

# Balancing Convergence and Diversity in Decomposition-Based Many-Objective Optimizers

Yuan Yuan, Hua Xu, Bo Wang, Bo Zhang, and Xin Yao, *Fellow, IEEE*

**Abstract**—The decomposition-based multiobjective evolutionary algorithms (MOEAs) generally make use of aggregation functions to decompose a multiobjective optimization problem into multiple single-objective optimization problems. However, due to the nature of contour lines for the adopted aggregation functions, they usually fail to preserve the diversity in high-dimensional objective space even by using diverse weight vectors. To address this problem, we propose to maintain the desired diversity of solutions in their evolutionary process explicitly by exploiting the perpendicular distance from the solution to the weight vector in the objective space, which achieves better balance between convergence and diversity in many-objective optimization. The idea is implemented to enhance two well-performing decomposition-based algorithms, i.e., MOEA, based on decomposition and ensemble fitness ranking. The two enhanced algorithms are compared to several state-of-the-art algorithms and a series of comparative experiments are conducted on a number of test problems from two well-known test suites. The experimental results show that the two proposed algorithms are generally more effective than their predecessors in balancing convergence and diversity, and they are also very competitive against other existing algorithms for solving many-objective optimization problems.

**Index Terms**—Convergence, decomposition, diversity, many-objective optimization, multiobjective optimization.

## I. INTRODUCTION

VARIOUS multiobjective evolutionary algorithms (MOEAs) have been proposed to effectively solve

Manuscript received August 4, 2014; revised December 6, 2014; March 2, 2015, and May 22, 2015; accepted May 28, 2015. Date of publication June 9, 2015; date of current version March 29, 2016. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316301; in part by the National Natural Science Foundation of China under Grant 61175110, Grant 61329302, and Grant 61305079; in part by the National S&T Major Projects of China under Grant 2011ZX02101-004; in part by the National Banking Information Technology Risk Management Projects of China; and in part by the China Scholarship Council. The work of X. Yao was supported in part by the Engineering and Physical Sciences Research Council under Grant EP/J017515/1 and in part by the Royal Society Wolfson Research Merit Award. This work was done while the first author visited the Centre of Excellence for Research in Computational Intelligence and Applications.

Y. Yuan, H. Xu, B. Wang, and B. Zhang are with State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: yyxhdy@gmail.com; xuhua@tsinghua.edu.cn; wbthu@outlook.com; dcszb@tsinghua.edu.cn).

X. Yao is with the Centre of Excellence for Research in Computational Intelligence and Applications, School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K. (e-mail: x.yao@cs.bham.ac.uk).

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/TEVC.2015.2443001

multiobjective optimization problems (MOPs). However, most of these algorithms have been evaluated and applied to MOPs with only two or three objectives [1], [2], despite the fact that optimization problems involving a high number of objectives indeed appear widely in real-world applications [3]–[5]. In the literature, such MOPs that have more than three objectives are often termed *many-objective optimization problems* (MaOPs) [6], [7].

Unfortunately, experimental and analytical results [8], [9] have indicated that MaOPs would pose serious difficulties to the existing MOEAs, particularly for the popular Pareto-dominance-based MOEAs, e.g., nondominated sorting genetic algorithm II (NSGA-II) [10] and strength Pareto evolutionary algorithm 2 (SPEA2) [11]. The primary reason for this is that the proportion of nondominated solutions in a population increases exponentially with the number of objectives and it would lead to the severe loss of Pareto-dominance-based selection pressure toward the Pareto front (PF). In the existing research, there generally exist three types of techniques to overcome the shortage of Pareto-dominance-based MOEAs.

The idea of the first type of approaches is to adopt new preference relations (relaxed forms of Pareto dominance or other ranking schemes) that could induce finer grain order in the objective space. So far, many alternative preference relations have been proposed for many-objective optimization, such as average and maximum ranking [12], favor relation [13], fuzzy Pareto dominance [14], expansion relation [15], preference order ranking [16],  $\theta$ -dominance [17], and so on [18]–[20]. The strength and weakness of different preference relations on MaOPs have been experimentally investigated to a certain degree by several similar studies [21]–[23].

The second type of approach intends to improve the diversity maintenance mechanism. The consideration is that since Pareto dominance could not exert sufficient selection pressure toward the PF in a high-dimensional objective space, then the selection should be almost solely based on the diversity, which is generally regarded as the secondary selection operator in MOEAs. However, most existing diversity maintenance criteria, e.g., crowding distance [10], prefer dominance resistant [8] solutions, which would bias the search toward the solutions with the poor proximity to the global PF, although these solutions may present good diversity over the objective space. Thus, the diversity preservation in many-objective scenario requires some care to avoid or weaken this phenomenon. Compared to the first type of research, the

work along this direction [24]–[26] has received much less attention.

In many-objective optimization, it is indeed very difficult for a MOEA to emphasize both convergence and diversity within a single population or archive. Motivated by this, the third type of approaches maintains two archives for convergence and diversity separately during the evolutionary search, and at least one of the two archives still uses Pareto dominance. The related work can be referred in [27] and [28].

Unlike Pareto-dominance-based MOEAs, another two classes of MOEAs, indicator- and decomposition-based MOEAs, have not been criticized much on the selection pressure produced in a high-dimensional objective space. However, they would encounter their own difficulties when handling MaOPs.

The indicator-based MOEAs aim to optimize a performance indicator that provides a desired ordering among sets that represent PF approximations. The hypervolume (HV) [1] may be the most popular performance indicator ever adopted due to its nice theoretical properties [29]. There have been a few well-established indicator-based MOEAs [30]–[32] available in the literature that employ the HV as the selection criterion. Nevertheless, the computational cost of HV grows exponentially with the number of objectives increasing. To address this issue, one of the alternative strategies is to estimate the rankings of solutions induced by the HV indicator without computing the exact indicator values [33]. Another strategy is to replace the HV with the other indicators that are much less computationally expensive but also have good theoretical properties, e.g.,  $R2$  [34] and  $\Delta_p$  [35]. This strategy has been implemented in some recent indicator-based MOEAs [36]–[38]. However, such kind of algorithms now still perform unsatisfactorily on MaOPs [17].

The decomposition-based MOEAs generally aggregate the objectives of a MOP into an aggregation function using a unique weight vector, and a set of weight vectors (or directions) will generate multiple aggregation functions, each of which defines a single-objective optimization problem. The diversity of the population is implicitly maintained by specifying a good spread of the weight vectors in the objective space. MOEA based on decomposition (MOEA/D) [39], [40] and multiple single-objective Pareto sampling (MSOPS) [41] are two most typical decomposition-based MOEAs. Very recently, Yuan *et al.* [42] proposed a new framework based on NSGA-II for many-objective optimization, called ensemble fitness ranking (EFR), which is more general than MSOPS. In terms of the form, EFR can be viewed as an extension to average ranking and maximum ranking [12]. The significant difference lies in that EFR employs more general fitness functions (aggregation functions) instead of objective functions. However, there is still deficiency for decomposition-based MOEAs to conduct many-objective search. Take MOEA/D for instance, it normally could approach the PF very well even if in the high-dimensional objective space, but it struggles to maintain the diversity in many-objective optimization, usually failing to achieve a good coverage of PF [20], [26], [43]. The reason is largely due to the nature of contour lines for the adopted aggregation functions, which will be further explained

in Section III. The other decomposition-based MOEAs, e.g., MSOPS and EFR, suffer from a similar problem, since they also drive the search using aggregation functions. Note that, as a major framework to design MOEAs, MOEA/D has spawned a large amount of research work (see [44]–[47]), but almost all the newly proposed MOEA/D variants are yet to be validated on MaOPs to evaluate their suitability in many-objective optimization. Moreover, several recent decomposition-based MOEAs, e.g., MOEA/D-M2M [47] and NSGA-III [26], adopt new decomposition scheme, which decomposes the objective space into subregions without using any aggregation function. Although these algorithms address the difficulty in diversity maintenance to some extent, they struggle to converge to the PF in many-objective optimization as the Pareto dominance is still relied on to provide the selection pressure.

This paper mainly focuses on the decomposition-based MOEAs that use aggregation functions to promote the convergence in many-objective optimization. Our contributions to this topic are summarized as follows.

- 1) We have proposed a better tradeoff between convergence and diversity in decomposition-based many-objective optimizers by exploiting the perpendicular distance from a solution to the weight vector in the objective space.
- 2) We have implemented this idea in MOEA/D and EFR, respectively, to improve their performance on MaOPs, resulting in two enhanced algorithms, i.e., a MOEA/D variant (MOEA/D-DU) with a distance-based updating strategy and a new EFR version (EFR-RR) with a ranking restriction scheme.
- 3) We have provided an optional online normalization procedure that can be incorporated into MOEA/D-DU and EFR-RR to effectively tackle scaling problems, which is a new practice [26], [48] and has not been extensively studied in the literature.
- 4) We have combined for the first time different existing ideas together synergistically in a single MOEA. For example, MOEA/D-DU inherits the merit of systematic sampling (the same as [26]), notion of neighborhood (the same as [40]), adaptive normalization (similar to [26] and [48]), single first replacement (the same as [48]), and the preferred order of replacement (introduced in this paper).
- 5) We have discussed about the similarities and differences between our proposals and the other existing decomposition-based approaches. Based on the extensive experimental studies, we have also given possible explanations on why some of these approaches are inferior to ours in many-objective optimization.
- 6) Based on our experimental studies, we have suggested a practice to compare the performance of MOEAs (with or without sophisticated normalization) on benchmark problems more fairly and reasonably.

The rest of this paper is organized as follows. In Section II, the background knowledge is introduced. In Section III, the basic idea is given based on the analysis of the drawback of original MOEA/D and EFR. Section IV describes in detail how to implement the basic idea to enhance MOEA/D and EFR in many-objective optimization, respectively. Section V presents

the test problems, performance metrics, and algorithm settings used for performance comparison. Section VI experimentally investigates the performance of enhanced algorithms from two aspects. In Section VII, the extensive experiments are carried out to compare the enhanced algorithms with state-of-the-art methods. Finally, Section VIII concludes this paper.

