

Received July 27, 2016, accepted August 10, 2016, date of publication September 7, 2016, date of current version November 18, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2605759

Interactive Reference Region Based Multi-Objective Evolutionary Algorithm Through Decomposition

**RUOCHEN LIU, (Member, IEEE), RUINAN WANG, WEN FENG, JUNJUN HUANG,
AND LICHENG JIAO, (Senior Member, IEEE)**

Key Laboratory of Intelligent Perception and Image Understanding Key of Ministry of Education, International Research Center for Intelligent Perception and Computation, Xidian University, Xi'an 710071, China

Corresponding author: R. Liu (ruochenliu@xidian.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61373111, Grant 61272279, Grant 61672405, and Grant 61203303, in part by the Fundamental Research Funds for the Central Universities under Grant K50511020014, Grant K5051302084, Grant K50510020011, Grant JB150227 and JBG160229, and in part by the Provincial Natural Science Foundation of Shaanxi of China under Grant 2014JM8321.

ABSTRACT Many evolutionary multi-objective optimization (EMOs) methodologies have been proposed and shown a great potential in approximating the entire Pareto front. While in real world, what decision makers (DMs) want is one or several solutions to satisfy their requirements. It has become a hot problem that dynamically using preference information provided by DMs during the optimization process guides the search of EMO algorithms. An interactive reference region-based evolutionary algorithm through decomposition is proposed, denoted as RR-MOEA/D in this paper, which focuses the search on the desire of DMs to save computational resources. MOEA/D, as a well-known multi-objective optimization algorithm, is used as a basic framework here. In MOEA/D, by dealing with the sub-problems in the preference region and ignoring uninterested ones, the solutions obtained can converge to the regions which the DM prefers on the Pareto front and the computational complexity can be saved to a great extent. At each interaction, a humanized and simple interactive condition is adopted so that the reference region can be changed in a very intuitive way if the DM is unsatisfied the results in the interactive process. A rapid interaction is designed and a set of rough solutions can be obtained quickly whenever the preference information is changed. The proposed algorithm is tested on several benchmark problems and the experimental results show that the proposed algorithm can take full use of preference information and successfully converge to the reference region due to its reasonable and simple interaction mechanism.

INDEX TERMS Interactive preference, multi-objective optimization problem, reference region selection mechanism.

I. INTRODUCTION

Evolutionary multi-objective optimization algorithms (EMOs) have been widely applied in many engineering areas. Multiple objectives with conflicts make it impossible to get a single optimal solution which can optimize all the objectives at the same time. As a result, the optimal solutions obtained by EMOS are trade-offs, known as Pareto optimal solutions, which are characterized by that some improvement in one objective must lead to deterioration in at least one other objective.

Without incorporating preference information, an approximation to the entire Pareto front (PF) is obtained by most

EMOs. However, in many practical applications, what a decision maker (DM) requires may be a single solution or a set of solutions in a preferred region instead of entire PF. It is very time-consuming and unnecessary to obtain the whole PF which DMs are not interested in. Meanwhile, many preference information based EMOS have been proposed to select the most desirable solutions on the Pareto-optimal front, and those can be mainly divided into three categories: prior, interactive and posterior. The apriori approaches require the DM provides his/her preference information in advance. Often this preference information is used as a variant of selection mechanism to substitute or compose the

tradition one. A disadvantage of these apriori approaches lies in that it is very difficult and confused for a DM to express his/her preference without related knowledge of available solutions or structure of the search space. On the other hand, for those posterior approaches, the DMs can choose those preferred solutions on the basis of the convergence and spread of solutions. Disadvantages and criticisms for the posterior methods include high computational costs and time consumption, and it will become more difficult when dealing with many-objective problems (more than three objectives). The interactive, as one of the preference methods, is a new trend. It can be described as a DM dynamically guides the searching process by providing their reference information in an interactive way until he/she is satisfied with the outcome. The interactive methods allow DMs to guide searching and modify their preferences at various stages during the optimization process. Moreover, ignoring those regions with little interest and applying reference information as guidance can effectively reduce the computational costs.

Multi-objective evolutionary algorithm based on decomposition (MOEA/D) [1] is selected as a basic MOEA framework in this paper. MOEA/D decomposes a multi-objective problem (MOP) into multiple single-objective subproblems in which each subproblem is a different aggregation of all the objectives. From this viewpoint, the aggregation can be described as different preference of objectives, in other words, different subproblems are generated under different preference condition. Therefore, it is a convenient way to find the regions where the DM is interested by solving the corresponding subproblems according to the DM's preferences.

In this paper, we propose an interactive reference region based multi-objective evolutionary algorithm through decomposition (denoted as RR-MOEA/D), which focus the search on the desire of DMs to save computational resources. Once the preference is given by a DM at interactive stage, the proposed algorithm reduces the computational cost by ignoring the regions the DM is not interested in and only optimizing the subproblems the DM prefers. Whereas during optimization process, weight vectors of subproblems preferred is adapted dynamically to avoid the solutions we obtain flying away from the preferred regions. At each interaction, our algorithm provides a set of current solutions to the DM and the reference region can be changed in a very intuitive way if the DM is unsatisfied. Therefore, a rapid interaction is designed in this paper. A set of rough solutions can be obtained quickly whenever the preference information change.

The major advantages of the proposed RR-MOEA/D algorithm are enumerated as follows:

- 1) Reduce the amount of calculation and save the computing resources by focusing the search on the preferred regions;
- 2) It needs not to calculate any achievement scalarizing functions (ASF) [2] used in other preference mechanisms;
- 3) A simple and intuitive way to set the reference information the DM prefers;

4) A fast way to get rough solutions in the reference regions can avoid wasting time in searching Pareto-optimal solutions in regions the DM is not preferred;

5) Giving uncertain and inconsistent human decisions as the DM does in real life are allowed and changing the preference information in the interaction stage is permitted.

The remainder of this paper is structured as follows. Section 2 introduces the related background. The proposed algorithm is presented in Section 3 in details. Section 4 presents experimental results and validates the performance of the proposed algorithm. Section 5 concludes the paper.

II. RELATED BACKGROUND

A. MULTI-OBJECTIVE OPTIMIZATION

Generally, a minimized multi-objective optimization problem (MOP) can be defined as follows:

$$\begin{aligned} \min F(\mathbf{x}) &= (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ s.t. \quad \mathbf{x} &\in \Omega \end{aligned} \quad (1)$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is the decision vector, n is the dimensionality of the decision space, and Ω is the feasible region in the decision space. $F(\mathbf{x})$ is a set of objective function consisting of m objective functions.

Very often, since the objectives in Eq. (1) conflict each other, it is impossible to get a single optimal solution which optimizes all the objectives at the same time. As a result, the optimal solutions we obtained are trade-offs, known as Pareto optimal solutions. Some important related terms are stated as follows:

Definition 1 (Pareto-Dominance): Let $\mathbf{x}_a, \mathbf{x}_b \in \Omega$, \mathbf{x}_a is said to Pareto dominate \mathbf{x}_b , denoted as $\mathbf{x}_a \succ \mathbf{x}_b$ if and only if:

$$\begin{cases} \forall i = 1, 2, \dots, m, \quad f_i(\mathbf{x}_a) \leq f_i(\mathbf{x}_b) \\ \exists j = 1, 2, \dots, m, \quad f_j(\mathbf{x}_a) < f_j(\mathbf{x}_b) \end{cases} \quad (2)$$

Definition 2 (Pareto-Optimal): A point is Pareto optimal to Eq. (1) if:

$$\neg \exists \mathbf{x} \in \Omega : \mathbf{x} \succ \mathbf{x}^* \quad (3)$$

The Pareto-optimal set (PS) consists of all the Pareto-optimal solutions and the set of all the Pareto optimal objective vectors is called Pareto front (PF) [3].

B. MULTI-OBJECTIVE OPTIMIZATION BASED ON DECOMPOSITION

Decomposition is a basic strategy in traditional multi-objective optimization. Relying on a decomposition strategy such as the weighted-sum method [3], the Tchebycheff approach [3] or the boundary intersection approach [4], [5], a multi-objective problem can be converted into a number of single-objective problems. In 2007, Zhang et al. [1] proposed a decomposition algorithm based evolutionary multi-objective algorithm using the Tchebycheff as the decomposition strategy. Since we use the Tchebycheff approach in our study, a brief introduction of the Tchebycheff approach is given in the following.

The Tchebycheff method plays an important role not only in converting a multi-objective problem into a number of single-objective problems [3] but also in locally approximating the underlying preference utility function [6]. Each subproblem is obtained by using the following equation [1], [3]:

$$\begin{aligned} \text{minimize } g^{te}(\mathbf{x} | \boldsymbol{\lambda}, \mathbf{z}^*) &= \max_{1 \leq i \leq m} \{ \lambda_i |f_i(\mathbf{x}) - \mathbf{z}_i^*| \} \\ \text{subject to } \mathbf{x} &\in \Omega \end{aligned} \quad (4)$$

Where, for $i \in \{1, 2, \dots, m\}$, \mathbf{z}_i^* is the ideal point, i.e., $\mathbf{z}_i^* = \min \{f_i(\mathbf{x}) | \mathbf{x} \in \Omega\}$. For any weight vector $\boldsymbol{\lambda}$, the optimal solutions of Eq. (4) corresponds to a Pareto-optimal solution of Eq. (1). All the optimal solutions obtained can approximate to the entire PF with proper weight vectors.

