Applied Soft Computing 70 (2018) 797–813

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Adaptive region adjustment to improve the balance of convergence and diversity in MOEA/D

Peng Wang [a], Bo Liao [a,*], Wen Zhu [a], Lijun Cai [a], Siqi Ren [a], Min Chen [b], Zejun Li [b], Keqin Li [a,c]

[a] College of Information Science and Engineering, Hunan University, Changsha 410082, Hunan, China
[b] College of Computer and Information Science, Hunan Institute of Technology, Hengyang, China
[c] Department of Computer Science, State University of New York, New Paltz, NY 12561, USA

ARTICLE INFO

Article history:
Received 9 October 2017
Received in revised form 15 April 2018
Accepted 13 June 2018
Available online 27 June 2018

Keywords:
Adaptive region adjustment
Convergence
Diversity
Evolutionary computation
Multiobjective optimization

ABSTRACT

The multiobjective evolutionary algorithm based on decomposition (MOEA/D), which decomposes a multiobjective optimization problem (MOP) into a number of optimization subproblems and optimizes them in a collaborative manner, becomes more and more popular in the field of evolutionary multiobjective optimization. The mechanism of balance convergence and diversity is very important in MOEA/D. In the process of optimization, the chosen solutions must be distinctive and as close as possible to the Pareto front. In this paper, we first explore the relation between subproblems and solutions. Then we propose the adaptive region adjustment strategy to balance the convergence and diversity based on the objective region partition concept. Finally, this strategy is embedded in the MOEA/D framework and then a simple but efficient algorithm is proposed. To demonstrate the effectiveness of the proposed algorithm, comprehensive experiments have been designed. The simulation results show the effectiveness of our proposed algorithms.

1. Introduction

Many real-world problems can be converted to multiobjective optimization problems (MOPs), which have several objectives that are conflict and need to be optimized simultaneously [1,2]. The formulation of a MOP could be expressed as follows:

$$\begin{aligned} &\text{minimize} \quad F(x) = (f_1(x), f_2(x), \ldots, f_m(x)) \\ &\text{subject to} \quad x \in \Omega \end{aligned} \tag{1}$$

where $x = (x_1, \ldots, x_n) \in R^n$ is a vector of decision variables; n is the total number of decision variables; Ω is the feasible decision space. $F(x)$ can be expressed as $F: \Omega \longrightarrow R^m$, where m is the number of objective functions, and R^m is the objective space.

Since the objectives defined in (1) are mutually conflicting, it might be no single point in Ω which is able to minimize all the objectives simultaneously. In practice, the best tradeoffs among the objectives can be defined by Pareto optimality. Let $u, v \in R^m$, u is said to dominate v, denoted as $u \prec v$, if and only if $u_i \leq v_i$, for every $i \in 1, 2, \ldots, m$, and there exists a $j \in 1, 2, \ldots, m$ which makes $u_j < v_j$. A solution $x^* \in \Omega$ is said to be Pareto optimal solution to (1) if there is no other solution $x \in \Omega$ such that $F(x)$ dominate $F(x^*)$. And $F(x^*)$ is called a Pareto optimal objective vector. The set of all Pareto optimal solutions is called the Pareto set (PS) and the set of all Pareto optimal objective vectors is called the Pareto front (PF), with $PS = \{x^* | F(x^*) < F(x), \forall x \in \Omega\}$ and $PF = \{F(x) | x \in PS\}$. PF is what we are looking for in solving multiobjective optimization problems [3].

Over the past decades, multiobjective evolutionary algorithms (MOEAs) have gained wide popularity for solving MOPs in the evolutionary computation (EC) community [4–8]. Compared with the traditional optimization methods, the MOEAs have the advantages that require very few assumptions on the problems and is able to obtain an approximation of the PF through a single run. In fact, the goal of most MOEAs is to find a finite set of solutions to approximate PF [1,9]. In the evolutionary process, the selection mechanism plays a key role in balancing convergence and diversity. Depending on different selection mechanisms, most existing MOEAs can be classified into three categories: (1) Pareto domination-based approach which uses Pareto dominance relation among solutions as the pri-

* Corresponding author.
E-mail addresses: wangpenglw@126.com (P. Wang), dragonbw@163.com (B. Liao), syzhuwen@163.com (W. Zhu), lijcai@hnu.edu.cn (L. Cai), siqirenzl@163.com (S. Ren), chenmin@hnit.edu.cn (M. Chen), lzjfox@163.com (Z. Li), lik@newpaltz.edu (K. Li).

https://doi.org/10.1016/j.asoc.2018.06.023
1568-4946/© 2018 Published by Elsevier B.V.

mary selection operation [6,10,11]. (2) Indicator-based approach which uses performance indicator which can measure convergence and diversity simultaneously to guide their selection [12–15]. (3) decomposition-based approach which decomposes a MOP into a number of scalar objective subproblems or several simple multiobjective subproblems and simultaneously optimizes them in a collaborative manner [9,16–22].

In this paper, we focus on the multiobjective evolutionary algorithm based on decomposition (MOEA/D) [9], which optimizes a set of single objective subproblems in a collaborative manner through the neighborhoods that are based on the relationship among such subproblems. In the MOEA/D evolution process, the subproblems generate the next generation solution by mating with the information about their neighborhood and then perform the select and replacement in their neighborhood. Given that any new solution of a subproblem can replace the current solution of its neighborhood, several subproblems may have the same solution, especially if the new solution is a better solution [23,24]. In this case, the diversity of solutions is reduced and is easily trapped in local optima.

In recent years there have been a number of works focusing on selection and replacement in order to maintain population diversity [1]. In the MOEA/D-DE [25], the authors believed that the replacement neighborhood is smaller than the selection (mating) neighborhood is conducive to maintaining the diversity of the population. In the MOEA/D-AGR [26], the authors held that it is very advantageous to have a small replacement neighborhood in the early stage of the evolutionary process and a large replacement neighborhood in the latter part of the evolutionary process. Li et al. established a stable matching model to coordinate the relationship between solutions and subproblems in MOEA/D-STM [27], and presented MOEA/D-IR [28] which defined mutual preferences between solutions and extended relating the preference of between subproblems and solutions. In MOEA/D-ACD [29], Wang et al. proposed an adaptive constraint decomposition method based on MOEA/D. In [30], Gee et al. proposed an online diversity metric, which estimated the diversity loss of a solution to the whole population by a notion of maximum relative diversity loss (MRDL). In [31], Wu et al. considered that the original MOEA/D-STM had a high risk of matching solutions and unfavorable sub-problems, and eventually led to unbalanced selection results. Therefore, they introduced the concept of incomplete preference lists into the MOEA/D-STM to compensate for the loss of population diversity and reduce imbalances.

To improve the balance between convergence and diversity in MOEA/D, we have considered this issue from a new perspective. From the above literature review, it can be found that the previous research work is often more focused on considering convergence, and there is not a mandatory diversity, which often leads to a lack of diversity. So first emphasize a diversity, and then consider the convergence will be beneficial. In this paper, we applies the adaptive region adjustment strategy (ARA), which divides the region based on the solution of the neighborhood of the subproblem, to strike a balance between diversity and convergence. Then, we embed this strategy into the MOEA/D and apply this method on a number of typical test problems for a comprehensive systematic experimental study.

The rest of this paper is organized as follows. Section 2 presents some background knowledge about MOEA/D and related work. In Section 3, the adaptive region adjustment strategy is first described. Then the ARA strategy is integrated into the MOEA/D framework, which is called the MOEA/D-ARA algorithm. The general experimental settings and test problems are described in Section 4. And the empirical results are presented and analyzed in Section 5. Finally, this paper is concluded in Section 6.

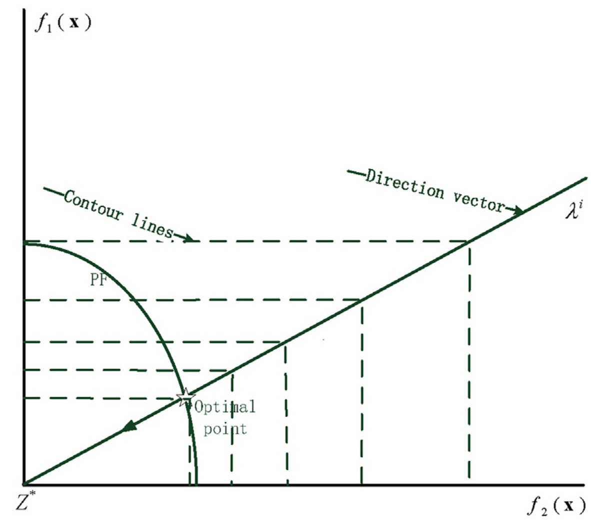

Fig. 1. Illustration of the Tchebycheff decomposition approach.

2. Background

2.1. Important components of MOEA/D

In this section, we mainly introduce two important components in MOEA/D framework. First, we introduce decomposition method, and then introduce the related concepts of neighborhood.

2.1.1. Decomposition approaches

In MOEA/D framework, a multi-objective problem is decomposed into a number of scalar optimization subproblems by using decomposition approaches and then optimizes them simultaneously. There are three popular decomposition approaches in the EMO community.

1. Weighted Sum Approach: In this approach, a convex combination of the different objectives is considered. The *i*th subproblem is defined as the following scalar optimization problem:

$$\begin{aligned} &\text{minimize} \quad g^{ws}(x|\lambda_i) = \sum_{j=1}^{m} \lambda_i^j f_j(x) \\ &\text{subject to} \quad x \in \Omega \end{aligned} \tag{2}$$

   This approach could work well for convex PFs (concave PFs in the case of maximization). However, it cannot obtained the entire PF in the case of nonconvex PFs [9].

2. Tchebycheff Approach: In this approach, the *i*th subproblem is defined as the following scalar optimization problem:

$$\begin{aligned} &\text{minimize} \quad g^{T}(x|\lambda_i, z^*) = \max\nolimits_{1\le j\le m}\{\lambda_i^j |f_j(x) - z_j^*|\} \\ &\text{subject to} \quad x \in \Omega \end{aligned} \tag{3}$$

   where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ is the weight vector of the scalar optimization subproblem, $\lambda_i \ge 0$ for all $i = 1, 2, \ldots, m$ and $\sum_{i=1}^{m} \lambda_i = 1$. In practice, when the weight value is equal to zero (λ_i), we usually set a very small value, such as 10^{-6}, to replace it. $z^* = (z_1^*, z_2^*, \ldots, z_m^*)$ is the reference point, $z_i^* = \min\{f_i(x)|x \in \Omega\} - \epsilon$ for each $i = 1, 2, \ldots, m$. $\epsilon > 0$ is a very small value, such as 10^{-6}. Under some mild conditions, the optimal solution of (3) is a Pareto optimal solution of (1) [3,9]. As shown in Fig. 1, it illustration of the Tchebycheff decomposition approach.

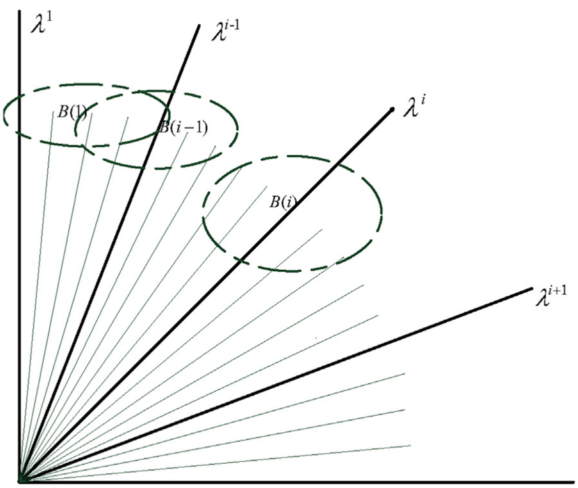

Fig. 2. Illustration of the neighborhood structure of a subproblem of size 5.

3. Penalty-based Boundary Intersection Approach: In this approach, the ith subproblem is defined as the following scalar optimization problem:

$$\begin{aligned} &\text{minimize} \quad g^{pbi}(x|\lambda_i, z^*) = d_1 + \theta d_2 \\ &\text{subject to} \quad x \in \Omega \end{aligned} \tag{4}$$

where $d_1 = \frac{\|(F(x)-z^*)^T \lambda_i\|}{\|\lambda_i\|}$, $d_2 = \|F(x) - (z^* - d_1 \frac{\lambda_i}{\|\lambda_i\|})\|$, and z^* is the reference point which is same as in the TCH approach, θ is a penalty parameter. One weakness with the PBI approach is that the parameter θ need to be properly tuned [9,1].

Among these methods, the Tchebycheff decomposition method is the most widely used as it has ability to solve multi-objective problems with non-convex Pareto optimal fronts [3,9]. In this paper, we also use the Tchebycheff decomposition method.

MOEA/D framework have attracted a lot of interest from researchers since it had been proposed, and many of MOEA/D-based variants of the algorithm were proposed, which includes a variant of the weight vector to design algorithms. In the [32], the authors argued that the simplex-lattice design cannot guarantee a set of uniform solutions, and proposed a new weight vector initialization method, called WS-transformation. This paper adopts this method as the decomposition method in MOEA/D. The WS-transformation maps the weight vector which is generated by the simplex-lattice design to its solution mapping vector. The form is as follows:

$$\lambda = WS(\lambda) = \left(\frac{\frac{1}{\lambda_1}}{\sum_{i=1}^{m} \frac{1}{\lambda_i}}, \frac{\frac{1}{\lambda_2}}{\sum_{i=1}^{m} \frac{1}{\lambda_i}}, \ldots, \frac{\frac{1}{\lambda_m}}{\sum_{i=1}^{m} \frac{1}{\lambda_i}} \right) \tag{5}$$

2.1.2. Neighborhood concepts

In MOEA/D framework, the concept of neighborhood is an important component. The decomposition method described above decomposes the multiobjective problem into a set of simple subproblems. The difference between these subproblems is that the values of λ are different, and between these λ have close relationship which is measured by their geometric distance to each other. In MOEA/D [9], the authors argued that if there is a close relationship among λ, then the corresponding subproblems of these λ have similar optimal solutions. Based on it, Zhang and Li proposed the neighborhood concepts, generally denoted as T. Given a subproblem, its neighborhood T represents the set of T most closely related subproblems of this subproblem. As shown in Fig. 2, it illustrates the neighborhood structure of a subproblem of size 5.

In the [33] and [34], the authors further discussed the concept of neighborhood, and divided into two neighborhoods, namely the mating neighborhood (T_m) and replacement neighborhood (T_r). The mating neighborhood is used to select parent solutions for reproduction operations to generate new solutions, and the replacement neighborhoods are used to determine which subproblem solution can be replaced by a new solution. In the [35] and [26], the mating and replacement neighborhoods were further discussed respectively.