## II. PRELIMINARIES AND BACKGROUND

In this section, some basic concepts in multiobjective optimization are first given. Then, we briefly introduce several most recent proposals in the literature that are somewhat similar to ours.

### A. Basic Concepts

A MOP can be mathematically defined as

$$\begin{aligned} \min \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ \text{subject to} \quad & \mathbf{x} \in \Omega \subseteq \mathbb{R}^n \end{aligned} \quad (1)$$

where  $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$  is a  $n$ -dimensional decision variable vector in the decision space  $\Omega$ ;  $\mathbf{f} : \Omega \rightarrow \Theta \subseteq \mathbb{R}^m$  is an objective vector consisting of  $m$  objective functions, which maps  $n$ -dimensional decision space  $\Omega$  to  $m$ -dimensional attainable objective space  $\Theta$ .

**Definition 1:** A vector  $\mathbf{u} = (u_1, u_2, \dots, u_k)^T$  is said to dominate another vector  $\mathbf{v} = (v_1, v_2, \dots, v_k)^T$ , denoted by  $\mathbf{u} \leq \mathbf{v}$ , iff  $\forall i \in \{1, 2, \dots, k\}, u_i \leq v_i$  and  $\exists j \in \{1, 2, \dots, k\}, u_j < v_j$ .

**Definition 2:** For a given MOP, a decision vector  $\mathbf{x}^* \in \Omega$  is said to be Pareto optimal, iff  $\nexists \mathbf{x} \in \Omega, \mathbf{f}(\mathbf{x}) \leq \mathbf{f}(\mathbf{x}^*)$ .

**Definition 3:** For a given MOP, the Pareto set, PS, is defined as

$$\text{PS} = \{\mathbf{x} \in \Omega | \mathbf{x} \text{ is Pareto optimal}\}. \quad (2)$$

**Definition 4:** For a given MOP, the PF is defined as

$$\text{PF} = \{\mathbf{f}(\mathbf{x}) \in \Theta | \mathbf{x} \in \text{PS}\}. \quad (3)$$

**Definition 5:** For a given MOP, the ideal point  $\mathbf{z}^*$  is a vector  $\mathbf{z}^* = (z_1^*, z_2^*, \dots, z_m^*)^T$ , where  $z_i^*$  is the infimum of  $f_i$ , for each  $i \in \{1, 2, \dots, m\}$ .

**Definition 6:** For a given MOP, the nadir point  $\mathbf{z}^{\text{nad}}$  is a vector  $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, z_2^{\text{nad}}, \dots, z_m^{\text{nad}})^T$ , where  $z_i^{\text{nad}}$  is the supremum of  $f_i$  over the PS, for each  $i \in \{1, 2, \dots, m\}$ .

The goal of multiobjective optimization is to find a set of nondominated objective vectors that are close to the PF (convergence), and also enable them to distribute well over the PF (diversity).

### B. Related Approaches

In this section, we would review several representative decomposition-based approaches relating to this paper.

- 1) Qi *et al.* [45] proposed an improved MOEA/D with the adaptive weight vector adjustment (MOEA/D-AWA), which mainly enhances MOEA/D in two aspects. First, MOEA/D-AWA uses a new weight vector initialization method based on the geometric analysis of the original Tchebycheff function. Second, MOEA/D-AWA adopts an adaptive weight vector adjustment strategy to deal

with problems having complex PFs. This strategy adjusts weights periodically so that weights of subproblems can be redistributed adaptively to obtain better uniformity of solutions.

- 2) Li *et al.* [46] proposed to use a stable matching (STM) model to coordinate the selection process in MOEA/D, and the resulting MOEA/D variant is referred to as MOEA/D-STM. In MOEA/D-STM, the subproblems and solutions are regarded as two sets of agents. A subproblem prefers solutions which can lower its aggregation function values, while a solution prefers subproblems whose corresponding direction vectors are close to it. To balance the preference of subproblems and solutions, a matching algorithm is used to assign a solution to each subproblem in order to balance the convergence and diversity of the evolutionary search. Unlike MOEA/D, MOEA/D-STM uses the generational scheme.
- 3) Deb and Himanshu [26] suggested a reference-point-based many-objective NSGA-II, called NSGA-III. Similar to NSGA-II, NSGA-III still employs the Pareto nondominated sorting to partition the population into a number of nondomination fronts. In the last accepted front, instead of using the crowding distance, the solutions are selected based on a niche-preservation operator, where the solutions associated with less crowded reference line have higher priority to be chosen. Overall, NSGA-III emphasizes the nondominated solutions yet close to the reference lines of a set of supplied reference points. A sophisticated online normalization procedure is incorporated into NSGA-III to effectively handle scaled test problems.

Just around the time when this paper was originally submitted, the following two proposals have been available in the literature.

- 1) Asafuddoula *et al.* [48] proposed an improved decomposition-based evolutionary algorithm (I-DBEA) for many-objective optimization. In I-DBEA, the child solution attempts to enter the population via replacement only if it is nondominated with respect to the individuals in the population. The child solution competes with all the solutions in the population in a random order until it makes a successful replacement or all the solutions have been competed. I-DBEA also includes an online normalization procedure similar to that in NSGA-III.
- 2) Wang *et al.* [49] presented a MOEA/D variant with the global replacement (MOEA/D-GR). Once a new solution is produced in MOEA/D-GR, it is associated with a subproblem on which it achieves the minimum aggregation function value. Then, a number of closest subproblems to the associated subproblem are selected to form the replacement neighborhood, and the new solution attempts to replace the current solutions to these selected subproblems.

Note that, MOEA/D-STM and MOEA/D-GR were studied and verified only on 2- and 3-objective problems. MOEA/D-AWA was studied on MaOPs, but the problems were just limited to degenerate problems. NSGA-III and I-DBEA



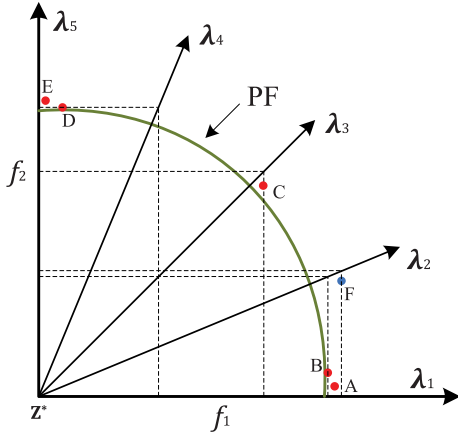


Fig. 1. Illustration of the distribution of solutions in the 2-D objective space.

were designed specially for many-objective optimization. Moreover, although NSGA-III and I-DBEA use the decomposition-based idea, they still use the Pareto dominance instead of aggregation functions to control the convergence, which differ significantly from the other mentioned decomposition-based approaches.

### III. BASIC IDEA

The Tchebycheff function is one of the most common type of aggregation functions used in decomposition-based MOEAs. In this paper, we adopt a modified version of the Tchebycheff function. Let  $\lambda_j = (\lambda_{j,1}, \lambda_{j,2}, \dots, \lambda_{j,m})^T$ ,  $j = 1, 2, \dots, N$ , be a set of uniformly spread weight vectors and  $\mathbf{z}^*$  be the ideal point, then the function for the  $j$ th subproblem can be defined as

$$\mathcal{F}_j(\mathbf{x}) = \max_{k=1}^m \left\{ \frac{1}{\lambda_{j,k}} |f_k(\mathbf{x}) - z_k^*| \right\} \quad (4)$$

where  $\lambda_{j,k} \geq 0$  for all  $k \in \{1, 2, \dots, m\}$  and  $\sum_{k=1}^m \lambda_{j,k} = 1$ . If  $\lambda_{j,k} = 0$ ,  $\lambda_{j,k}$  is set to  $10^{-6}$ .

This form of Tchebycheff function has two advantages over the original one [39]. First, the uniformly distributed weight vectors lead to the uniformly distributed search directions in the objective space. Second, each weight vector corresponds to a unique solution located on PF. The proof can be found in [45]. The two advantages alleviate the difficulty in diversity preservation to some extent.

However, even (4) is not perfect in practice. Ideally, if the optimal solution is obtained for each  $\mathcal{F}_j$  defined in (4), then the most desired diversity is meanwhile achieved. But it is indeed not the case for MOEA/D and EFR. On the one hand, the final solutions yielded by them are often just near optimal ones, which could not fully ensure the diversity of the final population. For example, Fig. 1 illustrates the distribution of obtained solutions (A–E) aided by five weight vectors ( $\lambda_1, \lambda_2, \dots, \lambda_5$ ) in the bi-objective space, where the dashed lines depict the contour lines of subproblems decomposed by (4). It can be seen that the solutions do not spread as well as the weight vectors. This is because that although B and D achieve good values on  $\mathcal{F}_2$  and  $\mathcal{F}_4$ , respectively, both of them deviate far from their corresponding directions ( $\lambda_2$  and  $\lambda_4$ ).

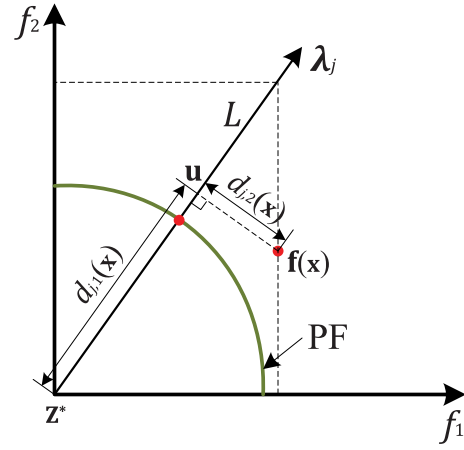


Fig. 2. Illustration of the perpendicular distance from the solution to the weight vector in the 2-D objective space.