As shown in Eq. (4), each subproblem can be translated as a different aggregation of all the objectives and different aggregation could be constructed as the different preference for objectives. By setting proper weights, solutions desired by DMs can be obtained by solving those related subproblems in the preference regions. Therefore, associating preference method with decomposition is a very natural thought. As the most common decomposition based multi-objective evolutionary algorithms, MOEA/D [1] has shown great potential. Here is a brief introduction of MOEA/D and its variants.

MOEA/D uses the Tchebycheff method as a decomposition method and converts a multi-objective problem into a set of subproblems and optimizes all the subproblems simultaneously in a single run. Different subproblems in MOEA/D have different weight vectors. For each weight vector, a neighborhood is defined as a set of its several closest weight vectors. Offspring of each subproblem is generated by using some genetic operators and its parents are randomly selected from its neighborhood. The solution of each subproblem is the best solution found so far in the neighborhood. That is to say, each subproblem optimized by its neighboring subproblems only obtain its optimal solution in evolving process, which makes that MOEA/D has such advantages as low computational complexity and fast convergence rate. MOEA/D-DE [7] is an improved version of MOEA/D, which has three improvements: 1) using a differential evolution (DE) [8] operator instead of SBX operator [9] in updating process; 2) setting a probability that parent solutions are selected from the neighborhood; 3) setting a upper limit number of solutions in neighborhood replaced by each child solution. All these three improvements aim at maintaining the population diversity. Since our algorithm works only in reference regions and has a small population size, keeping diversity is an important task. So, the proposed algorithm is mainly based on MOEA/D-DE in fact. As another new version of MOEA/D, MOEA/D-M2M [10] converts a multi-objective problem into several simple multi-objective problems. For each subproblem, any Pareto based EMOs can be used. MOEA/DD [11] suggests a hybrid paradigm which combines decomposition and dominance together. Most recently, an inverse model based MOEA (IM-MOEA) [12] is proposed, which uses a set of reference vectors to partition the objective space into a

number of subspaces and then the inverse models which maps the objective vectors into the decision vectors are built in each subspace to sample new solutions. More recently, a reference vector guide evolution algorithm for many-objective optimization (REVA) [13] is proposed, in which reference vectors is used to decompose the original multi-objective optimization problem into a number of single-objective subproblems. Additionally, a scalarization approach, termed the angle penalized distance, is adopted to balance convergence and diversity of the solutions in the high-dimensional objective space. NSGA-III [14] combines a decomposition strategy in MOEA/D [1] with non-dominated sorting approach from NSGA-II [15], and has been able to successfully find a well-converged and well-diversified set of solutions for many-objective problems.

C. PREFERENCE INFORMATION BASED EMO

The goal of evolutionary search is to help that the DM selects the solution which best matches his/her preferences within a representative set of Pareto optimal solutions. Various preference-based EMOs have been proposed in the last decade. Two surveys were provided in this topic [16], [17]. EMOs are categorized apriori, interactive, or a posteriori algorithms according to the treatment of preference. An apriori approach requires the preference information provided by the DM before the optimization stage and the evolutionary search is guided by the preference information. On the other hand, a posteriori approach firstly needs to approximate to the whole Pareto front, and then the DM selects their preferred solutions. With the character of the user involving in the optimization process, the interactive approaches offer a chance for the user to know about the structure of the search space and express his /her authentic preference correctly, and computational costs is reduced largely by progressively focusing on the most relevant areas and ignoring regions of little interest. We will give a simple review about the three approaches in the following section.

Some typical apriori methods are listed here: Wierzbicki [18] proposed a reference point based optimization of classical multi-criterion decision making algorithm which combines three approaches in a single multi-objective optimization framework and enables the DMs to refine the problem definition and to reduce the size of the objective space iteratively. Jaszkiewicz and Slowinski [19] proposed a light beam search procedure which connects the reference point with tools of multi-attribute decision analysis. This algorithm tries to overcome the inconvenience when the DM chooses the candidates for the best compromise. Deb et al. [20] hybridized the reference direction strategy with NSGA-II, in which, a reference direction can be determined by setting a starting point and a reference point, and reference solutions along the reference direction on PF can be obtained. Deb and Kumar [21] combined a reference point based reference information with NSGA-II in order to find a set of solutions instead of a single optimal solution and used a modified crowding distance to control the

density of solutions near reference points. Thiele et al. [22] proposed a preference based evolutionary approach which directly uses reference point information in fitness evaluation. Recently, Molina et al. [23] proposed another definition of Pareto dominance called g-dominance, in which, solutions satisfy all aspiration levels and solutions fulfilling none of the aspiration levels are preferred over those solutions satisfying some aspiration levels. All these algorithms mentioned above belong to the apriori method.

For the posteriori approach, in fact, many famous EMOs like MOEA/D [1], NSGA-II [15], SPEA-II [24], NNIA [25], and so on belong to it. After those algorithms obtain a set of compromised solutions which are closest to the true Pareto front and the DMs pick out the solutions they are interested in.

An interactive process aims to integrate the preference information provided by DMs into algorithm in an interactive way during the optimizing process. That is to say, some trade-off factors expressing the desires of the DMs are added into the optimization process to guide the latter search. According to [26], Tanino et al. [27] suggested the earliest attempt to incorporate the preference information expressed iteratively by the DM to evaluate solutions. Fonseca and Fleming [28] allowed the user to specify aspiration levels in form of a reference point, and used it to modify the EMO's rank scheme in order to focus the search. Kita et al. [29] interleaved generations of a Pareto ranking based evolutionary algorithm with the rank of the solutions provided by a DM. Todd and Sen [30] also tried to learn the user's utility function, instead of only considering linear weightings of objectives, they used the preference information provided by the DM to train an artificial neural network, which is then used to evaluate solutions in the evolutionary algorithm. Parmee et al. [31] described the concept of an interactive evolutionary design system which is related to multi-objective information gathering and subsequent design space redefinition. Instead of using linear programming to derive a weighting of the objectives most compatible with the pairwise comparisons as in [32], Phelps and Köksalan [33] used two evolutionary algorithms, one is to find the solutions, and another is to determine the most compatible ranking. Kamalian et al. [34] suggested using a posteriori evolutionary multi-objective optimization followed by an interactive evolutionary algorithm. Reference [35] allowed the user to modify the Pareto ranking computed automatically by changing the rank of some of the solutions. Thiele et al. [36] also used DM's preferences interactively expressed in the form of reference points. Avigad and Moshaiov [37] presented a new interactive concept based multi-objective evolutionary algorithm to handle Pareto-directed IC-MOPs. Said et al. [38] described an interactive run of r-NSGA-II algorithm in which is changed for a certain number of generations freely determined by the DM, and if the running result the DMs are dissatisfied, they can change the reference point. Deb et al. [39] presented a progressively interactive EMO approach which applied the constructed value function to search more preferred solutions.

And an interactive evolutionary algorithm can contribute to designing elegant object-oriented software which was discussed by Simons and Parmee [40]. Sun et al. [41] proposed a new surrogate-assisted interactive genetic algorithm to exploit the uncertainty of subjective fitness evaluations both in training the surrogates and in managing surrogates. Ruiz et al. [42] developed an interactive WASF-GA which is based on a preference-based evolutionary multiobjective optimization algorithm called WASF-GA [43], it required the DM to provide preference information at each iteration simply as a reference point and then the desired number of solutions is generated to represent the region of interest of the Pareto optimal front associated to the reference point given.

Among the interactive algorithms, those algorithms based on decomposition strategy are very few. We just found one references on this topic. iMOEA/D [44] is an interactive version of the MOEA/D algorithm [1], in which the preferred solutions are presented to the decision maker at intermediate generations, then, the searching process is guided to the neighbor regions of the preferred solution. It is obvious that both iMOEA/D and the proposed algorithm are based on MOEA/D although there are some little differences like initiation and updating of population. The biggest differences between the two algorithms lie in the way of determining preferred region and interactive method. In iMOEA/D, a generic polynomial utility function [45] is used to represent the human DM. During the stage of interactive, P current solutions are randomly presented to DM for choosing their favorite one; and the selected solution becomes the center of preferred weight region in the following optimization process. The detailed descriptions of about the way of determining preferred region and interactive method in the proposed algorithm is presented in Section 3.

III. THE PROPOSED APPROACH

In this section, the proposed reference region based multi-objective evolutionary algorithm through decomposition, is presented in details. If the final aim is to choose and implement a solution, then the goal of applying a multi-objective optimization method is to find a single, most preferred, final solution. However, in some cases, it may be preferable to find a set of solutions instead of one. This may be particularly true in case of robustness considerations when some aspects of uncertainty, imprecision or inconsistency in data or model are to be taken into account [26]. For most of preference based multi-objectives evolution algorithms, the preference information is treated as a variant of special selection mechanism to substitute or compose the traditional selection process [46], and the whole population converges close to the reference point or reference direction [44]. MOEA/D-M2M [10] converts a multi-objective problem into several simple multi-objective problems based on PF or PS shapes and searches and optimizes each simple multi-objective problem respectively. Inspired by all these, we consider that a reference region can be treated as a simple multi-objective problem and thus the search is focused on solving this problem only.