2.2. Related work and motivations

2.2.1. Selection process

In multiobjective optimization, convergence and diversity are two critical issues in search process, and a multiobjective evolutionary algorithm requires not only good convergence but also the ability to maintain diversity. The convergence means that the MOP function value of the solution for each subproblem should be as close as possible to the Pareto front, and the diversity means that the MOP function values of the solution of each subproblem are as uniform as possible on the Pareto front.

In the above section, we introduced the importance of keeping the balance of convergence and diversity in MOEA. Selection process, can be regarded as the process of choice solution, plays a key role in balancing the convergence and diversity. How to choose the solution directly affects the balance between convergence and diversity. Before performing the selection process, there is a reproduction operation that generates a solution set, which is used to select. As we all know, there is no single EA that outperforms all other EAs on different issues. Therefore, in addition to the most classical SBX and polynomial mutation operations in NSGAII [10] and MOEA/D [9], there are some studies that aim to modify reproduction operators to improve the performance of MOEAs. For example, reproduction operation based on DE [25,36], reproduction operation based on PSO, reproduction operation based on ACO [37], etc. There are multiple selection strategies used to maintain the diversity of solutions after the copy operation, such as the crowding distance sorting strategy in NSGAII [10], the niching strategy in NSGAIII [6], the stable matching (STM) model strategy in MOEA/D-STM [27], etc.

This article focuses on the selection process in MOEA/D. As the optimal solution of a subproblem is generally the intersection of the Pareto front and the weight vector, and the weight vectors are generally designed evenly distributed [9,32]. A unique and distinctive solution for each subproblem implies a promising solutions diversity and a well distribution along the PF [28]. In MOEA/D [9], the solution with the smallest aggregation function value is selected for the current subproblem in the its neighborhood. In MOEA/D-AGR [26], the authors proposed to influence the process of selecting solutions by adjusting the size of neighborhoods. In MOEA/D-ACD [29], in order to help balance the diversity and convergence, the authors proposed to add some constraints to the subproblem. In MOEA/D-MRDL [30], the selection solution is based on an online diversity evaluation.

2.2.2. Motivations

In MOEA/D, a multiobjective optimization problem (MOP) is decomposed into a number of optimization subproblems. Solving MOP can be converted to obtain a set of uniform solutions with Pareto approximation. In fact, the diversity of the solutions implied in MOEA/D algorithm framework depends on the uniform distribution of subproblems. In [32], the author found that the solution of the subproblem is the intersection of the Pareto front and the corresponding weight vector of the subproblem. If it can find the intersections or close enough to these intersections, then the solutions of the MOP will naturally have diversity. Therefore, in the

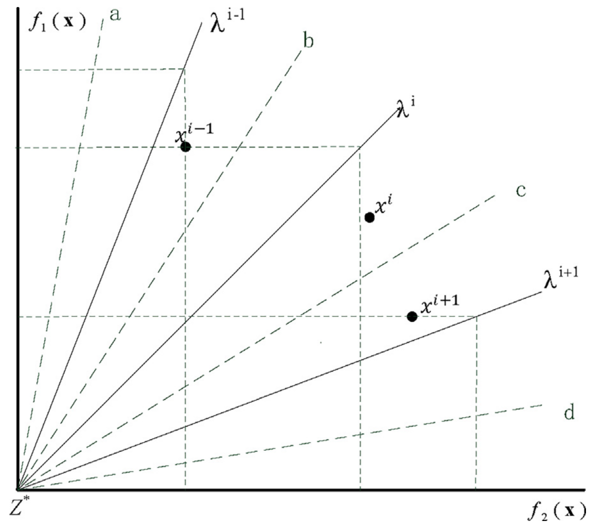

Fig. 3. Illustration of objective space partition.

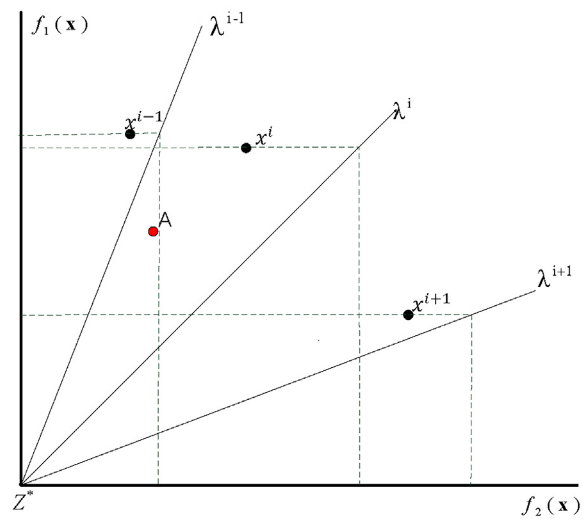

Fig. 4. Illustration of the distribution of solution.

evolutionary process, the solution of subproblem need be as close as possible to corresponding weight vector. One possible way to guarantee the diversity of solutions is to establish an one-to-one correspondence relation between subproblems and solutions in the evolutionary process. Motivated by this, a new MOEA/D variant MOEA/D-ARA is proposed which uses an adaptive region adjustment to improve the balance of convergence and diversity. In this paper, we use the objective region partition strategy to establish this correspondence relationship.

2.2.3. Objective region partition

As discussed in Section 2.2.2, a one-to-one correspondence relation between subproblems and solutions can guarantee the diversity of solutions. The diversity can be characterized by the distribution of solutions in the objective space. These solutions that are distributed across different regions within the objective space have a higher diversity. In fact, the subproblems divides the entire objective space into a number of sub-regions. Through this division, we can identify which region is located closest to the weight vector of the corresponding subproblem and then classify such region as the solution region of the subproblem. We use a special example to briefly explain this strategy. As shown in Fig. 3, the region of the vectors a and b is the solution region of weight vector λ^{i-1} and the solution x^{i-1} is the solution of subproblem λ^{i-1}. And the solution region of λ^i and λ^{i+1} are similar. It is easy to know from Fig. 1 that under this strategy x^i is assigned to subproblem λ^i instead of MOEAD algorithm where x^{i-1} is assigned to λ^i. Although x^i is a little worse than x^{i-1} in terms of the Tchebycheff function value for subproblem λ^i, but x^i is in fact more preferred. Without loss of generality, we provide two simple ways to measure and partition the objective space based on the weight vectors. The geometric relationship between a solution and a weight vector can be described in two simple ways, namely, by measuring the distance between the solution point and the weight vector [38] and by measuring the angle between the weight vector and the solution vector that connects the solution and reference point [20,29,39]. The distance is calculated as follows:

$$d(\lambda, F(x)) = F(x) - \frac{\lambda' \cdot F(x)}{\lambda' \cdot \lambda}\lambda \tag{6}$$

where x is the solution point, λ is the weight vector, λ' is the transpose of λ and $F(x)$ is the object function value. Given the difficulty in calculating the angle, we use to the sin value of the angle as a proxy for the angle value because the sin function is strictly monotonically increasing when the angle value ranges from 0 to $\frac{\pi}{2}$. The sin value of the angle is calculated as follows:

$$\sin\langle\lambda, F(x)\rangle = \sqrt{1 - \left(\frac{\lambda' \cdot F(x)}{|F(x)||\lambda|}\right)^2} \tag{7}$$

where all symbols have the same connotation as those in the previous formula.

The algorithm proposed in this paper uses the angle measurement, and the distance measurement is used to study the influence of the measurement on the algorithm.

3. Adaptive diversity adjustment mechanism

3.1. Algorithm proposed

3.1.1. Adaptive region adjustment

We investigate the one-to-one correspondence objective region partition strategy in the previous section to guarantee the diversity of solutions. However, there is another shortcoming in the MOEA/D framework that needs to be overcome, which is the problem of evolutionary imbalance between subproblems. This also leads to an imbalance between convergence and diversity. In order to distinguish with the imbalance discussed earlier, we call it vertical imbalance, and the former we call it horizontal imbalance which is similarity of solutions to different subproblems. Though the solution of subproblem with slow convergence is larger probability updated by the solution of adjacent subproblem with fast convergence is largely improved by keeping the diversity strategy, the evolutionary imbalance between subproblems also leads to another problem. The probability that the subproblem with slow convergence will be update is small, one of the reasons is probably because the subproblem optimization area is relatively small, which leads to some subproblems to find the optimal solution very early and others may remain in the random generation solution. As show in Fig. 4, the new solution A will replace x^{i-1} and x^i in the MOEAD, but only x^i in the diversity strategy. In fact, it is clear that in the next generation of population to retain A and x^{i-1} better than to retain A and x^i. We further improve by proposing the ARA strategy to overcome the drawback, in which the solution region of the subproblem is dynamically adjusted based on the comparison between the solutions of adjacent subproblems.

Definition 1. Given subproblem λ^i and its current solution x^i, we say that the set of x is the contour of current solution

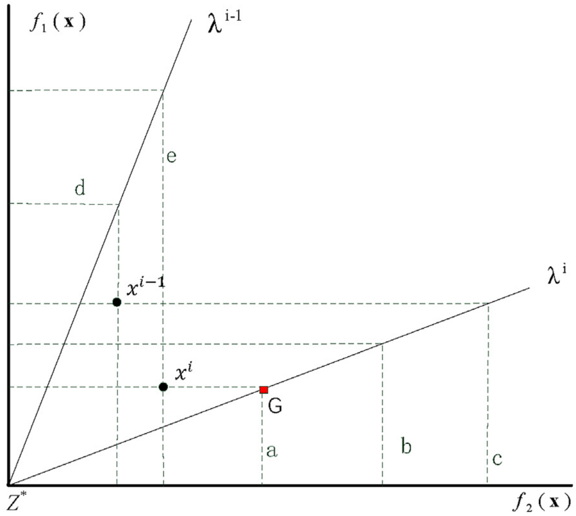

Fig. 5. Illustration of the solution on the contour line comparison.

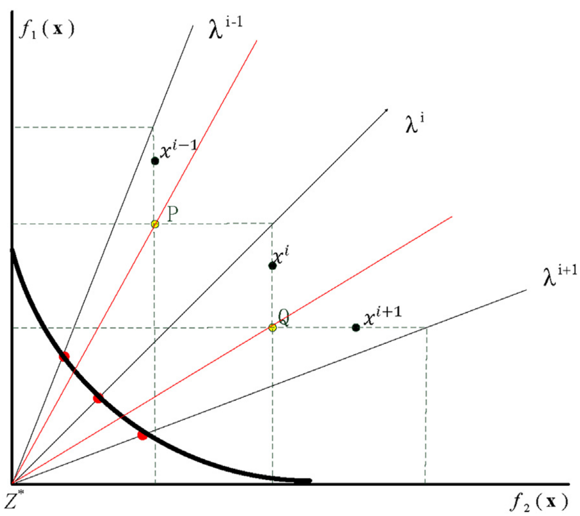

Fig. 6. Illustration of regional dynamic adjustment.

x^i to the subproblem λ^i (contour_x^i_λ^i), if x is satisfied contour_x^i_$\lambda^i = \{x | g^T(x|\lambda^i, z^*) = g^T(x^i|\lambda^i, z^*), \forall x \in \Omega\}$.

For a subproblem, whether a new solution replaces the current solution is to see if the formula (3) value is less than the value of the current solution. From a geometric perspective, the solution of a subproblem must be placed on the edge of the innermost square, in which the intersection of contour and subproblem corresponding vector acts as the vertex while reference point (z^*) is another diagonal vertex. As shown in Fig. 5, the G point is the intersection. In other words, for subproblem λ^i, the solution that can replace the current solution x^i is only possible in the rectangle with the line segment GZ^* as the diagonal. Further, for the order of the two solutions, different subproblems may not be the same. For subproblem λ^i, the solution on the contour line a (contour_x^i_λ^i) is better than that on the contour line b, and the solution on the contour line is superior to that on the contour line c (contour_x^{i-1}_λ^i). for subproblem λ^{i-1}, the solution on the contour line d (contour_x^{i-1}_λ^{i-1}) is better than that on the contour line e (contour_x^i_λ^{i-1}). Specifically, for subproblem λ^i, x^i is better than x^{i-1}, but for subproblem λ^{i-1}, x^{i-1} is better than x^i.

Definition 2. Given adjacent two subproblems λ^{i-1}, λ^i and their current contours contour_x^{i-1}_λ^{i-1}, contour_x^i_λ^i, we say that the intersection point of contour_x^{i-1}_λ^{i-1} and contour_x^i_λ^i is contour point (O_λ^{i-1}_λ^i)

Algorithm 3. Adaptive Region Adjustment

Input:
solution set S, subproblem set P, a new solution.
Output:
the new solution s^* may replace a old solution of solution set S

```
1:  for i ⟵ 1 to solution size do
2:    k ⟵ perm[i];
3:    for j ⟵ 1 to object number do
4:  if g(newsolution, lambda_[k]) < g(newsolution, lambda_[ind[j]]) then
5:    temp ⟵ Ind[j]; Ind[j] ⟵ k; k ⟵ temp
6:  end if
7:    end for
8:  end for
9:  for i ⟵ 1 to object number do do
10:   Min;
11:   for j ⟵ 1 to object number do do
12: if min > f[j] * lambda_[ind[j]][i] then
13:   min ⟵ f[j] * lambda_[ind[j]][i];
14: end if
15:   end for
16:   point[i] ⟵ min
17: end for
18: for i ⟵ 1 to object number do do
19:   if f_new < f[i] && g(newsolution, lambda_[ind[i]]) < g(point,
    lambda_[ind[i]]) then
20: solution of subproblem ind[i] replaced by new solution;
21:   end if
22: end for
```

In the ARA strategy, contour point is used as the division. For a subproblem, if the convergence speed of the current subproblem is slower than the adjacent subproblem, then the intersection is located far from the weight vector of corresponds to the subproblem. In this case, the probability that the current solution of the subproblem will be replaced is greater than that of the adjacent subproblem. As shown in Fig. 6, the solutions x^{i-1}, x^i and x^{i+1} correspond to the subproblems λ^{i-1}, λ^i and λ^{i+1}, respectively, while points P and Q denote O_λ^{i-1}_λ^i and O_λ^i_λ^{i+1} respectively. If using angle measurement, the current solution region of subproblem λ^i is surrounded by rays $\vec{Z^*P}$, $\vec{Z^*Q}$ and the contour line contour_x^i_λ^i. The pseudo-code of the ARA strategy is given in Algorithm 3.