On the other hand, the more important issue arises in the evolutionary process, where the selection of solutions would be misled if only depending on aggregation function values. For instance, in Fig. 1, there is another solution F, which is a little worse than B in terms of  $\mathcal{F}_2$  value. In MOEA/D, it is very possible for F to be replaced by B in the updating procedure. While in EFR, since B achieves the better ranking than F on  $\mathcal{F}_2$ , F may be removed in the environmental selection. But intuitively, F is in fact more preferred by the weight vector  $\lambda_2$ . It is worth noting that, in the early stage of evolution, the solutions are usually far away from the PF, and the misleading selection would be more likely to occur, which may pull the search to only a part of the PF.

Both situations mentioned above could be attributed to the fact that the solution which is distant to  $\lambda_j$  could also achieve relatively good  $\mathcal{F}_j$  value. The contour lines of (4) is responsible for this problem, and the problem would be aggravated in the high-dimensional objective space due to the sparse distribution of solutions and the exponentially growing HV.

Given all these, we are motivated to consider not only the aggregation function value of a solution but also its distance to the corresponding weight vector in MOEA/D and EFR. This practice is expected to force the solutions to stay close to the weight vectors, and explicitly maintain the desired distribution of solutions in the evolutionary process, leading to the better balance between convergence and diversity in many-objective optimization. It is worth mentioning that the penalty-based boundary intersection function [39] implicitly considers the closeness of a solution to the weight vector to certain extent, but it still has the problem mentioned above. In this paper, without loss of generality, we only consider the modified Tchebycheff function defined in (4).

Suppose that  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T$  is the objective vector of solution  $\mathbf{x}$ ,  $L$  is a line passing through  $\mathbf{z}^*$  with direction  $\lambda_j$ , and  $\mathbf{u}$  is the projection of  $\mathbf{f}(\mathbf{x})$  on  $L$ . Let  $d_{j,2}(\mathbf{x})$  be the perpendicular distance from the solution  $\mathbf{x}$  to the weight vector  $\lambda_j$  in the objective space, then it can be computed as

$$d_{j,2}(\mathbf{x}) = \|\mathbf{f}(\mathbf{x}) - \mathbf{z}^* - d_{j,1}(\mathbf{x})(\lambda_j / \|\lambda_j\|)\| \quad (5)$$

where  $d_{j,1}(\mathbf{x})$  is the distance between  $\mathbf{z}^*$  and  $\mathbf{u}$ , and can be obtained by

$$d_{j,1}(\mathbf{x}) = \left\| (\mathbf{f}(\mathbf{x}) - \mathbf{z}^*)^T \boldsymbol{\lambda}_j \right\| / \|\boldsymbol{\lambda}_j\|. \quad (6)$$

In Fig. 2, the perpendicular distance  $d_{j,2}(\mathbf{x})$  is illustrated in the 2-D objective space. In the following, we will describe how to exploit  $d_{j,2}(\mathbf{x})$  to enhance MOEA/D and EFR, respectively.

#### IV. PROPOSED ALGORITHMS

In this section, two enhanced algorithms, i.e., MOEA/D-DU and EFR-RR, are described in detail in Sections IV-A and IV-B, respectively. Section IV-C provides an optional normalization procedure that can be incorporated into the proposed algorithms. Section IV-D briefly analyzes the computational complexity of one generation of MOEA/D-DU and EFR-RR. In Section IV-E, we discuss the similarities and differences between the proposed algorithms and several existing ones in the literature.

##### A. Enhancing MOEA/D

The framework of the proposed MOEA/D-DU is depicted in Algorithm 1. First, a set of uniformly spread weight vectors  $\Lambda = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \dots, \boldsymbol{\lambda}_N\}$  is generated, where  $\boldsymbol{\lambda}_j$  determines the  $j$ th subproblem, i.e.,  $\mathcal{F}_j$ . In MOEA/D-DU, the Das and Dennis's systematic approach [50] is adopted to produce the diverse weight vectors just as in NSGA-III [26]. Then, a population of  $N$  solutions  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$  is initialized, where  $\mathbf{x}_j$  is the current solution to the  $j$ th subproblem. The ideal point  $\mathbf{z}^*$  is initialized in step 3. Because it is often very time-consuming to compute exact  $z_i^*$ , it is indeed estimated by the minimum value found so far for objective  $f_i$ , and is updated during the search. MOEA/D-DU still uses the same mating restriction scheme to generate child solutions as that in its predecessor [40], hence the neighborhood  $B(i)$  for each subproblem  $\mathcal{F}_i$  is determined in steps 4–6. Steps 7–18 are iterated until the termination criterion is met. At each iteration, for the solution  $\mathbf{x}_i$ , the mating solution  $\mathbf{x}_k$  is chosen from the neighborhood  $B(i)$  with a probability  $\delta$  and the whole population with a probability  $1 - \delta$  (see steps 9–12 of Algorithm 1). Then the genetic operators, i.e., simulated binary crossover (SBX) and polynomial mutation [51], are executed on them to generate a new solution  $\mathbf{y}$ , and finally  $\mathbf{y}$  is used to update the ideal point and the current population.

The updating strategy (step 17 of Algorithm 1) is the characteristic procedure in MOEA/D-DU, which is significantly different from the original MOEA/D. This strategy is presented in detail in Algorithm 2, which works as follows. Once a new solution  $\mathbf{y}$  is produced, its perpendicular distances to each weight vector  $\boldsymbol{\lambda}_j$ , i.e.,  $d_{j,2}(\mathbf{y})$ ,  $j = 1, 2, \dots, N$ , are computed respectively. Then  $K$  minimum distances are selected from all these  $N$  distances, where  $K \ll N$  is a parameter. Suppose the  $K$  minimum distances are  $d_{j_1,2}(\mathbf{y}), d_{j_2,2}(\mathbf{y}), \dots, d_{j_K,2}(\mathbf{y})$ , and are arranged in the nondecreasing order, i.e.,  $d_{j_1,2}(\mathbf{y}) \leq d_{j_2,2}(\mathbf{y}) \leq \dots \leq d_{j_K,2}(\mathbf{y})$ . The solution  $\mathbf{y}$  is compared with the solutions  $\mathbf{x}_{j_1}, \mathbf{x}_{j_2}, \dots, \mathbf{x}_{j_K}$  one by one. If one solution  $\mathbf{x}_{j_k}$ ,  $k \in \{1, 2, \dots, K\}$ , satisfies that  $\mathcal{F}_{j_k}(\mathbf{x}_{j_k})$  is greater than  $\mathcal{F}_{j_k}(\mathbf{y})$ , then the solution  $\mathbf{x}_{j_k}$  is replaced by the solution  $\mathbf{y}$ , and the

##### Algorithm 1 Framework of MOEA/D-DU

---

```

1: Generate a set of weight vectors  $\Lambda \leftarrow \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \dots, \boldsymbol{\lambda}_N\}$ 
2: Initialize the population  $P \leftarrow \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ 
3: Initialize the ideal point  $\mathbf{z}^* \leftarrow (z_1^*, z_2^*, \dots, z_m^*)^T$ 
4: for  $i \leftarrow 1$  to  $N$  do
5:    $B(i) \leftarrow \{i_1, i_2, \dots, i_T\}$ , where  $\boldsymbol{\lambda}_{i_1}, \boldsymbol{\lambda}_{i_2}, \dots, \boldsymbol{\lambda}_{i_T}$  are  $T$ 
     closest weight vectors to  $\boldsymbol{\lambda}_i$ .
6: end for
7: while the termination criterion is not satisfied do
8:   for  $i \leftarrow 1$  to  $N$  do
9:     if  $\text{rand}() < \delta$  then
10:       $E \leftarrow B(i)$ 
11:    else
12:       $E \leftarrow \{1, 2, \dots, N\}$ 
13:    end if
14:    Randomly select an index  $k \in E$  and  $k \neq i$ 
15:     $\mathbf{y} \leftarrow \text{GeneticOperators}(\mathbf{x}_i, \mathbf{x}_k)$ 
16:     $\text{UpdateIdealPoint}(\mathbf{y}, \mathbf{z}^*)$ 
17:     $\text{UpdateCurrentPopulation}(\mathbf{y}, \mathbf{z}^*, \Lambda, P, K)$ 
18:  end for
19: end while
20: return all the nondominated solutions in  $P$ 
```

---

##### Algorithm 2 UpdateCurrentPopulation( $\mathbf{y}, \mathbf{z}^*, \Lambda, P, K$ )

---

```

1: for  $j \leftarrow 1$  to  $N$  do
2:   Compute the perpendicular distance from  $\mathbf{y}$  to the
     weight vector  $\boldsymbol{\lambda}_j$ , i.e.,  $d_{j,2}(\mathbf{y})$ 
3: end for
4: Select  $K$  minimum distances from  $N$  distances  $d_{j,2}(\mathbf{y})$ ,
    $j = 1, 2, \dots, N$ , and obtain  $d_{j_1,2}(\mathbf{y}) \leq d_{j_2,2}(\mathbf{y}) \leq \dots \leq$ 
    $d_{j_K,2}(\mathbf{y})$ 
5: for  $k \leftarrow 1$  to  $K$  do
6:   if  $\mathcal{F}_{j_k}(\mathbf{y}) < \mathcal{F}_{j_k}(\mathbf{x}_{j_k})$  then
7:      $\mathbf{x}_{j_k} \leftarrow \mathbf{y}$ 
8:   return
9:   end if
10: end for
```

---

updating procedure is terminated. From the above, it can be seen that, MOEA/D-DU uses the steady-state form, where at most one solution in the current population can be replaced by the newly generated solution  $\mathbf{y}$ .

##### B. Enhancing EFR

EFR [42] adopts the generational scheme, and its framework is based on NSGA-II but with significant differences in the environmental selection phase.