As a result, a reference region based preference mechanism is designed in which the population evolves only in this region where the DM desires. Provided the reference information, it is possible to find the preference regions and then initialize the individuals in the preference regions and search in the region until approximating properly to the Pareto front. At the initial stage, MOEA/D [1] is used to generate the initial population. When the preference information is given, namely one or several reference regions are determined and those new population are formed in the preference regions corresponding to some subproblems. Whereas in the evolution process, the subproblems are updated by using MOEA/D-DE until the DM is satisfied. Here it should be emphasized that each individual solution belongs to different corresponding subproblem and each subproblem corresponds to certain weight. Thus collective and dynamic adapting the weight vectors in the course of evolution can ensure not only the distribution range but also the convergence in the preference region. At the interactive stage, a humanized and simple interactive condition is adopted in this paper. A set of rough solutions are presented to DM when the interactive condition is met. If the DM is satisfied with the present solutions, the process would be terminated; otherwise, the decision maker could change his/her prefer information, a similar process is started until the DM obtain the satisfied solutions. Therefore, the DM can control the preference information in a very intuitive and fast way.

A. PREFERENCE REGION

The first challenge in the proposed approach is how to determine the preference region. As we know, interaction involves the DM in the process of optimization for dynamically offering preference information. And the DM dynamically guides the algorithm to search in the preference region in the interaction processing. Therefore, an effective method should be found for determining the preference region when the interaction happens.

As a basic algorithm framework in this paper, MOEA/D [1] is characterized by 1) each subproblem is a independent single-objective optimization problem and the solutions of all the subproblems form the entire population; 2) all subproblems in MOEA/D spread uniformly in objective space and each subproblem and its neighbor subproblems form a preferred region. That is to say, finding the optimal subproblem which is nearest to the reference direction or the reference point and these individuals in its neighbor can form a preference region. In this paper, those reference directions are selected as the preferred information which are marked by owning more intuitive and better linked with weight vectors. As a result, the preferred regions can be determined by two factors in the proposed algorithm: one is the searching direction which is determined by an aspiration and a reservation point provided by DMs. another is radius which determines the range of reference region. So, once a DM offers a searching direction, we can find the optimal weight vector close to the searching direction and obtain the neighbor

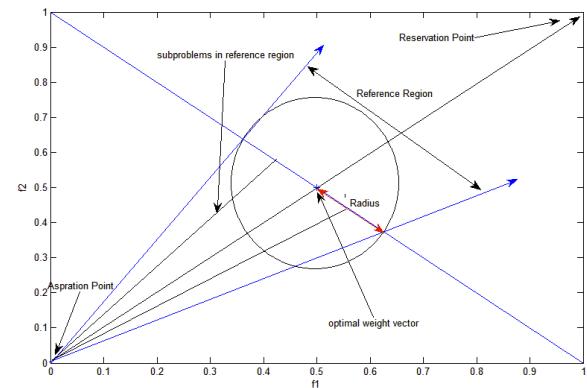


FIGURE 1. The illustration of determining the preference region.

of the optimal subproblem by computing the Euclidean distances between the optimal weight vector and other weight vectors. By setting the radius and reasonably initializing the weight vectors in this range, we can get subproblems in the preference region. Fig. 1 shows this process visually. Here, we can see that the vector determined by aspiration point and reservation point is just what we define as the reference direction. The radius determines the scope of reference region, and then we can find the optimal vector and other subproblems in this reference region.

Finding an optimal weight vector close to the searching direction is not trivial. In this paper, we employ the basic MOEA/D [1] to generate the initial population. To determine optimal weight vector, we connect each individual to the aspiration point so as to form a vector and compute the Cosine value between this vector and the searching direction:

$$\cos(\mathbf{a}, \mathbf{r}) = \frac{\mathbf{a} \cdot \mathbf{r}}{|\mathbf{a}| \times |\mathbf{r}|} \quad (5)$$

where vector \mathbf{a} is obtained by connecting individual \mathbf{x} with the aspiration point and vector \mathbf{r} is the searching direction. The bigger the Cosine values, the closer to the searching direction the individual is. We can select the subproblem's weight vector with the biggest Cosine value as the optimal weight vector and re-initialize all the subproblems in the reference region. Algorithm 1 shows the process of the initializing the weights and finding initial population in the reference region in details.

The human decision is characterized by uncertainty and inconsistency [45]. In the proposed algorithm, the reference information and the size of population are given by the decision maker. It is a common thing that the parameters the DM inputs might not well match with each other. Therefore, we should take account of various circumstances. If the number of subproblems (population size) is too small to cover entire preferred region, the subproblem in the region most close to the reference direction will become a priority (Algorithm 1: line 8-9). In this paper we set the parameter *penalty* to 1.5 to control the range of small population (it can also be changed in some specific application). Otherwise,

Algorithm 1 REFREGION**Iteration Parameters:**

num_v: the number of reference directions
PF: initial population
N: the size of population close to each reference direction
start: the vector of *num_v* aspiration points
fend: the vector of *num_v* reservation points
radius: determine the size of reference region

Output:

D₀ : initial population in reference region
W₀ : initial weight vectors in reference region

```

1. ref_dir  $\leftarrow$  fend – start
2. for i = 1, 2, ..., num_v do
3.   best_weighti  $\leftarrow$  max(cos_value(PF, ref_dir))
4.   range_weighti  $\leftarrow$  max(euc_dis(all_weight, best_weighti) < radius)
5.   L = length(range_weighti)
6.   for j = 1, 2, ..., L
7.     calculate the interval distance dj between the jth and the (j + 1)th weight
8.     if L > penalty*N(:, i)
9.       Wi  $\leftarrow$  init_weight(min_euc_dis(range_weighti, N(:, i)))
10.    else if L < penalty*N(:, i) & & L > N(:, i)
11.      Wi = range_weighti
12.    while L > N(:, i)
13.      find out the weight w with minimum interval distance d
14.      Wi  $\leftarrow$  delete w from Wi
15.      L – –
16.    end while
17.  else
18.    Wi = range_weighti
19.    while L < N(:, i)
20.      find out the weight w with maximum interval distance d
21.      Wi  $\leftarrow$  insert a middle weight between w and the following weight of w in Wi
22.      L ++
23.    end while
24.  end if
25. end for
26. Di  $\leftarrow$  max(cos_value(PF, ref_dir(:, i)), N(:, i))
27. D0  $\leftarrow$  (D1, ..., Dnum_v), W0  $\leftarrow$  (W1, ..., Wnum_v)
28. end for

```

the subproblems could be evenly distributed in the preference region to satisfy requirements (Algorithm 1: line 10-23). Consequently, the algorithm can keep diversity and robustness as well.

B. Interactive Condition

As one of the preference methods, the interactive ones allow a DM to dynamically guide the searching process until he/she is satisfied with the resulting outcome. Whereas a key problem

is when the current solutions are presented to the DM, in many interactive algorithms, the process of interacting with a DM will happen periodically [45], [46]. A long time interval between two interactions may lead to a waste of time in searching the region the DM does not favor. On the contrary, it is tiresome to bother the DM excessively and it is also a tiring work to find several preferred points from a large number of random points in the beginning. Determining when the algorithm interacts with the DM is important. We use an indicator to determine whether the rough result needs to be presented to the DM which is defined as follows:

$$\delta = \frac{1}{N} \sum_{i=1}^N \frac{f_t^i - f_{t-1}^i}{f_{t-1}^i} \quad (6)$$

Where *N* is the population size in the reference region, and *f_tⁱ* denotes the objective value of individual *i* at the *t*th generation. In this case, the function *f* is one with the minimum value of the individual *i*. Further, δ means the average improvement degree of function value that the current individuals are compared with their parent individuals. The precision of the solutions presented to the DM can be controlled by setting the value of δ . In other words, the parameter δ is determined by the DM, when the condition of *derta* $\leq \delta$ is satisfied, the interaction happens.

Let us make some detailed illustration about a few aspects of the Algorithm 1.

Step1: The reference direction(s) must be given by the DM, in which aspiration points are *start* and reservation points are *fend*.

Step3 to Step4: The two steps are designed to calculate the best weight corresponding to the reference direction and estimate the range weight around the preference direction simultaneously.

Step5 to Step 26 used to determine the initial population and the initial weight according to the relationship between the preference region weight.

C. RR-MOEA/D

The overall procedure of the proposed interactive preference region based multi-objective evolutionary algorithm through decomposition (RR-MOEA/D) is described in Algorithm 2.

Here, we give a more detailed explanation about several key steps.

Initialization: In order to get uniform distribution of the initial solutions and to make sure any region the DM prefers has enough corresponding individuals, the initial population is produced by limited iterations of MOEA/D instead of random initialization. In this way, we can easily obtain the individual and its corresponding weights in the preference region, and shift quickly into new initial population when DM's searching regions are changed.

Updating Strategy: The population in the reference region is updated by using MOEA/D until the termination criteria is satisfied. It is worth noticing that the weights need to

Algorithm 2 RR-MOEA/D

- Step1. Initialization:** run pre-determined generations MOEA/D to generate initial population;
- Step2. Parameter input:** ask the DM to input his/her preference information;
- Step3. Determining reference region:** determine the reference region using the preference information according to Algorithm 1;
- Step4. Updating:** update the population in the preference region by MOEA/D-DE until the stopping criterion is meet;
- Step5. Interaction:** If the DM is not satisfied with reference information, go to Step2 to reset the reference information; If the DM wants to get better results in this reference region, go to Step4; otherwise, output the results.

be updated every generation. Updating weights needs two indicators: update direction and update step. The update direction is determined by the subproblem farthest from the reference direction to the subproblem nearest from the reference direction. The update step is defined as a unit length (Algorithm 3: line 3). It is evitable that the population may flies away from the preferred region by dynamically adjusting the weights. The details of updating weights are stated in Algorithm 3.

