A Diversity-Enhanced Resource Allocation Strategy for Decomposition-Based Multiobjective Evolutionary Algorithm

Qiuzhen Lin, Genmiao Jin, Yueping Ma, Ka-Chun Wong, Carlos A. Coello Coello, *Fellow, IEEE*, Jianqiang Li, Jianyong Chen, *Member, IEEE*, and Jun Zhang, *Fellow, IEEE*

***Abstract*—The multiobjective evolutionary algorithm (MOEA) based on decomposition transforms a multiobjective optimization problem into a set of aggregated subproblems and then optimizes them collaboratively. Since these subproblems usually have different degrees of difficulty, resource allocation (RA) strategies have been reported to enhance performance, attempting to dynamically assign proper amounts of computational resources for the solution of each of these subproblems. However, existing schemes for decomposition-based MOEAs fully rely on the relative improvement of the aggregated functions to do this. This paper proposes a diversity-enhanced RA strategy for this kind of MOEA, depending on both relative improvement on aggregated function value and solution density around each subproblem to assign computational resources. Thus, one subproblem surrounded with fewer solutions in its neighboring area and more relative improvement on the aggregated function value will be allocated a higher probability for evolution. Our experimental results show the advantages of our proposed strategy over two popular RA strategies available for decomposition-based MOEAs, on tackling a set of complicated benchmark problems.**

***Index Terms*—Decomposition, multiobjective optimization, resource allocation (RA), solution density.**

I. Introduction

MULTIOBJECTIVE optimization problems (MOPs) widely arise in many application fields, such as economics [1], [2] and engineering design [3], [4]. Mathematically, an MOP without any constraint can be modeled as follows:

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

where $x = (x_1, \ldots, x_n)$ is a decision variables vector (n is the number of decision variables), $\Omega = [l_i, u_i]^n$ is the decision space (l_i and u_i are, respectively, the lower and upper bounds for the ith variable, $i \in [1, n]$), and $F : \Omega \to R^m$ defines m objective functions (R^m is the objective space).

Let $x_1, x_2 \in \Omega$, x_1 is said to dominate x_2, denoted by $x_1 \prec x_2$, if and only if $f_i(x_1) \le f_i(x_2)$ for $\forall i \in \{1, \ldots, m\}$ and $f_j(x_1) \ne f_j(x_2)$ for $\exists j \in \{1, \ldots, m\}$. A solution $x^* \in \Omega$ is said to be a Pareto-optimal or nondominated solution when no other solution $x \in \Omega$ can dominate x^*. The set of all Pareto-optimal (or nondominated) solutions composes the Pareto-optimal set (PS), and its corresponding set of objective function values is called the Pareto-optimal front (PF) [5]. Due to the conflicts among the objectives, no single solution is able to optimize them all at the same time. Therefore, we aim for the best possible tradeoffs among all the objectives.

Multiobjective evolutionary algorithms (MOEAs) have been found to be an effective and efficient tool for solving MOPs [6]. During the last decades, many competitive MOEAs have been designed [7]–[19]. According to the selection criteria, most MOEAs can be generally classified into three categories, i.e., Pareto domination-based MOEAs [7]–[10], indicator-based MOEAs [11]–[15], and decomposition-based MOEAs [16]–[19]. Particularly, since the publication of MOEA/D [16], decomposition-based MOEAs have become a very popular evolutionary framework for tackling MOPs. In this approach, an MOP is decomposed into a set of aggregated subproblems and then each subproblem is optimized on a collaborative manner. Its evolutionary framework has triggered a considerable amount of research [20]–[26].

There are several primary components in MOEA/D, e.g., weight vector generation, neighbor selection, subproblem selection, evolutionary operators, and population update. These components are frequently studied and have been enhanced by many MOEA/D variants. Regarding the weight vector generation, UMOEA/D [27] and MOEA/D-UDM [28] were proposed to produce the weight vectors with uniform

Manuscript received September 16, 2016; revised January 7, 2017, March 19, 2017, and May 24, 2017; accepted July 30, 2017. Date of publication September 6, 2017; date of current version July 17, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61402291 and Grant 61672358, in part by CONACyT under Grant 221551, and in part by the Early Career Scheme Grant from Research Grant Council under CityU Project 9048072 and under RGC Project 21200816. This paper was recommended by Associate Editor T. Ray. *(Corresponding author: Jianyong Chen.)*

Q. Lin, G. Jin, Y. Ma, J. Li, and J. Chen are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China (e-mail: jychen@szu.edu.cn).

K.-C. Wong is with the Department of Computer Science, City University of Hong Kong, Hong Kong.

C. A. Coello Coello is with the Department of Computer Science, CINVESTAV-IPN (Evolutionary Computation Group), Mexico City 07300, Mexico.

J. Zhang is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China.

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. The file contains figures and tables relevant to the paper. The material is 0.586 MB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCYB.2017.2739185

distribution and an arbitrary number of weight vectors can be generated to fit the population size. However, as pointed out in [29], the uniformly distributed weight vectors cannot guarantee to produce solutions with a uniform distribution. Thus, an evolutionary search strategy using uniformly distributed directions was accordingly proposed to solve the above problem [29]. Moreover, to fit the geometrical information of the PF, two adaptive control strategies for generating weight vectors were designed in [30] and [31], where the number of weight vectors is dynamically adjusted. For the neighbor selection mechanism, due to the significant impact of the neighborhood sizes (NSs) on MOEA/D, an adaptive selection strategy was presented in ENS-MOEA/D [32] to adaptively choose the preferred value of NSs based on the former successful experience. This way, different MOPs can be better solved using certain setting of NSs. On the subproblem selection, this can be modeled as a resource allocation (RA) strategy to assign computational resources for the subproblems. A dynamic RA (DRA) strategy was designed in MOEA/D-DRA [33]. This approach computes a utility function for each subproblem based on the improvement of the aggregated function values during the last evolutionary period, and then dynamically selects the subproblems for evolution based on this utility function. A generalized RA (GRA) strategy was further designed in MOEA/D-GRA [34]. This approach introduces a probability of improvement (PoI) vector for the subproblems and then assigns the computational resources according to this PoI vector. With respect to the evolutionary operators, differential evolution (DE) was introduced in MOEA/D-DE [35] to substitute simulated binary crossover (SBX). This way, the exploration capability of MOEA/D-DE is significantly enhanced, especially in solving some complicated MOPs with variable dependencies. Four DE search strategies were further merged in MOEA/D-FRRMAB [36] and a bandit-based adaptive operator selection strategy was designed to determine the application rates of different DE strategies on an online manner. Similarly, three DE strategies were also adopted in ADEMO/D [37] and four DE composite operator pools were used in MOEA/D-CDE [38], attempting to adaptively select the preferred DE operators according to the quality of historically found solutions. At last, considering the population update in MOEA/D, a stable matching (STM) model was reported in MOEA/D-STM [39] to match the subproblems and the solutions. It is the first generational version of MOEA/D and it also constitutes the first attempt to incorporate matching theory (a concept from economics) into the design of MOEAs. This approach struggles to balance the convergence and the population diversity during the evolutionary process. More recently, an inter-relationship (IR) model was built in MOEA/D-IR [40] based on the mutual-preferences of the subproblems and the solutions. Essentially, this IR model is a diversity first and convergence second strategy, which is different from the STM model that tries to maintain the balance of convergence and diversity. In [41], the replacement NS was also shown to be critical for population update, and an approach for dynamically adjusting this size was presented. Thus, it can spend much effort in maintaining the population diversity at the early stages of the search process and in speeding up convergence at the later phases of the search. Moreover, a dominance-based selection approach was further studied to be combined with a decomposition-based approach in MOEA/DD [42]. This approach aims to balance the convergence and the diversity when solving many-objective optimization problems. A systematic approach was proposed to generate widely spread weight vectors for a high-dimensional objective space and a mating restriction scheme was designed to fully exploit the mating parents chosen from the neighboring subregions.

This paper mainly concentrates on the subproblem selection scheme and designs a diversity-enhanced RA strategy for MOEA/D. As mentioned above, the RA strategies for MOEA/D have already been studied in MOEA/D-DRA and MOEA/D-GRA; however, they are completely dependent on the relative improvement of aggregated function values when computing the utility function in MOEA/D-DRA and the PoI vector in MOEA/D-GRA. These approaches only exploit the convergence status to allocate the computational resources for the subproblems. In our opinion, besides the convergence status, the diversity among the subproblems is also an important indicator to design an effective and efficient RA strategy [43]–[46]. For example, in some cases, the subproblems surrounded with numerous solutions should not be assigned too many computational resources, even though they may show significant improvement rates on aggregated function values. Therefore, an improved (diversity-enhanced) RA (IRA) strategy is proposed to consider both convergence (relative improvement on aggregated function values) and diversity (the solution density around the subproblem) of each subproblem. This approach can reasonably balance the convergence and the diversity for each subproblem when running the RA strategy. After embedding this IRA strategy into the framework of MOEA/D, a novel MOEA/D variant named MOEA/D-IRA is presented. Some complicated test MOPs are used to test the performance of MOEA/D-IRA, and the experimental results indicate that MOEA/D-IRA outperforms MOEA/D-DE [35], two MOEA/D variants with an RA strategy (MOEA/D-DRA [33] and MOEA/D-GRA [34]), and one recently proposed MOEA/D variant (MOEA/D-IR [40]).

The rest of this paper is organized as follows. In Section II, we introduce some background knowledge including the decomposition approach of MOEA/D and the two RA strategies used in MOEA/D-DRA and MOEA/D-GRA. The details of our proposed algorithm MOEA/D-IRA are described in Section III, where our IRA strategy is introduced. All the experimental studies are presented in Section IV, including the parameters settings of the compared algorithms, the comparison of the results of our algorithm with respect to those of several MOEA/D variants, and the parameter sensitivity analysis in our IRA strategy. Finally, this paper is concluded in Section V with some future research topics.

II. BACKGROUND

A. Decomposition Approach

In MOEA/D, a decomposition approach is used to transform an MOP into a number of single-objective optimization subproblems and then they can be optimized collaboratively to attain the entire PF. Several decomposition approaches are commonly applied in many MOEA/D variants, such as the weighted sum approach, the Tchebycheff approach, and the boundary intersection method [16]. In this paper, we use the Tchebycheff approach to construct the aggregated functions, since this scheme is used in many MOEA/D variants [32], [33], [36]. The Tchebycheff approach can be defined as follows:

$$\min_{x\in\Omega} g^{\mathrm{tch}}(x|\lambda, z^*) = \max_{1\le i\le m}\left\{|f_i(x) - z_i^*|/\lambda_i\right\} \tag{2}$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)$ is the weight vector (also the direction vector) with $\lambda_i \ge 0 (i \in \{1, \ldots, m\})$ and $\sum_{i=1}^{m} \lambda_i = 1$. In practice, λ_i is set to a very small number (e.g., 10^{-6}), in case $\lambda_i = 0$. $z^* = \{z_1^*, \ldots, z_m^*\}$ is the ideal point with $z_i^* = \{\min f_i(x)|x \in \Omega\}$ for each $i = 1, \ldots, m$. For a solution x^* in PF of (1), there exists a weight vector λ that satisfies that x^* is also the optimal solution of the subproblem in (2), such that, using a set of uniformly distributed weight vectors, the optimal solutions for (2) can compose a number of Pareto-optimal solutions for (1).