At the  $t$ th generation of EFR, the parent population  $P_t$  (of size  $N$ ) and the offspring population  $Q_t$  (of size  $N$ ) are first merged as the population  $U_t = P_t \cup Q_t$  (of size  $2N$ ). Then the best  $N$  solutions would be chosen from  $U_t$  as the next population, which can be described as follows. Each solution  $\mathbf{x}$  in  $U_t$  is associated with  $N$  different fitness values  $\mathcal{F}_1(\mathbf{x}), \mathcal{F}_2(\mathbf{x}), \dots, \mathcal{F}_N(\mathbf{x})$ . In this paper, the fitness function  $\mathcal{F}_j$ ,  $j \in \{1, 2, \dots, N\}$ , is set as the modified Tchebycheff function defined in (4), and is determined in the initialization phase

**Algorithm 3** MaximumRanking( $U_t, K$ )

---

```

1:  $\{L_1, L_2, \dots, L_N\} \leftarrow \{\emptyset, \emptyset, \dots, \emptyset\}$ 
2: for each solution  $\mathbf{x} \in U_t$  do
3:    $R_g(\mathbf{x}) \leftarrow +\infty$ 
4:   Compute the set  $B(\mathbf{x})$ 
5:   for each  $j \in B(\mathbf{x})$  do
6:      $L_j \leftarrow L_j \cup \mathbf{x}$ 
7:   end for
8: end for
9: for  $j \leftarrow 1$  to  $N$  do
10:  Sort  $L_j$  according to  $\mathcal{F}_j$  in nondecreasing order
11:  for  $k \leftarrow 1$  to  $|L_j|$  do
12:     $\mathbf{x} \leftarrow L_j[k]$ 
13:    if  $k < R_g(\mathbf{x})$  then
14:       $R_g(\mathbf{x}) \leftarrow k$ 
15:    end if
16:  end for
17: end for
18: return  $R_g$ 

```

---

of EFR. For each fitness function  $\mathcal{F}_j$ , all the solutions in  $U_t$  are sorted in the nondecreasing order in accordance with values of such function, and each solution has its own ranking position. When all  $N$  fitness functions are considered, for each solution  $\mathbf{x}$ , it has  $N$  ranking positions, denoted by the vector  $\mathbf{R}(\mathbf{x}) = (r_1(\mathbf{x}), r_2(\mathbf{x}), \dots, r_N(\mathbf{x}))^T$ , where  $r_j(\mathbf{x})$  is the ranking position of  $\mathbf{x}$  on the fitness function  $\mathcal{F}_j$ . Then the ensemble ranking is used to give the global rank  $R_g(\mathbf{x})$  to each solution  $\mathbf{x}$  based on  $\mathbf{R}(\mathbf{x})$ . There are three alternative ensemble ranking schemes provided in [42]. Here we only consider the maximum ranking, which computes  $R_g(\mathbf{x})$  as

$$R_g(\mathbf{x}) = \min_{j=1}^N r_j(\mathbf{x}). \quad (7)$$

According to  $R_g$ , the merged population  $U_t$  can be partitioned into a number of solution sets  $\{F_1, F_2, \dots\}$  (analogous to “fronts” in NSGA-II), where the solutions in  $F_i$  have the  $i$ th minimum  $R_g$  value. Unlike NSGA-II, EFR just randomly chooses solutions in the last accepted front.

From the above description, it can be seen that all the solutions in  $U_t$  will be involved in the ranking on each of the fitness function. However, as mentioned in Section III, it may cause the misleading selection of solutions. To alleviate the problem, we propose a new version of EFR, called EFR-RR, where a ranking restriction scheme is introduced, i.e., a solution is only allowed to be ranked on the fitness functions whose corresponding weight vectors are close to it in the objective space. The detail procedure is explained as the following. For each solution  $\mathbf{x} \in U_t$ , a set  $B(\mathbf{x})$  with the size  $K$  ( $K \ll N$ ) is defined, which contains the indexes of  $K$  closest weight vectors to  $\mathbf{x}$  among all  $N$  weight vectors, in terms of perpendicular distance given by (5). The solution  $\mathbf{x}$  will only participate in the ranking on the fitness functions  $\mathcal{F}_j$ , where  $j \in B(\mathbf{x})$ . Thus, each solution  $\mathbf{x}$  only has  $K$  ranking positions, and each fitness function only assigns ranks to a part of solutions. In EFR-RR,

**Algorithm 4** Framework of EFR-RR

---

```

1: Generate a set of weight vectors  $\Lambda \leftarrow \{\lambda_1, \lambda_2, \dots, \lambda_N\}$ 
2: Initialize the population  $P_0 \leftarrow \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$ 
3: Initialize the ideal point  $\mathbf{z}^* \leftarrow (z_1^*, z_2^*, \dots, z_m^*)^T$ 
4:  $t \leftarrow 0$ 
5: while the termination criterion is not met do
6:    $Q_t \leftarrow \text{MakeOffspringPopulation}(P_t)$ 
7:    $U_t \leftarrow P_t \cup Q_t$ 
8:    $R_g \leftarrow \text{MaximumRanking}(U_t, K)$ 
9:    $\{F_1, F_2, \dots\} \leftarrow \text{GetFronts}(U_t, R_g)$ 
10:   $P_{t+1} \leftarrow \emptyset$ 
11:   $i \leftarrow 1$ 
12:  while  $|P_{t+1}| + |F_i| < N$  do
13:     $P_{t+1} \leftarrow P_{t+1} \cup F_i$ 
14:     $i \leftarrow i + 1$ 
15:  end while
16:  Randomly shuffle  $F_i$ 
17:   $P_{t+1} \leftarrow P_{t+1} \cup F_i[1 : (N - |P_{t+1}|)]$ 
18:   $t \leftarrow t + 1$ 
19: end while

```

---

the global rank of the solution  $\mathbf{x}$  is given as

$$R_g(\mathbf{x}) = \min_{j \in B(\mathbf{x})} r_j(\mathbf{x}). \quad (8)$$

If  $K = N$ , (8) is equivalent to (7).

In Algorithm 3, we describe the procedure of maximum ranking in EFR-RR. Further, we summarize the whole procedure of EFR-RR in Algorithm 4.

*C. Optional Normalization Procedure*

Normalization is particularly useful for an algorithm to solve problems having a PF whose objective values are differently scaled. For example, Zhang and Li [39] showed that even a naive normalization procedure can significantly improve the performance of MOEA/D on scaled test problems.

In essence, the task of normalization is to estimate  $\mathbf{z}^*$  and the  $\mathbf{z}^{\text{nad}}$ , because the objective of  $f_i(\mathbf{x})$  can be replaced by (9) in the normalization

$$\tilde{f}_i(\mathbf{x}) = (f_i(\mathbf{x}) - z_i^*) / (z_i^{\text{nad}} - z_i^*). \quad (9)$$

Generally,  $z_i^*$  can be effectively estimated by the best value found so far for objective  $f_i$ . However, the estimation of  $\mathbf{z}^{\text{nad}}$  is a much more difficult task since it necessitates information about the whole PF [52].

In this paper, we provide an online normalization procedure similar to that in NSGA-III, which can be incorporated into the proposed algorithms to deal with scaled test problems.

The difference between normalization presented here and that in NSGA-III lies in that we use a slightly different achievement scalarizing function to identify extreme points. Suppose  $S_t$  is the population to be normalized. In our normalization, the extreme point in the objective axis  $f_j$  is identified by finding the solution  $\mathbf{x} \in S_t$  that minimizes the following achievement scalarizing function with weight vector

$\mathbf{w}_j = (w_{j,1}, w_{j,2}, \dots, w_{j,m})^T$  being the axis direction:

$$\text{ASF}(\mathbf{x}, \mathbf{w}_j) = \max_{i=1}^m \left\{ \frac{1}{w_{j,i}} \left| \frac{f_i(\mathbf{x}) - z_i^*}{z_i^{\text{nad}} - z_i^*} \right| \right\} \quad (10)$$

where  $j \in \{1, 2, \dots, m\}$ ,  $w_{j,i} = 0$ , when  $i \neq j$  and  $w_{j,j} = 1$ . In (10), for  $w_{j,i} = 0$ , we replace it with a small number  $10^{-6}$ , and  $z_i^{\text{nad}}$  is the intercept of the constructed hyperplane with  $i$ th objective axis in previous one generation.

It is worth pointing out that  $m$  extreme points may fail to constitute an  $m$ -dimensional hyperplane. Even if the hyperplane is built, it is also likely to obtain negative intercepts in some axis directions. The original NSGA-III study [26] did not indicate how to deal with these cases. In our normalization, when these cases occur, we assign  $z_i^{\text{nad}}$  to the largest value of  $f_i$  in the nondominated solutions of  $S_t$ , for each  $i \in \{1, 2, \dots, m\}$ . For other details of the normalization, please refer to [26].

### D. Computational Complexity

For MOEA/D-DU, the major computational costs are in the updating procedure depicted in Algorithm 2. Steps 1–3 require  $O(mN)$  computations to compute  $d_{j,2}(\mathbf{y})$ ,  $j = 1, 2, \dots, N$ . In step 4,  $O(N \log K)$  computations are needed to select  $K$  minimum distances and sort them. Steps 8–13 at most require  $O(mK)$  computations. Thus, the overall complexity of MOEA/D-DU to generate  $N$  trial solutions in one generation is  $O(mN^2)$ , given that  $m$  is generally larger than  $\log K$  for many-objective optimization.

As for EFR-RR, the major computation costs lie in the calculation of global ranks. First, the computation of perpendicular distances totally requires  $O(mN^2)$  computations. To obtain  $B(\mathbf{x})$  for each solution  $\mathbf{x}$  totally requires  $O(N^2 \log K)$  computations. Suppose that  $C_j$  is the number of solutions that are involved in the ranking on the aggregation function  $\mathcal{F}_j$ , then the ranking on all the aggregation functions requires  $O(\sum_{j=1}^N C_j \log C_j)$  computations. Because  $\sum_{j=1}^N C_j \log C_j < N^2 \log N$  and generally  $m > \log N$  in the many-objective scenario, the overall complexity in the worst case is  $O(mN^2)$ .

