2(\overrightarrow{x}) = 1 - \sqrt{x_1} + \frac{2}{|J2|}\sum_{j\in J_2} y_j^2$$

where $J_1 = \{j|j$ is odd and $(2 \leq j \leq n\}$ and
$J_2 = \{j|j$ is even and $2 \leq j \leq n\}$ and

$$y_j = \begin{cases} x_j - [0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n}) + \\ 0.6x_1]\cos(6\pi x_1 + \frac{j\pi}{n}) j \in J_1 \\ x_j - [0.3x_1^2\cos(24\pi x_1 + \frac{4j\pi}{n}) + \\ 0.6x_1]\sin(6\pi x_1 + \frac{j\pi}{n}) j \in J_2 \end{cases}$$

The search space is $[0, 1] \times [-1, 1]^{n-1}$.

UF3: The two objectives to be minimized:

$$f_1(\overrightarrow{x}) = x_1 + \frac{2}{|J_1|}(4\sum_{j\in J_1} y_j^2 2\prod_{j\in J_1}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 2),$$

$$f_2(\overrightarrow{x}) = 1 - \sqrt{x_1} + \frac{2}{|J_2|}(4\sum_{j\in J_2} y_j^2 2\prod_{j\in J_2}\cos(\frac{20y_j\pi}{\sqrt{j}}) + 2)$$

where J_1 and J_2 are the same as those of $UF1$,
and $y_j = x_j - x_1^{0.5(1.0+\frac{3(j2)}{n2})}, j = 2, ..., n,$.
The search space is $[0, 1]^n$.

UF4: The two objectives to be minimized:

$$f_1(\overrightarrow{x}) = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1} h(y_j),$$

$$f_2(\overrightarrow{x}) = 1 - x_1^2 + \frac{2}{|J_2|}\sum_{j\in J_2} h(y_j)$$

where $J_1 = \{j|j$ is odd and $(2 \leq j \leq n\}$ and
$J_2 = \{j|j$ is even and $2 \leq j \leq n\}$
$y_i = x_j\sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, ..., n$ and $h(t) = \frac{|t|}{1+e^{2|t|}}$.
The search space is $[0, 1] \times [-2, 2]^{n-1}$.

UF7: The two objectives to be minimized:

$$f_1(\overrightarrow{x}) = \sqrt[5]{x_1} + \frac{2}{J_1}\sum_{j\in J_1} y_j^2,$$

$$f_2(\overrightarrow{x}) = 1 - \sqrt[5]{x_1} + \frac{2}{J_2}\sum_{j\in J_2} y_j^2$$

where $J_1 = \{j|j$ is odd and $(2 \leq j \leq n\}$ and $J_2 = \{j|j$ is even and $2 \leq j \leq n\}$
$y_i = x_j\sin(6\pi x_1 + \frac{j\pi}{n}), j = 2, ..., n.$
The search space is $[0, 1] \times [-1, 1]^{n-1}$.

5.2. Results and Discussion

At the step 7 of DMEA-II, the estimated ideal point of the non-dominated solutions are identified in M and determine a system of n rays R. We replace this step with one of interactive functions in algorithms: 1, 2, 3 to guide the evolutionary process to make the population toward the DM's preferred region. Some typical snapshots for the experiments with several test problems are show in Figures: 5 to 14.

Through experiments with 10 test functions, the paper indicates some features of the interactive method:

1. By applying a niching to control external archive and next generation and replacing some rays in DM's preferred region, obtain solutions are converged to DM's preferred region in objective space.
2. The final solutions are distributed uniformly outside DM's preferred region, except DM's unexpected region (region that is the furthest from DM's preferred region). It means DMEA-II with interactive still be balanced in maintaining two properties: convergence and spreading of population and indirectly balance between exploration and exploitation.
3. The effect of the interactive method with 'rays redistribution' guides the evolutionary process strongly converged to DM's preferred region.