In Algorithm 3, lines 1 to 8 to determine that the location of the new solution is located among which several subproblems, the general m is equal to the objective dimension. Line 1 and line 2 loop through each subproblem in a random manner to compute the angle of the subproblem and the new solution. The *perm* array is used to store random sorted subproblem numbers. And the *ind* array is used to store the nearest m subproblems from the new solution. Line 3 loops through the *ind* array to update it. The *g* function is used to calculate the angle. Line 4 calculates the angle of the new solution to the current subproblem, the angle of the new solution to the subproblem in the ind array, and then compares them. If the angle of the new solution to the current subproblem is smaller than the angle of the new solution to the subproblem in the ind array, the subproblem in the ind array is replaced with the current subproblem in the line 5. Lines 9 to 17 are used to calculate contour point. The *point* array is used to store contour point. The meaning of the loop in line 9 is to traverse each object and find the smallest function value of each object. Line 10 loops through each subproblem in the *ind* array to find the minimum function value. Line 18 to 22 update subproblems if conditions are met. For a subproblem, the solution of the subproblem is replaced by the new solution only if the value of the Chebyshev function of the new solution is smaller than that of the old solution and the angle of the new solution to the subproblem is smaller than the angle of the contour point to the subproblem.

3.1.2. Computational complexity analysis

In this section, we will analyze the computational complexity of the ARA strategy in three steps. First, we identify some of the most suitable subproblems for the new solution. This process is described in lines 1 to 8 of Algorithm 3. It needs to find out m suitable subproblems in the N subproblems, so requires $O(MN)$ computations. Second, we find the minimum component of the current solution of these subproblems to form the contour point. This process is described in lines 9 to 17 of Algorithm 3 and requires $O(MM)$ computations. Third, we find the subproblem of the region where the new solution is located and then compare the object value. If the value of this subproblem is less than the object value of its current solution, then this subproblem is replaced and it needs constant time $O(M)$. M denotes the number of objects, and this value must be far less than that of N, which denotes population size. So the strategy ARA requires a total of $O(MN)$ computations.

3.1.3. Discussion

In this section we compare MOEA/D-ARA with the general MOEA/D framework. In MOEA/D-ARA, because of the partition of the solution to the subproblem and ARA strategy, it is almost impossible for different subproblems to have the same solution and the diversity of solutions is preserved. More generally, MOEA/D-ARA can also ensure that the difference between solutions is as large as possible through a repulsive relationship between similar solutions. Fig. 6 shows that the new solution produced by MOEA/D-ARA can replace the old solution without reducing the area. The ARA strategy also makes the area of this solution similar to that of the sector, thereby preventing overlapping of solution areas between subproblems. In other words, that is to strengthen the diversity at the same time did not weaken its convergence. In MOEA/D-ARA we choose all subproblems as a replacement neighborhood in order to find a highly suitable subproblem for the new solution. In this way, we can avoid some new solutions that are unsuitable for the current subproblem. Otherwise, the solution to the other subproblem is abandoned in this process, thereby wasting computational resources.

In order to enhance the readability of this paper, we further compare MOEA/D-ARA proposed in this paper with the MOEA/D-SAS [20], which discuss issues of diversity and convergence. The MOEA/D-ARA and the MOEA/D-SAS are both discussing how to improve the balance between diversity and convergence. Both MOEA/D-ARA and MOEA/D-SAS use the value of the angle between the solution and the subproblem to measure the relationship between the solution and the subproblem. However, these two algorithms have different purposes for solving it. In MOEA/D-SAS, the L nearest solutions for each subproblem are found sequentially, and then all the solutions are divided into different subsets according to the relationship between the solutions and the subproblems. The closer the solution to the sub-problem is, the more likely it is to be reserved for the next generation. Further, in MOEA/D-SAS, the two points in the objective space are calculated, and the largest angle in the smallest angle of the selected next-generation solution set is selected. In MOEA/D-ARA, the relationship between the subproblem and the solution is to determine which subproblem is more suitable for the current solution, and the current solution can only replace the solution to the subproblem. Therefore, it is more likely to find a potential subproblem that can be replaced.

3.2. Incorporation into MOEA/D

In this section, we present a MOEA/D variant. It uses the ARA strategy presented in the previous section, and we call it MOEA/D-ARA, which differs from MOEA/D-DRA [40] in that the ARA strategy is applied during the selection process. The pseudo-code of MOEA/D-ARA is showed in Algorithm 4, and then some important components of MOEA/D-ARA are further more illustrated. The parameters of input in Algorithm 4, MOP is the demand solution problem, Maxevaluations is the stopping criterion, N is the number of population or subproblems considered in MOEA/D, T_m is the number of the weight vectors whose corresponding solutions are used to select mating parent solutions in the neighborhood of each weight vector, T_r is the number of the weight vectors whose corresponding solutions may be replaced by new solutions in the neighborhood of each weight vector and δ is the probability that parents are selected from the neighborhood.

Algorithm 4. MOEA/D-ARA

$$P = \begin{cases} B(i), & \text{if } rand < \delta \\ \{1, 2, \ldots, N\} & \text{otherwise} \end{cases}$$

Input:
MOP, Maxevaluations, N, T_m, T_r, δ.

Output:
An approximated POF $\{f_1, f_2, \ldots, f_N\}$
An approximated POS $\{x_1, x_2, \ldots, x_N\}$

1: Initialize the weight vectors λ' by applying WS-transformation Eq. (2) on the evenly weight vectors, and then compute him and T closest weight vectors to each weight vector without the euclidean distance;
2: Initialize the population $S \longleftarrow \{x_1, x_2, \ldots, x_N\}$, by uniformly random sampling the decision space, and then compute the objective function value fi of each solution.
3: Initialize the z by setting $z_k = \min_{j=1,2,\ldots,N} f_k^f$ where $k = 1, 2, \ldots, m$.
4: set gen = 0 and $\pi_i = 1$ for all $i = 1, 2 \ldots, N$.
5: Let all indices of the subproblems whose objectives are MOP individual objectives f_i form the initial I. By using ten-tournament selection based on π_i, select other $\frac{N}{5} - m$ indices and add them to I
6: **for** $i \longleftarrow 1$ to size(I) **do**
7: Selection of Mating/Update Range: Uniformly randomly generate a number rand from (0,1). Then set
$$P = \begin{cases} B(i), & \text{if } rand < \delta \\ \{1, 2, \ldots, N\} & \text{otherwise} \end{cases}$$
8: ReproductionSet $r_1 = i$ and Randomly select three solutions x^{r_2}, x^{r_3} from P; then use $x^{r_1}, x^{r_2}, x^{r_3}$ to generate a solution $\bar{y}$ by a DE operator. And use a mutation operator on $\bar{y}$ with probability p_m to produce a new solution y. Evaluate the F-function value of y;
9: Update the current ideal objective vector z^*: for each $j = 1, 2, \ldots, m$, if $z_j < f_j(y)$, then set $z_j = f_j(y)$.
10: Update the current solutions: use Algorithm 3 update the current solutions.
11: **end for**
12: Stopping Criteria if the stopping criteria is satisfied, then stop and output solutions and them F-function value.
13: iteration++. If mod($iteration$, 50) = 0 then compute Δ^i, the relative decrease of the objective for each subproblem I during the last 50 iteration, update as follows:
$$\pi^i = \begin{cases} 1, & \text{if } \Delta^i > 0.001 \\ (0.95 + 0.05 \times \frac{\Delta^i}{0.001}) \times \pi^i & \text{otherwise} \end{cases}$$
Go to Line 6.

3.2.1. Initialization of the algorithm

Evenly spread weight vectors are usually preferred as we do not have a priori knowledge on how the weight vectors affect the distribution of final output solutions. In MOEA/D framework, there is no easy to generate the evenly spread weight vectors when the number of objectives is large. Here, We initialize a set of weight vectors $\lambda = \{\lambda_1, \lambda_2, \ldots, \lambda_N\}$ that are evenly spread in the objective space and set the number of weight vectors be equal to the population size. In this paper, we use the method proposed in [41] to generate the evenly spread weight vectors. Each element of a weight vector w takes a value from $\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\}$, where H is the number of divisions along each coordinate, and the number of weight vectors is $N = C_{m-1}^{N+m-1}$. After the generation of λ, we get the weight vectors $\bar{\lambda}$ by applying WS-transformation Eq. (3) on the evenly weight vec-

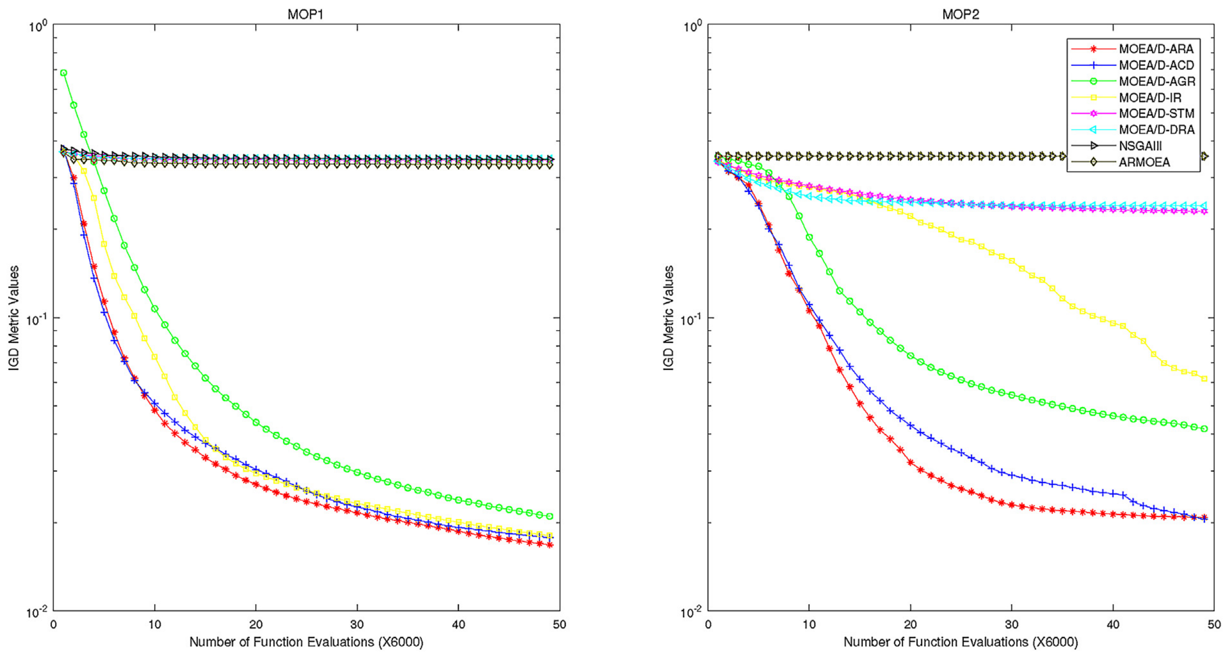

Fig. 7. Evolution of the median IGD metric values versus the number of function evaluations.

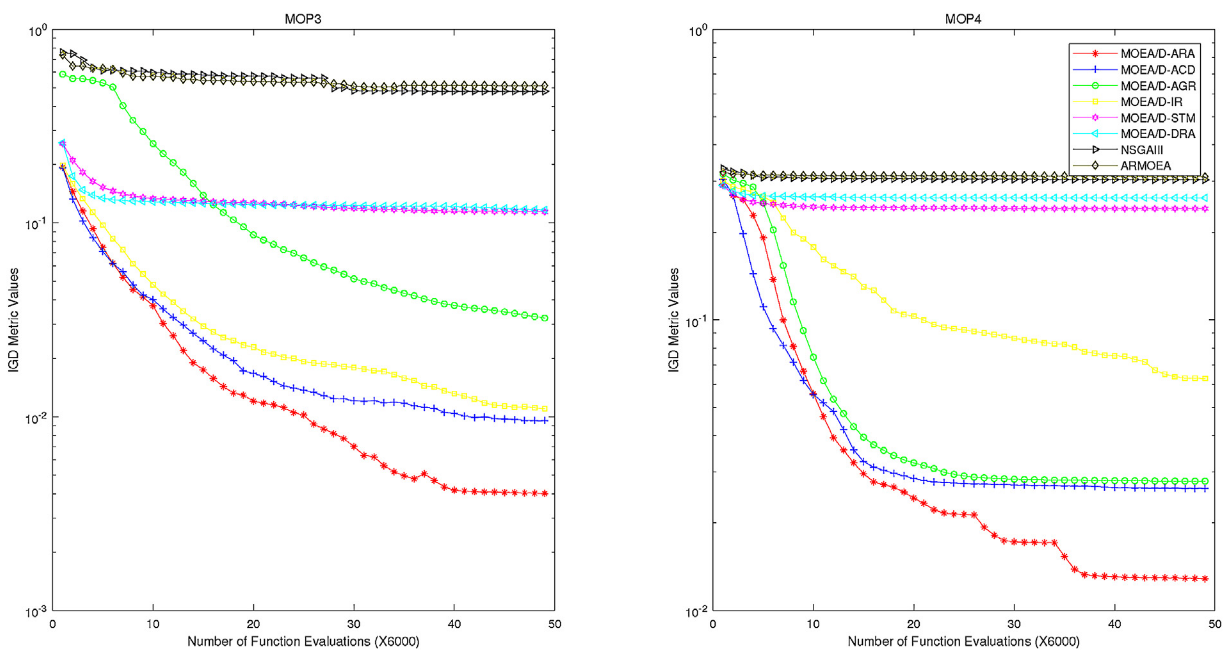

Fig. 8. Evolution of the median IGD metric values versus the number of function evaluations.

tors and compute the Euclidean distance between any two weight vectors of $\bar{\lambda}$. The neighborhood set of each weight vector λ^i, $i \in \{1, 2, \ldots, N\}$, is defined as $B(i) = \{i_1, i_2, \ldots, i_T\}$, where $\lambda^{i_1}, \lambda^{i_2}, \ldots, \lambda^{i_T}$ are the $T\,(1 \le T \le N)$ closest weight vectors of λ^i.

The initial population $S = x_1, \ldots, x_N$ can be randomly sampled from via a uniform distribution as no prior knowledge about the search can be used. After the generation of S, we compute the objective function value f_i of each solution. Since the ideal objective vector is usually unknown a priori, here we use the set of the minimum F-function value of each objective to approximately replace the objective vector.