B. Resource Allocation Strategies for MOEA/D

In the original MOEA/D [16], all the subproblems are treated equally and assigned with the same amounts of computational resources. This equal assignment strategy is not suitable for all kinds of MOPs, as the subproblems decomposed from various MOPs may have different difficulties. The RA strategy in MOEA/D can alleviate the above problem, e.g., MOEA/D-DRA [33] and MOEA/D-GRA [34] were designed following this direction. They are introduced below, respectively.

1) MOEA/D-DRA: A DRA strategy was designed in MOEA/D-DRA [33], which assigns the computational resources according to the relative improvement of the aggregated function values for each subproblem. In this approach, the subproblems with high improvement rates in the previous search phase will be allocated with more computational resources, as this indicates that these subproblems can be easily enhanced and can be further optimized. Otherwise, subproblems which are found to have low improvement rates in the previous search phase, may be very hard to improve and, therefore, receive less computational resources. To achieve the above purpose, a utility function π^i is computed in MOEA/D-DRA for the ith subproblem, as follows:

$$\pi^i = \begin{cases} 1 & \text{if } \Delta^i > 0.001 \\ \left(0.95 + 0.05 \times \frac{\Delta^i}{0.001}\right) \times \pi^i & \text{otherwise} \end{cases} \tag{3}$$

where Δ^i is the relative improvement of the objective function value in subproblem i, which is defined as

$$\Delta^i = \frac{g^{\mathrm{tch}}\left(x_{t-\Delta t}^i|\lambda^i, z^*\right) - g^{\mathrm{tch}}\left(x_t^i|\lambda^i, z^*\right)}{g^{\mathrm{tch}}\left(x_{t-\Delta t}^i|\lambda, z^*\right)} \tag{4}$$

where t is the current generation, Δt is the updating period, and $g^{\mathrm{tch}}(\cdot)$ is the decomposition approach as introduced in (2). In (4), the utility function π^i is updated with the period of Δt generations. $x_{t-\Delta t}^i$ is the solution of ith subproblem before Δt generations and x_t^i is the one at current generation t. Initially, π^i is set to 1, and then, if the computed value of Δ^i is smaller than 0.001, it indicates that the subproblem is hard to be enhanced. Thus, the value of π^i will be reduced in order to save computational resources.

To pick a set I of subproblems for evolution, MOEA/D-DRA first selects m indexes of the subproblems whose objectives are, respectively, m objectives f_i in order to form an initial set I, and then other $\lfloor N/5 \rfloor - m$ subproblems (N is the number of weight vectors) are selected into I by using 10-tournament selection based on π^i.

2) MOEA/D-GRA: Following the work of MOEA/D-DRA, a GRA strategy was designed in MOEA/D-GRA [34]. In this approach, a PoI vector is maintained and each subproblem is uniquely associated with a PoI element. A larger value of PoI indicates a higher probability that the corresponding subproblem will be selected to be further improved. That is to say, at each generation, the subproblems are selected using the probabilities in this PoI vector. This way, the computational resources can be assigned to the subproblems with high PoI values. The PoI vector is updated as follows:

$$p^i = \frac{\Delta^i + \varepsilon}{\max_{j=1,\ldots,N}\left\{\Delta^j\right\} + \varepsilon} \tag{5}$$

where $i = 1, \ldots, N$, Δ^i is defined in (4), and ε is a small value to avoid the numerator or denominator to be zero. It is worth noting that once none of the subproblems can be further improved during the previous Δt generation, i.e., $\max\{\Delta^j\} = 0 (j = 1, \ldots, N)$, this PoI vector will be reinitialized with $p^i = 1$ for $i = 1, \ldots, N$, so that all the subproblems will have an equal probability of being selected for being evolved.

In MOEA/D-GRA, when selecting the subproblems for evolution, uniformly distributed random values in the range [0, 1] are generated to compare with the probabilities in the PoI vector. That is to say, if the probability p^i in the PoI vector is larger than the uniformly distributed random number in [0, 1], the corresponding subproblem i will be selected for evolution at this generation. During the evolution, this PoI vector is updated with a period of Δt generations using (5).

C. Short Discussion of MOEA/D-DRA and MOEA/D-GRA

The RA strategies in MOEA/D-DRA and MOEA/D-GRA were all designed based on the relative improvement Δ^i of the aggregated function for each subproblem. The main difference between MOEA/D-DRA and MOEA/D-GRA is the formula to estimate the difficulties of the subproblems, i.e., π^i used in MOEA/D-DRA and p^i used in MOEA/D-GRA. In MOEA/D-DRA, a set of subproblems is selected for evolution according to their utility functions π^i at each generation. This kind of selection in MOEA/D-DRA can also be realized in MOEA/D-GRA by simply setting $p^i = 1$ or $p^i = 0$. Therefore, MOEA/D-GRA can be seen as an extension of MOEA/D-DRA.

It was experimentally validated that the relative improvement on the aggregated function values is an effective and efficient indicator to dynamically assign the computational

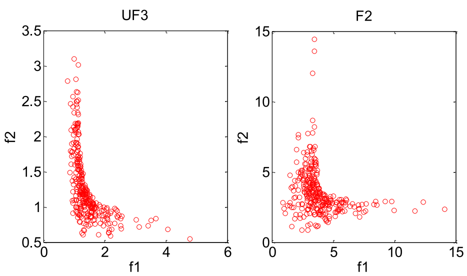


Fig. 1. Solutions obtained by MOEA/D-GRA at the tenth generation.

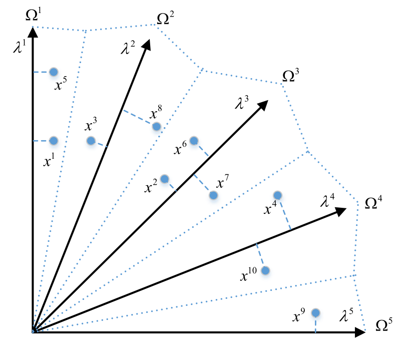


Fig. 2. Solution density of the weight vectors.

resources for the subproblems [34]. However, the utility functions used in DRA and GRA were naturally designed only depending on the convergence aspect. This may lead to the case that some subproblems are assigned with too many computational resources when they are surrounded by numerous solutions. Therefore, it is more reasonable to also consider the assignment of computational resources from another aspect (i.e., diversity), especially on some complicated problems. Thus, an IRA (diversity-enhanced) strategy is designed in this paper, which takes into account both convergence and diversity of each subproblem. This way, computational resources can be more reasonably assigned in MOEA/D. The details of our IRA strategy will be described next.

III. Proposed MOEA/D-IRA

In this section, the details of MOEA/D-IRA are introduced. First, the IRA strategy is presented and then the pseudo-code of MOEA/D-IRA is also provided to facilitate its implementation.

A. *Diversity-Enhanced Resource Allocation Strategy*

The allocation of computational resources in MOEA/D is an important issue. Although MOEA/D-DRA and MOEA/D-GRA have already been designed to alleviate the above problem, they only adopt the relative improvement of aggregated function values in (4) to dynamically assign the computational resources to the subproblems. Essentially, these two strategies only consider the convergence ability and ignore the distribution of solutions among the suproblems. In some cases, the subproblems may be surrounded with numerous solutions due to an uneven distribution in objective space, which often happens at the early stages of the search. In Fig. 1, the intermediate populations obtained by MOEA/D-GRA at the tenth generation are plotted with a population size of 300 when solving complex test problems, such as UF3 [47] and F2 [35]. As many solutions were located in the central area of the plots, most of them were evolved to further improve the subproblems in this area. This search behavior improves these subproblems quickly and thus assigns large amounts of computational resources to them using the GRA approach, which may significantly lower the population diversity. For such cases, it is more reasonable to consider the diversity also as an essential indicator in the RA strategy. It is worth mentioning that our IRA strategy is composed by two parts; one is the convergence indicator using the relative improvement on aggregated function values, and the other is the diversity indicator based on the number of solutions around the subproblems. This way, the IRA strategy would like to assign more computational resources to the subproblems surrounded with less solutions, which helps to enhance these subproblems. On the other hand, as the improvement of subproblems under the framework of MOEA/D is mainly dependent on the information of neighboring subproblems, by assigning more computational resources to the subproblems with low solution density, their enhancement will in turn facilitate the better solving of other subproblems.

Inspired by Li *et al.* [40], the solution density, sd, around each subproblem is adopted here to estimate the diversity circumstance of each subproblem. Some well-known MOEAs, such as MOEA/DD [42] and NSGA-III [48], also adopt the similar solution association method to maintain population diversity. As shown in Fig. 2, each subregion Ω^i is associated with the direction vector λ^i of a subproblem i. The solution x belongs to the subregion Ω^i only when the direction vector λ^i is closest to the solution among all the direction vectors based on the perpendicular distance in the following equation:

$$d_{\perp}(x, \lambda) = F'(x) - \frac{\lambda^T F'(x)}{\lambda^T \lambda}\lambda \tag{6}$$

where $F'(x)$ is the normalized objective vector of x, and its normalized objectives $f'_k(x)$ ($k = 1, 2, \ldots, m$, and m is the total number of objectives) of x are obtained using

$$f'_k(p_i) = \frac{f_k(p_i) - f_k \min}{f_k \max - f_k \min} \tag{7}$$

where $f_k \max$ and $f_k \min$ are, respectively, the maximum and minimum values of the kth objective found in the population. Then, the sd^i of subproblem i is the number of solutions in subregion Ω^i. For example, in Fig. 2, the direction vector closest to the solutions x^2, x^6, and x^7 is λ^3, thus they belong to the subregion Ω^3 and the solution density of subproblem 3 is 3, i.e., $sd^3 = 3$. By integrating the relative improvement on aggregated function value and the solution density on each subproblem, our IRA strategy is defined as

$$p^i = \beta \times \frac{\Delta^i + \varepsilon}{\max_{j=1,2,\ldots,N}\left\{\Delta^j\right\} + \varepsilon} + (1-\beta) \times \left(1 - \frac{sd^i}{\max_{j=1,2,\ldots,N}\left\{sd^j\right\}}\right) \tag{8}$$

Algorithm 1 UPDATE_P (S, S', λ)

Input: current solution set $S = \{x^1, x^2, \ldots, x^N\}$, previous solution set S' before Δt generations, the weight vectors $\lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^N\}$
Output: selection probability vector P

1: **for** $i = 1$ **to** N **do**
2: $sd^i = 0$;
3: **end for**
4: **for** $i = 0$ **to** N **do**
5: **for** $j = 0$ **to** N **do**
6: calculate the perpendicular distance $d(i,j) = d_{\perp}(x^i, \lambda^j)$ using (6);
7: **end for**
8: find the subproblem k as: $k = \arg_{j=1,\ldots,N} \min\{d(i,j)\}$;
9: sd^k++;
10: calculate the relative improvement Δ^i using (4);
11: $S' = S$;
12: **end for**
13: **for** $i = 0$ **to** N **do**
14: compute selection probability p^i for each subproblem using (8);
15: **end for**
16: **return** P

where $\varepsilon = 1.0 \times 10^{-50}$ is a small value to guarantee a valid division, Δ^i is the relative improvement on aggregated function value as defined in (4), sd^i is the solution density as described above, N is the number of direction vectors, and β is a control parameter to adjust the weights of the two parts.