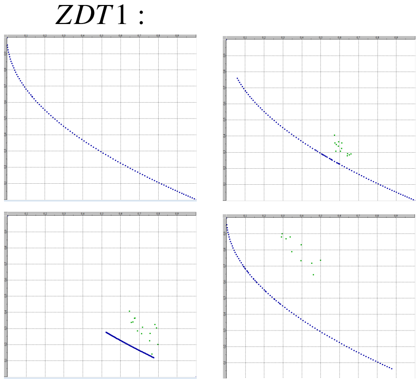

Fig. 5. Visualization of the interactive method on ZDT1 in orders: (1^{st}: Without interactive, 2^{nd}: Rays replacement, 3^{rd}: Rays redistribution, 4^{th}: Value added Niching).

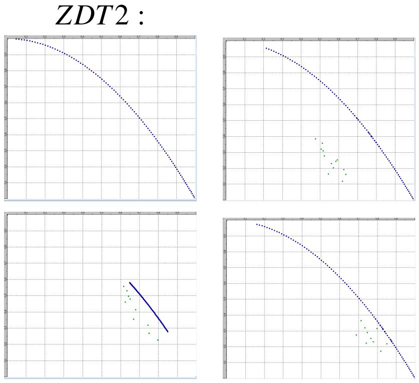

Fig. 6. Visualization of the interactive method on ZDT2 in orders: (1^{st}: Without interactive, 2^{nd}: Rays Replacement, 3^{rd}: Rays Redistribution, 4^{th}: Value Added Niching).

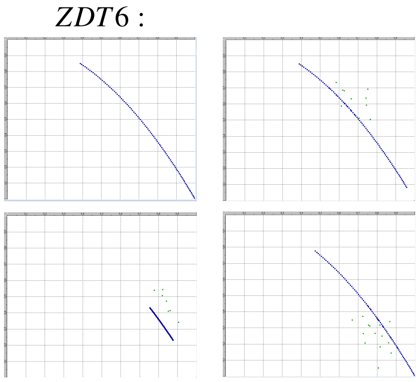

Fig. 9. Visualization of the interactive method on ZDT6 in orders: (1^{st}: Without interactive, 2^{nd}: Rays Replacement, 3^{rd}: Rays Redistribution, 4^{th}: Value Added Niching).

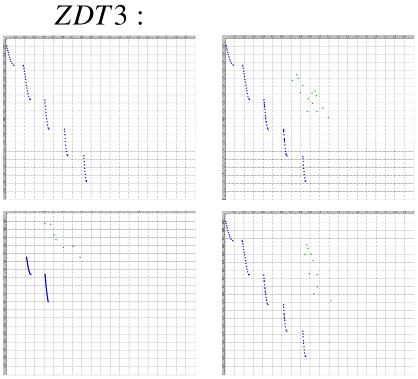

Fig. 7. Visualization of the interactive method on ZDT3 in orders: (1^{st}: Without interactive, 2^{nd}: Rays Replacement, 3^{rd}: Rays Redistribution, 4^{th}: Value Added Niching).

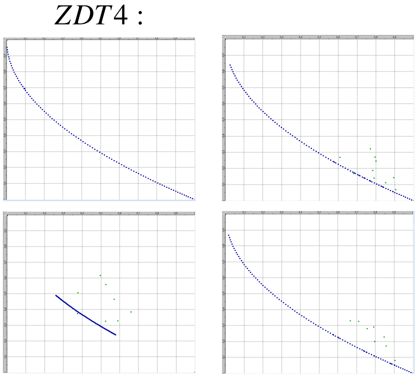

Fig. 8. Visualization of the interactive method on ZDT4 in orders: (1^{st}: Without interactive, 2^{nd}: Rays Replacement, 3^{rd}: Rays Redistribution, 4^{th}: Value Added Niching).