3.2.2. Reproduction and evolutionary strategies

A very important operation in the evolutionary algorithm is the reproduction operation. The reproduction operation is to generate the offspring population. In this paper, the reproduction operation is the same as done in [25], which use the differential evolution (DE) operator [42] and polynomial mutation [43]. Specifically, the process of generating an offspring solution $x^i_{new} = \{x^i_1, x^i_2, \ldots, x^i_n\}$ is as follows:

$$u^i_j = \begin{cases} x^{r_1}_j + F \times (x^{r_2}_j - x^{r_3}_j), & \text{if } rand < CR \text{ or } j = j_{rand} \\ x^i_j & \text{otherwise} \end{cases} \quad (8)$$

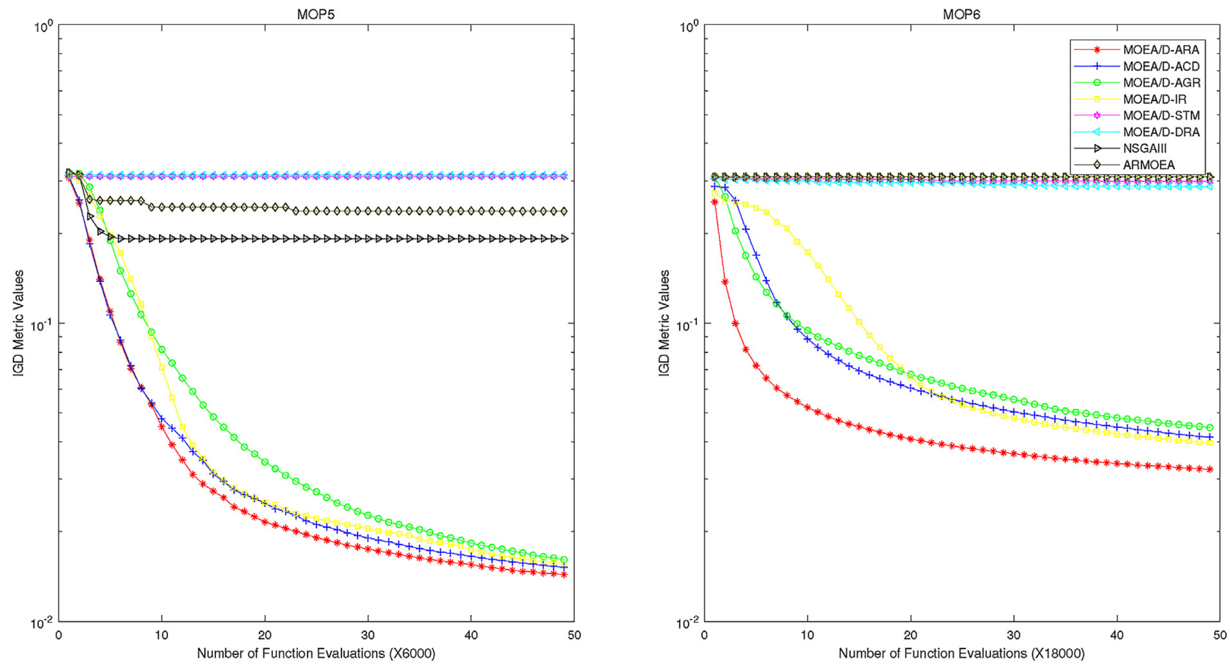

Fig. 9. Evolution of the median IGD metric values versus the number of function evaluations.

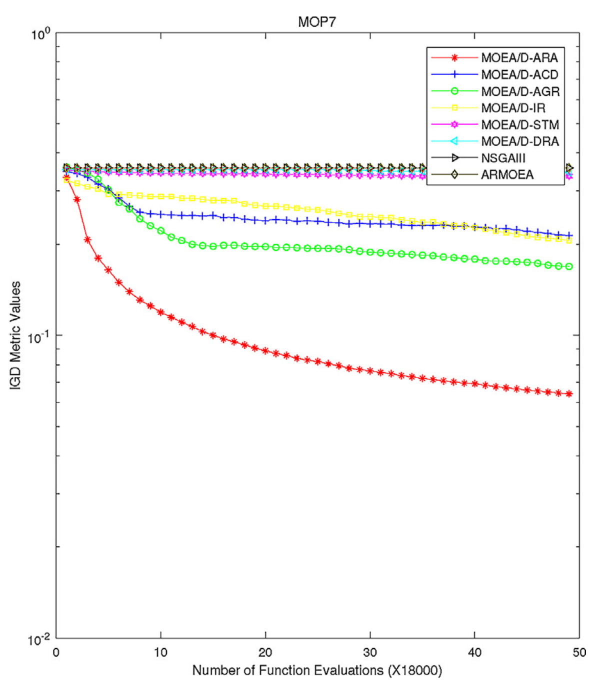

Fig. 10. Evolution of the median IGD metric values versus the number of function evaluations.

where $x_j^{r_1}$ is the current solution and $x_j^{r_2}$, $x_j^{r_3}$ are two solutions randomly chosen from neighborhood, CR and F are two control parameters of differential evolution operator, $rand$ is a random real number which is uniformly sampled from 0 to 1, j_{rand} is a random integer uniformly chosen from 1 to n. On the basis of this intermediate solution, we use polynomial mutation to obtain solution x_{new}^i. The jth decision variable of x_{new}^i is calculated as follows:

$$x_j^i = \begin{cases} u_j^i + \sigma_j \times (b_j - a_j), & \text{if } rand < P_m \\ u_j^i & \text{otherwise} \end{cases} \quad (9)$$

and σ_j of the above formula is calculated as follows:

$$\sigma_j = \begin{cases} (2 \times rand)^{\frac{1}{\eta+1}} - 1, & \text{if } rand < 0.5 \\ 1 - (2 - 2 \times rand)^{\frac{1}{\eta+1}} & \text{otherwise} \end{cases} \quad (10)$$

where the distribution index η and mutation rate p_m are two control parameters, and a_j, b_j are respectively lower and upper bounds of the jth decision variable.

4. Experiment settings

To validate the optimization performance of our proposed algorithm, seven state-of-the-art MOEAs were used here for comparative studies. The details are as follows.

4.1. The comparison algorithm

1. MOEA/D-DRA [40]: It is a variant of MOEA/D, which won the CEC2009 MOEA competition. And different from MOEA/D, it dynamically allocates the computational resources to different subproblems based on their utility values.
2. MOEA/D-STM [27]: It is a variant of MOEA/D, which uses a stable matching model to adjust the selection process of MOEA/D.
3. MOEA/D-IR [28]: It is a recently proposed MOEA/D variant, which uses incorporating interrelationship by defining mutual preferences between subproblems and solutions for selection in MOEA/D.
4. MOEA/D-AGR [26]: It is a recently proposed MOEA/D variant, which applies a global replacement (GR) scheme and adaptive scheme to adjusting T_r size for MOEA/D replacement mechanism
5. MOEA/D-ACD [29]: It is a recently proposed MOEA/D variant, which imposes some constraints on the subproblems to help MOEA/D balance the population diversity and convergence in an appropriate manner.
6. NSGA-III [6]: It is the most popular Pareto-based MOEA, which suggest a reference-point-based many-objective evolutionary algorithm following NSGA-II framework. It emphasizes that the

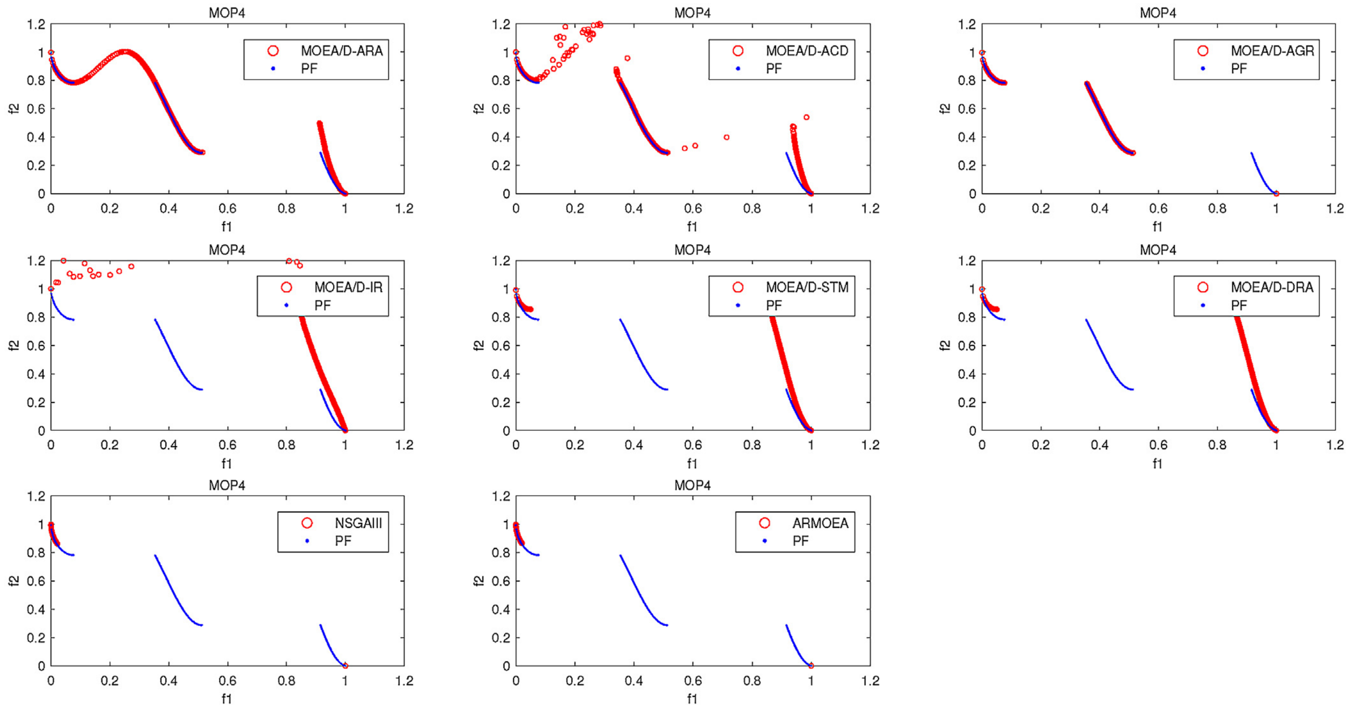

Fig. 11. Plots of the final solutions with the lowest IGD-metric values found by MOEA/D-ARA, MOEA/D-ACD, MOEA/D-AGR, MOEA/D-IR, MOEA/D-STM, MOEA/D-DRA, NSGAIII and ARMOEA in 30 runs in the objective space on MOP4.

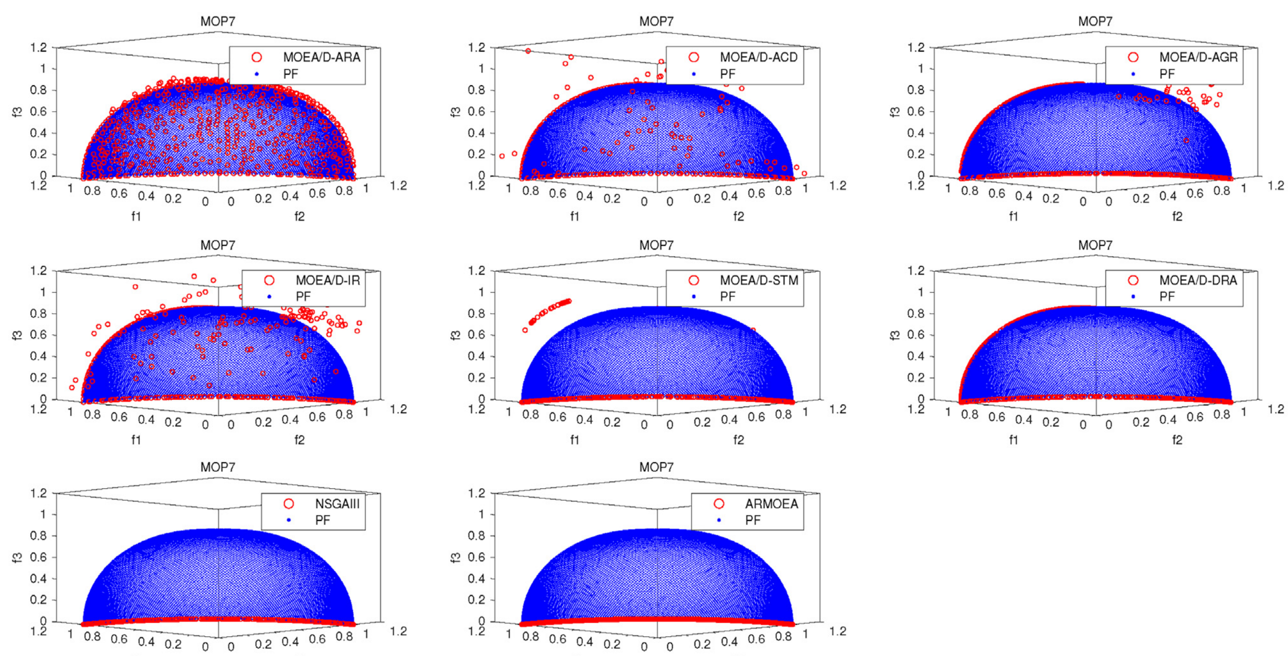

Fig. 12. Plots of the final solutions with the lowest IGD-metric values found by MOEA/D-ARA, MOEA/D-ACD, MOEA/D-AGR, MOEA/D-IR, MOEA/D-STM, MOEA/D-DRA, NSGAIII and ARMOEA in 30 runs in the objective space on MOP7.

solutions are both non-dominated and close to a set of supplied reference points.

7. ARMOEA [15]: It propose an MOEA based on an enhanced inverted generational distance indicator, in which an adaptation method is suggested to adjust a set of reference points based on the indicator contributions of candidate solutions in an external archive.

4.2. Testing problems

In order to study the MOEAD-ARA algorithm proposed in this paper, seventeen unconstrained MOP test instances are used as the benchmark problems. Specifically, UF1 to UF10 are used as the benchmark in CEC2009 MOEA competition [40], which UF1-UF7 are bi-objective problems and UF8-UF10 are tri-objective problems. MOP1-MOP5 are bi-objective problems and MOP6-MOP7 are tri-objective problems [4,44]. The number of decision variables of

Table 1
Performance comparisons of IGD values on MOP1-MOP7 test instances.