A selection probability p^i $(i = 1, \ldots, N)$ is associated with each subproblem i and then the computational resources are accordingly assigned based on this probability. That is to say, once a randomly generated real value in [0, 1] is smaller than p^i, the subproblem i will be selected for evolution in this generation. Based on the observation of (8), it is easy to find out that our IRA strategy is actually a weighted sum of the convergence and diversity factors, as controlled by the parameter β. According to (8), the subproblem, which has been improved significantly over the last Δt generations and has less surrounding solutions, will be assigned with a higher selection probability for evolution in the next generation. Moreover, it is noted that when the parameter β is set to 1.0 in (8), our IRA strategy degenerates into the GRA strategy in [34]. Therefore, our IRA strategy is more comprehensive than the GRA strategy and it is an improvement of GRA, as the diversity indicator is further exploited for RA.

The pseudo-code of the update of the selection probability using our IRA strategy is given in Algorithm 1. Lines 1–3 in Algorithm 1 are used to initialize the solution density of each subproblem to be zero. Then, for each solution index i in line 4, the perpendicular distances between the solution x^i and all the weight vectors $\lambda^j (j = 1, 2, \ldots, N)$ are computed in lines 5–7. After that, in lines 8 and 9, the weight vector λ^k closest to the solution x^i is found and the solution density of subproblem k is increased by one. This way, the solution density is an integer not smaller than zero. The relative improvement Δ^i is calculated in line 10 based on the solution set S' before Δt generations and the current solution set S. In line 11, the solution set S' is updated by the current solution set S to compute the value Δ^i for the next iteration.

Algorithm 2 MOEA/D-IRA

1: Initialize the population $S = \{x^1, x^2, \ldots, x^N\}$, the saved population $S' = S$, the weight vectors $\lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^N\}$, the ideal point $z^* = \{z^1, z^2, \ldots, z^N\}$;
2: set $e = 0$, $gen = 0$, $A = \{1, 2, \ldots, N\}$, $p^i = 0.5$ for each $i = 1, 2, \ldots, N$;
3: **for** $i = 0$ **to** N **do**
4: initialize neighbor $B(i) = \{i_1, i_2, \ldots, i_T\}$ where $\lambda^{i_1}, \lambda^{i_2}, \ldots, \lambda^{i_T}$ are the T closet weight vectors to λ^i;
5: **end for**
6: **for** $i = 0$ **to** T **do**
7: compute the neighbor selection probability pn^i using (9);
8: **end for**
9: **while** $e < max\_evaluations$ **do**
10: **for** $i = 0$ **to** N **do**
11: **if** $rand(0, 1) \leq p^i$ **then**
12: **if** $rand(0, 1) \leq \delta$ **then**
13: $E = B(i)$;
14: **else**
15: $E = A$;
16: **end if**
17: select two solutions x^{r_1}, x^{r_2} from E based on pn;
18: generate an offspring v^i using x^{r_1}, x^{r_2}, x^i by DE;
19: get a new solution y^i by executing polynomial mutation on v^i;
20: evaluate the objective values of y^i;
21: update the ideal point z^*;
22: update the population using y^i;
23: $e = e + 1$;
24: **end if**
25: **end for**
26: $gen = gen + 1$;
27: **if** $mod(gen, \Delta t) == 0$ **then**
28: update P = UPDATE_P(S, S', λ) using **Algorithm 1**;
29: **end if**
30: **end while**

At last, line 14 updates the selection probability vector P using both relative improvement Δ^i and solution density sd^i as defined in (8).

B. Algorithmic Framework of MOEA/D-IRA

Based on the above IRA strategy, the algorithmic framework of MOEA/D-IRA is introduced. The pseudo-code of MOEA/D-IRA is given in Algorithm 2. Regarding the other important parts of the algorithm, a detailed introduction is given below.

1) Initialization: First, in line 1, an initial population $S = x^1, x^2, \ldots, x^N$ is randomly sampled in decision space and the saved population S' (used for computing the relative improvement after Δt generations) is initialized as S. As the exact ideal point is unknown in advance, an approximated point is used instead, which can be obtained as the minimum function value of each objective, i.e., $z_i^* = \min\{f_i(x)|x \in S\}$ for all $i = \{1, \ldots, m\}$. The weight vectors $\lambda = \{\lambda^1, \lambda^2, \ldots, \lambda^N\}$ are initialized as a set of evenly distributed vectors with the constraints $\sum_{j=1}^{m} \lambda_j^i = 1$ and $\lambda_j^i \geq 0$ for all $i = \{1, \ldots, N\}$.

Second, in line 4, in order to initialize the neighbors of weight vector λ^i, the Euclidean distances between λ^i and other weight vectors are computed, and then the neighbors of λ^i are included in a set $B(i) = \{i_1, i_2, \ldots, i_T\}$, where $\lambda^{i_1}, \lambda^{i_2}, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i. Based on this procedure,

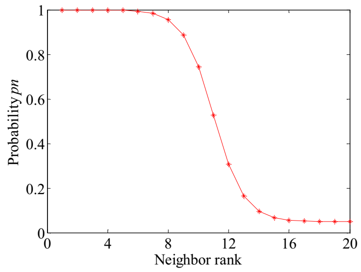


Fig. 3. Dynamic change of selection probability pn with different neighbor ranks.

a neighbor rank is also obtained based on the Euclidean distances among the weight vectors. The closest neighbor for each subproblem is λ^{i_1} with the neighbor rank 1, while its farthest neighbor is λ^{i_T} with the neighbor rank T. Then, a probability pn is used to select the parent solutions for applying the DE operator, as follows:

$$pn^i = pn_{\min} + \left(1 - pn_{\min}\right)\left(1 - \frac{1}{1 + 0.05 \times \exp\left(-20 \times \frac{i}{T} - 0.7\right)}\right) \tag{9}$$

where $pn_{\min}$ is a minimum probability to ensure that each neighbor has the opportunity to be selected; T is the neighbor size; and i is the neighbor rank based on the Euclidian distance of weight vectors. The closer neighbors have more chance to be selected using (9), as shown in line 7. Fig. 3 illustrates the dynamic change of probability pn with different neighbor ranks. It shows that the closer neighbors have higher opportunity to participate in the DE evolution than the farther neighbors.

2) Reproduction: Reproduction is an important component to generate an offspring population. There are a lot of reproduction operators, such as SBX [5] and DE [49], [50]. In lines 18 and 19 of Algorithm 2, the DE operator and polynomial-based mutation [3] are used to generate new solutions, as shown in (10)–(12). The main process of this operation is introduced below.

At first, a subproblem i is selected for evolution according to the selection probability p^i. If this subproblem i is selected for evolution, the candidate set E for selecting parent solutions is set to the neighbor set $B(i)$ or the entire population based on the parameter δ. Then, in line 17, when a solution is randomly selected from the candidate set E, a random real-value is further produced to check if it is larger than the probability pn in (9). If yes, this solution is selected as the parent solution for applying the DE operator. Only the parents in the neighborhood have to be selected with a probability pn, while the parents from the entire population are all selected randomly for the sake of keeping diversity. With two selected parent solutions and the solution for current subproblem i, the DE operator is run to generate a new solution $v^i = v_1^i, \ldots, v_n^i$, as follows:

$$v_j^i = \begin{cases} x_j^i + F \times \left(x_j^{r1} - x_j^{r2}\right) & \text{if } rand < CR \text{ or } j = j_{rand} \\ x_j^i & \text{otherwise} \end{cases} \tag{10}$$

where CR and F are two control parameters of the DE operator, $rand$ is a random real number uniformly sampled from [0, 1], j_{rand} is a random integer uniformly selected from $[1, n]$, and x^{r1} and x^{r2} are two selected solutions from E. After that, polynomial-based mutation is further implemented on v^i to obtain the offspring solution y^i, as follows:

$$y_j^i = \begin{cases} v_j^i + \Delta_j \times \left(u_j - l_j\right) & \text{if } rand < p_m \\ v_j^i & \text{otherwise} \end{cases} \tag{11}$$

with

$$\Delta_j = \begin{cases} \left[2r + (1-2r)\left(\frac{u_j - v_j}{u_j - l_j}\right)^{\eta+1}\right]^{1/(\eta+1)} - 1 & \text{if } r < 0.5 \\ 1 - \left[2 - 2r + (2r-1)\left(\frac{v_j - l_j}{u_j - l_j}\right)^{\eta+1}\right]^{1/(\eta+1)} & \text{otherwise} \end{cases} \tag{12}$$

where $j \in \{1, \ldots, n\}$, r is a random real number uniformly sampled from [0, 1], η is the distribution index, p_m is the mutation probability, and l_j and u_j are, respectively, the lower and upper bounds of the jth decision variable.

3) Updating: After generating a new solution, the population should be updated by using a replacement strategy to discard the inferior old solution and keep the good new one. In MOEA/D [16] and some variants [33], [35], several solutions in the neighbors or the entire population can be replaced by the new one. In line 22 of Algorithm 2, the new solution only replaces the solution of the matched subproblem based on its relative improvement. By calculating the improvement of this solution to each subproblem, the one with the largest improvement rate will be replaced. Compared to the random replacement strategy [16], this strategy is particularly effective when the subproblem that can be greatly improved by the new solution is not in its neighbor set [34]. On the other hand, in line 28, the selection probability P of each subproblem is updated within a period of Δt generations by using Algorithm 1.

IV. EXPERIMENTS

In this section, the relevant experimental design for performance analysis of the proposed algorithm is provided. The test problems, parameters settings, and performance measures used in our experiments are introduced first. Then, the comparison of results of our proposed MOEA/D-IRA with respect to four competitive MOEA/D variants (i.e., MOEA/D-DE, MOEA/D-IR, MOEA/D-DRA, and MOEA/D-GRA) are provided. Moreover, the contributions of two components [the IRA strategy and the mating parent selection strategy using (9)] are experimentally studied in MOEA/D-IRA, and two existing RA strategies (DRA and GRA) are used to compare the IRA strategy, under the framework of MOEA/D-IRA. At last, the impact of the parameters settings in our algorithm is analyzed and a suggestion for setting the parameter values is also provided based on the experiments.

A. Test Problems and Parameters Settings

In our experiments, 19 unconstrained test MOPs were used to assess the performance of our proposed algorithm, including ten UF instances (UF1–UF10) from the CEC2009 MOEA competition [47] and nine F instances [35], which were widely used to test the comprehensive performance of several MOEA/D variants [33], [34], [36], [39], [40]. The test instances adopted in this paper have different features and their PS shapes are very complicated. It is noted that UF1–UF7, F1–F5, and F7–F9 are bi-objective problems, while UF8–UF10 and F6 are three-objective problems. The number of decision variables is set to 30 for F1–F5, F9 and all UF test problems, and is set to 10 for F6–F8.

The parameters in all the compared algorithms are set as follows.