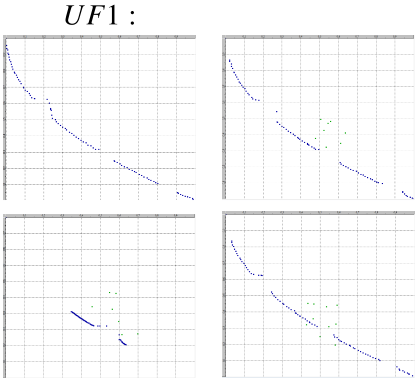

Fig. 10. Visualization of the interactive method on UF1 in orders: (1^{st}: Without interactive, 2^{nd}: Rays Replacement, 3^{rd}: Rays Redistribution, 4^{th}: Value Added Niching).

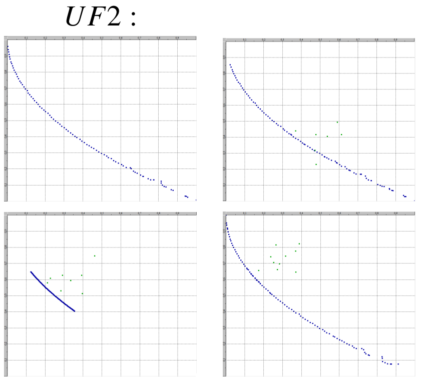

Fig. 11. Visualization of the interactive method on UF2 in orders: (1^{st}: Without interactive, 2^{nd}: Rays Replacement, 3^{rd}: Rays Redistribution, 4^{th}: Value Added Niching).

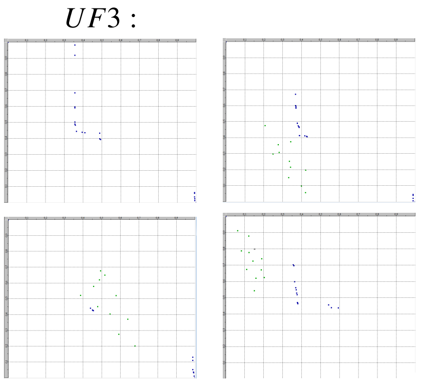

Fig. 12. Visualization of the interactive method on UF3 in orders: (1^{st}: Without interactive, 2^{nd}: Rays Replacement, 3^{rd}: Rays Redistribution, 4^{th}: Value Added Niching).

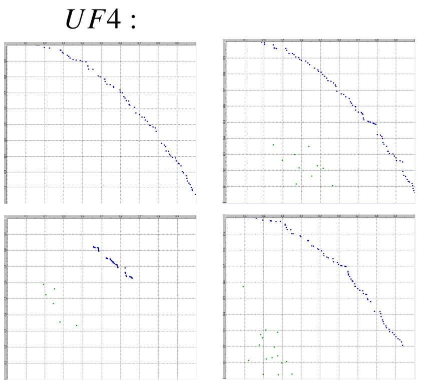

Fig. 13. Visualization of the interactive method on UF4 in orders: (1^{st}: Without interactive, 2^{nd}: Rays Replacement, 3^{rd}: Rays Redistribution, 4^{th}: Value Added Niching).

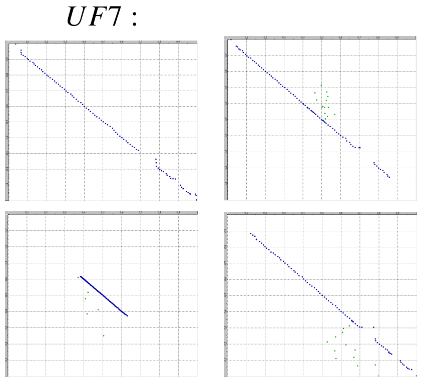

Fig. 14. Visualization of the interactive method on UF7 in orders: (1^{st}: Without interactive, 2^{nd}: Rays Replacement, 3^{rd}: Rays Redistribution, 4^{th}: Value Added Niching).

By using the interactive method with proposed approaches on DMEA-II and applying a niching in step 9 and step 10, the guiding technique through the using of reference points is used make the population to be converged to the DM's preferred region. It ensures convergence and spreading of population and concept to use two kind of improvement directions in DMEA. With the interactive method help DM to get the most preferred solutions.