### E. Discussion

After describing the details of MOEA/D-DU and EFR-RR, we would like to discuss main similarities and differences between the proposed algorithms and the ones listed in Section II-B. It should be pointed out that all these algorithms employ a set of weight vectors to guide the search process.

1) *MOEA/D-DU Versus MOEA/D-AWA*: Although both of them make sure that the uniformly spread weight vectors lead to uniformly spread search directions in the objective space, they use slightly different means to implement this. MOEA/D-AWA adopts a new weight vector initialization method, whereas MOEA/D-DU uses a modified form of aggregation function. More importantly, MOEA/D-AWA aims to adjust the weights periodically so as to obtain a better uniformity of solutions on problems having complex PFs, whereas MOEA/D-DU intends to maintain a better balance between convergence and diversity using a set of fixed weight vectors. In other words, MOEA/D-AWA emphasizes weight adjustment

strategy, whereas MOEA/D-DU emphasizes a new strategy to update the population.

2) *MOEA/D-DU Versus MOEA/D-STM*: Both of them use the modified version of Tchebycheff function and employ  $d_{j,2}(\mathbf{x})$  to balance the convergence and diversity. But they differ significantly in the implementation mechanism. In MOEA/D-STM, the aggregation function value and  $d_{j,2}(\mathbf{x})$  are regarded as preferences of two kinds of agents (subproblem and solution), respectively. Then the deferred acceptance procedure is used to find a STM of preferences between subproblems and solutions and thus to select half of solutions from the combined population. In MOEA/D-DU, the compromise between aggregation function value and  $d_{j,2}(\mathbf{x})$  is achieved in a more straightforward way, i.e., the newly generated solution first chooses its  $K$  closest weight vectors in terms of  $d_{j,2}(\mathbf{x})$ , and it only has the chance to compete with the solutions corresponding to these  $K$  weight vectors. Moreover, MOEA/D-STM uses the generational scheme, while MOEA/D-DU is a steady-state algorithm. In Section VII-A, we would show that, in many-objective optimization, the performance of MOEA/D-STM cannot compare with that of MOEA/D-DU.

3) *MOEA/D-DU Versus I-DBEA*: Both of them enhance MOEA/D with major modifications in the updating procedure and need to compute  $d_{j,2}(\mathbf{x})$  in the evolutionary process. And also, they both adopt the steady-state form. However, the idea of MOEA/D-DU is in essence different from that of I-DBEA. For I-DBEA, although it uses  $d_{j,1}(\mathbf{x})$  besides  $d_{j,2}(\mathbf{x})$  to select solutions, the simple precedence of  $d_{j,2}(\mathbf{x})$  over  $d_{j,1}(\mathbf{x})$  makes  $d_{j,1}(\mathbf{x})$  work with very low probability since  $d_{j,2}(\mathbf{x})$  is a real value and it can almost always distinguish two solutions. Thus, similar to NSGA-III, I-DBEA also emphasizes nondominated solutions yet close to the reference lines. In contrast to I-DBEA, MOEA/D-DU does not rely on any Pareto dominance like the original MOEA/D, and it pulls the population toward PF by simultaneously minimizing a number of aggregation functions. Further more,  $d_{j,2}(\mathbf{x})$  plays quite different roles in MOEA/D-DU and I-DBEA. In I-DBEA,  $d_{j,2}(\mathbf{x})$  is used as a metric to directly compare solutions. Whereas in MOEA/D-DU,  $d_{j,2}(\mathbf{x})$  is used to select solutions that have the chance to be replaced by the newly generated solution, and the solutions are still compared in terms of aggregation function values. In Section VII-A, we would show that MOEA/D-DU is particularly competitive to I-DBEA for solving MaOPs.

4) *MOEA/D-DU Versus MOEA/D-GR*: Both of them conduct a careful selection of solutions that can be replaced by the new solution. However, the two algorithms have rather different original intentions. MOEA/D-GR aims to find a most suitable subproblem for the newly produced solution, and the neighborhood concept of subproblems defined in MOEA/D is still used in the replacement scheme of MOEA/D-GR. Unlike MOEA/D-GR, MOEA/D-DU does not use the neighborhood of subproblems any longer in the updating procedure. Instead, it takes a pure geometric view and just exploits  $d_{j,2}(\mathbf{x})$  to avoid the preservation of solutions that achieves the good aggregation function value but far from the region represented by the corresponding weight vector as much as possible, and so as to maintain a desired distribution of solutions



in high-dimensional objective space. In Section VII-A, we would demonstrate the superiority of MOEA/D-DU over MOEA/D-GR on MaOPs.

5) *EFR-RR Versus NSGA-III*: Both of them use the generational scheme and split the combined population into a number of “fronts” to select solutions. However, there exists significant difference between them in how to rank solutions in the environmental selection phase. In NSGA-III, the Pareto dominance is still used to promote the convergence, and a niche-preservation operator aided by a set of weight vectors is adopted to select solutions in the last accepted front, intending to maintain the population diversity. In contrast to NSGA-III, EFR-RR promotes the convergence using the aggregation functions instead of Pareto dominance. Note that EFR-RR also performs the niching procedure with the aid of weight vectors, thus each weight vector in EFR-RR serves the purpose of promoting both convergence and diversity. Whereas in NSGA-III, each weight vector is mainly used to emphasize the diversity. Even when it comes to niching methodology, there is a distinction between NSGA-III and EFR-RR. In NSGA-III, each solution can only be associated with a sole weight vector; but in EFR-RR, each solution can be associated with several weight vectors. In Section VII-B, we would show that EFR-RR is compared favorably with NSGA-III on scaled problems.

## V. EXPERIMENTAL DESIGN

This section is devoted to the experimental design for investigating the performance of MOEA/D-DU and EFR-RR. First, the test problems and performance metrics used in the experiments are given. Then, we list eight MOEAs that are used to validate the proposed algorithms. Finally, the experimental settings adopted in this paper are provided.

### A. Test Problems

For comparison purposes, we use the test problems from DTLZ [53] and WFG [54] test suites. We divide the problems into two groups.

The first group of problems are all normalized test problems including DTLZ1–DTLZ4, DTLZ7, and WFG1–WFG9. Note that the objective values of the original DTLZ7 and WFG1–WFG9 problems are scaled differently. Since their true PFs are known, we modify their objective functions as

$$f_i \leftarrow (f_i - z_i^*) / (z_i^{\text{nad}} - z_i^*), \quad i = 1, 2, \dots, m. \quad (11)$$

Thus, for each of normalized DTLZ7 and WFG1–WFG9 problems, the ideal point and the nadir point are  $\mathbf{0}$  and  $\mathbf{1}$ , respectively. This group of problems are used to test the performance of the algorithms without explicit normalization procedure. The main features of these problems are summarized in Table I.

The second group of problems are all scaled test problems, including scaled DTLZ1 and DTLZ2 problems [26] and the original WFG4–WFG9 problems. The scaled DTLZ1 and DTLZ2 are the modifications of DTLZ1 and DTLZ2, respectively. To illustrate, if the scaling factor is  $10^i$ , the

TABLE I  
FEATURES OF THE TEST PROBLEMS

Problem	Features
DTLZ1	Linear, Multi-modal
DTLZ2	Concave
DTLZ3	Concave, Multi-modal
DTLZ4	Concave, Biased
DTLZ7	Mixed, Disconnected, Multi-modal
WFG1	Mixed, Biased
WFG2	Convex, Disconnected, Multi-modal, Non-separable
WFG3	Linear, Degenerate, Non-separable
WFG4	Concave, Multi-modal
WFG5	Concave, Deceptive
WFG6	Concave, Non-separable
WFG7	Concave, Biased
WFG8	Concave, Biased, Non-separable
WFG9	Concave, Biased, Multi-modal, Deceptive, Non-separable

TABLE II  
SCALING FACTORS FOR SCALED DTLZ1  
AND DTLZ2 PROBLEMS

No. of Objectives ( $m$ )	Scaling Factor	
	DTLZ1	DTLZ2
2	$10^i$	$10^i$
5	$10^i$	$10^i$
8	$3^i$	$3^i$
10	$2^i$	$3^i$
13	$1.2^i$	$2^i$

scaled DTLZ1 modifies the objective functions of the original DTLZ1 as

$$f_i \leftarrow 10^{i-1} f_i, \quad i = 1, 2, \dots, m. \quad (12)$$

This group of test problems are used to test the ability of the algorithms with normalization procedure to deal with differently scaled objective values.

All these problems can be scaled to any number of objectives and decision variables. In this paper, we consider the number of objectives  $m \in \{2, 5, 8, 10, 13\}$ . For DTLZ problems, the number of decision variables is given by  $n = m + k - 1$  where  $m$  is the number of objectives,  $k$  is set to 5 for DTLZ1, 10 for DTLZ2–DTLZ6, and 20 for DTLZ7. For all WFG problems, the number of decision variables is set to 24, and the position-related parameter is set to  $m - 1$ . The scaling factors used for scaled DTLZ1 and DTLZ2 with different number of objectives are shown in Table II.

### B. Performance Metrics

The performance metrics are needed to evaluate the performance of the concerned algorithms. The inverted generational distance (IGD) [55] is one of the most widely used metrics in multiobjective scenario, providing a combined information about convergence and diversity of a solution set. However, in high-dimensional objective space, a huge number of uniformly distributed points on the PF are required to calculate IGD in a reliable manner [56]. In this paper, we use another



TABLE III  
SETTING OF div IN DCI

No. of Objectives ( $m$ )	5	8	10	13
$div$	10	6	5	5

very popular metric, HV [1], as the primary comparison criterion. The HV metric is strictly Pareto compliant [55], and its nice theoretical qualities make it a rather fair metric [31]. HV can measure both convergence and diversity of a solution set in a sense, and larger values means better quality.