Problem	IGD	ARA	ACD	AGR	IR	STM	DRA	NSGAIII	ARMOEA
MOP1	Mean	**1.661E−02**	1.755E−02	2.047E−02	1.784E−02	3.440E−01	3.469E−01	3.4205E−01	3.435E−01
	Std	1.30E−03	1.56E−03	5.23E−03	1.31E−03	3.46E−02	3.37E−02	4.79E−03	3.78E−03
	Rank	1	2−	4−	3−	5−	6−	7−	8−
MOP2	Mean	**2.073E−02**	2.080E−02	4.101E−02	6.053E−02	2.305E−01	2.402E−01	3.509E−01	3.549E−01
	Std	4.45E−02	4.21E−02	5.79E−02	7.98E−02	4.94E−02	5.51E−02	1.06E−02	2.26E−16
	Rank	1	2 ≈	4−	3−	6−	5−	7−	8−
MOP3	Mean	**4.005E−03**	8.716E−03	2.517E−02	1.071E−02	1.255E−01	1.288E−01	4.875E−01	5.068E−01
	Std	6.09E−03	1.09E−02	5.07E−02	2.41E−02	5.98E−02	5.95E−02	4.69E−02	3.56E−02
	Rank	1	2 ≈	4−	3 ≈	5−	6−	7−	8−
MOP4	Mean	**1.649E−02**	3.039E−02	3.366E−02	7.044E−02	2.496E−01	2.759E−01	2.948E−01	3.058E−01
	Std	1.48E−02	3.91E−02	3.60E−02	7.04E−02	9.24E−03	3.16E−02	1.49E−02	3.23E−01
	Rank	1	2 ≈	3−	4−	5−	6−	7−	8−
MOP5	Mean	**1.423E−02**	1.506E−02	1.586E−02	1.540E−02	3.105E−01	3.139E−01	2.003E−01	2.401E−01
	Std	9.95E−04	1.19E−03	2.10E−03	1.50E−03	1.40E−02	1.05E−02	1.87E−02	2.87E−02
	Rank	1	2−	4−	3−	7−	8−	5−	6−
MOP6	Mean	**3.181E−02**	4.053E−02	4.388E−02	3.902E−02	2.959E−01	2.848E−01	3.093E−01	3.093E−01
	Std	8.15E−04	1.30E−03	1.30E−03	1.49E−03	1.89E−02	3.08E−02	4.22E−06	2.93E−07
	Rank	1	3−	4−	2−	6−	5−	7−	8−
MOP7	Mean	**6.198E−02**	2.102E−01	1.642E−01	1.973E−01	3.288E−01	3.402E−01	3.570E−01	3.570E−01
	Std	2.45E−03	3.59E−02	3.74E−02	3.72E−02	3.25E−02	2.52E−02	7.01E−06	2.42E−07
	Rank	1	4−	2−	3−	5−	6−	8−	7−
	Total rank	7	17	25	21	39	41	48	53
	Final rank	1	2	4	3	5	6	7	8

Wilcoxons rank sum test at a 0.05 significance level is performed between MOEA/D-ARA and each of the other competing algorithms. − and + denotes that the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-ARA, respectively. The best mean is bold. MOP1 to MOP5 have two objectives and MOP6, MOP7 have three objectives.

Table 2
Performance comparisons of HV values on MOP1-MOP7 test instances.

Problem	HV	ARA	ACD	AGR	IR	STM	DRA	NSGAIII	ARMOEA
MOP1	Mean	**6.436E−01**	6.425E−01	6.389E−01	6.421E−01	1.139E−01	1.087E−01	1.340E−01	1.310E−01
	Std	1.69E−03	2.01E−03	5.38E−03	1.63E−03	6.78E−02	6.37E−02	1.11E−02	8.77E−03
	Rank	1	2−	4−	3−	5−	6−	7−	8−
MOP2	Mean	**3.105E−01**	3.076E−01	2.880E−01	2.595E−01	6.786E−02	5.947E−02	2.357E−03	0.000E+00
	Std	4.50E−02	4.44E−02	6.80E−02	9.54E−02	4.74E−02	4.38E−02	6.93E−03	0.00E+00
	Rank	1	2−	3−	4−	5−	6−	7−	8−
MOP3	Mean	**2.085E−01**	2.015E−01	1.949E−01	2.010E−01	9.943E−02	9.670E−02	0.000E+00	0.000E+00
	Std	9.31E−03	1.67E−02	3.71E−02	2.73E−02	5.26E−02	5.30E−02	0.00E+00	0.00E+00
	Rank	1	2 ≈	4−	3 ≈	5−	6−	7−	8−
MOP4	Mean	**4.980E−01**	4.802E−01	4.797E−01	4.326E−01	1.763E−01	1.576E−01	1.447E−01	1.342E−01
	Std	1.94E−02	4.87E−02	5.39E−02	9.41E−02	2.06E−02	2.02E−02	1.48E−02	6.83E−03
	Rank	1	2 ≈	3−	4−	5−	6−	7−	8−
MOP5	Mean	**6.458E−01**	6.448E−01	6.438E−01	6.442E−01	3.536E−01	3.536E−01	3.551E−01	3.544E−01
	Std	1.20E−03	1.56E−03	1.98E−03	2.06E−03	6.25E−17	6.25E−17	8.40E−03	1.00E−03
	Rank	1	2−	4−	3−	7−	8−	5−	6−
MOP6	Mean	**7.924E−01**	7.840E−01	7.774E−01	7.841E−01	5.091E−01	5.253E−01	4.978E−01	4.991E−01
	Std	9.11E−04	1.52E−03	1.39E−03	1.65E−03	2.83E−02	4.59E−02	1.14E−04	1.62E−05
	Rank	1	3−	4−	2−	6−	5−	8−	7−
MOP7	Mean	**4.046E−01**	3.367E−01	3.446E−01	3.235E−01	2.251E−01	2.208E−01	2.123E−01	2.139E−01
	Std	2.40E−03	1.98E−02	5.15E−02	3.63E−02	2.53E−02	2.37E−02	2.23E−04	9.38E−06
	Rank	1	4−	2−	3−	5−	6−	8−	7−
	Total rank	7	17	24	22	38	43	49	52
	Final rank	1	2	4	3	5	6	7	8

Wilcoxons rank sum test at a 0.05 significance level is performed between MOEA/D-ARA and each of the other competing algorithms. − and + denotes that the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-ARA, respectively. The best mean is bold. MOP1 to MOP5 have two objectives and MOP6, MOP7 have three objectives.

UF1-UF10 is set to 30; for MOP1-MOP7, the number of objectives is set to 10. MOP1-MOP7 problems decision variables range from 0 to 1. And UF1-UF10 decision variables range from −1 to 1 except that the first variable (x_1) is from 0 to 1.

4.3. The parameter settings

The parameters of MOEA/D-DRA, MOEA/D-AGR, MOEA/D-STM, MOEA/D-IR, MOEA/D-ACD, NSGAII and SPEA2 are set according to their corresponding references [40,26–29,6,15] respectively. The detailed parameter settings of our proposed MOEA/D-ARA are summarized as follows.

- The population size N: It is set to be N=300 for MOP1 to MOP5 problems and UF1 to UF7, $N = 600$ for MOP6 to MOP7 and UF8 to UF10 problems.
- Number of runs and termination condition: Each algorithm was run 30 times independently for each test instance on a Linux operating system computer with a configuration of Intel Core2 Duo CPU 2.4 GHz processor and 16 GB memory. The algorithms termination after a preset number of function evaluations. The maximal number of function evaluations is set to be 300 000 for MOP1 to MOP5 and UF1 to UF7 problems, and 900 000 for MOP6 to MOP7 and UF8-UF10 problems.
- The neighborhood size: In MOEA/D-ARA, $T_m = 20$ and $T_r = N$.

Table 3
Performance comparisons of IGD values on UF1-UF10 test instances.

Problem	IGD	ARA	ACD	AGR	IR	STM	DRA	NSGAIII	ARMOEA
UF1	Mean	**1.644E−03**	1.798E−03	2.448E−03	1.865E−03	1.741E−03	1.809E−03	8.699E−02	1.052E−01
	Std	8.18E−05	5.51E−05	1.90E−04	1.07E−04	7.33E−05	1.65E−04	1.48E−02	1.64E−02
	Rank	1	4−	6−	5−	2−	3−	7−	8−
UF2	Mean	**3.033E−03**	6.695E−03	6.899E−03	4.823E−03	6.418E−03	5.438E−03	2.271E−02	3.291E−02
	Std	1.02E−03	1.50E−03	1.26E−03	1.76E−03	1.44E−03	8.10E−03	4.70E−03	1.32E−02
	Rank	1	5−	6−	3−	4−	2−	7−	8−
UF3	Mean	4.248E−03	7.235E−03	**3.302E−03**	4.448E−03	9.960E−03	1.637E−02	1.500E−01	2.675E−01
	Std	2.62E−03	4.58E−03	2.10E−03	2.59E−03	1.25E−02	2.41E−02	3.96E−02	4.76E−02
	Rank	2	5−	1 +	3 ≈	4 ≈	6−	7−	8−
UF4	Mean	5.457E−02	6.125E−02	5.413E−02	5.616E−02	6.328E−02	5.874E−02	**4.068E−02**	4.086E−02
	Std	2.48E−03	4.96E−03	2.69E−03	3.33E−03	4.79E−03	4.53E−03	4.79E−04	3.88E−04
	Rank	4	7−	3 ≈	5−	8−	6−	1 +	2 +
UF5	Mean	3.054E−01	3.381E−01	2.448E−01	3.077E−01	2.822E−01	3.130E−01	**2.253E−01**	2.877E−01
	Std	1.43E−01	6.47E−02	2.20E−02	8.72E−02	9.23E−02	1.53E−01	4.57E−02	9.15E−02
	Rank	4	8−	3 ≈	7 ≈	5 ≈	6 ≈	1 +	2 +
UF6	Mean	1.333E−01	2.163E−01	**9.310E−02**	9.719E−02	2.177E−01	2.106E−01	1.258E−01	1.530E−01
	Std	1.40E−01	1.66E−01	1.04E−01	4.79E−02	1.69E−01	1.49E−01	1.82E−02	7.58E−02
	Rank	4	6−	1 +	2 ≈	8−	7−	3−	5−
UF7	Mean	**1.826E−03**	3.175E−03	2.875E−03	3.743E−03	2.631E−03	6.773E−03	3.447E−02	1.060E−01
	Std	1.86E−04	1.88E−03	4.26E−04	3.95E−03	1.76E−03	7.08E−03	8.62E−03	1.22E−01
	Rank	1	5−	6−	2−	4−	3−	7−	8−
UF8	Mean	**2.465E−02**	4.025E−02	3.405E−02	2.524E−02	4.367E−02	3.506E−02	4.896E−01	2.356E−01
	Std	4.16E−04	5.13E−03	6.66E−03	8.49E−04	4.38E−03	2.34E−03	8.48E−02	3.26E−04
	Rank	1	5−	3−	2−	6−	4−	8−	7−
UF9	Mean	**4.997E−02**	7.625E−02	5.830E−02	5.910E−02	5.613E−02	1.057E−01	1.201E−01	9.710E−02
	Std	4.86E−02	5.52E−02	5.28E−02	5.45E−02	5.06E−02	5.34E−02	6.26E−02	5.04E−02
	Rank	1	5−	3−	2 ≈	4−	7−	8−	6−
UF10	Mean	**2.859E−01**	4.335E−01	4.720E−01	3.061E−01	3.155E−01	3.168E−01	3.258E−01	3.207E−01
	Std	8.45E−02	8.56E−02	8.74E−02	6.03E−02	2.76E−02	6.59E−02	6.45E−02	1.22E−01
	Rank	1	7−	8−	2 ≈	4−	3−	6 ≈	5 ≈
	Total rank	20	57	40	33	49	46	55	59
	Final rank	1	7	3	2	5	4	6	8

Wilcoxons rank sum test at a 0.05 significance level is performed between MOEA/D-ARA and each of the other competing algorithms. − and + denotes that the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-ARA, respectively. The best mean is bold. UF1 to UF7 have two objectives and UF8 to UF10 have three objectives.

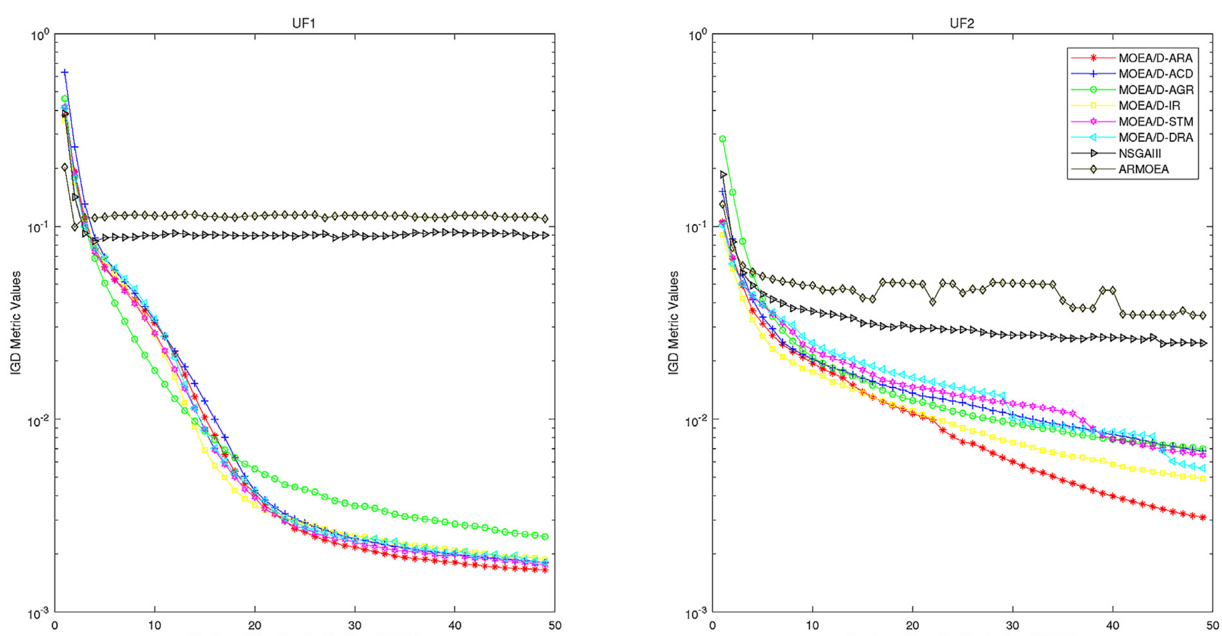

Fig. 13. Evolution of the median IGD metric values versus the number of function evaluations.

- Control parameters: As recommended in [25], we set $CR = 1.0$ and $F = 0.5$ for the DE operator [42] and $\eta = 20$ and $p_m = 1/n$ for the polynomial mutation [43] for reproduction operators. And $\delta = 0.9$ for probability to select in the neighborhood.

4.4. The evaluation indicators

In this paper experimental studies, we adopt the following two widely used performance metrics.