6. Applying the interactive method to a Spam-Email Detection System

In this section, we present our application of the proposed interactive method to a real application: a Spam-Email Detection System, which relies on the well-known Spam-Assassin architecture. SpamAssassin is a common anti-spam system that developed 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 $/^\wedge \backslash s * DearFriend \backslash b/i$ (*).

- **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 $/^\wedge \setminus s * DearFriend \setminus b/i$. 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 [17].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 [18, 19, 20, 21]. 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. We first describe the problem formulation and then the system using interactive method proposed above.

6.1. Problem formulation

In recent years, the spread of spams is increasing considerably and seems to be uncontrollable. Stopping spammers has drawn an increasingly number of anti-spam approaches. 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 Spam Detection approach is to maximize the SDR and to minimize the FAR as much as possible. The key point of problem is that SDR is correlated with FAR. Thus, the higher rate of detecting spam an approach brings the higher probability to alarm a ham (non-spam mail) 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. This motivate us to consider multi-objectivity in this work. For the problem, the objective is also to find a set of ideal scores x where $x = (x_0, .., x_N), N = (30, 50, 100), x_0 \in [2, 5], (x_1...x_N) \in [0, 2]$.

The threshold T is corresponding to x_0 while the rest of the set are the score for rules.

The objective function is designed to run on the dataset S (spam) and H (ham).

- $S = s_1, s_2, .., s_K$
- $H = h_1, h_2, .., h_L$

The set of N rules is pre-designed based on the framework in [22].
$R = r_1, r_2, .., r_N$
Each rule might match with some spams or hams through the matching function: Where $r \in R; e \in \{S, H\}$ At threshold T, the function to detect spam is implemented as follows: Then the two

Algorithm 4: Spam Detection Function.

```
Input: e is an email
Output: 1 if e is spam, otherwise 0.
$score = 0;$
For i= 1 to N
    • $score+ = m(r, e) * x_i$
If($score \geq T$)
    • Then return 1.
Else return 0.
```

objectives SDR and FAR are compute against the formula:

$$SDR = \frac{\sum_{i=0}^{K} is\_spam(s_i)}{K}$$

$$FAR = \frac{\sum_{i=0}^{H} is\_spam(h_i)}{H}$$

In order to make the objectives are consistent, the SDR objective of this specific problem should be reformulate as (1 SDR) to get the maximum.

6.2. *Results of applying the interactive to SEDA*

This is a complicated multi-objective problem and with a large POF. If a DM is present, he/she might wish to impose their preference on the results. This is the case where we can use our proposed method. To apply the proposed interactive method to the DMEA-II for the Spam Email Detection System (SEDA), we need to define the set of reference points from DM (in the objective space). For example, a set of reference points are given is : $P^r = \{p^r_1, p^r_2, ..., p^r_m\}$. Here, m is number of reference point, $p^r_i(fSDR_i, fFAR_i)$ is the reference point i^{th} with SDR value is $fSDR_i$, FAR value is $fFAR_i$. We experiment with cases of 30, 50 and 100 rules respectively.

For the description of the system, the proposed Spam Email Detection System model integrated with DMEA-II is shown in Fig. 15).

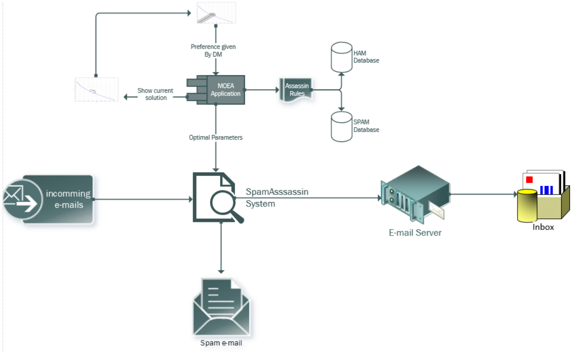

Fig. 15. Illustration of the proposed Spam Email Detection System which is integrated with DMEA-II.