1) *MOEA/D-DE:* The NS $T = 20$, the probability to select the neighbors as the candidate set for evolution $\delta = 0.9$, and the updated size $n_r = 2$. The other parameters are set the same as in [35].
2) *MOEA/D-DRA:* The NS $T = 0.1N$ (N is the number of weight vectors), the probability to select the neighbors as the candidate set for evolution $\delta = 0.9$, and the updated size $n_r = 0.01N$. The other parameters are set the same as in [33].
3) *MOEA/D-IR:* The NS $T = 20$, and the probability to select the neighbors as the candidate set for evolution $\delta = 0.9$. The other parameters are set the same as in [40].
4) *MOEA/D-GRA:* The NS $T = 20$, the probability to select the neighbors as the candidate set for evolution $\delta = 0.8$, and the updating period $\Delta t = 20$. The other parameters are set the same as in [34].
5) *MOEA/D-IRA:* The NS $T = 20$, the probability to select the neighbors as the candidate set for evolution $\delta = 0.8$, the updating period $\Delta t = 20$, the weight parameter$\beta = 0.98$ in (8), the minimum selection probability in (9) $pn_{\min} = 0.05$.

The population size N was set to 300 for all bi-objective test MOPs and to 600 for all three-objective test MOPs. The adopted weight vectors can be downloaded from the website of Dr. K. Li (http://www.cs.bham.ac.uk/∼likw/publications.html). The maximum allowable number of function evaluations was set to 150 000 for F1–F5 and F7–F9, and to 300 000 for F6 and UF1–UF10. All the compared algorithms performed 51 independent runs on each test problem.

B. Performance Measures

In this paper, in order to provide a comprehensive assessment on the performance for the compared MOEA/D variants, two widely used performance measures, i.e., inverted generational distance (IGD) [51] and hypervolume (HV) [52], were adopted. They can simultaneously measure the convergence and the population diversity of the obtained approximation set. When calculating IGD, 1000 points were uniformly sampled from the true PF for the bi-objective test problems, while 10 000 points were sampled for the three-objective ones. A lower value of IGD indicates that the obtained set is closer to the true PF and more uniformly distributed along the true PF. Regarding the computation of HV, it is more appropriate to set the reference point slightly larger than the worst value of each objective on the true PF, so that the convergence and the diversity of the approximation set can be well balanced [52]. Thus, the reference point was set to $(2.0, 2.0)^T$ for bi-objective UF and F instances, and set to $(2.0, 2.0, 2.0)^T$ for the three-objective instances. A larger value of HV indicates a better quality of P for approximating the entire true PF.

C. Performance Comparisons With Other MOEA/D Variants

In this section, the performance of MOEA/D-IRA is compared to four MOEA/D variants, i.e., MOEA/D-DE [35], MOEA/D-DRA [33], MOEA/D-GRA [34], and MOEA/D-IR [40] which is based on the framework of MOEA/D-DRA. Tables I and II, respectively, provide the results of all the algorithms on UF and F instances after performing 51 independent runs, regarding IGD and HV. The best mean result for each problem is highlighted in boldface with gray background. In order to have a statistically sound conclusion, Wilcoxon's rank sum test with a 5% significance level was conducted to compare the significance of difference between the results obtained by MOEA/D-IRA and the other algorithms. In Tables I and II, "−," "+," and "∼," respectively, denote that the results obtained by the corresponding algorithm are worse than, better than or similar to those of MOEA/D-IRA.

As observed from Table I, MOEA/D-IRA is found to be advantageous when compared to its competitors with respect to IGD. Among the 19 test instances, MOEA/D-IRA is able to perform best on 16 test problems, while the other compared algorithms could only obtain the best results on at most two test problems. The comparisons of MOEA/D-IRA with other algorithms are summarized in the last row of Table I, where "−/∼/+" gives the total number of test problems in which MOEA/D-IRA performs better than, similarly to, and worse than the corresponding algorithm. Considering the comparisons with MOEA/D-DE, MOEA/D-IR, and MOEA/D-DRA, MOEA/D-IRA has shown an absolute advantage, as it outperforms them on at least 17 test problems; whereas, MOEA/D-IRA only performs worse than MOEA/D-IR on UF8 and F8, and worse than MOEA/D-DE on F8, as revealed by the Wilcoxon's rank sum test. When compared to MOEA/D-GRA, MOEA/D-IRA performs better on 14 test problems and obtains statistically similar results on three test problems as indicated by Wilcoxon's rank sum test, and it is outperformed only on UF3 and F8.

Therefore, when considering all the test problems adopted, it is reasonable to draw a conclusion that our algorithm presents a superior performance over MOEA/D-DE, MOEA/D-IR, MOEA/D-DRA, and MOEA/D-GRA with respect to IGD. Such advantages of MOEA/D-IRA are mainly brought by the IRA strategy combined with the convergence and diversity indicators in (8).

TABLE I
PERFORMANCE COMPARISON OF SEVERAL COMPETITIVE MOEA/D VARIANTS USING IGD VALUES ON UF AND F TEST INSTANCES

Test Instance	MOEA/D-DE	MOEA/D-IR	MOEA/D-DRA	MOEA/D-GRA	MOEA/D-IRA
UF1	1.92E-03(1.77E-04)(-)	2.08E-03(2.04E-04)(-)	2.96E-03(5.92E-04)(-)	1.79E-03(1.10E-04)(-)	**1.57E-03**(6.67E-05)
UF2	6.54E-03(1.91E-03)(-)	5.17E-03(1.80E-03)(-)	7.57E-03(2.29E-03)(-)	4.44E-03(1.91E-03)(-)	**2.66E-03**(4.36E-04)
UF3	1.08E-02(1.21E-02)(-)	5.52E-03(4.45E-03)(-)	3.48E-02(3.68E-02)(-)	**2.94E-03**(1.96E-03)(+)	3.28E-03(1.76E-03)
UF4	6.14E-02(3.98E-03)(-)	5.82E-02(3.62E-03)(-)	6.29E-02(4.95E-03)(-)	5.45E-02(3.59E-03)(~)	**5.35E-02**(3.33E-03)
UF5	3.00E-01(8.87E-02)(-)	2.83E-01(5.90E-02)(-)	3.27E-01(1.00E-01)(-)	2.40E-01(7.57E-02)(~)	**2.27E-01**(4.20E-02)
UF6	2.46E-01(2.21E-01)(-)	1.51E-01(9.20E-02)(-)	2.60E-01(2.22E-01)(-)	1.56E-01(1.66E-01)(-)	**8.01E-02**(3.00E-02)
UF7	2.64E-03(4.22E-04)(-)	3.11E-03(2.41E-03)(-)	3.95E-03(4.00E-03)(-)	2.07E-03(1.38E-04)(-)	**1.71E-03**(1.10E-04)
UF8	5.98E-02(7.12E-03)(-)	**3.98E-02**(1.04E-02)(+)	5.65E-02(1.60E-02)(-)	6.33E-02(1.21E-02)(-)	4.86E-02(1.51E-02)
UF9	5.78E-02(3.71E-02)(-)	5.31E-02(4.52E-02)(-)	1.02E-01(5.31E-02)(-)	4.10E-02(3.44E-02)(-)	**3.22E-02**(2.37E-02)
UF10	4.77E-01(5.06E-02)(-)	4.68E-01(7.24E-02)(-)	4.25E-01(8.69E-02)(-)	5.67E-01(7.83E-02)(-)	**3.69E-01**(5.71E-02)
F1	1.36E-03(2.83E-05)(-)	1.38E-03(2.18E-05)(-)	1.80E-03(1.58E-04)(-)	1.36E-03(3.19E-05)(-)	**1.34E-03**(1.93E-05)
F2	3.08E-03(4.72E-04)(-)	3.27E-03(4.15E-04)(-)	6.19E-03(1.47E-02)(-)	2.59E-03(2.10E-04)(-)	**2.08E-03**(1.18E-04)
F3	9.38E-03(1.47E-02)(-)	3.43E-03(1.90E-03)(-)	2.39E-02(4.06E-02)(-)	2.61E-03(9.58E-04)(-)	**1.99E-03**(4.81E-04)
F4	4.47E-03(1.58E-03)(-)	2.03E-03(1.19E-04)(-)	5.15E-03(8.42E-03)(-)	2.21E-03(1.81E-04)(-)	**1.80E-03**(8.15E-05)
F5	1.15E-02(7.53E-03)(-)	9.60E-03(2.56E-03)(-)	1.29E-02(1.19E-02)(-)	6.90E-03(1.78E-03)(-)	**4.97E-03**(1.22E-03)
F6	2.89E-02(7.32E-04)(-)	2.21E-02(2.01E-04)(-)	2.95E-02(8.33E-04)(-)	2.78E-02(4.32E-04)(-)	**2.20E-02**(2.30E-04)
F7	3.68E-03(5.06E-03)(-)	2.25E-03(4.21E-04)(-)	2.38E-03(4.30E-04)(-)	1.87E-03(2.72E-04)(~)	**1.81E-03**(2.47E-04)
F8	7.37E-02(3.66E-02)(+)	5.78E-02(2.52E-02)(+)	1.33E-01(6.20E-02)(-)	**3.38E-02**(2.32E-02)(+)	9.83E-02(4.31E-02)
F9	4.25E-03(2.05E-03)(-)	3.21E-03(5.39E-04)(-)	4.15E-03(5.45E-04)(-)	2.49E-03(2.35E-04)(-)	**1.99E-03**(1.56E-04)
Total	18-/1+	17-/2+	19-	14-/3~/2+	

+, - and ~ denote that the performance of corresponding algorithm is significantly better than, worse than, and similar to MOEA/D-IRA respectively by Wilcoxon's rank sum test with $\alpha = 0.05$