For calculating HV, the choice of reference point is a crucial issue. In our experiments, following the recommendation in [57] and [58], we set the reference point to be  $1.1\mathbf{z}^{\text{nad}}$ , where  $\mathbf{z}^{\text{nad}}$  can be analytically obtained for each test problem. Moreover, according to the practice in [7] and [19], the solutions that do not dominate reference point are discarded for the HV calculation. In the event of scaled problems, the objective values of the obtained solution set and the reference point are first normalized using  $\mathbf{z}^{\text{nad}}$  and  $\mathbf{z}^*$  ( $\mathbf{z}^*$  is  $\mathbf{0}$  for all the adopted test problems) before computing HV. Thus, for an  $m$ -objective problem,  $HV \in [0, 1.1^m - V_m]$  in our experiments, where  $V_m$  is the HV of the region enclosed by the exact normalized PF and the coordinate axes. For problems having no more than ten objectives, HV is exactly calculated using the recently proposed WFG algorithm [59]. As for problems with 13 objectives, the HV is approximated by the Monte Carlo simulation proposed in [33], and 10 000 000 sampling points are used to ensure the accuracy.

Additionally, we use generational distance (GD) [60] and diversity comparison indicator (DCI) [61] as two assistant performance metrics. GD indicates how close the obtained solution set between the true PF, and can only reflect the convergence of an algorithm. For GD, smaller value is preferred. DCI is recently proposed for assessing the diversity of PF approximations in many-objective optimization.  $DCI \in [0, 1]$  only has a relative sense, and larger value means better. To calculate DCI, a parameter  $div$  (the number of divisions) is required to set the grid environment. Following the guidance in [61], we set  $div$  as shown in Table III.

### C. MOEAs for Comparison

Eight MOEAs are considered to evaluate MOEA/D-DU and EFR-RR, four out of which are MOEA/D variants.

The first MOEA/D variant is a slightly modified version of MOEA/D-DE [40]. For a fair comparison, we replace the differential evolution (DE) operator in MOEA/D-DE with the recombination schemes in MOEA/D-DU. Another reason for this is that the results in [26] have indicated that DE operator seems not quite suitable for MaOPs. In our experimental studies, this MOEA/D version is referred as MOEA/D for short, which is the base algorithm of MOEA/D-DU.

The other three compared MOEA/D variants are MOEA/D-STM [46], MOEA/D-GR [49], and I-DBEA [48], since they are somewhat similar to MOEA/D-DU. Based on the same reasons mentioned above, we replace the DE operators in MOEA/D-STM and MOEA/D-GR with the recombination schemes in MOEA/D-DU.

The original EFR [42] is also involved in comparison because it is the predecessor of EFR-RR.

Moreover, two nondecomposition-based MOEAs are adopted for comparison, i.e., grid-based evolutionary algorithm (GrEA) [20] and shift-based density estimation (SDE) [25], both of which were proposed specially for many-objective optimization and compared favorably with several state-of-the-art many-objective optimizers. For SDE, the version SPEA2+SDE is used here as it achieves the best overall performance among the three considered versions (NSGA-II+SDE, SPEA2+SDE, and PESA-II+SDE) in [25].

All seven MOEAs mentioned above except MOEA/D-STM and I-DBEA do not incorporate any explicit normalization techniques in their original studies. MOEA/D-STM uses a naive normalization procedure, while I-DBEA uses a sophisticated one. The normalization is not the major contribution of this paper. So, for a fair comparison, we just remove the normalization procedures in MOEA/D-STM and I-DBEA. The seven MOEAs together with MOEA/D-DU and EFR-RR are compared on normalized test problems in Section VII-A. This practice is to eliminate the influence of differently scaled objective values and the normalization procedure on the performance of MOEAs and purely verify the effectiveness of the proposed strategies.

To investigate the performance on scaled test problems, we further incorporate the normalization procedure provided in Section IV-C into MOEA/D-DU and EFR-RR, and compare them in Section VII-B with NSGA-III [26], one of whose features is to include an ingenious normalization procedure to handle differently scaled objective values.

Except GrEA and SDE, all the other concerned MOEAs use the same approach [50] to generate structured weight vectors. MOEA/D, MOEA/D-STM, MOEA/D-GR, MOEA/D-DU, EFR, and EFR-RR all use the modified Tchebycheff function defined in (4).

All MOEAs including MOEA/D-DU and EFR-RR are implemented in the jMetal framework [62], and run on an Intel Core i7 2.9 GHz processor with 8 GB of RAM.

### D. Experimental Settings

The experimental settings include general settings and parameter settings. The general settings are as follows.

- 1) *Number of Runs and Termination Criterion*: Each algorithm is run 30 times independently on each test instance, and the average metric values are recorded. The algorithm terminates after  $20\,000 \times m$  function evaluations per run.
- 2) *Significance Test*: To test the difference for statistical significance in some cases, the Wilcoxon signed-rank test [63] at a 5% significance level is conducted on the metric values obtained by two competing algorithms.

As for parameter settings, we first list several common ones.

- 1) *Population Size*: The population size is the same with the number of weight vectors in all the algorithms except GrEA and SDE. Because of the generation mode of weight vectors, the population size for these algorithms cannot be arbitrary and is controlled by a parameter  $H$  ( $N = C_{H+m-1}^{m-1}$ ). As for GrEA and SDE, the population

TABLE IV  
SETTING OF THE POPULATION SIZE

No. of Objectives ( $m$ )	Divisions ( $H$ )	Population Size ( $N$ )
2	99	100
5	6	210
8	3, 3	240
10	3, 2	275
13	2, 2	182

TABLE V  
PARAMETER SETTINGS FOR CROSSOVER AND MUTATION

Parameter	Value
Crossover probability ( $p_c$ )	1.0
Mutation probability ( $p_m$ )	$1/n$
Distribution index for crossover ( $\eta_c$ )	20
Distribution index for mutation ( $\eta_m$ )	20

size can be set to any positive integer. But for ensuring a fair comparison, all the algorithms adopt the same population size for each problem instance. Table IV lists the population sizes used for varying number of objectives. To avoid only producing boundary weight vectors, the two layers of weight vectors [26] are adopted for problems having more than five objectives.

- 2) *Parameter for Crossover and Mutation*: All the algorithms use the SBX crossover and polynomial mutation to produce new solutions. The settings are shown in Table V. In EFR, EFR-RR, I-DBEA, and NSGA-III, a larger distribution index for crossover ( $\eta_c = 30$ ) is used according to [26], [42], and [48].
- 3) *Neighborhood Size  $T$  and Probability  $\delta$* : In MOEA/D, MOEA/D-STM, MOEA/D-GR and MOEA/D-DU,  $T$  is set to 20 and  $\delta$  is set to 0.9.
- 4) *Parameter  $K$  in MOEA/D-DU and EFR-RR*:  $K$  is set to 5 for MOEA/D-DU and 2 for EFR-RR. In Section VI-A, the influence of  $K$  will be investigated.

MOEA/D, MOEA/D-GR, GrEA, and SDE have their specific parameters. In MOEA/D, the maximal number of solutions replaced by each child solution ( $n_r$ ) is set to 1. In MOEA/D-GR, the size of replacement neighborhood ( $T_r$ ) is set to 5 and only one solution can be replaced by the child solution. In GrEA, the setting of grid division ( $div$ ) for each test instance is summarized in Table VI, which is adjusted according to guidelines provided in [20]. In SDE, the archive size is set as the same with the population size.

## VI. ANALYSIS OF THE PERFORMANCE OF ENHANCED ALGORITHMS

In this section, the performance of enhanced algorithms is to be analyzed. First, the influence of parameter  $K$  on the performance of MOEA/D-DU and EFR-RR is examined. Then, we investigate the ability of proposed algorithms to balance the convergence and diversity in many-objective optimization.

### A. Influence of Parameter $K$

$K$  is a major control parameter to balance the convergence and diversity in MOEA/D-DU and EFR-RR. To study their

TABLE VI  
SETTING OF GRID DIVISION IN GRE

Problem	No. of Objectives ( $m$ )	Grid Division ( $div$ )
DTLZ1	2, 5, 8, 10, 13	18, 15, 14, 16, 19
DTLZ2	2, 5, 8, 10, 13	19, 11, 12, 15, 10
DTLZ3	2, 5, 8, 10, 13	19, 19, 19, 19, 20
DTLZ4	2, 5, 8, 10, 13	19, 10, 13, 15, 14
DTLZ7	2, 5, 8, 10, 13	20, 13, 12, 11, 9
WFG1	2, 5, 8, 10, 13	18, 17, 15, 13, 10
WFG2	2, 5, 8, 10, 13	19, 16, 18, 20, 16
WFG3	2, 5, 8, 10, 13	20, 19, 16, 16, 12
WFG4	2, 5, 8, 10, 13	18, 13, 12, 11, 9
WFG5	2, 5, 8, 10, 13	19, 12, 13, 15, 10
WFG6	2, 5, 8, 10, 13	20, 12, 13, 12, 10
WFG7	2, 5, 8, 10, 13	19, 12, 13, 14, 9
WFG8	2, 5, 8, 10, 13	20, 14, 12, 12, 14
WFG9	2, 5, 8, 10, 13	19, 13, 13, 15, 10

sensitivity to  $K$ , we have tested the settings of  $K \in [1, 20]$  with a step size 1 for both of them on all the normalized problem instances. All other parameters except  $K$  are kept the same as in Section V-D.

Due to space limitation, Fig. 3 only shows the variation of HV values across all  $K$  values on the normalized DTLZ4, DTLZ7, WFG3, and WFG9 problems with 5, 8, 10, and 13 objectives. From Fig. 3, we can observe the following.

- 1)  $K$  exerts the similar impact on the performance of MOEA/D-DU and EFR-RR for each test instance.
- 2) The most suitable setting of  $K$  not only depends on what problem to be solved but also depends on the number of objectives the problem has.
- 3) For some problems, e.g., DTLZ4, the performance of MOEA/D-DU or EFR-RR is not very sensitive to the settings of  $K$ . Both of them can perform quite stably over a wide range of  $K$  values.
- 4) For some problems, e.g., WFG3 and WFG9, different setting of  $K$  may only lead to a slight variation of performance.
- 5) For other problems, e.g., DTLZ7, a more careful setting of  $K$  should be conducted. In such cases, a slight variation of  $K$  may result in a huge change of their performance.