Interaction: At this stage, the current population is presented to the DM. If the DM is unsatisfied with the result, the preference information can be reset. A new population is produced according to the new desire of the DM and a mixed population combining the current population with the initial population. The DM can make tentative decisions before he/she finds his/her desired regions. When the reference region is determined, the DM can reset the precision of the solutions as the stopping criteria to get the final output results.

Algorithm 3 Updating Weights**Iteration Parameters:**

num_v: the number of reference directions

PF: initial population

N: the size of population close to each reference direction

start: the vector of *num_v* aspiration points

fend: the vector of *num_v* reservation points

radius: determine the size of reference region

Output:

W_t: weights after updating

1. $\text{ref_dir} \leftarrow \text{fend} - \text{start}$
2. **for** $i = 1, 2, \dots, \text{num_v}$ **do**
3. $\text{step}^i \leftarrow 2 \times \text{radius}^i / N^i$
4. $\text{best}^i \leftarrow \max(\cos(D_t, \text{ref_dir}(:, i)))$
5. $\text{worst}^i \leftarrow \min(\cos(D_t, \text{ref_dir}(:, i)))$
6. $\text{update_direction}^i \leftarrow \text{best}^i - \text{worst}^i$
7. $\text{W}_t \leftarrow \text{update_weight}(\text{update_direction}^i, \text{step}^i, \text{W}_t)$
8. **end for**

In the algorithm, the size of the neighborhood is decided by the size of population in the reference region and the range of reference region. If a small population or a quite small range which the DM is interested in, all the subproblems are treated as a neighborhood. In this case, the algorithm is simplified due to avoiding calculating the Euclidean distances between all of the weight vectors.

Given the preference direction(s), RR-MOEA/D can control its searching regions in which the DM is interested. By optimizing the subproblems in those regions, those solutions can be gotten which are converged to the preferred regions.

IV. EXPERIMENTS AND RESULTS

In order to validate the performance of the proposed algorithm, we apply it to solving two 2-objective ZDT problems and two 3-objective DTLZ problems respectively. All experiments were carried out on a personal computer with AMD A8-5550M APU with Radeon (tm) HD Graphics (2.10GHz) and 4 GB of RAM, Running Windows7.

A. BENCHMARK FUNCTIONS AND EXPERIMENTAL SETTING

In our experiments, we use ZDT2and ZDT4 [47] and three-objective benchmark problems, DTLZ1 and DTLZ2 [21], to test our algorithm, which are presented in Table 1.

The parameters in RR-MOEA/D are set as follows: the initial population size of MOEA/D-DE is set to 100 and 300 for the two objective and three objective problems respectively and MOEA/D-DE is executed 50 generations before inputting the preference information. The algorithm will stop after 250 generations. The size of population in the preference region *N* and the parameter of controlling interactive condition δ depend on the decision maker. Here, in order to illustrate experimental result, the size of *radius* is set as 0.005 and *N* is set to 30 and 60 for the two objective and three objective problems and the *penalty* is set as 1.5. The impact of the parameters provided by the DM on the performance of the algorithm is analyzed in Section 4.3.

B. EXPERIMENTAL RESULT AND ANALYSIS**1) RESULTS OF TWO-OBJECTIVE TEST PROBLEMS**

ZDT2 is a two-objective problem with 30variables and its PF is non-convex as shown in Fig. 2. Four cases based on different preference information are analyzed in order to verify the performance of RR-MOEA/D.

Case 1: The reference direction is defined by two points (0,0) and (1,1), we suppose the DM prefers the middle region of the PF in this case.

Case 2: The reference direction is between (0,0) and (1,0), which means one objective gets more preferences.

Case 3: The same as Case 2 and the aspiration point and reservation point is (0,0) and (0,1), respectively.

Case 4: To show the performance of algorithm when the DM sets more than one reference directions, we consider two

TABLE 1. Test functions used in the study.

Test Function	Definition
ZDT2	$f_1(x) = x_1$ $f_2(x) = g(x) \cdot \left[1 - \left(\frac{f_1(x)}{f_2(x)} \right)^2 \right]$ $g(x) = 1 + \frac{9}{(n-1)} \left(\sum_{i=2}^n x_i \right)$ $n = 30, \quad x = (x_1, \dots, x_n)^T \in [0,1]^n$ $f_1(x) = x_1$ $f_2(x) = g(x) \sqrt{\frac{f_1(x)}{f_2(x)}}$
ZDT4	$where \quad g(x) = 1 + 10(n-1) + \sum_{i=2}^n \left[x_i^2 - 10 \cos(4\pi x_i) \right]$ $n = 10, \quad x = (x_1, \dots, x_n)^T \in [0,1] \times [-5,5]^{n-1}$ $f_1(x) = 0.5 \times (1 + g(x)) x_1 x_2$ $f_2(x) = 0.5 \times (1 + g(x)) x_1 (1 - x_2)$ $f_3(x) = 0.5 \times (1 + g(x)) (1 - x_1)$
DTLZ1	$where \quad g(x) = 100(n-2) + 100 \sum_{i=3}^n \{(x_i - 0.5) - \cos[20\pi(x_i - 0.5)]\}$ $n = 10, \quad x = (x_1, \dots, x_n)^T \in [0,1]^n$ $f_1(x) = (1 + g(x)) \cos(\frac{x_1\pi}{2}) \cos(\frac{x_2\pi}{2})$ $f_2(x) = (1 + g(x)) \cos(\frac{x_1\pi}{2}) \sin(\frac{x_2\pi}{2})$ $f_3(x) = (1 + g(x)) \sin(\frac{x_1\pi}{2})$
DTLZ2	$where \quad g(x) = \sum_{i=3}^n x_i^2,$ $n = 10, \quad x = (x_1, \dots, x_n)^T \in [0,1]^2 \times [-1,1]^{n-2}$

reference directions: D_1 is determined by (0, 0) and (1, 2); and D_2 is determined by (0, 0) and (2, 1).

Fig. 2 shows that RR-MOEA/D has a good performance for all these four cases. For different kind of reference region required by the DM, the algorithm can obtain a set of reasonable and uniformly distributed solutions in the preferred region. Also, the algorithm can deal with more than one reference region simultaneously, and all preferred solutions can be obtained in each reference region.

Another two objective problem is ZDT4, which is hard due to its many local optimal solutions and its convex PF with 10 variables. Similar with ZDT2 problem, we also adopt the four cases above to test ZDT4 problem (see in Fig. 3).

It is obvious in Fig. 3 that the proposed algorithm can deal with the convex problem as well and get the ideal solutions in any region the DM prefers. Uniform distribution and convergence in the preference region on ZDT4, a hard problem of

ZDT test problems, show a good performance of the proposed algorithm.

2) RESULTS OF THREE-OBJECTIVE TEST PROBLEMS

DTLZ1 is a 10-variable, three-objective problem. Its efficient frontier is known and is given by $\sum_{i \in [1,m]} f_i = 0.5$, as shown in Fig. 4. Fig. 4(a) shows the obtained solutions with one reference direction when the reference direction is given by two points (0, 0, 0) and (0.5, 0.5, 0.5). To show the performance with more than one preference region, Fig. 4(b) shows the obtained solutions with two reference direction regions. Two reference directions are: $\{(0,0,0), (0.5,0.5,0.3)\}$, $\{(0,0,0), (0.1,0.1,0.5)\}$.

DTLZ2 has a convex PF with 10 variables and its efficient frontier satisfies $\sum_{i \in [1,m]} f_i^2 = 1$. Similar with DTLZ1 problem, we use one and two reference directions to show the performance. The first reference direction is from point (0,0,0)

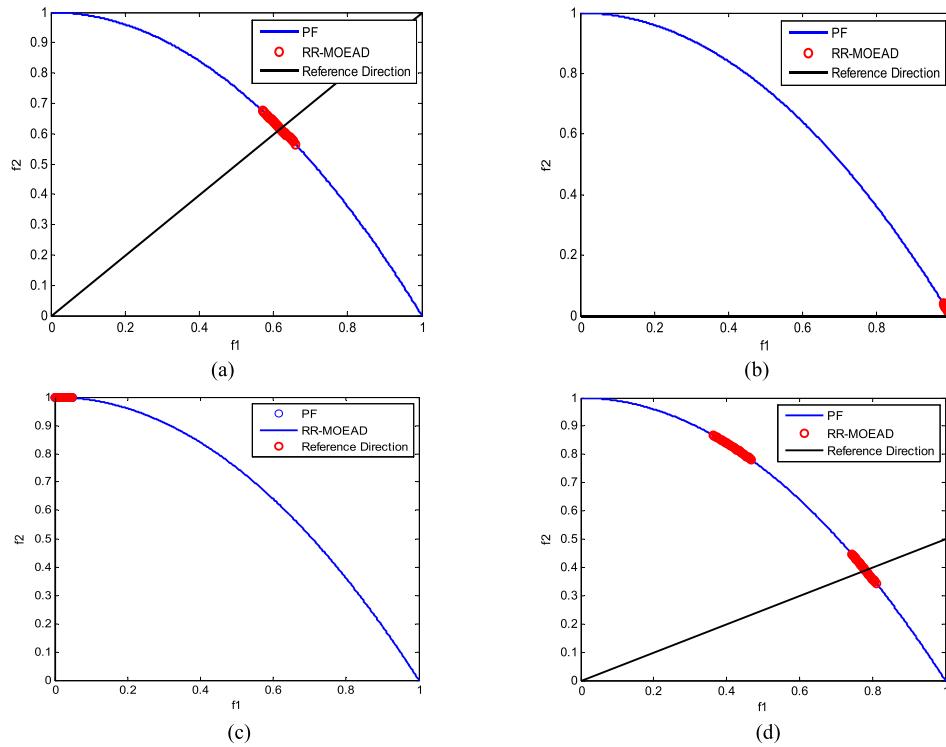


FIGURE 2. Preferred solutions obtained by RR-MOEA/D on ZDT2: (a) case 1; (b) case 2; (c) case 3; (d) case 4.