TABLE II
PERFORMANCE COMPARISON OF SEVERAL COMPETITIVE MOEA/D VARIANTS USING HV VALUES ON UF AND F TEST INSTANCES

Test Instance	MOEA/D-DE	MOEA/D-IR	MOEA/D-DRA	MOEA/D-GRA	MOEA/D-IRA
UF1	3.6563(2.99E-03)(-)	3.6540(3.54E-03)(-)	3.6517(5.90E-03)(-)	3.6590(1.31E-03)(-)	**3.6614** (9.98E-04)
UF2	3.6434(1.57E-02)(-)	3.6438(2.06E-02)(-)	3.6406(1.27E-02)(-)	3.6461(1.76E-02)(-)	**3.6580** (5.04E-03)
UF3	3.6216(6.52E-02)(-)	3.6535(1.35E-02)(-)	3.5348(1.50E-01)(-)	**3.6611**(3.38E-03)(+)	3.6577 (8.63E-03)
UF4	3.1491(1.80E-02)(-)	3.0984(8.73E-02)(-)	3.1360(2.52E-02)(-)	3.1766(1.48E-02)(~)	**3.1793** (1.48E-02)
UF5	2.6191(2.30E-01)(-)	2.5578(3.23E-01)(-)	2.4826(2.48E-01)(-)	2.9125(1.71E-01)(-)	**2.9594** (1.31E-01)
UF6	2.8021(3.82E-01)(-)	2.8103(3.47E-01)(-)	2.7303(3.94E-01)(-)	2.9943(3.55E-01)(-)	**3.1647** (7.15E-02)
UF7	3.4832(8.32E-03)(-)	3.4768(3.31E-02)(-)	3.4759(4.38E-02)(-)	3.4911(3.26E-03)(-)	**3.4946** (2.07E-03)
UF8	7.3175(1.92E-02)(-)	**7.3857**(2.65E-02)(~)	7.3415(2.42E-02)(-)	7.3362(1.86E-02)(-)	7.3806 (2.60E-02)
UF9	7.5037(1.58E-01)(-)	7.6110(2.10E-01)(-)	7.3680(2.44E-01)(-)	7.6495(1.53E-01)(-)	**7.7207** (1.05E-01)
UF10	3.4414(2.34E-01)(-)	3.5588(4.96E-01)(-)	3.6822(3.26E-01)(-)	3.5668(3.35E-01)(-)	**4.6251** (3.84E-01)
F1	3.6634(3.85E-04)(-)	**3.6638**(2.64E-04)(~)	3.6611(1.57E-03)(-)	3.6636(3.82E-04)(-)	3.6638 (2.96E-04)
F2	3.6455(1.02E-02)(-)	3.6459(9.69E-03)(-)	3.6356(5.85E-02)(-)	3.6536(3.56E-03)(-)	**3.6589** (1.39E-03)
F3	3.6202(6.97E-02)(-)	3.6482(1.99E-02)(-)	3.5735(1.39E-01)(-)	3.6537(9.48E-03)(-)	**3.6591** (8.71E-03)
F4	3.6533(7.38E-03)(-)	3.6602(1.14E-03)(-)	3.6451(4.78E-02)(-)	3.6593(1.65E-03)(-)	**3.6614** (1.01E-03)
F5	3.6273(4.29E-02)(-)	3.6312(2.46E-02)(-)	3.6243(4.98E-02)(-)	3.6396(1.76E-02)(-)	**3.6541** (4.89E-03)
F6	7.4221(2.37E-03)(-)	7.4432(9.92E-04)(-)	7.4226(2.71E-03)(-)	7.4255(1.49E-03)(-)	**7.4452** (5.03E-04)
F7	3.6187(4.72E-02)(-)	3.6431(8.05E-03)(-)	3.6468(7.41E-03)(~)	3.6458(7.20E-03)(-)	**3.6496** (7.78E-03)
F8	3.4354(8.49E-02)(~)	3.1821(2.29E-01)(-)	3.3456(1.17E-01)(-)	**3.5267**(5.32E-02)(+)	3.4319 (7.72E-02)
F9	3.3133(1.18E-02)(-)	3.3134(2.03E-02)(-)	3.3148(8.07E-03)(-)	3.3239(1.45E-03)(-)	**3.3265** (1.22E-03)
Total	18-/1~	17-/2~	18-/1~	16-/1~/2+	

+, - and ~ denote that the performance of corresponding algorithm is significantly better than, worse than, and similar to MOEA/D-IRA respectively by Wilcoxon's rank sum test with $\alpha = 0.05$.

Table II further lists the experimental results of all the compared algorithms with respect to HV. As observed from Table II, similar conclusions are drawn. First, MOEA/D-IRA outperforms others as it performs best on most of the test problems adopted. MOEA/D-IRA obtains the best results on 15 out of 19 test problems, while MOEA/D-IR and MOEA/D-GRA only achieve the best results on two test problems. Second, MOEA/D-IRA performs better on most cases when, respectively, compared to MOEA/D-DE, MOEA/D-IR, MOEA/D-DRA, and MOEA/D-GRA. Only MOEA/D-GRA outperforms MOEA/D-IRA on UF3 and F8, while the other competitors cannot surpass MOEA/D-IRA on any test problem. Besides that, MOEA/D-IRA obtains statistically similar results to MOEA/D-DE on F8, to MOEA/D-IR on UF8 and F1, to MOEA/D-DRA on F7, and to MOEA/D-GRA on UF4. As summarized in the last row of Table II, our algorithm performs better than or similarly to MOEA/D-DE, MOEA/D-IR, MOEA/D-DRA, and MOEA/D-GRA on 19, 19, 19, and 17 out of 19 test problems. Therefore, it is further confirmed by using HV that our algorithm shows advantages when tackling the UF and F test problems.

In order to have a deeper understanding about the performance of our algorithm, Figs. 4–6 provide the plots of the approximation set obtained by MOEA/D-DE, MOEA/D-IR, MOEA/D-DRA, MOEA/D-GRA, and MOEA/D-IRA on UF2, UF9, and F5, respectively, in which the true PFs are

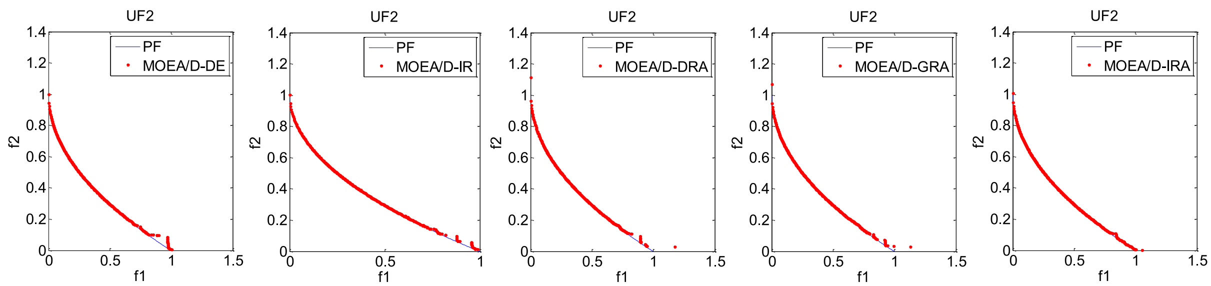


Fig. 4. Comparison of approximation sets on UF2.

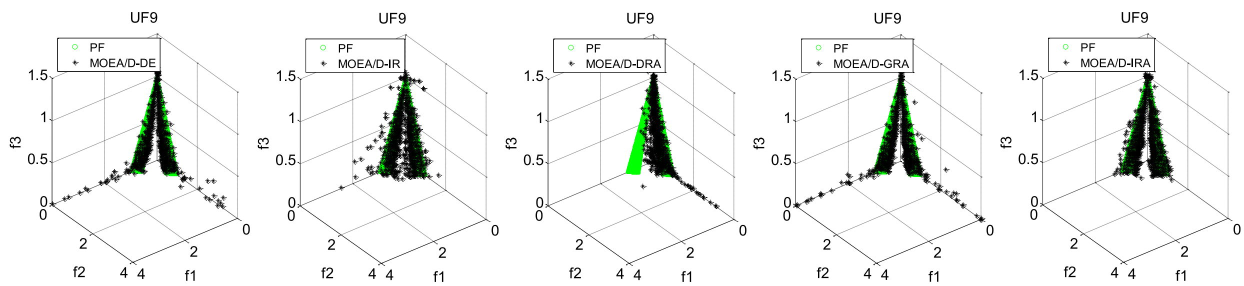


Fig. 5. Comparison of approximation sets on UF9.

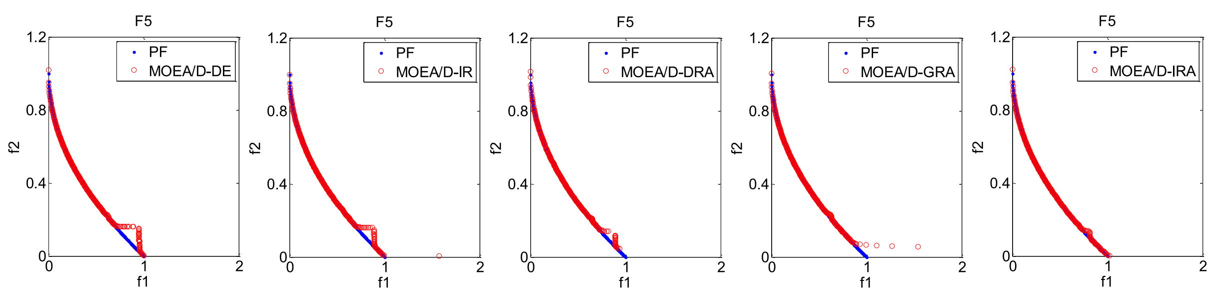


Fig. 6. Comparison of approximation sets on F5.

also illustrated for comparison. These plotted solutions were obtained from one run with the median IGD value from 51 runs. As observed from Fig. 4, MOEA/D-DE, MOEA/D-IR, MOEA/D-DRA, and MOEA/D-GRA fail to find a set of uniformly distributed solutions to cover the entire true PF of UF2, as they miss some Pareto-optimal solutions at one end of the true PF. MOEA/D-IRA performs much better as it smoothly covers the entire true PF of UF2. Regarding UF9 in Fig. 5, the approximation set obtained by MOEA/D-IRA can also evenly approximate the true PF, while MOEA/D-DE, MOEA/D-IR, MOEA/D-DRA, and MOEA/D-GRA find some extreme solutions far away from the true PF. In Fig. 6, MOEA/D-IRA also provides more evenly distributed solutions, while the other competitors fail to find some Pareto-optimal solutions on some parts of the true PF. From these observations, it is clear that the final solution sets found by MOEA/D-IRA are closer to the true PFs and are more uniformly distributed along the true PFs when compared to the other competitors. Moreover, to provide an overview of the evolutionary progress for all the competitors, their convergence curves regarding the mean IGD values on all the UF problems are provided in Fig. S.1 of the supplementary file. This information is not included in this paper due to page limitations. From these plots, MOEA/D-IRA outperforms the competitors on most cases, and is able to gradually reduce the IGD values and get closer to the true PF on most of UF problems as the search progresses.

To further study the performance of MOEA/D-IRA under a limited computational load, one more experiment was conducted with 20% of maximum function evaluations (i.e., 30 000 for F1–F5 and F7–F9, and 60 000 for F6 and UF1–UF10) for all the compared algorithms. Due to page limitations, the IGD and HV results are provided in Tables S.I and S.II of the supplementary file. As the final solutions in this case do not fully converge to the true PFs, the advantages of MOEA/D-IRA over the other competitors are not so obvious as shown in Tables I and II. This is because all the subproblems still have the potential to be further enhanced and an equal-probability selection for them can also improve their aggregated function values. However, our IRA approach still works effectively under this limited computational load, as it performs better than all the competitors from the one-by-one comparisons shown in the last row of Tables S.I and S.II in the supplementary file.