The results for the real case study on the SEDA are shown in Figs: 16, 17, 18 by Rays Replacement approach, Figs: 19, 20, 21 by Rays Redistribution and Figs: 22, 23, 24 by Value Added Niching approach. In each figure the left graph indicates POF without interaction while the right one showed POF with interaction (after several repetitions). In each experiment, DM gave several reference points and those reference points represent the DM's preferred region in the objective space. By the proposal interactive method, DM expects to obtain more solutions for final decision in the area of their preferred region (on 1-SDR and FAR values).

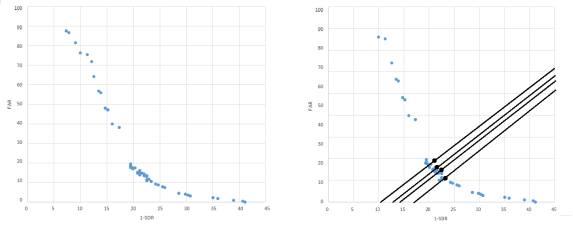

Fig. 16. Results for the proposal interactive method with Rays Replacement approach for SEDA in case of 30 rules. Before (left) and After (right) the interactive process.

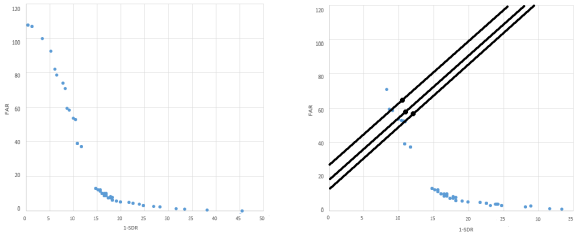

Fig. 17. Results for the proposal interactive method with Rays Replacement approach for SEDA in case of 50 rules. Before (left) and After (right) the interactive process.

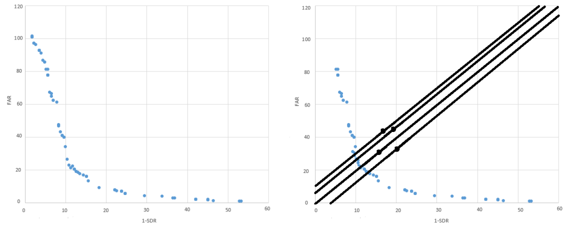

Fig. 18. Results for the proposal interactive method with Rays Replacement approach for SEDA in case of 100 rules. Before (left) and After (right) the interactive process.

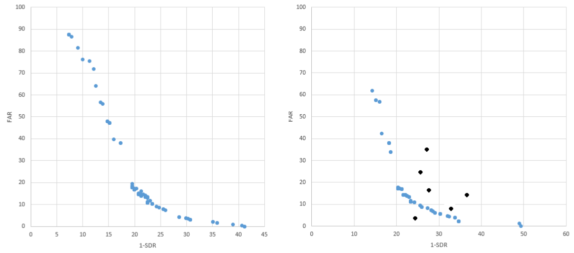

Fig. 19. Results for the proposal interactive method with Rays Redistribution approach for SEDA in case of 30 rules. Before (left) and After (right) the interactive process.

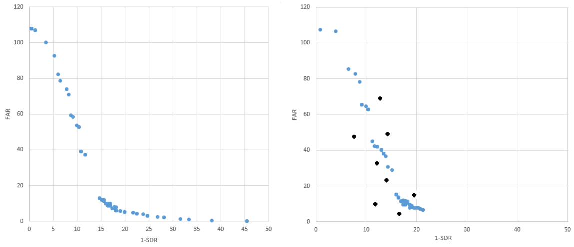

Fig. 20. Results for the proposal interactive method with Rays Redistribution approach for SEDA in case of 50 rules. Before (left) and After (right) the interactive process.