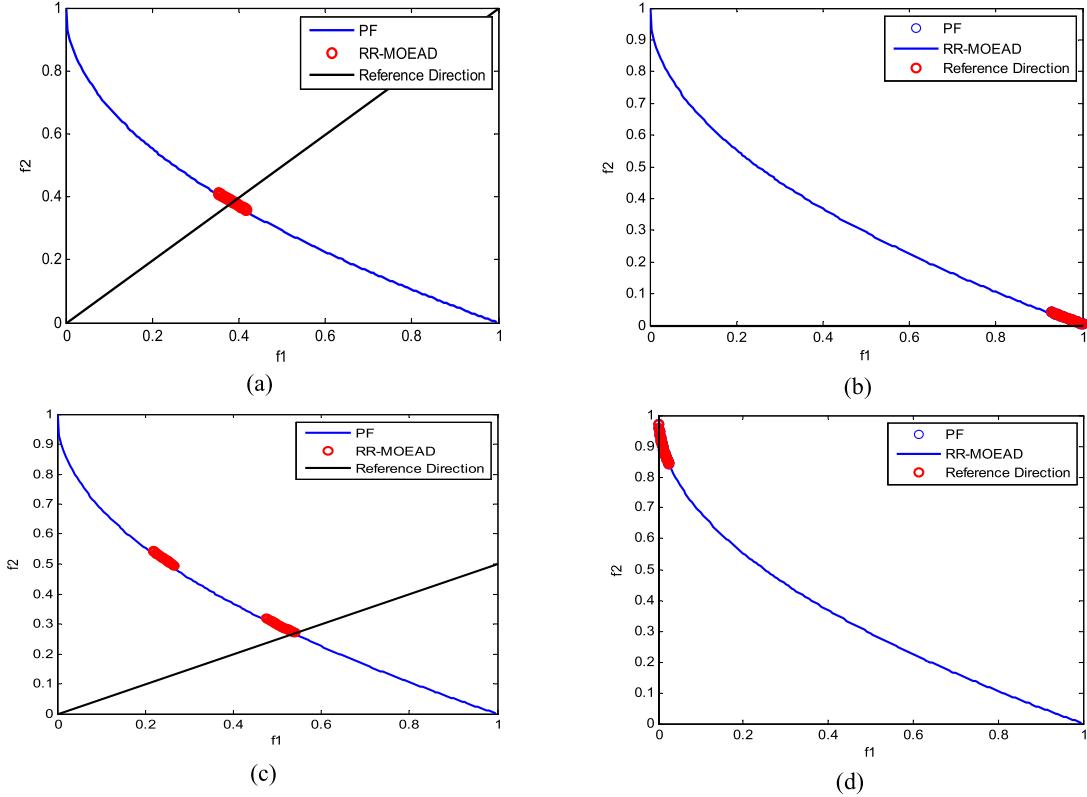


FIGURE 3. Preferred solutions obtained by RR-MOEA/D on ZDT4 : (a) case 1; (b) case 2; (c) case 3; (d) case 4.

to point (0.7, 0.8, 0.5). The two reference directions in Fig. 5(b) are $\{(0,0,0), (1,1,0.8)\}$, $\{(0,0,0), (0.4,0.4,0.1)\}$, respectively.

The final results of three-objective problem are shown in Fig. 4 and Fig. 5. Obviously our algorithm also can well converge to the true PF within the referred region the decision

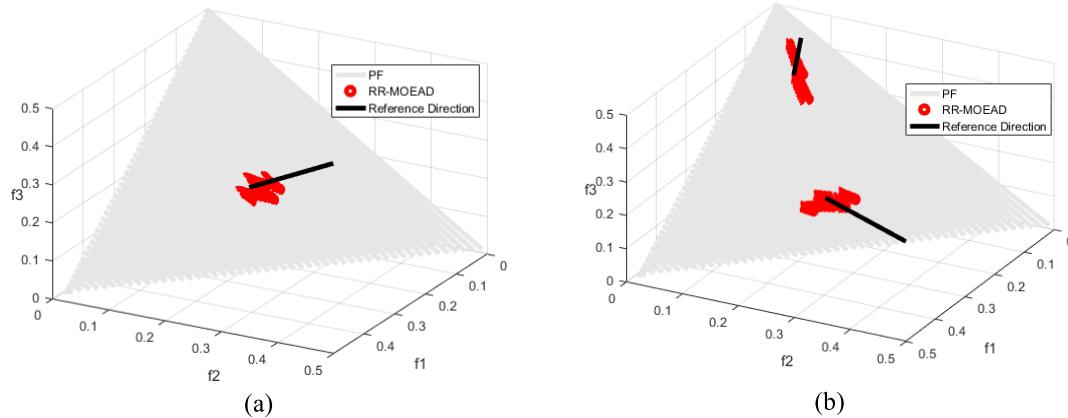


FIGURE 4. Preferred solutions obtained by RR-MOEA/D on DTLZ1 (a) one reference region; (b) two reference regions.

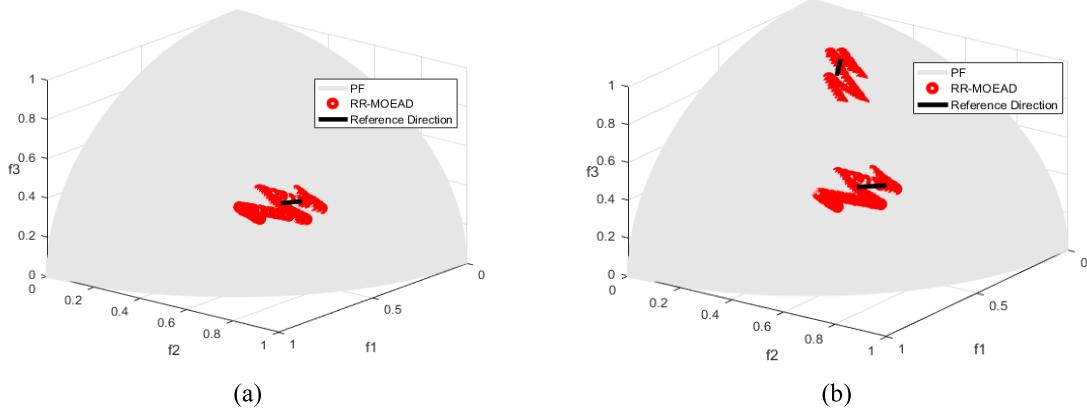


FIGURE 5. Preferred solutions obtained by RR-MOEA/D on DTLZ2 (a) one reference region; (b) two reference regions.

maker favors. It needs to point out here that the shape of obtained solutions is determined by the way of initializing the weights in MOEA/D.

C. DISCUSSION

1) THE DISCUSSION OF INTERACTION PERFORMANCE

As an interactive method, the interaction performance of the algorithm is vitally importance. A simple interaction is presented in the proposed algorithm. It is not an easy task

that an exact preference region is required at the beginning, so the algorithm allows the decision maker to pick up a reference direction as a trial. When the interaction condition is met, these current solutions which are rough solutions in the reference region are presented to the DM. If the DM is interested in the reference region, our algorithm will go on with finding the solutions approximate to PF in the reference region. Otherwise, searching direction can be changed by the DM. Here is an example to illustrate that we have plenty ways to change the searching directions. Supposed we set an initial searching direction from $(0, 0)$ to $(1, 1)$ and the size of preference region $radius = 0.005$ for ZDT2 problem, a set of rough solutions in the reference region are presented to the DM (see in Fig. 6) when the interaction condition is satisfied (Eq. 6), in which, $\delta = 0.001$. If the DM is unsatisfied with the reference region he/she set earlier, they change their preference information as follows:

Case 1: The DM changes the initial searching region to a near preference region(e.g. from $(0,0) \rightarrow (1,1)$ to $(0,0) \rightarrow (1,1.1)$).

Case 2: The reference region is changed far from the initial reference region by the DM(e.g. from $(0,0) \rightarrow (1,1)$ to $(0,0) \rightarrow (0.3,0.8)$).

Case 3: A new preference region is added (e.g. from $(0,0) \rightarrow (1,1)$ to $(0,0) \rightarrow (0.5,1)$ and $(0,0) \rightarrow (1,1)$).