The above comparisons clearly show us that MOEA/D-IRA is a more effective algorithm to solve some complicated test problems when compared to MOEA/D-DE, MOEA/D-IR, MOEA/D-DRA, and MOEA/D-GRA. The outstanding performance of MOEA/D-IRA is mainly

TABLE III
PERFORMANCE COMPARISON OF DIFFERENT MOEA/D-IRA VARIANTS USING IGD AND HV VALUES ON UF AND F TEST INSTANCES

	IGD			HV		
Instance	Variant-I	Variant-II	MOEA/D-IRA	Variant-I	Variant-II	MOEA/D-IRA
UF1	1.71E-03(1.03E-04) (-)	1.60E-03(8.27E-05) (-)	**1.57E-03**(6.67E-05)	3.6608(8.35E-04) (-)	3.6612(7.95E-04) (~)	**3.6614** (9.98E-04)
UF2	3.44E-03(6.81E-04) (-)	5.26E-03(1.40E-03) (-)	**2.66E-03**(4.36E-04)	3.6547(7.35E-03) (-)	3.6504(9.94E-03) (-)	**3.6580** (5.04E-03)
UF3	**2.55E-03**(1.47E-03) (+)	5.78E-03(4.13E-03) (-)	3.28E-03(1.76E-03)	**3.6603**(6.58E-03) (+)	3.6548(8.70E-03) (-)	3.6577 (8.63E-03)
UF4	5.51E-02(3.30E-03) (-)	5.44E-02(3.98E-03) (~)	**5.35E-02**(3.33E-03)	3.1773(1.39E-02) (~)	3.1783(1.49E-02) (~)	**3.1793** (1.48E-02)
UF5	2.32E-01(2.87E-02) (~)	2.75E-01(7.15E-02) (-)	**2.27E-01**(4.20E-02)	**2.9600**(9.04E-02) (~)	2.7862(1.72E-01) (-)	2.9594 (1.31E-01)
UF6	1.17E-01(1.36E-01) (~)	8.26E-02(4.14E-02) (~)	**8.01E-02**(3.00E-02)	3.0889(2.91E-01) (~)	3.1578(1.27E-01) (~)	**3.1647** (7.15E-02)
UF7	1.91E-03(1.14E-04) (-)	1.88E-03(1.21E-04) (-)	**1.71E-03**(1.10E-04)	3.4936(2.61E-03) (-)	3.4939(1.99E-03) (-)	**3.4946** (2.07E-03)
UF8	5.47E-02(2.03E-02) (~)	5.89E-02(1.35E-02) (-)	**4.86E-02**(1.51E-02)	7.3677(3.58E-02) (~)	7.3509(2.37E-02) (-)	**7.3806** (2.60E-02)
UF9	4.98E-02(4.97E-02) (~)	4.76E-02(3.50E-02) (-)	**3.22E-02**(2.37E-02)	7.6435(2.13E-01) (~)	7.6291(1.51E-01) (-)	**7.7207** (1.05E-01)
UF10	4.70E-01(8.87E-02) (-)	4.56E-01(8.82E-02) (-)	**3.69E-01**(5.71E-02)	4.0771(5.35E-01) (-)	3.9939(4.56E-01) (-)	**4.6251** (3.84E-01)
F1	1.34E-03(2.47E-05) (~)	1.36E-03(2.57E-05) (-)	**1.34E-03**(1.93E-05)	3.6638(3.66E-04) (~)	3.6636(3.20E-04) (-)	**3.6638** (2.96E-04)
F2	2.36E-03(1.51E-04) (-)	2.48E-03(3.70E-04) (-)	**2.08E-03**(1.18E-04)	3.6577(1.25E-03) (-)	3.6576(1.88E-03) (-)	**3.6589** (1.39E-03)
F3	2.19E-03(2.38E-04) (-)	2.59E-03(8.31E-04) (-)	**1.99E-03**(4.81E-04)	**3.6593**(3.64E-03) (+)	3.6585(5.96E-03) (-)	3.6591 (8.71E-03)
F4	2.01E-03(1.23E-04) (-)	3.59E-03(1.60E-03) (-)	**1.80E-03**(8.15E-05)	3.6608(1.05E-03) (-)	3.6551(7.80E-03) (-)	**3.6614** (1.01E-03)
F5	5.75E-03(9.28E-04) (-)	9.72E-03(2.54E-03) (-)	**4.97E-03**(1.22E-03)	3.6512(8.00E-03) (-)	3.6430(9.79E-03) (-)	**3.6541** (4.89E-03)
F6	**2.19E-02**(1.92E-04) (+)	2.20E-02(1.83E-04) (~)	2.20E-02(2.30E-04)	**7.4453**(4.31E-04) (~)	7.4450(4.81E-04) (-)	7.4452 (5.03E-04)
F7	**1.73E-03**(2.09E-04) (~)	7.97E-03(1.17E-02) (-)	1.81E-03(2.47E-04)	**3.6502**(6.94E-03) (~)	3.5782(1.01E-01) (-)	3.6496 (7.78E-03)
F8	**8.27E-02**(5.08E-02) (~)	1.25E-01(3.74E-02) (-)	9.83E-02(4.31E-02)	**3.4591**(8.18E-02) (~)	3.3740(7.14E-02) (-)	3.4319 (7.72E-02)
F9	2.28E-03(1.45E-04) (-)	6.40E-03(1.30E-02) (-)	**1.99E-03**(1.56E-04)	3.3259(9.24E-04) (-)	3.3178(2.64E-02) (-)	**3.3265** (1.22E-03)
Total	10-/7~/2+	16-/3~		8-/9~/2+	16-/3~	

+, - and ~ denote that the performance of corresponding algorithm is significantly better than, worse than, and similar to MOEA/D-IRA respectively by Wilcoxon's rank sum test with $\alpha = 0.05$.

due to the utilization of a diversity-enhanced RA strategy, which helps to assign the computational resources more reasonably by considering both convergence and diversity for each subproblem.

D. Comparisons With Different MOEA/D-IRA Variants

In order to analyze the contributions of the two components (i.e., the IRA strategy and the mating parent selection strategy) proposed in MOEA/D-IRA, two variants of MOEA/D-IRA (i.e., variant-I and variant-II) are used for performance comparison. Variant-I is implemented by removing the mating parent selection strategy from MOEA/D-IRA and only uses the original random selection of the mating parents. Variant-II is designed by removing the IRA strategy from MOEA/D-IRA and all the subproblems are equally evolved in one generation. All the mean IGD and HV results of variant-I, variant-II, and MOEA/D-IRA from 51 runs are, respectively, listed in Table III.

From the IGD results in Table III, it is clear that MOEA/D-IRA shows a superior performance when compared to both variant-I and variant-II. Most of the best IGD values are obtained by MOEA/D-IRA. Variant-I is best on four test problems, while variant-II cannot perform best on any test problem. These results indicate that the two strategies are all effective to enhance the performance of MOEA/D-IRA on most of the test problems adopted. More specifically, MOEA/D-IRA outperforms variant-I and variant-II on 10 and 16 out of 19 test problems, respectively. The Wilcoxon's rank sum test also shows that MOEA/D-IRA performs similarly to variant-I and variant-II on seven and three test problems, respectively. Only variant-I is able to outperform MOEA/D-IRA on two test problems (i.e., UF3 and F6). Therefore, the effectiveness of the two above strategies is validated using IGD.

Regarding the HV results in Table III, MOEA/D-IRA performs best on 13 out of 19 test problems, while variant-I is best on the rest six test problems. As revealed by the Wilcoxon's rank sum test, MOEA/D-IRA obtains statistically similar results to variant-I and variant-II on nine and three test problems, and outperforms variant-I and variant-II on 8 and 16 test problems, respectively. That is to say, MOEA/D-IRA performs better than or similarly to variant-I and variant-II on 17 and 19 out of 19 test problems. Thus, the two proposed strategies still show their usefulness using HV, as they improve performance on about half of the test problems, but only deteriorate it on two test problems. Therefore, the advantage of using these two strategies in MOEA/D-IRA is further confirmed by HV.

E. Comparisons of Different RA Strategies

To further analyze the advantages of different RA strategies, such as DRA, GRA, and IRA, some experiments were conducted here. MOEA/D-IRA has a different evolutionary behavior with respect to MOEA/D-DRA and MOEA/D-GRA, i.e., the mating parent selection when running DE. So, in order to have a fair comparison of these different RA strategies, DRA and GRA are also embedded into MOEA/D-IRA to substitute our IRA strategy, making two new variants as denoted by DRA-I and GRA-I. It is noted that, except for the RA strategy, DRA-I and GRA-I share the same evolutionary procedures as MOEA/D-IRA.

Table IV presents the comparison of results of DRA-I, GRA-I, and MOEA/D-IRA using IGD and HV. Regarding the IGD results in Table IV, it is clear that MOEA/D-IRA also presents a superior performance over DRA-I and GRA-I. MOEA/D-IRA achieves most of the best IGD values, while DRA-I performs best on UF8 and F8, and GRA-I only obtains the best performance on UF1. This also indicates

TABLE IV
PERFORMANCE COMPARISON OF DIFFERENT RA STRATEGIES USING IGD AND HV VALUES ON UF AND F TEST INSTANCES

	IGD			HV		
Instance	DRA-I	GRA-I	MOEA/D-IRA	DRA-I	GRA-I	MOEA/D-IRA
UF1	1.86E-03(1.49E-04)(-)	**1.56E-03**(6.94E-05) (~)	1.57E-03(6.67E-05)	3.6585(2.19E-03)(-)	**3.6615**(7.87E-04) (~)	3.6614 (9.98E-04)
UF2	3.69E-03(2.13E-03) (-)	2.82E-03(5.57E-04) (~)	**2.66E-03**(4.36E-04)	3.6527(1.29E-02)(~)	3.6579(4.65E-03) (~)	**3.6580** (5.04E-03)
UF3	9.52E-03(9.09E-03)(-)	4.08E-03(2.73E-03) (~)	**3.28E-03**(1.76E-03)	3.6309(4.73E-02)(-)	3.6561(1.14E-02) (~)	**3.6577** (8.63E-03)
UF4	5.72E-02(4.24E-03) (-)	5.40E-02(3.11E-03) (~)	**5.35E-02**(3.33E-03)	3.1660(1.76E-02)(-)	**3.1798**(1.38E-02) (~)	3.1793 (1.48E-02)
UF5	3.06E-01(1.23E-01) (-)	2.47E-01(7.56E-02) (~)	**2.27E-01**(4.20E-02)	2.5662(3.02E-01)(-)	2.9216(1.78E-01) (~)	**2.9594** (1.31E-01)
UF6	2.30E-01(1.56E-01)(-)	1.10E-01(1.02E-01) (~)	**8.01E-02**(3.00E-02)	2.7593(3.07E-01)(-)	3.0964(2.11E-01) (-)	**3.1647** (7.15E-02)
UF7	2.79E-03(2.61E-03)(-)	1.75E-03(1.09E-04) (-)	**1.71E-03**(1.10E-04)	3.4819(3.42E-02)(-)	3.4938(2.60E-03) (~)	**3.4946** (2.07E-03)
UF8	**4.60E-02**(7.93E-03)(~)	7.26E-02(1.54E-02) (-)	4.86E-02(1.51E-02)	7.3755(1.45E-02)(-)	7.3331(1.91E-02) (-)	**7.3806** (2.60E-02)
UF9	1.06E-01(5.21E-02)(-)	4.57E-02(3.90E-02) (-)	**3.22E-02**(2.37E-02)	7.3550(2.37E-01)(-)	7.6419(1.66E-01) (-)	**7.7207** (1.05E-01)
UF10	4.06E-01(6.68E-02)(-)	4.67E-01(8.26E-02) (-)	**3.69E-01**(5.71E-02)	3.6915(5.38E-01)(-)	3.9574(4.56E-01) (-)	**4.6251** (3.84E-01)
F1	1.47E-03(3.32E-05)(-)	1.35E-03(2.41E-05) (~)	**1.34E-03**(1.93E-05)	3.6637(3.32E-04)(~)	**3.6639**(2.50E-04) (~)	3.6638 (2.96E-04)
F2	3.57E-03(5.41E-03)(-)	9.92E-03(1.45E-02) (-)	**2.08E-03**(1.18E-04)	3.6486(3.21E-02)(-)	3.6280(6.34E-02) (-)	**3.6589** (1.39E-03)
F3	3.80E-03(4.40E-03)(-)	4.33E-03(9.45E-03) (-)	**1.99E-03**(4.81E-04)	3.6488(3.04E-02)(-)	3.6502(4.30E-02) (-)	**3.6591** (8.71E-03)
F4	1.92E-03(1.35E-04)(-)	2.41E-03(1.09E-04) (-)	**1.80E-03**(8.15E-05)	3.6602(1.50E-03)(-)	3.6586(1.25E-03) (-)	**3.6614** (1.01E-03)
F5	1.03E-02(1.24E-02)(-)	1.26E-02(4.07E-03) (-)	**4.97E-03**(1.22E-03)	3.6252(5.64E-02)(-)	3.6364(1.99E-02) (-)	**3.6541** (4.89E-03)
F6	2.81E-02(5.24E-04)(-)	2.90E-02(6.51E-04) (-)	**2.20E-02**(2.30E-04)	7.4272(1.89E-03)(-)	7.4246(1.66E-03) (-)	**7.4452** (5.03E-04)
F7	2.20E-03(4.56E-04)(-)	1.93E-03(3.67E-04) (~)	**1.81E-03**(2.47E-04)	3.6400(9.17E-03)(-)	3.6464(1.13E-02) (~)	**3.6496** (7.78E-03)
F8	**7.78E-02**(5.86E-02)(+)	1.11E-01(4.77E-02) (~)	9.83E-02(4.31E-02)	3.3934(1.50E-01)(~)	3.4037(7.94E-02) (~)	**3.4319** (7.72E-02)
F9	2.86E-03(5.79E-04)(-)	3.47E-03(1.77E-03) (-)	**1.99E-03**(1.56E-04)	3.3226(2.67E-03)(-)	3.3230(5.44E-03) (-)	**3.3265** (1.22E-03)
Total	17-/1~/1+	10-/9~		16-/3~	10-/9~	