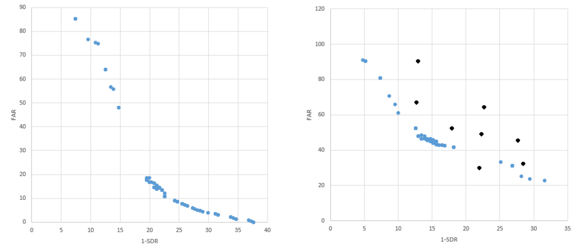

Fig. 21. Results for the proposal interactive method with Rays Redistribution approach for SEDA in case of 100 rules. Before (left) and After (right) the interactive process.

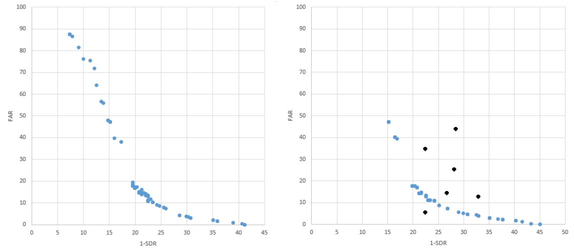

Fig. 22. Results for the proposal interactive method with Value Added Niching approach for SEDA in case of 30 rules. Before (left) and After (right) the interactive process.

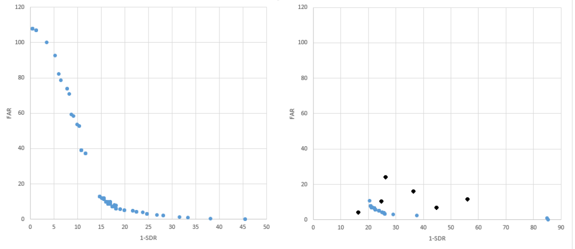

Fig. 23. Results for the proposal interactive method with Value Added Niching approach for SEDA in case of 50 rules. Before (left) and After (right) the interactive process.

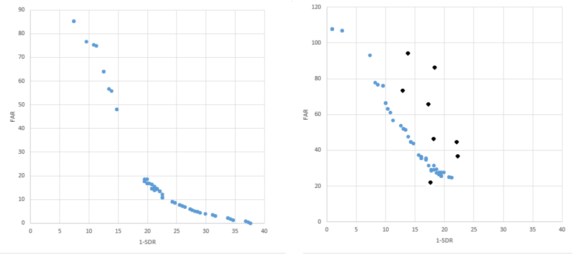

Fig. 24. Results for the proposal interactive method with Value Added Niching approach for SEDA in case of 100 rules. Before (left) and After (right) the interactive process.

It is observed from the figures that the original POF in all three cases are quite large (the left graphs). With the present of DM, POFs were contracted towards the areas of preference. That is the effect of reference points , which was used to create the reference rays in DMEA-II. If the loop is continued, the estimated POFs will approach the POF's part that is closest to the preferred area.

7. Conclusion

In this paper, we proposed an interactive method using multi reference points with direction based multi-objective evolutionary optimization-II (DMEA-II). In our alternative method, there are three ray based approaches are used: In Rays Replacement, the furthest rays from DM's preferred region are replaced by new rays that generated from set of reference points. In Rays Redistribution, the system of rays will be redistributed to be in DM's preferred region. And in Value Added Niching, based on the distances from non-dominated solutions in archive to DM's preferred region, the niching values for the solutions is increased to be priority selected. By applying a niching with those ray based approaches, the final solutions strongly converged to the DM's preferred region. It ensures convergence and spreading of population and concept to use two kind of improvement directions. With the interactive method help DM to get the most preferred solutions and concept of using two kind of improvement directions: Spread direction and Convergence direction.

By applying this method to a real application such as an Spam Email Detection System, according to the results, when solving the problem using DMEA-II with the proposed interactive method, it achieved more efficient results but also created a set of ready-to-use rule scores. These scores supports different levels of the trade-off between SDR and FAR. By obtained solutions, it gives users more flexibility and efficiency for system configuration.

Acknowledgment

Our experiments were done at Software Technology Lab of Faculty of Information Technology, Le Quy Don Technical University.