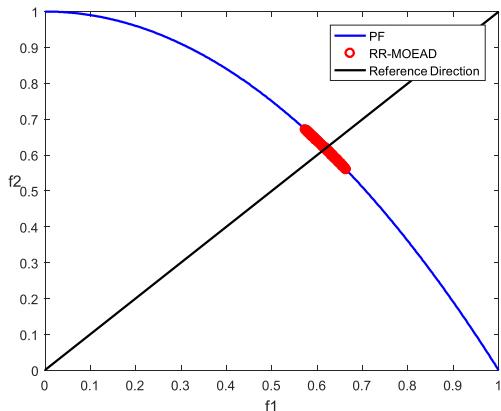
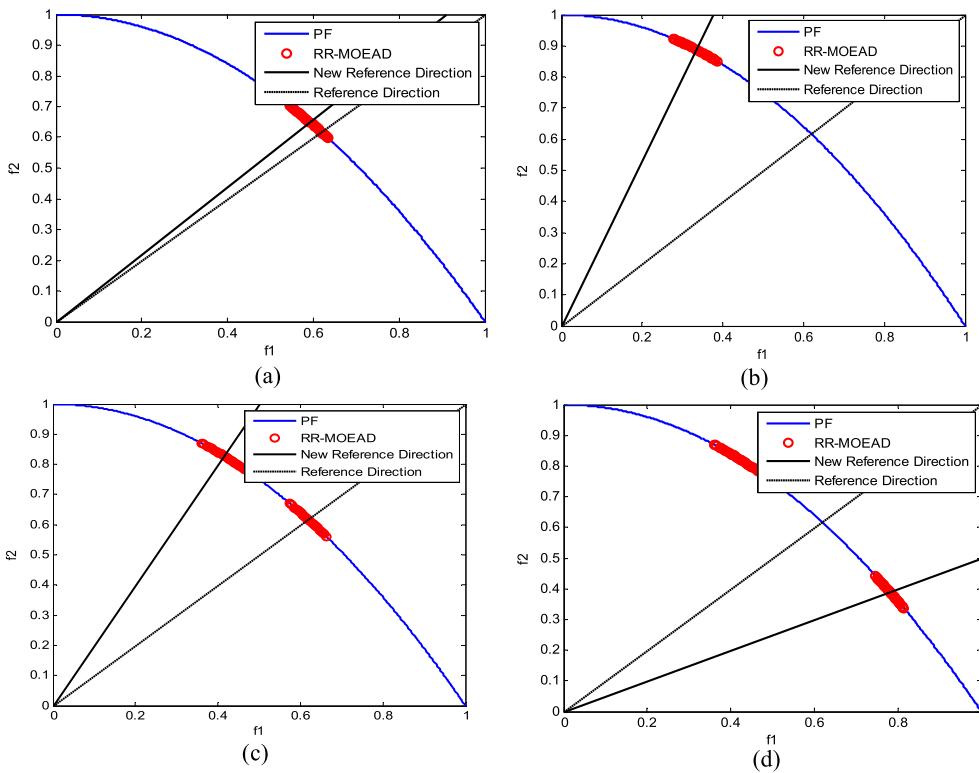


FIGURE 6. Initial solutions obtained in the interactive stage of ZDT2 ($(0, 0) \rightarrow (1, 1)$).

**FIGURE 7.** Solutions obtained of ZDT2 after different interactions: (a) case 1; (b) case 2; (c) case 3; (d) case 4.**TABLE 2.** Mean generations and its standard variance with different δ .

δ	Problems	Reference direction	Mean generation \pm standard deviation
0.001	ZDT1	(0,0)→(1,1)	14.50 \pm 10.72
	ZDT2	(0,0)→(1,1)	17.20 \pm 22.40
	ZDT3	(0,0)→(0.15,0.4)	13.40 \pm 9.60
	ZDT4	(0,0)→(1,1)	7.70 \pm 2.23
	ZDT6	(0,0)→(1,1)	10.20 \pm 3.73
0.01	ZDT1	(0,0)→(1,1)	15.40 \pm 5.93
	ZDT2	(0,0)→(1,1)	25.20 \pm 10.40
	ZDT3	(0,0)→(0.15,0.4)	13.60 \pm 8.08
	ZDT4	(0,0)→(1,1)	35.10 \pm 17.37
	ZDT6	(0,0)→(1,1)	40.50 \pm 20.82
0.1	ZDT1	(0,0)→(1,1)	40.10 \pm 7.59
	ZDT2	(0,0)→(1,1)	42.60 \pm 10.92
	ZDT3	(0,0)→(0.15,0.4)	47.80 \pm 13.43
	ZDT4	(0,0)→(1,1)	76.00 \pm 25.40
	ZDT6	(0,0)→(1,1)	86.90 \pm 29.88

Case 4: The DM changes an initial searching region to two different searching regions (e.g. from (0,0) → (1,1) to (0,0) → (0.5,1) and (0,0) → (1,0.5)).

The final results of these four cases are shown in Fig. 7. If the DM is not satisfied with the searching region he/she set before, our algorithm also can fast capture

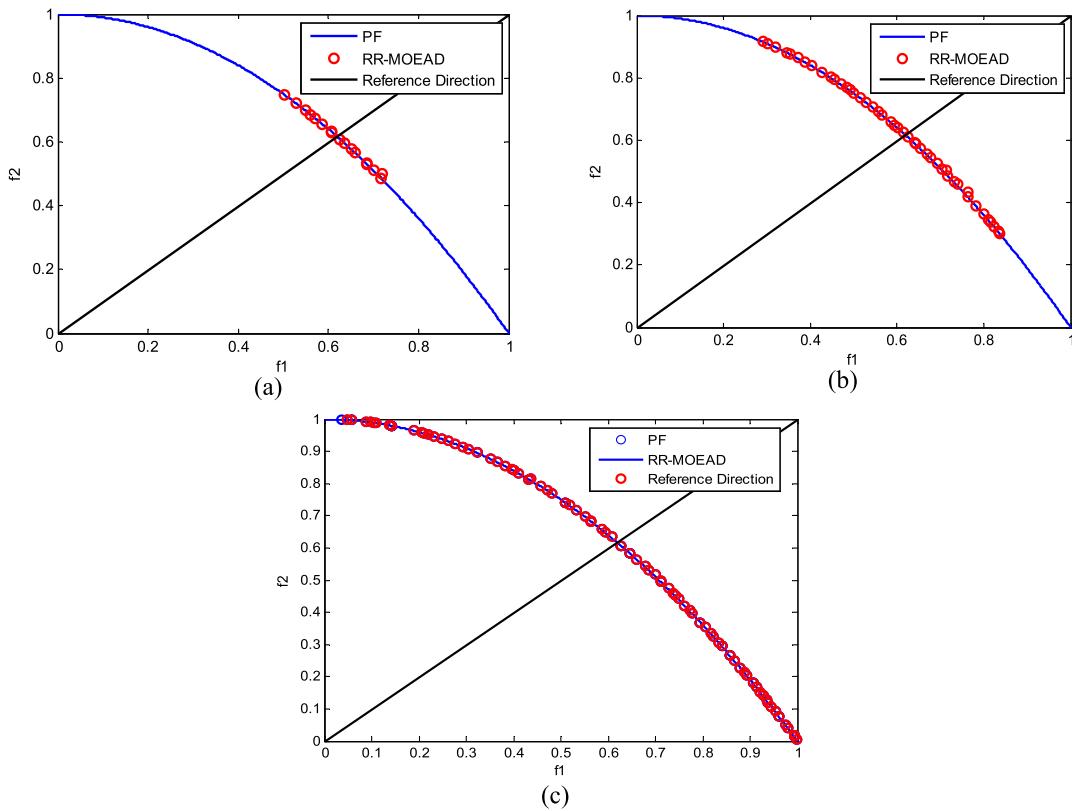


FIGURE 8. Solutions obtained on ZDT2 with the $\text{radius}=0.5$: (a) $N = 20$; (b) $N = 50$; (c) $N = 100$.

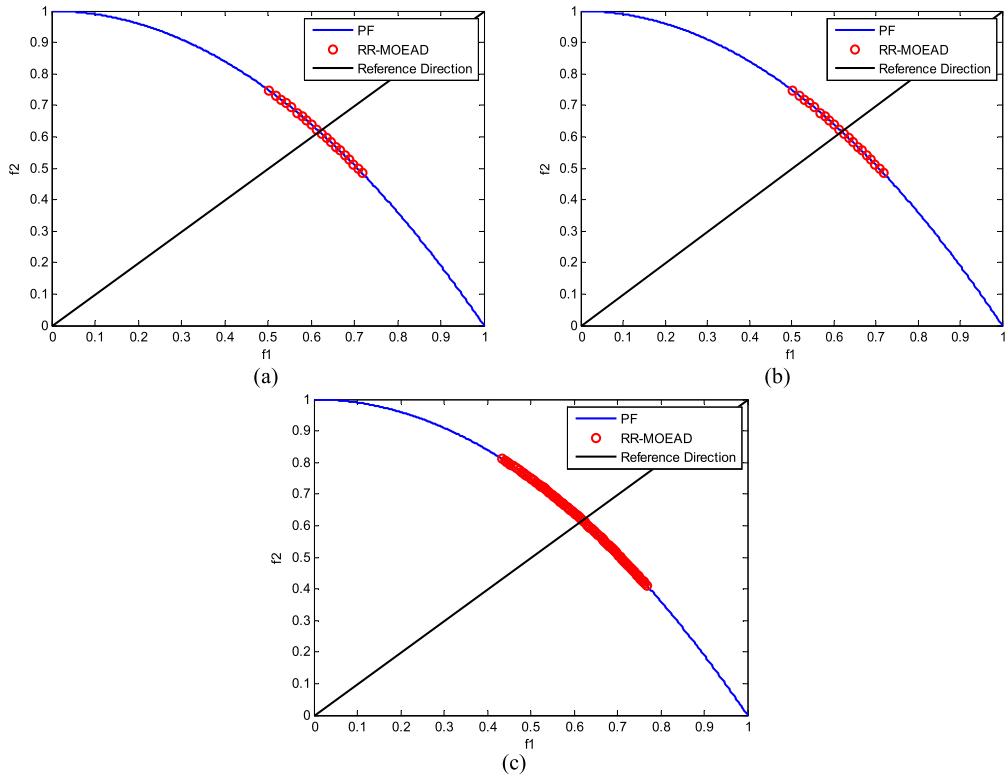


FIGURE 9. Solutions obtained on ZDT2 with the $\text{radius}=0.05$: (a) $N = 20$; (b) $N = 50$; (c) $N = 100$.

the new preference region and well converge to the PF within the new preferred region to meet various user requirements.