+, - and ~ denote that the performance of corresponding algorithm is significantly better than, worse than, and similar to MOEA/D-IRA respectively by Wilcoxon's rank sum test with $\alpha = 0.05$.

that our IRA strategy is more effective than GRA and DRA, to enhance MOEA/D on tackling most of the test MOPs adopted. More specifically, MOEA/D-IRA outperforms DRA-I and GRA-I on 17 and 10 out of 19 test problems, respectively. The Wilcoxon's rank sum test also indicates that MOEA/D-IRA performs similarly to DRA-I and GRA-I on one and nine test problems, respectively. DRA-I only beats MOEA/D-IRA on F8. Therefore, the superior performance of IRA over DRA and GRA is justified using IGD. As the only difference of IRA from GRA and DRA is the extra diversity indicator, this validates the statement that the combination of convergence and diversity indicators in the RA strategy can be more reasonable to assign the computational resources to the subproblems.

Based on the HV results in Table IV, MOEA/D-IRA performs best on 16 out of 19 test problems. GRA-I is best on three test problems, while DRA-I cannot perform best on any test problem. As revealed by the Wilcoxon's rank sum test, MOEA/D-IRA obtains statistically similar results to DRA-I and GRA-I on three and nine test problems, and outperforms DRA-I and GRA-I on 16 and 10 test problems, respectively. That is to say, MOEA/D-IRA performs better than or similarly to both DRA-I and GRA-I on all the 19 test problems. Thus, the proposed IRA strategy is shown to be very effective as it improves performance on about half of the test problems, but does not deteriorate on any test problem. Therefore, the advantage of our proposed IRA is also confirmed by HV.

F. Parameter Sensitivity Analysis of Our IRA Strategy

In the proposed IRA strategy, the selection of parameter β will significantly affect the performance of MOEA/D-IRA, as it is an important factor to control the weights of the convergence indicator (i.e., the relative improvement of each subproblem) and the diversity indicator (i.e., the solution density around each subproblem) when computing the selection probability of each subproblem. An appropriate setting of β can properly balance the convergence and the diversity, which benefits the performance enhancement of our algorithm.

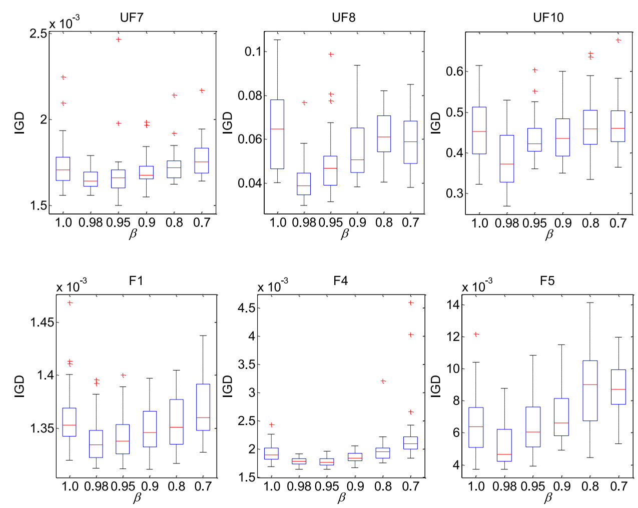


Fig. 7. Parameter sensitivity studies of β.

To study the impact of parameter β in MOEA/D-IRA, we adopted different β values (i.e., 0.7, 0.8, 0.9, 0.95, 0.98, and 1.0) for performance comparison. Since the convergence indicator is still the main contributor to distinguish the difficulties of the subproblems while the diversity indicator is only used as a complement for enhancement, the values of β were set to start from 0.7. The other parameters of MOEA/D-IRA were set the same as mentioned in Section IV-A. For each value of β, 19 test problems were independently run 51 times. Due to page limitations, only the boxplots of the IGD values obtained by six β values on some typical test problems, such as UF7, UF8, UF10, F1, F4, and F5, are provided in Fig. 7.

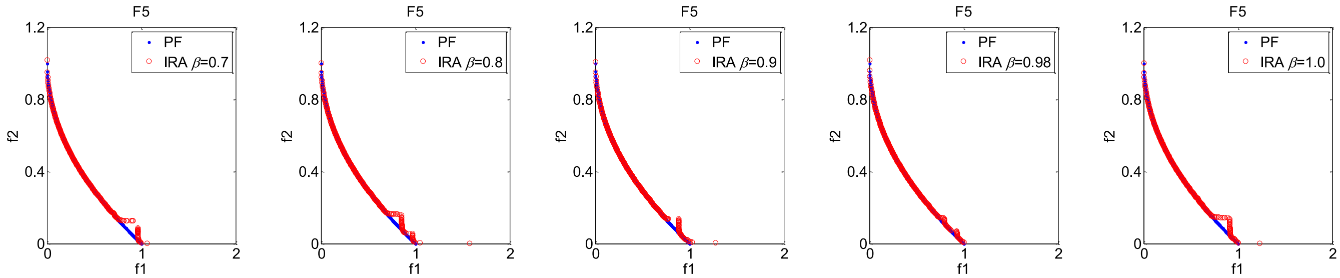


Fig. 8. Plots of final solutions obtained by MOEA/D-IRA with different β on F5.

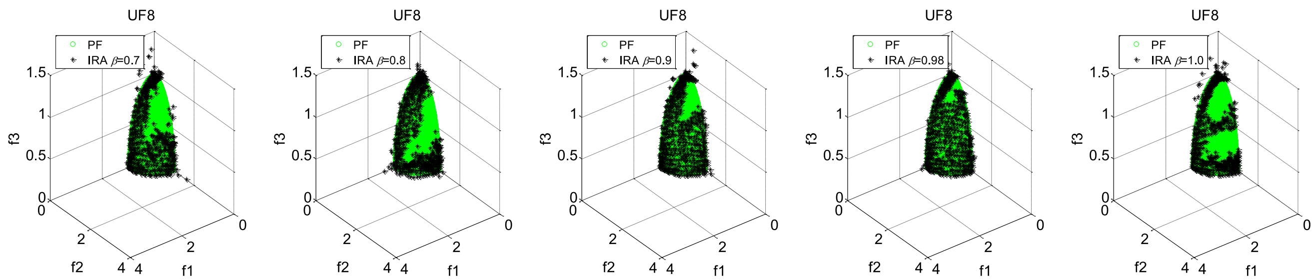


Fig. 9. Plots of final solutions obtained by MOEA/D-IRA with different β on UF8.

As observed from Fig. 7, it is found that MOEA/D-IRA is sensitive to the setting of β. Generally, the performance of MOEA/D-IRA with a large β value is superior to that with a small β value. The IGD values are generally reduced when the values of $\beta(\beta < 1.0)$ are increased. Particularly, when the value of β is set to 1.0 (i.e., removing the diversity indicator), the optimization performance would obviously deteriorate, as its IGD value becomes larger when compared to that obtained by a large value of β less than 1.0 (e.g., 0.98 and 0.95). To visually show their performance, Figs. 8 and 9, respectively, plot the final solutions obtained by MOEA/D-IRA with different values of β, in solving F5 and UF8. It is noted that these plotted solutions were obtained from one run with the median IGD value from 51 runs. On F5, only MOEA/D-IRA with $\beta = 0.98$ can fully approach the true PF, while MOEA/D-IRA with other values of β may miss some parts of the true PF. Regarding UF8, MOEA/D-IRA with different β values may fail to approach some regions of the true PF, as UF8 is more difficult and has three optimization objectives. However, based on the observation of the plots, MOEA/D-IRA with $\beta = 0.98$ covers more regions of the true PF on UF8. This also indicates that the diversity indicator plays an important role in MOEA/D-IRA as a supplement for the convergence indicator. Based on the comparisons of MOEA/D-IRA with different values of β, it is found that a value in the range (0.9, 1.0) is more appropriate for setting β in MOEA/D-IRA. This value makes the diversity indicator work effectively in most cases without having a significant negative effect on convergence, so as to properly keep the balance between convergence and diversity.

G. More Discussion of Our IRA Strategy

Due to page limitations, further discussions of our IRA strategy are provided in the supplementary file of this paper, in order to study the effectiveness of our IRA strategy on a generational version of MOEA/D (i.e., MOEA/D-STM [39]) and on solving other types of test problems (i.e., MOP test problems [19]).

V. Conclusion

This paper proposes a diversity-enhanced RA strategy for decomposition-based MOEAs. The convergence indicator, i.e., the relative improvement of aggregated function value, is still the main factor in our IRA strategy, while the diversity indicator, i.e., the solutions density around each subproblem, is used as a complement to make the resource assignment more reasonable. This way, more computational resources will be assigned to search the sparse area and the subproblems around this region will be enhanced. Such enhancement will also help to improve the neighboring subproblems as MOEA/D is essentially a co-evolutionary framework. Based on the combination of the two above indicators, the proposed IRA strategy can properly balance the convergence and the diversity. After assessed on 19 complicated test MOPs, our algorithm shows advantages over four competitive MOEA/D variants, i.e., MOEA/D-DE, MOEA/D-IR, MOEA/D-DRA, and MOEA/D-GRA, on solving most of the test problems adopted.