Overall, although the performance of MOEA/D-DU and EFR-RR varies with different  $K$  values, they can achieve the balance between convergence and diversity under the proper setting of  $K$ . In general, our experiments suggest that it may be good to adjust  $K$  between  $[1, 8]$  since they can usually achieve the optimal or near optimal performance with the setting of  $K$  in this range. When tuning  $K$  for the considered problem instance, if the solutions do not converge well, larger  $K$  value may be more suitable; if the diversity of solutions is not desirable, smaller  $K$  value is recommended.

### B. Investigation of Convergence and Diversity

This section is devoted to investigate the convergence and diversity of the proposed algorithms using GD and DCI metrics, respectively. Here, we only consider normalized DTLZ1–DTLZ4 and WFG4–WFG9 problems. Because the PFs of these problems are all regular geometries, GD value

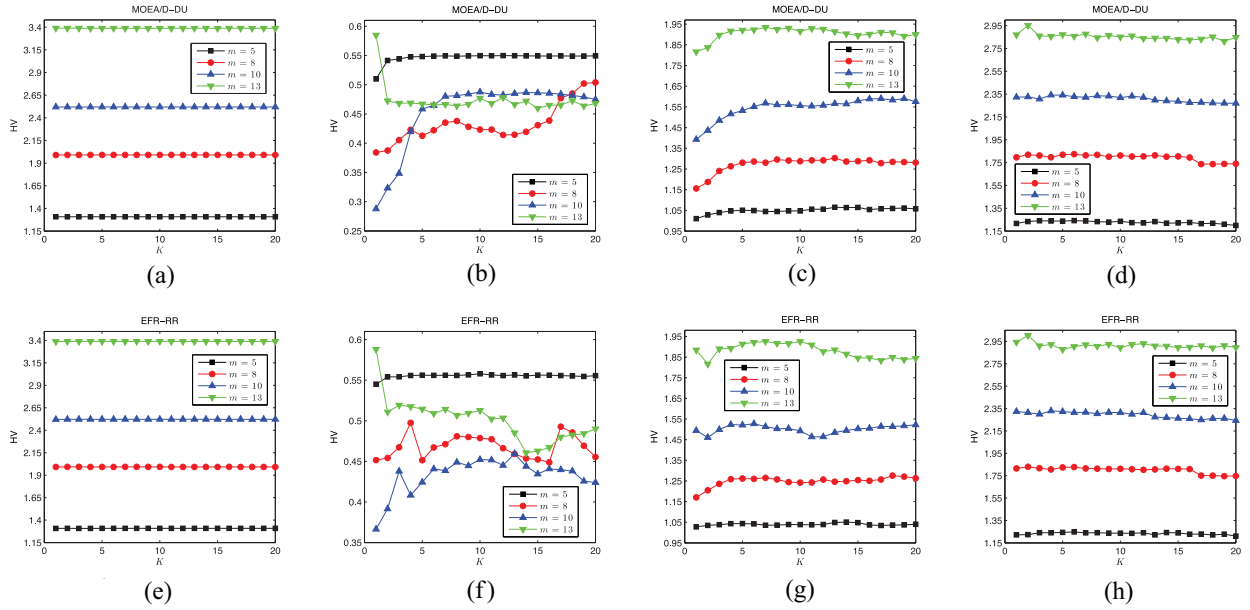


Fig. 3. Examination of the influence of  $K$  on the performance of MOEA/D-DU, EFR-RR for the normalized (a) and (e) DTLZ4, (b) and (f) DTLZ7, (c) and (g) WFG3, and (d) and (h) WFG9 test problems with varying number of objectives  $m$ . The figures show the average HV of 30 independent runs each.

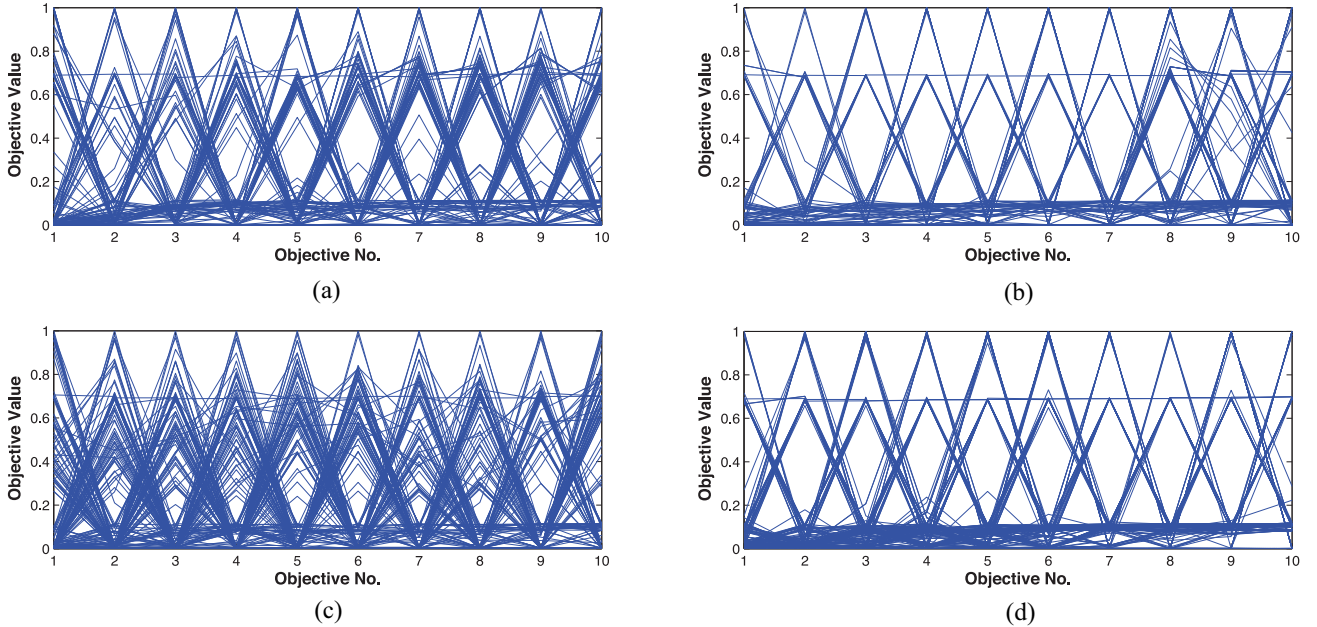


Fig. 4. Final nondominated solution set of (a) MOEA/D-DU, (b) MOEA/D, (c) EFR-RR, and (d) EFR on the normalized 10-objective WFG4 instance, shown by parallel coordinates.

can be easily determined analytically without sampling points on the PF. The PF shape of DTLZ1 is a hyperplane, while that of the other problems is a hypersphere.

We decide to compare our MOEA/D-DU and EFR-RR to their corresponding predecessors (MOEA/D and EFR) in the many-objective scenario. Table VII provides the average GD and DCI results. From Table VIII, both MOEA/D-DU and EFR-RR are generally better than their predecessors at maintaining the diversity, which achieves the better DCI results in almost all the concerned instances. For the convergence, MOEA/D-DU is still superior overall to MOEA/D and it obtains better GD results in most cases; EFR-RR is especially competitive to EFR and they perform better in terms of

average GD with respect to different problem instances. It is worth noting that, compared with EFR, EFR-RR obtains much poorer GD and DCI results on 5-objective DTLZ3, which has a huge number of local PFs. The exact reasons for EFR-RRs underperformance on this instance is the subject of future investigation.

To describe the distribution of solutions in high-dimensional objective space more intuitively, Fig. 4 plots the final nondominated solutions of MOEA/D-DU, MOEA/D, EFR-RR, and EFR in a single run on 10-objective WFG4 instance by parallel coordinates. This particular run is associated with the result closest to the average HV value. As can be seen from Fig. 4, both MOEA/D-DU and EFR-RR can find a widely



TABLE VII  
COMPARISON BETWEEN MOEA/D-DU AND MOEA/D (EFR-RR AND EFR) IN TERMS OF AVERAGE GD AND DCI VALUES ON NORMALIZED TEST PROBLEMS. THE BETTER AVERAGE RESULT FOR EACH INSTANCE IS HIGHLIGHTED IN BOLDFACE