A fast way to obtain a number of rough solutions in the reference region to interact with the DM is another distinguishing characteristic. As mentioned earlier, δ is provided

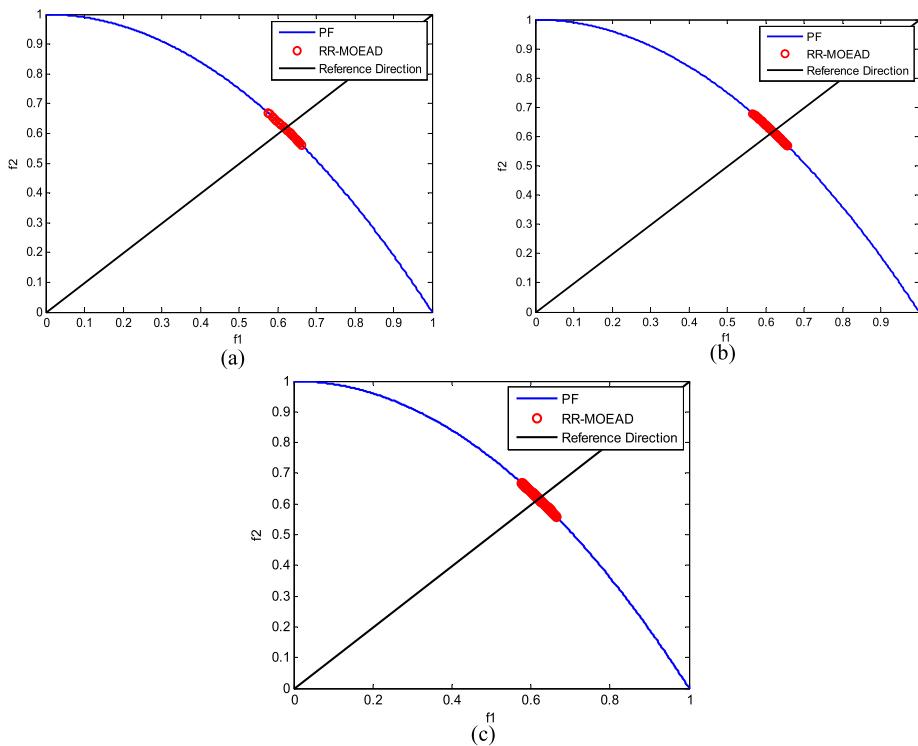


FIGURE 10. Solutions obtained on ZDT2 with the $radius=0.005$: (a) $N = 20$; (b) $N = 50$; (c) $N = 100$.

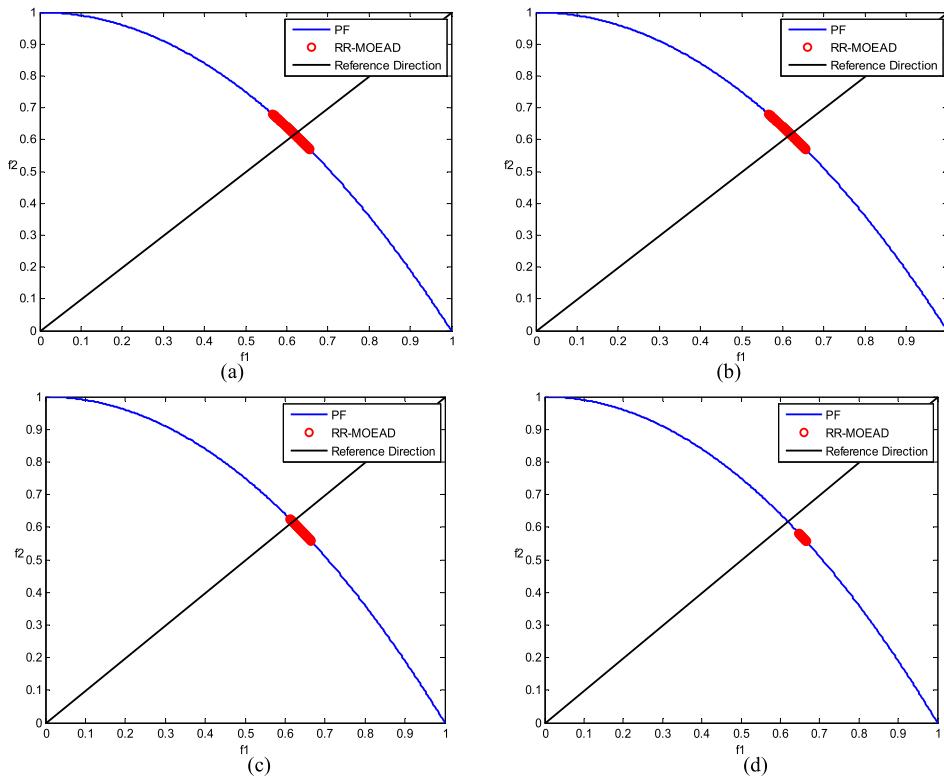


FIGURE 11. Solutions obtained on ZDT2 problem with the $radius=0.005$: (a) $penalty=1.0$; (b) $penalty=1.5$; (c) $Penalty=2.0$; (d) $penalty=3.0$.

by the decision maker to control the accuracy of the solution. According to the interactive condition used in our study, rapidly getting most solutions in DM's preferred region is presented to the DM. In Table 2, the mean generations needed are

shown when the interaction happens for ZDT test problems based on 10 independent runs (δ is set as 0.1, 0.01, 0.001). It is easy to see that it just needs no more than 20 generations for the proposed algorithm to find a rough set of solution when

$\delta = 0.001$, even when $\delta = 0.1$, the proposed algorithm can find some rough solutions in no more than 100 generations. It is obvious that RR-MOEAD is a fast algorithm to meet the interactive condition and this means the algorithm can quickly get the rough solutions in the region preferred by the DM.

2) THE DISCUSSION OF PARAMETERS (RADIUS, N AND PENALTY)

At interactive stage, RR-MOEAD allows the DM to input different preference information to satisfy different preference requirements. In order to analyze the impact of the input parameters like the population size N , the size of preferred region i.e. the *radius* and the *penalty* on the performance of the algorithm, we set the population size as 20, 50 and 100 respectively when the *radius* size is fixed as 0.5, 0.05, and 0.005 for 2-objective ZDT2 problem and perform a series of experiments. The aspiration point and the reservation point are set to (0, 0) and (1, 1).

Further, in order to determine the effect of the parameter *penalty* on the experiment's results. We set the *penalty* as 1.0, 1.5, 2.0 and 3.0 respectively when the value of *radius* is fixed as 0.005 and the population is 30 for ZDT2, the aspiration point and the reservation point are set to (0,0) and (1,1).

From Fig. 8 Fig. 10, we can see that the larger the *radius* is, it is better to set the *radius* as a large value. When DMs provide a small value for the parameter *radius*, it is not necessary to set a large value for the parameter N . in this paper, we set *radius* = 0.005 and N is set to 30 and 60 for the two objective and three objective problems.

Fig. 11 shows the solutions obtained with different value of the *penalty*. We can conclude that if the value of the *penalty* is set around 1.5, the entire reference region would be well converged to near the reference direction. Meanwhile, the *penalty* being fixed as a greater value, the converged front will deviate the reference direction. So, in the experimental study, we set it as 1.5. From here, we can draw a conclusion that the more detailed information the DM provide, the more accurate solutions our algorithm can find. For example, if the smaller value of radius and penalty are given, the front obtained would be well converged to the reference direction. No matter what reference region the DM needs, our algorithm can get a well-spaced front well converged to the region most preferred.

V. CONCLUSIONS

In this paper, an interactive reference region based evolutionary algorithm through decomposition named RR-MOEAD is proposed. At the beginning of the proposed algorithm, MOEA/D is used to generate initial solutions before the preference desired by the decision maker are given. We can initialize the suitable subproblems in the reference regions according to the DM's requirements. By dealing with the subproblems in the reference region, the solutions obtained can converge to the regions of the Pareto front which the DM prefers.

During the stage of interaction, a simple interactive condition is adopted in this paper. The proposed algorithm can get a set of rough solutions in one or several reference regions quickly and avoid wasting time in searching the region which the DM is not interested in. Solutions obtained are presented to DM when the interactive condition is met. If the DM is unsatisfied with these solutions, the new preference information can be reset. Therefore, the decision maker can control the preference information in a very intuitive way. In addition, RR-MOEAD allows the DM to reset different kinds of preference information and it can quickly obtain a set rough solution according to new preference information.

It should be emphasized that all of subproblems in the preference region are determined by weight vectors in the evolving stage. It is inevitable that the population converges to the region on PF where the user is undesired by dynamically adjusting the corresponding weights. If the number of sub problems (population size) is so small to cover entire preference region, the subproblems in the region most close to the reference direction will become a priority. In other words, RR-MOEAD allows the DM to give uncertain and inconsistent human decision as he or she does in real life.