Different subproblems decomposed from MOPs may emphasize convergence or diversity when allocating the computational resources. In our future work, an adaptive RA strategy will be further studied for MOEA/D algorithms, without setting any extra parameters. On the other hand, some adaptive control approaches will also be studied in MOEA/D-IRA, such as the multiple evolutionary operators that dynamically allocate the computational resources to the preferred operator.

References

[1] M. G. C. Tapia and C. A. C. Coello, "Applications of multi-objective evolutionary algorithms in economics and finance: A survey," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Singapore, Sep. 2007, pp. 532–539.

[2] A. Ponsich, A. L. Jaimes, and C. A. C. Coello, "A survey on multiobjective evolutionary algorithms for the solution of the portfolio optimization problem and other finance and economics applications," *IEEE Trans. Evol. Comput.*, vol. 17, no. 3, pp. 321–344, Jun. 2013.

[3] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Informat.*, vol. 26, no. 4, pp. 30–45, 1996.

[4] A. Ropponen, R. Ritala, and E. N. Pistikopoulos, "Optimization issues of the broke management system in papermaking," *Comput. Chem. Eng.*, vol. 35, no. 11, pp. 2510–2520, 2011.

[5] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.

[6] A. Zhou *et al.*, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm Evol. Comput.*, vol. 1, no. 1, pp. 32–49, 2011.

[7] Z. Liang *et al.*, "A double-module immune algorithm for multi-objective optimization problems," *Appl. Soft Comput.*, vol. 35, pp. 161–174, Oct. 2015.

[8] N. Srinivas and K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithms," *Evol. Comput.*, vol. 2, no. 3, pp. 221–248, 1994.

[9] K. Deb, A. Pratap, S. Agarwal, 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.

[10] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimisation and Control With Application to Industrial Problems (EUROGEN)*, K. Giannakoglou *et al.*, Eds. Athens, Greece: CIMNE, 2002, pp. 95–100.

[11] M. Basseur and E. Zitzler, "Handling uncertainty in indicator-based multiobjective optimization," *Int. J. Comput. Intell. Res.*, vol. 2, no. 3, pp. 255–272, 2006.

[12] S. Jiang, J. Zhang, Y.-S. Ong, A. N. Zhang, and P. S. Tan, "A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2202–2213, Oct. 2015.

[13] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.

[14] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.

[15] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications," *Theor. Comput. Sci.*, vol. 425, pp. 75–103, Mar. 2012.

[16] 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.

[17] Y.-Y. Tan, Y.-Y. Jiao, H. Li, and X.-W. Wang, "A modification to MOEA/D-DE for multiobjective optimization problems with complicated Pareto sets," *Inf. Sci.*, vol. 213, pp. 14–38, Dec. 2012.

[18] L. Ke, Q. Zhang, and R. Battiti, "MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant colony," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec. 2013.

[19] 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. 2014.

[20] Y.-L. Li, Y.-R. Zhou, Z.-H. Zhan, and J. Zhang, "A primary theoretical study on decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 20, no. 4, pp. 563–576, Aug. 2016.

[21] R. Wang, Q. Zhang, and T. Zhang, "Decomposition-based algorithms using Pareto adaptive scalarizing methods," *IEEE Trans. Evol. Comput.*, vol. 20, no. 6, pp. 821–837, Dec. 2016.

[22] S. Jiang and S. Yang, "An improved multiobjective optimization evolutionary algorithm based on decomposition for complex Pareto fronts," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 421–437, Feb. 2016.

[23] K. Li, Q. Zhang, and R. Battiti, "Hybridization of decomposition and local search for multiobjective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 10, pp. 1808–1820, Oct. 2014.

[24] X. Y. Cai, Y. X. Li, Z. Fan, and Q. Zhang, "An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 508–523, Aug. 2015.

[25] L. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Constrained subproblems in a decomposition-based multiobjective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 20, no. 3, pp. 475–480, Jun. 2016.

[26] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.

[27] Y. Y. Tan, Y. C. Jiao, H. Li, and X. K. Wang, "MOEA/D+ uniform design: A new version of MOEA/D for optimization problems with many objectives," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1648–1660, 2013.

[28] X. Ma *et al.*, "MOEA/D with uniform decomposition measurement for many-objective problems," *Soft Comput.*, vol. 18, no. 12, pp. 2541–2564, 2014.

[29] R. Wang, T. Zhang, and B. Guo, "An enhanced MOEA/D using uniform directions and a pre-organization procedure," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Cancún, Mexico, 2013, pp. 2390–2397.

[30] S. Jiang, Z. Cai, J. Zhang, and Y.-S. Ong, "Multiobjective optimization by decomposition with Pareto-adaptive weight vectors," in *Proc. 7th Int. Conf. Nat. Comput. (ICNC)*, vol. 3. Shanghai, China, 2011, pp. 1260–1264.

[31] Y. Qi *et al.*, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, Jun. 2014.

[32] S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood sizes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 442–446, Jun. 2012.

[33] Q. Zhang, W. Liu, and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Trondheim, Norway, 2009, pp. 203–208.

[34] A. Zhou and Q. Zhang, "Are all the subproblems equally important? Resource allocation in decomposition-based multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 52–64, Feb. 2016.

[35] 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.

[36] K. Li, Á. Fialho, S. Kwong, and Q. Zhang, "Adaptive operator selection with bandits for a multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 114–130, Feb. 2014.

[37] S. M. Venske, R. A. Gonçalves, and M. R. Delgado, "ADEMO/D: Multiobjective optimization by an adaptive differential evolution algorithm," *Neurocomputing*, vol. 127, pp. 65–77, Mar. 2014.

[38] Q. Lin *et al.*, "Adaptive composite operator selection and parameter control for multiobjective evolutionary algorithm," *Inf. Sci.*, vol. 339, pp. 332–352, Apr. 2016.

[39] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 6, pp. 909–923, Dec. 2014.

[40] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2076–2088, Oct. 2015.

[41] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for MOEA/D," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 474–486, Feb. 2016.

[42] 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.

[43] Y. Lei, M. Gong, J. Zhang, W. Li, and L. Jiao, "Resource allocation model and double-sphere crowding distance for evolutionary multi-objective optimization," *Eur. J. Oper. Res.*, vol. 234, no. 1, pp. 197–208, 2014.

[44] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, 2002.

[45] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.

[46] K. Li *et al.*, "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm," *Inf. Sci.*, vol. 182, no. 1, pp. 220–242, 2012.

[47] Q. F. Zhang *et al.*, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., Tech. Rep. CES-487, 2009.

[48] 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 evolutionary computation," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[49] W. Gong and Z. Cai, "Differential evolution with ranking-based mutation operators," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 2066–2081, Dec. 2013.

[50] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[51] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[52] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.

Qiuzhen Lin received the B.S. degree from Zhaoqing University, Zhaoqing, China, the M.S. degree from Shenzhen University, Shenzhen, China, in 2007 and 2010, respectively, and the Ph.D. degree from the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, in 2014.

He is currently a Lecturer with the College of Computer Science and Software Engineering, Shenzhen University. He has published over 20 research papers. His current research interests include artificial immune system, multiobjective optimization, and dynamic system.

Genmiao Jin received the B.S. degree from East China Jiaotong University, Nanchang, China, in 2012, and the M.S. degree from Shenzhen University, Shenzhen, China, in 2017.

He is currently a Software Engineer with China Merchants Bank, Shenzhen.

Yueping Ma received the B.S. degree from Zhejiang Normal University, Jinhua, China, in 2014, and the M.S. degree from Shenzhen University, Shenzhen, China, in 2017.

He is currently a Research Assistant with Shenzhen University. His current research interests include evolutionary computation and recommender system.

Ka-Chun Wong received the B.Eng. degree in computer engineering and the M.Phil. degree from the Chinese University of Hong Kong, Hong Kong, in 2008 and 2010, respectively, and the Ph.D. degree from the Department of Computer Science, University of Toronto, Toronto, ON, Canada, in 2015.

He is as Assistant Professor with the City University of Hong Kong. His current research interests include bioinformatics, computational biology, evolutionary computation, data mining, machine learning, and interdisciplinary research.

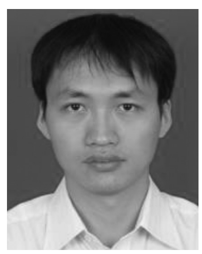
Carlos A. Coello Coello (M'98–SM'04–F'11) received the Ph.D. degree in computer science from Tulane University, New Orleans, LA, USA, in 1996.

He is currently a Professor (CINVESTAV-3F Researcher) with the Computer Science Department, CINVESTAV-IPN, Mexico City, Mexico. He has authored and co-authored over 450 technical papers and book chapters. He has also co-authored a book entitled *Evolutionary Algorithms for Solving Multi-Objective Problems* (2nd edition, Springer, 2007). His publications currently report over 34 000 citations in Google Scholar with an H-index of 76. His current research interests include evolutionary multiobjective optimization and constraint-handling techniques for evolutionary algorithms.

Dr. Coello Coello was a recipient of the 2007 National Research Award from the Mexican Academy of Sciences in the area of exact sciences, and the 2013 IEEE Kiyo Tomiyasu Award and the 2012 National Medal of Science and Arts in the area of physical, mathematical, and natural sciences. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and serves on the editorial board of 12 other international journals. He is a member of ACM and the Mexican Academy of Science.

Jianqiang Li received the B.S. and Ph.D. degrees from the South China University of Technology, Guangzhou, China, in 2003 and 2008, respectively.

He is an Associate Professor with the College of Computer and Software Engineering, Shenzhen University, Shenzhen, China. He led a project of the National Natural Science Foundation and the Natural Science Foundation of Guangdong Province, China. His current research interests include robotic, hybrid systems, Internet of Things, and embedded systems.



Jianyong Chen (M'11) received the Ph.D. degree from the City University of Hong Kong, Hong Kong, in 2003.

He is a Professor with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He was a Senior Engineer of network technology with ZTE Corporation, Shenzhen, from 2003 to 2006. He was the Vice Chairman of the International Telecommunication Union-Telecommunication (ITU-T) SG17 from 2004 to 2012, and has edited three recommendations developed in ITU-T SG17. He has published over 30 papers and holds over 30 patents in the field of artificial intelligence and information security. His current research interests include artificial intelligence and information security.

Jun Zhang (M'02–SM'08–F'17) received the Ph.D. degree in electrical engineering from the City University of Hong Kong, Hong Kong, in 2002.

From 2004 to 2016, he was a Professor with Sun Yat-sen University, Guangzhou, China. Since 2016, he has been with the South China University of Technology, Guangzhou, where he is currently a Cheung Kong Chair Professor. His current research interests include computational intelligence, cloud computing, big data, high performance computing, data mining, wireless sensor networks, operations research, and power electronic circuits. He has authored seven research books and book chapters, and over 100 technical papers in the above areas.

Prof. Zhang was a recipient of the China National Funds for Distinguished Young Scientists from the National Natural Science Foundation of China in 2011, and the First-Grade Award in Natural Science Research from the Ministry of Education, China, in 2009. He is currently an Associate Editor of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, and the IEEE TRANSACTIONS ON CYBERNETICS. He is the Founding and Current Chair of the IEEE Guangzhou Subsection, the IEEE Beijing (Guangzhou) Section Computational Intelligence Society Chapters, and ACM Guangzhou Chapter.