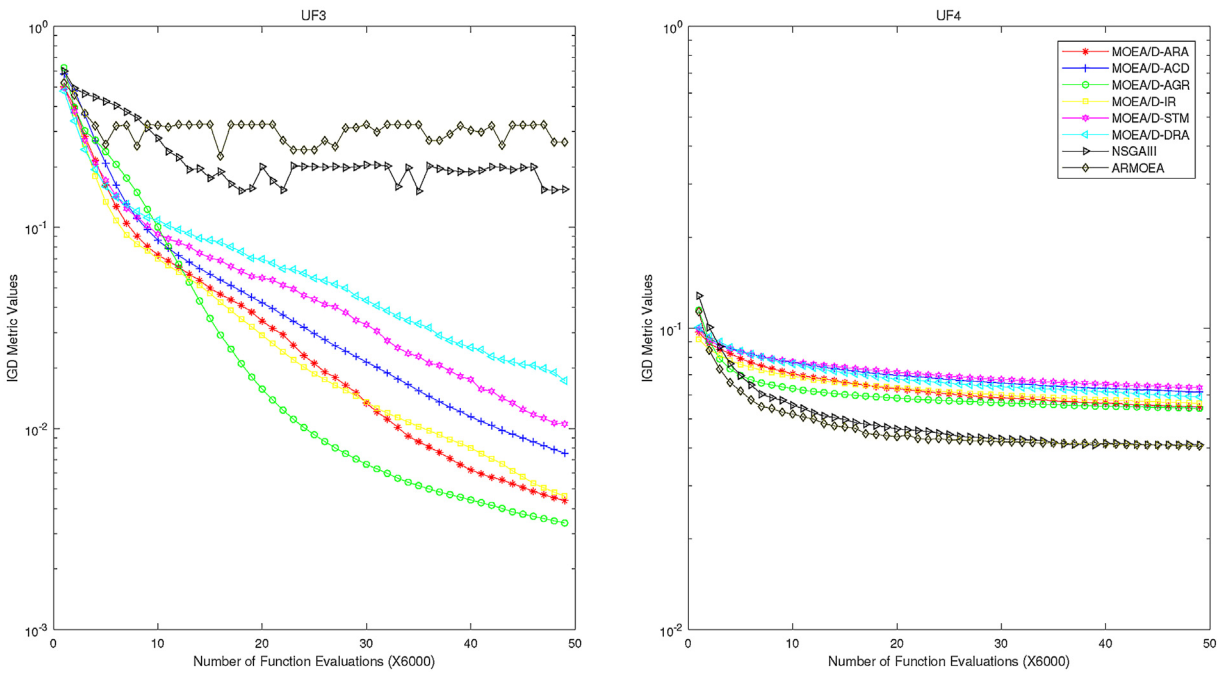

Fig. 14. Evolution of the median IGD metric values versus the number of function evaluations.

Table 4
Performance comparisons of HV values on UF1-UF10 test instances.

Problem	HV	ARA	ACD	AGR	IR	STM	DRA	NSGAIII	ARMOEA
UF1	Mean	**6.638E−01**	6.635E−01	6.624E−01	6.634E−01	6.636E−01	6.636E−01	5.530E−01	5.322E−01
	Std	2.26E−04	1.11E−04	2.72E−04	2.34E−04	1.40E−04	2.30E−04	1.55E−02	1.59E−02
	Rank	1	4−	6−	5−	2−	3−	7−	8−
UF2	Mean	**6.620E−01**	6.570E−01	6.569E−01	6.597E−01	6.574E−01	6.596E−01	6.368E−01	6.313E−01
	Std	1.28E−03	2.28E−03	2.09E−03	2.14E−03	2.02E−03	8.01E−03	5.36E−03	7.81E−03
	Rank	1	5−	6−	3−	4−	2−	7−	8−
UF3	Mean	6.576E−01	6.531E−01	**6.600E−01**	6.564E−01	6.512E−01	6.467E−01	4.765E−01	3.731E−01
	Std	5.91E−03	8.12E−03	3.94E−03	7.20E−03	1.56E−02	1.97E−02	3.80E−02	4.30E−02
	Rank	2	5−	1 +	3 ≈	4 ≈	6−	7−	8−
UF4	Mean	2.530E−01	2.442E−01	2.544E−01	2.509E−01	2.439E−01	2.477E−01	2.751E−01	**2.753E−01**
	Std	3.58E−03	6.59E−03	3.64E−03	5.18E−03	5.11E−03	5.68E−03	5.29E−04	3.32E−04
	Rank	4	7−	3 ≈	5 ≈	8−	6−	2 +	1 +
UF5	Mean	5.352E−02	5.481E−02	2.586E−02	6.428E−02	6.528E−02	1.262E−01	**1.705E−01**	1.661E−01
	Std	7.96E−02	6.76E−02	4.53E−02	7.81E−02	8.48E−02	9.72E−02	9.15E−02	5.78E−02
	Rank	7	5 ≈	8 ≈	4 ≈	6 ≈	3 +	1 +	2 +
UF6	Mean	2.044E−01	1.810E−01	2.188E−01	2.338E−01	1.975E−01	2.059E−01	**2.551E−01**	2.585E−01
	Std	6.91E−02	1.06E−01	9.13E−02	5.59E−02	7.77E−02	7.06E−02	4.68E−02	4.71E−02
	Rank	5	8 ≈	3 +	4 ≈	7 ≈	6 ≈	1 +	1 +
UF7	Mean	**4.967E−01**	4.950E−01	4.950E−01	4.944E−01	4.957E−01	4.929E−01	4.458E−01	3.917E−01
	Std	3.28E−04	1.74E−03	6.98E−04	4.27E−03	1.18E−03	5.10E−03	1.38E−02	8.03E−02
	Rank	1	5−	6−	2−	3−	4−	7−	8−
UF8	Mean	**4.373E−01**	4.014E−01	4.149E−01	4.363E−01	3.936E−01	4.115E−01	2.150E−01	2.071E−01
	Std	1.27E−03	1.35E−02	1.18E−02	1.61E−03	8.23E−03	3.32E−03	1.65E−02	3.99E−04
	Rank	1	5−	3−	2−	6−	4−	7−	8−
UF9	Mean	**7.254E−01**	6.849E−01	7.154E−01	7.151E−01	7.058E−01	6.424E−01	6.147E−01	6.281E−01
	Std	6.95E−02	7.66E−02	7.66E−02	7.71E−02	6.84E−02	7.17E−02	5.09E−02	4.65E−02
	Rank	1	4−	2 ≈	3 ≈	5−	6−	8−	7−
UF10	Mean	1.159E−01	6.255E−02	8.067E−04	9.745E−02	1.090E−01	**1.313E−01**	1.171E−02	7.947E−02
	Std	5.63E−02	3.04E−02	1.52E−03	3.60E−02	2.59E−02	2.98E−02	1.04E−02	9.64E−02
	Rank	3	6−	8−	5 ≈	4 ≈	1 ≈	2 ≈	7−
	Total rank	26	54	46	36	49	41	50	58
	Final rank	1	7	4	2	5	3	6	8

Wilcoxons rank sum test at a 0.05 significance level is performed between MOEA/D-ARA and each of the other competing algorithms. − and + denotes that the performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-ARA, respectively. The best mean is bold. UF1 to UF7 have two objectives and UF8 to UF10 have three objectives.

1. Inverted Generational Distance [45–47]: Inverted generational distance is adopted to assess the algorithm performance and can provide reliable information on both the diversity and convergence of obtained solutions. Let P^* be a set of uniformly distributed Pareto optimal points along the true Pareto front in the objective space. Let P be an approximate set to the true Pareto front obtained by an algorithm. The IGD measures the gap between P^* and P, which is calculated as follows:

$$IGD(P, P^*) = \frac{\sum_{x \in P^*} d(x, P)}{|P^*|} \tag{11}$$

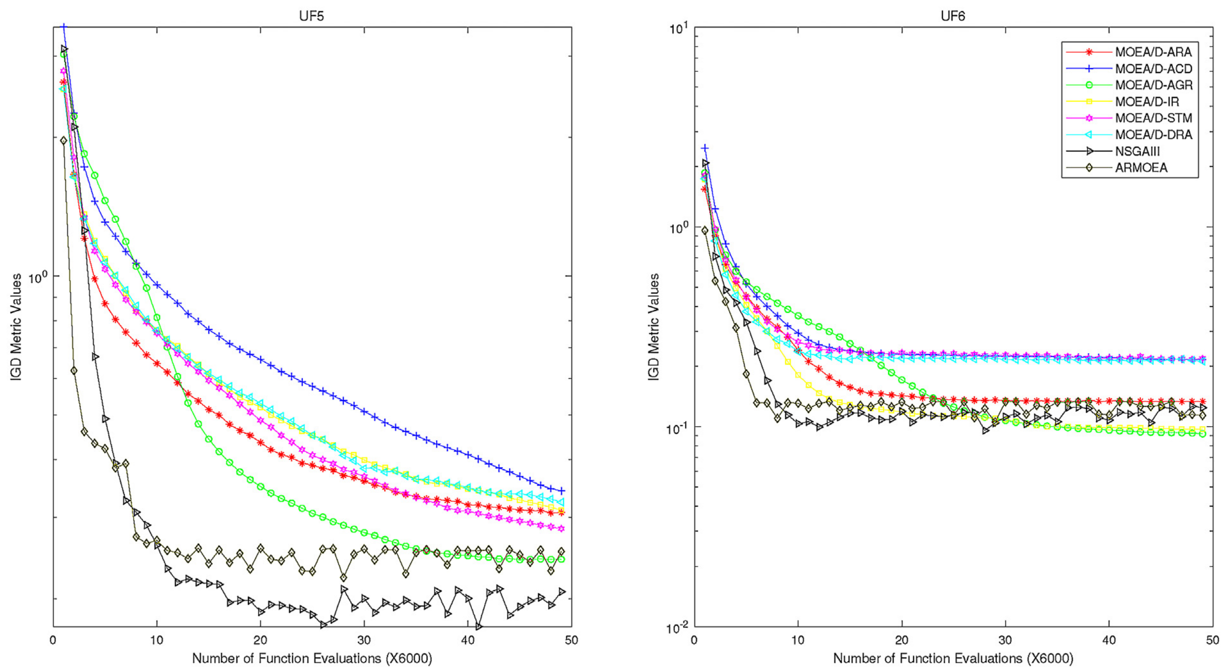

Fig. 15. Evolution of the median IGD metric values versus the number of function evaluations.

where $d(x, P)$ is the Euclidean distance between the point x from P^* and the nearest member of P, and $|P^*|$ is the cardinality of P^*. When $|P^*|$ is large enough, $IGD(P, P^*)$ can measure both the uniformity and the convergence of P.

2. HV Metric [48]: Hypervolume metric measures the size of the objective space dominated by the approximated solutions P and bounded by a reference point $z^r = (z_1^r, z_2^r, \ldots, z_m^r)^T$ in the objective space that is dominated by all Pareto-optimal objective vectors. And the HV metric is computed as follows:

$$HV(P) = Leb\left(\bigcup [f_1(x), z_1^r] \times \ldots \times [f_m(x), z_m^r]\right) \quad (12)$$

where Leb() indicates the Lebesgue measure. In our experiments, z^r is set to $(1.0, 1.0)^T$ for bi-objective test instances and $(1.0, 1.0, 1.0)^T$ for tri-objective test instances.

To certain extent, both IGD and HV can measure the convergence and diversity of P simultaneously. The smaller the IGD value or the higher the HV value, the better the quality of P for approximating the entire PF. The comparison results data for the experiment is presented in the corresponding table, where the best mean metric values are highlighted in bold. Wilcoxons rank sum test at a 5% significance level is conducted to compare the significance of difference between two algorithms for statistically sound conclusions.

5. Experiment results

In this section, according to the experimental design described in Section 4, we study and compare the performance of the proposed MOEA/D-ARA algorithm with that of the MOEA/D-DRA, MOEA/D-STM, MOEA/D-IR, MOEA/D-AGR, MOEA/D-ACD, NSGAIII and ARMOEA algorithms. The experiment can be divided into four parts. Firstly, the performance of MOEA/D-ARA on the MOP1-MOP7 and UF1-UF10 instances is compared with those of the seven other algorithms in Sections 5.1 and 5.2, respectively. Then, the potential rationality of our proposed ARA are investigated by comparing this strategy with two other variants of MOEA/D-ARA in Section 5.3. Finally, the effects of different measurements are empirically studied in Section 5.4.

5.1. Performance comparisons on MOP1-MOP7 instances

MOP1-MOP7 instances are proposed benchmark problems in recent years, which are modified from the ZDT and DTLZ instances [49,50]. Some Pareto solutions to these problems are easy to get, but some others are extremely hard to find. These observations can be attributed to the fact that the complex decision-making space of these problems increases the tendency for the population to be trapped in some special areas. It has been argued in [4] that such tendency can be easily addressed by increasing the population diversity, thereby highlighting the importance of balancing convergence and diversity during the search process. Tables 1 and 2 compare the performances of all eight MOEAs on the IGD and HV metrics, respectively. MOEA/D-ARA outperforms the other MOEAs in all MOP1-MOP7 problems. Based on the Wilcoxon rank-sum test in Table 1, MOEA/D-ARA and MOEA/D-ACD do not show any differences in MOP2, MOP3 and MOP4, and MOEA/D-ARA and MOEA/D-IR do not show any differences in MOP3. Meanwhile, the Wilcoxon rank-sum tests in Table 2, It show not any significant differences between MOEA/D-ARA and MOEA/D-ACD in MOP3 and MOP4, and between MOEA/D-ARA and MOEA/D-IR in MOP3.

Figs. 7–10 show the median IGD metric for each algorithm on MOP instances along with the evolution of the number of function evaluations, which react to our proposed algorithms and contrast algorithms to optimize the search process, such as whether the optimization is slow or whether it stops at a local optimum. MOEA/D-ARA shows the best performance throughout the evolutionary process. Specifically, although the convergence rate of MOEA/D-ARA at an early stage may be slower than some algorithms, but in the end will show better results. and clearly outperforms the other MOEAs in the whole evolutionary process of MOP6 and MOP7. In terms of convergence, MOEA/D-ARA demonstrates a slow convergence rate in the early stages of MOP2 and MOP3, but shows a fast convergence rate in the middle stage. By contrast, the other MOEAs demonstrate a fast and slow convergence in the early and late stages, respectively.

Here we present the solutions of the algorithms in two more difficult problems, which reflects the convergence and diversity of our algorithm and the comparison algorithm in the final solutions of these two tests. Fig. 11 shows the distribution of the final solutions with the lowest IGD-metric values found by MOEA/D-ARA, MOEA/D-ACD, MOEA/D-AGR, MOEA/D-IR, MOEA/D-STM, MOEA/D-DRA, NSGAIII and ARMOEA in 30 runs in the objective space on MOP4 which have very complex Pareto front. From Fig. 11, we can see that all the algorithms are not very close to the whole Pareto front. But the MOEA/D-ARA algorithm is the best performer, and it gets more Pareto solutions and converges better than other algorithms. Fig. 12 shows the distribution of the final solutions with the lowest IGD-metric values found by MOEA/D-ARA, MOEA/D-ACD, MOEA/D-AGR, MOEA/D-IR, MOEA/D-STM, MOEA/D-DRA, NSGAIII and ARMOEA in 30 runs in the objective space on MOP7. From Fig. 12 we can see that only MOEA/D-ARA can approximate the entire PF well in these comparison algorithms. The convergence of the algorithms MOEA/D-ACD and MOEA/D-IR is not enough, and many solutions are located far away from the PF. Meanwhile, the distributions of the algorithms MOEA/D-AGR, MOEA/D-DRA, MOEA/STM, NSGAIII and ARMOEA are not enough, and almost all of solutions are centered on the edge of the PF.