Both MOPs with two and three objectives are adopted to test the performance of RR-MOEAD. According to the results of experimental study, the proposed algorithm can successfully converge to different regions the DM most prefers. It is obvious that our algorithm can realize a fast and simple interactive process and copes with different preference information including uncertain and inconsistent human decision. In our future work, we will try to find a better way to construct the preference regions. As we known, there are still many aspects of interaction strategy to worth further exploration. Dealing with interaction information better and more humanized is our another work.

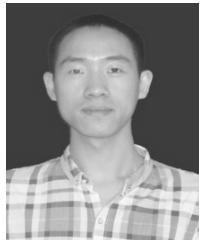
REFERENCES

- [1] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [2] K. Sindhya, A. B. Ruiz, and K. Miettinen, "A preference based interactive evolutionary algorithm for multi-objective optimization: PIE," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), vol. 6576. Berlin, Germany: Springer-Verlag, 2011, pp. 212–225.
- [3] K. Miettinen, *Nonlinear Multiobjective Optimization*. Norwell, MA, USA: Kluwer, 1999.
- [4] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.
- [5] A. Messac, A. Ismail-Yahaya, and C. A. Mattson, "The normalized normal constraint method for generating the Pareto frontier," *Struct. Multidisciplinary Optim.*, vol. 25, no. 2, pp. 86–98, Jul. 2003.
- [6] R. F. Dell and M. H. Karwan, "An interactive MCDM weight space reduction method utilizing a Tchebycheff utility function," *Naval Res. Logistics*, vol. 37, no. 2, pp. 263–277, Apr. 1990.
- [7] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.
- [8] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
- [9] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 3, pp. 115–148, 1995.

- [10] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2013.
- [11] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.
- [12] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evolutionary algorithm using Gaussian process-based inverse modeling," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 838–856, Dec. 2015.
- [13] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, 2016.
- [14] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
- [15] K. Deb, S. Agarwal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [16] C. A. C. Coello, "Handling preferences in evolutionary multiobjective optimization: A survey," in *Proc. Congr. Evol. Comput.*, Jul. 2000, pp. 30–37.
- [17] L. Rachmawati and D. Srinivasan, "Preference incorporation in multi-objective evolutionary algorithms: A survey," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2006, pp. 3385–3391.
- [18] A. P. Wierzbicki, "The use of reference objectives in multiobjective optimization," in *Multiple Criteria Decision Making Theory and Application*. Heidelberg, Germany: Springer, 1980, pp. 468–486.
- [19] A. Jaszkiewicz and R. Słowiński, "The 'light beam search' approach—An overview of methodology applications," *Eur. Jour. Oper. Res.*, vol. 113, no. 2, pp. 300–314, Mar. 1999.
- [20] K. Deb and J. Sundar, "Reference point based multi-objective optimization using evolutionary algorithms," *Int. J. Comput. Intell. Res.*, vol. 2, no. 3, pp. 273–286, 2006.
- [21] K. Deb and A. Kumar, "Interactive evolutionary multi-objective optimization and decision-making using reference direction method," in *Proc. 9th Annu. Conf. Genet. Evol. Comput. (GECCO)*, 2007, pp. 781–788.
- [22] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evol. Comput.*, vol. 17, no. 3, pp. 411–436, 2009.
- [23] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. C. Coello, and R. Caballero, "g-dominance: Reference point based dominance for multi-objective metaheuristics," *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 685–692, Sep. 2009.
- [24] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Evolutionary Methods for Design, Optimization and Control With Applications to Industrial Problems*. K. C. G. Koglow, D. T. Tsahalis, J. Périaux, and T. Fogarty, Eds. Heidelberg, Germany: Springer, 2001, pp. 95–100.
- [25] M. G. Gong, L. Jiao, H. Du, and L. Bo, "Multiobjective immune algorithm with nondominated neighbor-based selection," *Evol. Comput.*, vol. 16, no. 2, pp. 225–255, 2008.
- [26] J. Branke, K. Deb, and K. Miettinen, *Multiobjective Optimization: Interactive and Evolutionary Approaches*, vol. LNCS 5252. Berlin, Germany: Springer-Verlag, 2008.
- [27] T. Tanino, M. Tanaka, and C. Hojo, "An interactive multicriteria decision making method by using a genetic algorithm," *Manage. Sci.*, vol. 22 no. 6, pp. 652–663, 1976.
- [28] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 28, no. 1, pp. 26–37, Jan. 1998.
- [29] M. Shibuya, H. Kita, and S. Kobayashi, "Integration of multi-objective and interactive genetic algorithms and its application to animation design," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. IEEE SMC*, Oct. 1999, pp. 646–651.
- [30] D. S. Todd and P. Sen, "Directed multiple objective search of design spaces using Genetic Algorithms and neural networks," in *Proc. Genetic Evol. Comput. Conf.*, 1999, pp. 1738–1743.
- [31] I. C. Parmee, D. Cvetković, A. Watson, and C. Bonham, "Multiobjective satisfaction within an interactive evolutionary design environment," *Evol. Comput.*, vol. 8, no. 2, pp. 197–222, Jun. 2000.
- [32] H. J. C. Barbosa and A. M. S. Barreto, "An interactive genetic algorithm with co-evolution of weights for multiobjective problems," in *Proc. Genetic Evol. Comput. Conf.*, 2001, pp. 203–210.
- [33] S. Phelps and M. Köksalan, "An interactive evolutionary metaheuristic for multiobjective combinatorial optimization," *Manage. Sci.*, vol. 49, no. 12, pp. 1726–1738, 2003.
- [34] R. Kamalian, H. Takagi, and A. M. Agogino, "Optimized design of MEMS by evolutionary multi-objective optimization with interactive evolutionary computation," in *Proc. GECCO*, Berlin, Germany, vol. 3103. 2004, pp. 1030–1041.
- [35] R. Kamalian, Y. Zhang, H. Takagi, and A. M. Agogino, "Evolutionary synthesis of micromachines using supervisory multiobjective interactive evolutionary computation," in *Proc. ICMLC, LNAI3930*, 2006, pp. 428–437.
- [36] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based interactive evolutionary algorithm for multiobjective optimization," Dept. Inf. Technol. Electr. Eng., Aalto Univ. School Bus., Helsinki, Finland, Tech. Rep., 2007.
- [37] G. Avigad and A. Moshaiov, "Interactive evolutionary multiobjective search and optimization of set-based concepts," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 4, pp. 1013–1027, Aug. 2009.
- [38] L. B. Said, S. Bechikh, and K. Ghedira, "The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 801–818, Oct. 2010.
- [39] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius, "An interactive evolutionary multiobjective optimization method based on progressively approximated value functions," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 723–739, Oct. 2010.
- [40] C. L. Simons and I. C. Parmee, "Elegant object-oriented software design via interactive, evolutionary computation," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 1797–1805, Nov. 2012.
- [41] X. Sun, D. Gong, Y. Jin, and S. Chen, "A new surrogate-assisted interactive genetic algorithm with weighted semisupervised learning," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 685–698, Apr. 2013.
- [42] *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), vol. 9019. Switzerland, Springer International Publishing, 2015, pp. 249–263.
- [43] A. B. Ruiz, R. Saborido, and M. Luque, "A preference-based evolutionary algorithm for multiobjective optimization: The weighting achievement scalarizing function genetic algorithm," *J. Global Optim.*, vol. 62, no. 1, pp. 101–129, May 2015.
- [44] M. G. Gong, F. Liu, W. Zhang, L. Jiao, and Q. Zhang, "Interactive MOEA/D for multi-objective decision making," in *Proc. 13th Annu. Conf. Genetic Evol. Comput.*, 2011, pp. 721–728.
- [45] R. Battiti and A. Passerini, "Brain-computer evolutionary multiobjective optimization: A genetic algorithm adapting to the decision maker," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 671–687, Oct. 2010.
- [46] C. Fonseca and P. J. Fleming, "Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization," In *Proc. 5th Int. Conf. Genet. Algorithms*, 1993, pp. 416–423.
- [47] E. Zitzler, L. Thiele, and K. Deb, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evol. Comput.*, vol. 8, no. 2, pp. 173–195, Jun. 2000.



RUOCHEN LIU (M'07) received the Ph.D. degree from Xidian University, Xian, China, in 2005. She is currently an Associate Professor with the Intelligent Information Processing Innovative Research Team of the Ministry of Education of China, Xidian University. Her research interests are broadly in the area of computational intelligence. Her areas of special interest include artificial immune systems, evolutionary computation, data mining, and optimization.



RUINAN WANG received the B.S. degree from the Hebei University of Science and Technology, Hebei, China. He is currently pursuing the M.S. degree from Xidian University. His current research focuses on multi-objective optimization and data mining.



JUNJUN HUANG received the B.S. degree from the Huaiyin Institute of Technology, Huai'an, China. He is currently pursuing the M.S. degree with Xidian University. His current research focuses on multi-objective optimization and image processing.



WEN FENG received the B.S. degree from Xidian University, Xi'an, China, where she is currently pursuing the M.S. degree. Her current research focuses on multi-objective optimization.



LICHENG JIAO (SM'89) received the Ph.D. degree from Xian Jiaotong University, Xian, China, in 1990. He is currently a Professor and the Dean of the Electronic Engineering School, Xidian University, China. His current research focuses on intelligent information processing.

• • •