References

[1] L. Thiele, K. Miettinen, P. J. Korhonen, J. Molina, A preference based evolutionary algorithm for multi-objective optimization (2009) 411–436.

[2] K. Deb, A. Kumar, Interactive evolutionary multi-objective optimization and decision-making using reference direction method, in: GECCO '07, 2007, pp. 781–788.

[3] K. Deb, A. Sinha, P. J. Korhonen, J. Wallenius., An interactive evolutionary multi-objective optimization method based on progressively approximated value functions.

[4] K. Deb, J. Sundar, Reference point based multi-objective optimization using evolutionary algorithms, in: GECCO '06: Proceedings of the 8th annual conference on Genetic and Evolutionary Computation, ACM Press, New York, NY, USA, 2006, pp. 635–642.

[5] M. Gong, F. Liu, W. Zhang, L. Jiao, Q. Zhang, Interactive moea/d for multi-objective decision making, in: GECCO' 2011, 2011, pp. 721–728.

[6] E. B. J., D. K., M. K., S. R., Berlin, Consideration of partial user preferences in evolutionary multi-objective optimization. multi-objective optimization: interactive and evolutionary approaches, OR Spectrum.

[7] E. Petri, M. Kaisa, Trade-off analysis approach for interactive nonlinear multiobjective optimization, in: OR Spectrum, 2011, pp. 1–14.

[8] B. V., B. J., E. P., G. S., M. J., R. F., Slowinski, Interactive multi-objective optimization from a learning perspective. multi-objective optimization: interactive and evolutionary approaches, OR Spectrum.

[9] L. Nguyen, L. T. Bui, A decomposition-based interactive method formulti-objective evolutionary algorithms, The Journal on Information Technologies and Communications (JITC) 2 (2).

[10] L. Nguyen, L. T. Bui, A multi-point interactive method for multi-objective evolutionary algorithms., in: The fourth International Conference on Knowledge and Systems Engineering (KSE 2012), Da Nang, Vietnam, 2012.

[11] L. Nguyen, L. T. Bui, H. Abbass, Dmea-ii: the direction-based multi-objective evolutionary algorithm-ii, Soft Computing (2013) 1–16doi:10.1007/s00500-013-1187-3.

[12] W. A., The use of reference objectives in multi-objective optimisation, MCDMtheory and Application, Proceedings. No. 177 in Lecture notes in economics and mathematical systems (1980) 468–486.

[13] E.-G. T. A. P. W. Jos R. Figueira, Arnaud Liefooghe, A parallel multiple reference point approach for multi-objective optimization, European Journal of Operational Research, vol. 205 (2010) 390–400.

[14] K. Deb, Multiobjective Optimization using Evolutionary Algorithms, John Wiley and Son Ltd, New York, 2001.

[15] E. Zitzler, L. Thiele, K. Deb, Comparision of multiobjective evolutionary algorithms: Emprical results, Evol. Comp 8 (1) (2000) 173–195.

[16] S. Z.-P. S. W. L. S. T. Q. Zhang, A. Zhou, Multiobjective optimization test instances for the cec 2009 special session and competition.

[17] S. A., SpamAssassin, O'Reilly, 2004.

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

[19] A. López-Herrera, E. Herrera-Viedma, 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.

[20] I. Yevseyeva, V. Basto-Fernandes, D. Ruano-Ordás, J. R. Méndez, Optimising anti-spam filters with evolutionary algorithms, Expert Systems with Applications.

[21] V. Basto-Fernandes, I. Yevseyeva, J. R. Méndez, Optimization of anti-spam systems with multiobjective evolutionary algorithms, Inf. Resour. Manage. J. 26 (1) (2000) 54–67.

[22] F. J.-V. T. M.T. Vu, Q.A. Tran, Multilingual rules for spam detection, Proceedings of the 7th International Conference on Broadband and Biomedical Communications (IB2COM 2012) (2012) 106–110.