5.2. Performance comparisons on UF1-UF10 instances

Tables 3 and 4 compare the IGD and HV metric values of MOEA/D-ARA with those of the other seven MOEAs, respectively. The experimental results clearly identify MOEA/D-ARA as a promising solution to UF1-UF10 instances. The Wilcoxon rank-sum tests reveal that MOEA/D-ARA achieves better results than the other MOEAs in most cases. Table 3 shows that, under the IGD metric, MOEA/D-ARA outperforms the other MOEAs on UF1, UF2, UF7, UF8, UF9 and UF10. On the UF3 and UF6, the MOEA/D-ARA did not achieve the best performance, but it is only in the best algorithm performance behind in the statistical significance of the wilcoxons rank sum test. Table 4 shows that MOEA/D-ARA outperforms MOEA/D-AGR, MOEA/D-IR, MOEA/D-ACD, MOEA/D-DRA, MOEA/D-DE on the instances UF1, UF2, UF7, UF8 and UF9. Although the MOEA/D-ARA did not achieve the best performance on instance UF10, but it has no statistically significant difference with the best. From these two tables we can find that our algorithm is deficient in the case of non-continuous Pareto.

Figs. 13–17 show the median IGD metric for each algorithm on UF1-UF10 instances along with the evolution of the number of function evaluations, which react to our proposed algorithms and contrast algorithms to optimize the search process, such as whether the optimization is slow or whether it stops at a local optimum. Although the MOEA/D-ARA algorithm converges slowly at an early stage on instances UF1, UF2, UF7, UF8, UF9 and UF10, it eventually achieves the best results. On the instances UF3 and UF6 also achieved good results.

5.3. Performance comparisons with other variants

To further investigate the potential rationality of the ARA strategy. we have extracted two variants from this strategy with each variant having a unique choice of operation.

Here we present two variants, one is Variant-I, in which each solution randomly selects a subproblem as an update in all subproblems. In this case, the strategy proposed in this paper is useless. By contrast, the other is Variant-II which uses a fixed regional division rather than a dynamic one where in each solution must first determine the subproblem in the solution area before the value of the function can be replaced. Using the same parameter settings as in Section 4.3, these two variants have been experimentally compared with MOEA/D-ARA on MOP1-MOP7 and UF1-UF10 test instances. Here we use a non-overlapping fixed regional division for Variant-II. Tables 5 and 6 show the experimental results for the IGD and HV measurements, respectively. Variants-I behave badly because of the very low efficiency of the random selection operations. As a result, many favorable solutions are abandoned and many computing resources are wasted. In the 17 instances, Variant-II shows a poorer performance than MOEA/D-ARA except in the three instances of UF5, UF6 and UF9 in Table 5. Such poor performance may be attributed to the use of a fixed regional division, so that the subproblems are not synchronized leads some subproblems to slow convergence. Moreover, the parameters must be adjusted accordingly before dividing the area in Variant-II.

Table 5
Performance comparisons of MOEA/D-ARA and two variants of IGD values.

	MOEA/D-ARA	Variant-I	Variant-II
MOP1	**1.661E−02**(1.30E−03)	3.334E−01(5.04E−02)	1.899E−02(1.74E−03)
MOP2	**2.073E−02**(4.45E−02)	2.449E−01(5.17E−02)	6.230E−02(9.58E−02)
MOP3	**4.005E−03**(6.09E−03)	1.385E−01(2.80E−02)	2.016E−02(3.31E−02)
MOP4	**1.649E−02**(1.48E−02)	3.081E−01(2.75E−02)	2.008E−02(1.76E−02)
MOP5	**1.423E−02**(9.95E−04)	3.147E−01(1.51E−02)	1.594E−02(1.49E−03)
MOP6	**3.181E−02**(8.15E−04)	3.059E−01(3.52E−06)	4.101E−02(9.21E−04)
MOP7	**6.198E−02**(2.45E−03)	3.510E−01(5.37E−06)	9.085E−02(3.82E−03)
UF1	**1.644E−03**(8.18E−05)	5.276E−02(4.85E−03)	1.893E−03(1.66E−04)
UF2	**3.033E−03**(1.02E−03)	4.378E−02(8.15E−03)	5.699E−03(1.12E−03)
UF3	**4.248E−03**(2.62E−03)	1.760E−01(4.47E−02)	5.513E−03(4.79E−03)
UF4	**5.457E−02**(2.48E−03)	8.268E−02(8.77E−03)	5.761E−02(4.05E−03)
UF5	3.054E−01(1.43E−01)	6.208E−01(1.22E−01)	**2.602E−01**(2.53E−02)
UF6	1.333E−01(1.40E−01)	4.957E−01(2.01E−01)	**1.054E−01**(1.11E−01)
UF7	**1.826E−03**(1.86E−04)	4.177E−02(5.41E−03)	2.074E−03(1.42E−04)
UF8	**2.465E−02**(4.16E−04)	1.549E−01(3.30E−02)	2.904E−02(1.44E−03)
UF9	4.997E−02(4.86E−02)	1.905E−01(1.05E−02)	**4.605E−02**(4.18E−02)
UF10	**2.859E−01**(8.45E−02)	9.564E−01(3.42E−01)	4.419E−01(7.79E−02)

The numbers are the mean for run 30 times independently and the numbers in parentheses are the standard deviations.

Table 6
Performance comparisons of MOEA/D-ARA and two variants of HV values.

	MOEA/D-ARA	Variant-I	Variant-II
MOP1	**6.436E−01**(1.69E−03)	1.331E−01(9.32E−02)	6.401E−01(2.15E−03)
MOP2	**3.105E−01**(4.50E−02)	5.448E−02(3.64E−02)	2.549E−01(1.19E−01)
MOP3	**2.085E−01**(9.31E−03)	9.171E−02(2.28E−02)	1.888E−01(3.57E−02)
MOP4	**4.980E−01**(1.94E−02)	1.349E−01(1.79E−02)	4.933E−01(2.39E−02)
MOP5	**6.458E−01**(1.20E−03)	3.536E−01(5.55E−17)	6.429E−01(2.00E−03)
MOP6	**7.924E−01**(9.11E−04)	4.942E−01(4.84E−05)	7.812E−01(1.13E−03)
MOP7	**4.046E−01**(2.40E−03)	2.124E−01(4.82E−05)	3.763E−01(2.95E−03)
UF1	**6.638E−01**(2.26E−04)	5.776E−01(8.75E−03)	6.634E−01(2.33E−04)
UF2	**6.620E−01**(1.28E−03)	6.039E−01(6.23E−03)	6.582E−01(1.77E−03)
UF3	**6.576E−01**(5.91E−03)	3.782E−01(6.28E−02)	6.558E−01(8.99E−03)
UF4	**2.530E−01**(3.58E−03)	2.149E−01(9.90E−03)	2.502E−01(4.71E−03)
UF5	**5.352E−02**(7.96E−02)	1.341E−03(7.22E−03)	1.974E−02(4.93E−02)
UF6	**2.044E−01**(6.91E−02)	7.007E−02(7.85E−02)	1.987E−01(9.03E−02)
UF7	**4.967E−01**(3.28E−04)	4.278E−01(1.05E−02)	4.962E−01(2.49E−04)
UF8	**4.373E−01**(1.27E−03)	2.702E−01(3.49E−02)	4.259E−01(4.04E−03)
UF9	**7.254E−01**(6.95E−02)	5.440E−01(3.19E−03)	7.244E−01(5.81E−02)
UF10	**1.159E−01**(5.63E−02)	6.122E−03(1.87E−02)	4.470E−02(2.64E−02)

The numbers are the mean for run 30 times independently and the numbers in parentheses are the standard deviations.

5.4. Effects of different measurements

Based on the distance measurements, we propose a variant that uses the Euclidean distance measurement of the solution point-to-subproblem vector to replace the angle measurement of the solution and subproblem vectors. Using the same parameter settings described in Section 5.3, this variant has been experimentally compared with the MOEA/D-ARA on the MOP1-MOP7 and UF1-UF10 test instances. Tables 7 and 8 show the experimental results for the IGD and HV measurements, respectively. In the 17 instances, Variant-III shows a poorer performance than MOEA/D-ARA except in the three instances of UF5, UF6 and UF9 and in the UF6 and UF9 two instances shown in Table 7 respectively. Such performance has been mainly attributed to the fact that the measurement in

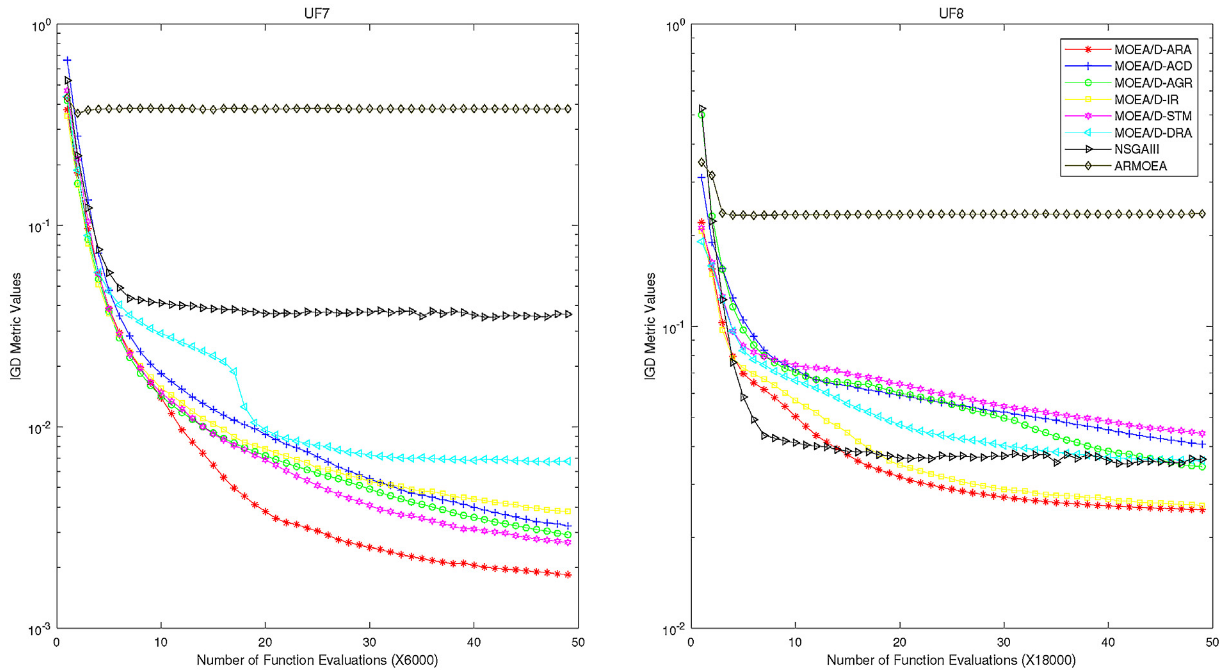

Fig. 16. Evolution of the median IGD metric values versus the number of function evaluations.

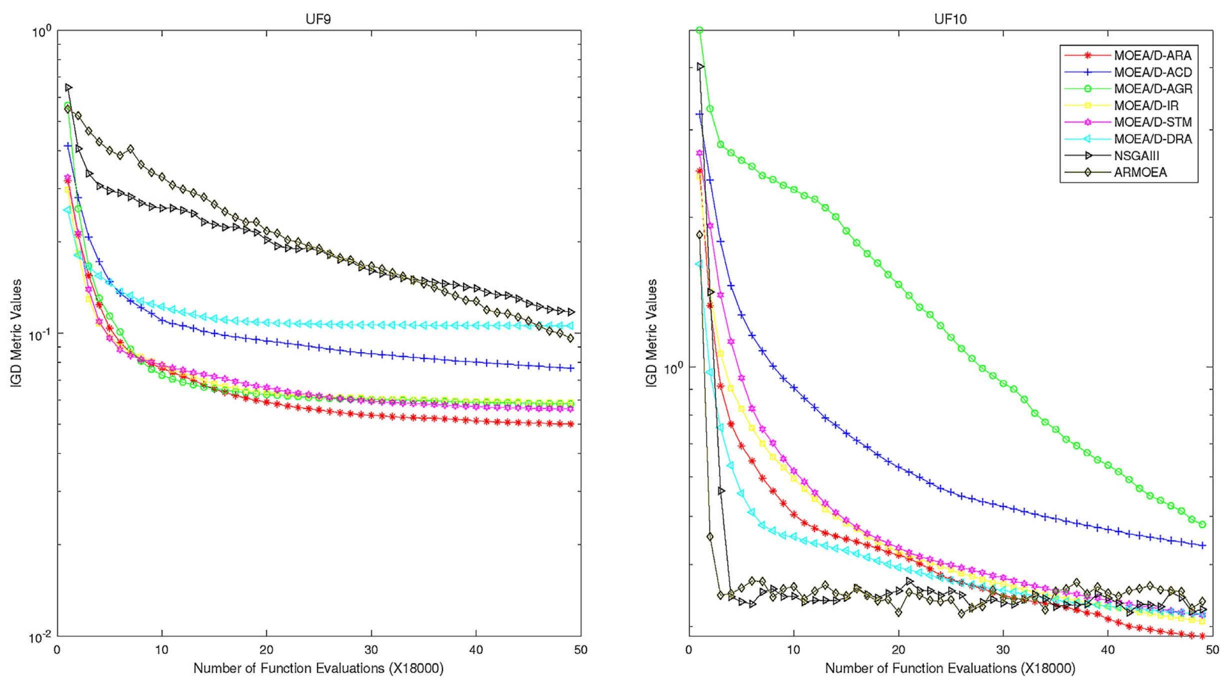

Fig. 17. Evolution of the median IGD metric values versus the number of function evaluations.

variant III leads to overlaps of the solution region of subproblems, which may result in similar or identical solutions for different subproblems, thereby reducing the diversity of these solutions and increasing their tendency to fall into the local optima.

6. Conclusion

In MOEA/D, MOP is decomposed into a number of scalar optimization subproblems by using decomposition approaches and then optimizes them simultaneously. In the evolutionary process, diversity and convergence are required at the same time is necessary. It is therefore advisable to balance as much as possible when designing a selection and replacing strategy. This paper presents a simple yet effective attempt along this direction. To guarantee diversity, we force different subproblems to correspond to different solutions in the region division process. On this basis, we proposed ARA strategy to adjust solutions of different subproblems and to ensure the balance between their diversity and convergence. We find that the convergence will not be weakened when diversity is guaranteed. Extensive experimental studies have been conducted to compare our proposed MOEA/D-ARA with other peer MOEAs

Table 7
Performance comparisons of MOEA/D-ARA and variant-III of IGD values.

	MOEA/D-ARA	Variant-III
MOP1	**1.661E−02**(1.30E−03)	1.790E−02(1.92E−03)
MOP2	**2.073E−02**(4.45E−02)	5.464E−02(8.84E−02)
MOP3	**4.005E−03**(6.09E−03)	1.215E−02(2.97E−02)
MOP4	**1.649E−02**(1.48E−02)	2.623E−02(4.39E−02)
MOP5	**1.423E−02**(9.95E−04)	1.726E−02(7.71E−03)
MOP6	**3.181E−02**(8.15E−04)	3.407E−02(8.45E−04)
MOP7	**6.198E−02**(2.45E−03)	6.271E−02(2.31E−03)
UF1	**1.644E−03**(8.18E−05)	1.899E−03(1.83E−04)
UF2	**3.033E−03**(1.02E−03)	5.961E−03(8.40E−03)
UF3	**4.248E−03**(2.62E−03)	8.023E−03(1.39E−02)
UF4	**5.457E−02**(2.48E−03)	5.666E−02(5.05E−03)
UF5	3.054E−01(1.43E−01)	**2.767E−01**(8.45E−02)
UF6	1.333E−01(1.40E−01)	**1.092E−01**(1.46E−01)
UF7	**1.826E−03**(1.86E−04)	1.994E−03(2.01E−04)
UF8	**2.465E−02**(4.16E−04)	2.528E−02(2.53E−02)
UF9	**4.997E−02**(4.86E−02)	**3.865E−02**(3.89E−02)
UF10	**2.859E−01**(8.45E−02)	3.087E−01(7.99E−02)

The numbers are the mean for run 30 times independently and the numbers in parentheses are the standard deviations.

Table 8
Performance comparisons of MOEA/D-ARA and variant-III of HV values.

	MOEA/D-ARA	Variant-III
MOP1	**6.436E−01**(1.69E−03)	6.414E−01(2.44E−03)
MOP2	**3.105E−01**(4.50E−02)	2.672E−01(1.07E−01)
MOP3	**2.085E−01**(9.31E−03)	2.005E−01(2.97E−02)
MOP4	**4.980E−01**(1.94E−02)	4.855E−01(6.03E−02)
MOP5	**6.458E−01**(1.20E−03)	6.417E−01(7.74E−03)
MOP6	**7.924E−01**(9.11E−04)	7.896E−01(9.03E−04)
MOP7	**4.046E−01**(2.40E−03)	3.989E−01(3.03E−03)
UF1	**6.638E−01**(2.26E−04)	6.635E−01(2.29E−04)
UF2	**6.620E−01**(1.28E−03)	6.591E−01(7.97E−03)
UF3	**6.576E−01**(5.91E−03)	6.557E−01(1.03E−02)
UF4	**2.530E−01**(3.58E−03)	2.515E−01(6.26E−03)
UF5	**5.352E−02**(7.96E−02)	1.256E−02(4.26E−02)
UF6	2.044E−01(6.91E−02)	**2.124E−01**(7.93E−02)
UF7	**4.967E−01**(3.28E−04)	4.965E−01(3.55E−04)
UF8	**4.373E−01**(1.27E−03)	4.358E−01(1.05E−03)
UF9	7.254E−01(6.95E−02)	**7.408E−01**(5.55E−02)
UF10	**1.159E−01**(5.63E−02)	1.146E−01(4.93E−02)

The numbers are the mean for run 30 times independently and the numbers in parentheses are the standard deviations.

as well as to study their evolutionary processes and performances under different selection strategies and measures.

In the future, we will study some of the follow-up studies as follows.

1. The many-objective optimization problem and complex PF problem have become major concerns in evolutionary multiobjective optimization community [6]. It is very interested in extending the algorithm proposed in this paper to these topics.
2. Our studies show that the balance of convergence of different subproblems can benefit the evolution process. On this basis, we will conduct further studies on the balance of convergence of different subproblems.
3. It is important to balance the population diversity and convergence in MOEA, especially when different MOPs require a balance of inconsistencies. The future we will study how to intelligently choose a different balance according to different MOPs, which should be very important for applying machine learning algorithm.

Acknowledgements

This study is supported by the Program for New Century Excellent Talents in university (Grant No.NCET-10-0365), National Nature Science Foundation of China (Grant Number: 11171369, 61272395, 61370171,61300128, 61472127, 61572178, 61672214,61672223 and 61772192).

References

[1] A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A survey of multiobjective evolutionary algorithms based on decomposition, IEEE Trans. Evol. Comput. 21 (3) (2017) 440–462, http://dx.doi.org/10.1109/TEVC.2016.2608507.
[2] A. Zhou, B. Qu, H. Li, S. Zhao, P.N. Suganthan, Q. Zhang, Multiobjective evolutionary algorithms: a survey of the state of the art, Swarm Evol. Comput. 1 (1) (2011) 32–49, http://dx.doi.org/10.1016/j.swevo.2011.03.001.
[3] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer, Boston, MA, USA, 1999.
[4] H. Liu, F. Gu, Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, IEEE Trans. Evol. Comput. 18 (3) (2014) 450–455, http://dx.doi.org/10.1109/TEVC.2013.2281533.
[5] I. Giagkiozis, P.J. Fleming, Methods for multi-objective optimization: an analysis, Inf. Sci. 293 (2015) 338–350, http://dx.doi.org/10.1016/j.ins.2014.08.071.
[6] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (4) (2014) 577–601.
[7] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evol. Comput. 20 (5) (2016) 773–791, http://dx.doi.org/10.1109/TEVC.2016.2519378.
[8] R. Wang, J. Xiong, H. Ishibuchi, G. Wu, T. Zhang, On the effect of reference point in MOEA/D for multi-objective optimization, Appl. Soft Comput. 58 (2017) 25–34, http://dx.doi.org/10.1016/j.asoc.2017.04.002.
[9] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731, http://dx.doi.org/10.1109/TEVC.2007.892759.
[10] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.
[11] S. Jiang, S. Yang, A strength pareto evolutionary algorithm based on reference direction for multiobjective and many-objective optimization, IEEE Trans. Evol. Comput. 21 (3) (2017) 329–346, http://dx.doi.org/10.1109/TEVC.2016.2592479.
[12] N. Beume, B. Naujoks, M.T.M. Emmerich, SMS-EMOA: multiobjective selection based on dominated hypervolume, Eur. J. Oper. Res. 181 (3) (2007) 1653–1669, http://dx.doi.org/10.1016/j.ejor.2006.08.008.
[13] J. Bader, E. Zitzler, Hype: an algorithm for fast hypervolume-based many-objective optimization, Evol. Comput. 19 (1) (2011) 45–76.
[14] H. Wang, Y. Jin, X. Yao, Diversity assessment in many-objective optimization, IEEE Trans. Cybern. 47 (6) (2017) 1510–1522.
[15] Y. Tian, R. Cheng, X. Zhang, F. Cheng, Y. Jin, An indicator based multi-objective evolutionary algorithm with reference point adaptation for better versatility, IEEE Trans. Evol. Comput. PP (99) (2017), http://dx.doi.org/10.1109/TEVC.2017.2749619, 1–1.
[16] W.K. Mashwani, A. Salhi, Multiobjective memetic algorithm based on decomposition, Appl. Soft Comput. 21 (2014) 221–243, http://dx.doi.org/10.1016/j.asoc.2014.03.007.
[17] K. Michalak, The effects of asymmetric neighborhood assignment in the MOEA/D algorithm, Appl. Soft Comput. 25 (2014) 97–106, http://dx.doi.org/10.1016/j.asoc.2014.07.029.
[18] A. Zhou, Q. Zhang, Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms, IEEE Trans. Evol. Comput. 20 (1) (2016) 52–64, http://dx.doi.org/10.1109/TEVC.2015.2424251.
[19] H. Zhang, A. Zhou, S. Song, Q. Zhang, X.-Z. Gao, J. Zhang, A self-organizing multiobjective evolutionary algorithm, IEEE Trans. Evol. Comput. 20 (5) (2016) 792–806.
[20] X. Cai, Z. Yang, Z. Fan, Q. Zhang, Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and many-objective optimization, IEEE Trans. Cybern. 47 (9) (2017) 2824–2837, http://dx.doi.org/10.1109/TCYB.2016.2586191.
[21] W.K. Mashwani, A. Salhi, O. Yeniay, M.A. Jan, R.A. Khanum, Hybrid adaptive evolutionary algorithm based on decomposition, Appl. Soft Comput. 57 (2017) 363–378, http://dx.doi.org/10.1016/j.asoc.2017.04.005.
[22] J. Luo, Y. Yang, X. Li, Q. Liu, M. Chen, K. Gao, A decomposition-based multi-objective evolutionary algorithm with quality indicator, Swarm Evol. Comput. 39 (2018) 339–355, http://dx.doi.org/10.1016/j.swevo.2017.11.004.
[23] I. Giagkiozis, R.C. Purshouse, P.J. Fleming, Generalized decomposition, in: R.C. Purshouse, P.J. Fleming, C.M. Fonseca, S. Greco, J. Shaw (Eds.), Evolutionary Multi-Criterion Optimization – 7th International Conference, EMO 2013, Sheffield, UK, March 19–22, 2013. Proceedings, Vol. 7811 of Lecture Notes in Computer Science, Springer, 2013, pp. 428–442, http://dx.doi.org/10.1007/978-3-642-37140-0_33.

[24] H. Li, D. Landa-Silva, An adaptive evolutionary multi-objective approach based on simulated annealing, Evol. Comput. 19 (4) (2011) 561–595, http://dx.doi.org/10.1162/EVCO_a_00038.
[25] H. Li, Q. Zhang, Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2) (2009) 284–302, http://dx.doi.org/10.1109/TEVC.2008.925798.
[26] Z. Wang, Q. Zhang, A. Zhou, M. Gong, L. Jiao, Adaptive replacement strategies for MOEA/D, IEEE Trans. Cybern. 46 (2) (2016) 474–486, http://dx.doi.org/10.1109/TCYB.2015.2403849.
[27] K. Li, Q. Zhang, S. Kwong, M. Li, R. Wang, Stable matching-based selection in evolutionary multiobjective optimization, IEEE Trans. Evol. Comput. 18 (6) (2014) 909–923, http://dx.doi.org/10.1109/TEVC.2013.2293776.
[28] K. Li, S. Kwong, Q. Zhang, K. Deb, Interrelationship-based selection for decomposition multiobjective optimization, IEEE Trans. Cybern. 45 (10) (2015) 2076–2088, http://dx.doi.org/10.1109/TCYB.2014.2365354.
[29] L. Wang, Q. Zhang, A. Zhou, M. Gong, L. Jiao, Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm, IEEE Trans. Evol. Comput. 20 (3) (2016) 475–480, http://dx.doi.org/10.1109/TEVC.2015.2457616.
[30] S.B. Gee, K.C. Tan, V.A. Shim, N.R. Pal, Online diversity assessment in evolutionary multiobjective optimization: a geometrical perspective, IEEE Trans. Evol. Comput. 19 (4) (2015) 542–559, http://dx.doi.org/10.1109/TEVC.2014.2353672.
[31] M. Wu, K. Li, S. Kwong, Y. Zhou, Q. Zhang, Matching-based selection with incomplete lists for decomposition multiobjective optimization, IEEE Trans. Evol. Comput. 21 (4) (2017) 554–568, http://dx.doi.org/10.1109/TEVC.2017.2656922.
[32] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, J. Wu, MOEA/D with adaptive weight adjustment, Evol. Comput. 22 (2) (2014) 231–264, http://dx.doi.org/10.1162/EVCO_a_00109.
[33] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization, in: Proc. IEEE Congr. Evol. Comput. (CEC), Trondheim, Norway, 2009, pp. 2508–2515, http://dx.doi.org/10.1109/CEC.2009.4983256.
[34] H. Ishibuchi, N. Akedo, Y. Nojima, Relation between neighborhood size and MOEA/D performance on many-objective problems, in: Evolutionary Multi-Criterion Optimization (EMO) (LNCS 7811), Springer, Berlin, Germany, 2013, pp. 459–474.
[35] S. Jiang, S. Yang, An improved multiobjective optimization evolutionary algorithm based on decomposition for complex pareto fronts, IEEE Trans. Cybern. 46 (2) (2016) 421–437, http://dx.doi.org/10.1109/TCYB.2015.2403131.
[36] A. Moraglio, J. Togelius, S. Silva, Geometric differential evolution for combinatorial and programs spaces, Evol. Comput. 21 (4) (2013) 591–624, http://dx.doi.org/10.1162/EVCO_a_00099.
[37] L. Ke, Q. Zhang, R. Battiti, MOEA/D-ACO: a multiobjective evolutionary algorithm using decomposition and antcolony, IEEE Trans. Cybern. 43 (6) (2013) 1845–1859, http://dx.doi.org/10.1109/TSMCB.2012.2231860.
[38] Y. Yuan, H. Xu, B. Wang, B. Zhang, X. Yao, Balancing convergence and diversity in decomposition-based many-objective optimizers, IEEE Trans. Evol. Comput. 20 (2) (2016) 180–198, http://dx.doi.org/10.1109/TEVC.2015.2443001.
[39] L.L.Z.W. Wanliang Wang, Senliang Ying, W. Li, An improved decomposition-based multiobjective evolutionary algorithm with a better balance of convergence and diversity, Appl. Soft Comput. 57 (2017) 627–641.
[40] Q. Zhang, W. Liu, H. Li, The performance of a new version of MOEA/D on cec09 unconstrained mop test instances, Proc. IEEE Congr. Evol. Comput. (CEC) (2009) 203–208, http://dx.doi.org/10.1109/CEC.2009.4982949.
[41] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim. 8 (3) (1998) 631–657.
[42] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31, http://dx.doi.org/10.1109/TEVC.2010.2059031.
[43] K. Deb, M. Goyal, A combined genetic adaptive search (geneas) for engineering design, Comput. Sci. Informat 26 (4) (1996) 30–45.
[44] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable test problems for evolutionary multiobjective optimization, Evolutionary Multiobjective Optimization, Theor. Adv. Appl. (2005) 105–145.
[45] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V.G. da Fonseca, Performance assessment of multiobjective optimizers: an analysis and review, IEEE Trans. Evol. Comput. 7 (2) (2003) 117–132, http://dx.doi.org/10.1109/TEVC.2003.810758.
[46] O. Schutze, X. Esquivel, A. Lara, C.A.C. Coello, Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization, IEEE Trans. Evol. Comput. 16 (4) (2012) 504–522, http://dx.doi.org/10.1109/TEVC.2011.2161872.
[47] Q. Zhang, A. Zhou, Y. Jin, A. RM-MED, A regularity model-based multiobjective estimation of distribution algorithm, IEEE Trans. Evol. Comput. 12 (1) (2008) 41–63, http://dx.doi.org/10.1109/TEVC.2007.894202.
[48] E. Zitzler, L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach, IEEE Trans. Evol. Comput. 3 (4) (1999) 257–271, http://dx.doi.org/10.1109/4235.797969.
[49] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evol. Comput. 8 (2) (2000) 173–195, http://dx.doi.org/10.1162/106365600568202.
[50] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Trans. Evol. Comput. 10 (5) (2006) 477–506, http://dx.doi.org/10.1109/TEVC.2005.861417.