Problem	$m$	GD		DCI		GD		DCI	
		MOEA/D-DU	MOEA/D	MOEA/D-DU	MOEA/D	EFR-RR	EFR	EFR-RR	EFR
DTLZ1	5	<b>0.000020</b>	0.004803	<b>0.969315</b>	0.896647	0.000668	<b>0.000188</b>	<b>0.943488</b>	0.920854
	8	0.000136	<b>0.000048</b>	<b>0.992567</b>	0.972053	0.000170	<b>0.000073</b>	<b>0.989848</b>	0.949016
	10	0.000022	<b>0.000004</b>	<b>0.991502</b>	0.959821	<b>0.000215</b>	0.000579	<b>0.988164</b>	0.922184
	13	<b>0.000029</b>	0.000084	<b>0.992594</b>	0.936280	0.000173	<b>0.000039</b>	<b>0.994281</b>	0.945934
DTLZ2	5	<b>0.000023</b>	0.000054	<b>0.939564</b>	0.836140	<b>0.000052</b>	0.000105	<b>0.933494</b>	0.854702
	8	0.000199	<b>0.000103</b>	<b>0.985478</b>	0.968888	0.000429	<b>0.000211</b>	<b>0.985609</b>	0.925318
	10	0.000087	<b>0.000059</b>	<b>0.976237</b>	0.930026	0.000573	<b>0.000148</b>	<b>0.978438</b>	0.876171
	13	<b>0.000218</b>	0.000328	<b>0.990947</b>	0.942113	0.000354	<b>0.000286</b>	<b>0.998750</b>	0.964441
DTLZ3	5	0.011119	<b>0.009032</b>	<b>0.814999</b>	0.701524	0.312651	<b>0.002989</b>	0.007747	<b>0.995126</b>
	8	0.003099	<b>0.000492</b>	<b>0.982177</b>	0.945281	<b>0.003595</b>	0.031573	<b>0.985280</b>	0.926237
	10	0.001925	<b>0.000282</b>	<b>0.961370</b>	0.914810	0.005250	<b>0.001787</b>	<b>0.984499</b>	0.857481
	13	<b>0.001035</b>	0.001727	<b>0.975368</b>	0.804104	<b>0.000481</b>	0.019179	<b>0.997120</b>	0.939957
DTLZ4	5	<b>0.000016</b>	0.000027	<b>0.957419</b>	0.934750	<b>0.000024</b>	0.000027	<b>0.955842</b>	0.946478
	8	<b>0.000026</b>	0.000041	<b>0.982202</b>	0.980584	<b>0.000034</b>	0.000048	0.986258	<b>0.990275</b>
	10	<b>0.000014</b>	0.000021	<b>0.998616</b>	0.981274	<b>0.000022</b>	0.000030	<b>0.998315</b>	0.995138
	13	<b>0.000020</b>	0.000037	<b>0.993399</b>	0.952210	<b>0.000019</b>	0.000023	<b>0.995285</b>	0.993149
WFG4	5	<b>0.000036</b>	0.000108	<b>0.958943</b>	0.237922	0.000098	<b>0.000058</b>	<b>0.960109</b>	0.286770
	8	<b>0.000488</b>	0.000632	<b>0.993236</b>	0.951922	<b>0.000457</b>	0.000663	<b>0.985042</b>	0.931296
	10	<b>0.000131</b>	0.000199	<b>0.989878</b>	0.907584	<b>0.000192</b>	0.000208	<b>0.998226</b>	0.848858
	13	<b>0.000217</b>	0.000229	<b>0.991063</b>	0.934429	<b>0.000214</b>	0.000226	<b>0.996185</b>	0.967426
WFG5	5	<b>0.001296</b>	0.001421	<b>0.883937</b>	0.771605	<b>0.001341</b>	0.001361	<b>0.816306</b>	0.735464
	8	<b>0.001131</b>	0.001155	<b>0.998100</b>	0.962539	<b>0.001180</b>	0.001283	<b>0.987310</b>	0.925904
	10	<b>0.000703</b>	0.000832	<b>0.986966</b>	0.934614	<b>0.000792</b>	0.000878	<b>0.998997</b>	0.841463
	13	0.000879	<b>0.000861</b>	<b>0.996643</b>	0.982889	<b>0.000907</b>	0.000949	<b>0.998150</b>	0.964958
WFG6	5	0.001173	<b>0.001152</b>	<b>0.926435</b>	0.315379	0.001284	<b>0.001275</b>	<b>0.974729</b>	0.187833
	8	<b>0.001263</b>	0.001637	<b>0.998935</b>	0.970405	<b>0.001310</b>	0.001522	<b>0.989279</b>	0.927994
	10	<b>0.000789</b>	0.000937	<b>0.993133</b>	0.932979	<b>0.000910</b>	0.001007	<b>0.999700</b>	0.819165
	13	<b>0.001172</b>	0.001184	<b>0.996104</b>	0.962610	<b>0.001171</b>	0.001207	<b>0.999021</b>	0.961705
WFG7	5	<b>0.000066</b>	0.000136	<b>0.899289</b>	0.480338	0.000099	<b>0.000064</b>	<b>0.945729</b>	0.332050
	8	<b>0.000552</b>	0.000966	<b>0.996412</b>	0.965031	<b>0.000271</b>	0.000445	<b>0.989559</b>	0.930570
	10	<b>0.000128</b>	0.000168	<b>0.975515</b>	0.954160	<b>0.000096</b>	0.000186	<b>0.999086</b>	0.838123
	13	<b>0.000786</b>	0.002391	<b>0.974228</b>	0.759269	0.000374	<b>0.000372</b>	<b>0.992426</b>	0.924018
WFG8	5	<b>0.003256</b>	0.014374	<b>0.982783</b>	0.125524	<b>0.003480</b>	0.006584	<b>0.990924</b>	0.095518
	8	0.005102	<b>0.005095</b>	<b>0.959361</b>	0.819171	0.005643	<b>0.005110</b>	<b>0.968154</b>	0.830303
	10	0.005566	<b>0.003776</b>	<b>0.988397</b>	0.717678	0.006110	<b>0.004606</b>	<b>0.996009</b>	0.756254
	13	0.004611	<b>0.001271</b>	<b>0.981147</b>	0.665580	0.003793	<b>0.001848</b>	<b>0.989844</b>	0.677784
WFG9	5	<b>0.000704</b>	0.001363	<b>0.980032</b>	0.152752	0.001065	<b>0.001042</b>	<b>0.978193</b>	0.190482
	8	0.001224	<b>0.001114</b>	<b>0.997196</b>	0.963975	0.001212	<b>0.001026</b>	<b>0.990183</b>	0.922299
	10	<b>0.000668</b>	0.000675	<b>0.997815</b>	0.927091	0.000927	<b>0.000584</b>	<b>0.998744</b>	0.880118
	13	<b>0.000945</b>	0.001150	<b>0.994202</b>	0.947416	<b>0.000961</b>	0.001285	<b>0.996398</b>	0.955634

distributed solutions over  $f_i \in [0, 1]$  for all ten objectives and a better tradeoff among them is clear from the plot, whereas both MOEA/D and EFR fail to cover most of intermediate solutions.

Based on the above comparisons, it can be concluded that MOEA/D-DU and EFR-RR can generally achieve better tradeoff between convergence and diversity than their corresponding predecessors.

## VII. COMPARISON WITH STATE-OF-THE-ART ALGORITHMS

In this section, we compare the proposed MOEA/D-DU and EFR-RR with several state-of-the-art MOEAs according to the experimental design described in Section V.

### A. Comparison on Normalized Test Problems

In this section, the algorithms without explicit normalization are compared on the first group of problems, i.e., normalized test problems. Table VIII shows the comparative results of nine algorithms in terms of HV metric on normalized DTLZ test problems, and Table IX on normalized WFG test problems. To have statistically sound conclusions, the significance test is conducted between MOEA/D-DU (EFR-RR) and each of seven other algorithms. In the two tables, the results significantly outperformed by MOEA/D-DU, EFR-RR, and both of them are marked in different symbols, respectively. Table X provides a summary of significance test of HV results between the proposed MOEA/D-DU (EFR-RR) and the other algorithms on all the considered 70 test instances. In this table, Alg<sub>1</sub> versus Alg<sub>2</sub>,

TABLE VIII  
PERFORMANCE COMPARISON ON NORMALIZED DTLZ PROBLEMS WITH RESPECT TO THE AVERAGE HV VALUES.  
THE BEST AVERAGE HV VALUE FOR EACH INSTANCE IS HIGHLIGHTED IN BOLDFACE

Problem	$m$	MOEA/D- DU	EFR- RR	MOEA/D	MOEA/D- STM	MOEA/D- GR	I-DBEA	EFR	GrEA	SDE
DTLZ1	2	0.704279	0.704371	0.703886 <sup>‡</sup>	0.703975 <sup>‡</sup>	<b>0.704426</b>	0.667992 <sup>‡</sup>	0.703793 <sup>‡</sup>	0.678608 <sup>‡</sup>	0.702987 <sup>‡</sup>
	5	<b>1.577413</b>	1.573057	1.561594 <sup>‡</sup>	1.553199 <sup>‡</sup>	1.577212	1.572915 <sup>‡</sup>	1.568136 <sup>‡</sup>	1.378445 <sup>‡</sup>	1.545282 <sup>‡</sup>
	8	2.131065	<b>2.137607</b>	2.095140 <sup>‡</sup>	2.072317 <sup>‡</sup>	2.094544 <sup>‡</sup>	2.136977	2.072935 <sup>‡</sup>	2.121597 <sup>‡</sup>	2.105726 <sup>‡</sup>
	10	2.591995	2.592416	2.587703 <sup>‡</sup>	2.584714 <sup>‡</sup>	2.587288 <sup>‡</sup>	<b>2.592680</b>	2.573180 <sup>‡</sup>	2.560445 <sup>‡</sup>	2.561582 <sup>‡</sup>
	13	3.389976	3.450434	3.280537 <sup>‡</sup>	3.198809 <sup>‡</sup>	3.387834*	<b>3.450678</b>	3.376324 <sup>‡</sup>	2.189600 <sup>‡</sup>	3.396960*
DTLZ2	2	<b>0.420129</b>	0.420127	0.420128	0.420122 <sup>‡</sup>	0.420128	0.418982 <sup>‡</sup>	0.420128 <sup>‡</sup>	0.406731 <sup>‡</sup>	0.419984 <sup>‡</sup>
	5	<b>1.307144</b>	1.306897	1.279784 <sup>‡</sup>	1.256106 <sup>‡</sup>	1.305507 <sup>‡</sup>	1.305292 <sup>‡</sup>	1.285600 <sup>‡</sup>	1.303783 <sup>‡</sup>	1.298814 <sup>‡</sup>
	8	1.950643	1.980951	1.880637 <sup>‡</sup>	1.802214 <sup>‡</sup>	1.799788 <sup>‡</sup>	1.972858*	1.780040 <sup>‡</sup>	<b>2.008651</b>	1.995382
	10	2.499082	2.503177	2.469220 <sup>‡</sup>	2.437519 <sup>‡</sup>	2.428536 <sup>‡</sup>	2.506564	2.384168 <sup>‡</sup>	<b>2.518827</b>	2.505607
	13	3.123026	3.323667	3.009885 <sup>‡</sup>	2.951134 <sup>‡</sup>	2.984305 <sup>‡</sup>	3.379538	3.122809*	3.306616*	<b>3.380496</b>
DTLZ3	2	0.396144	<b>0.411314</b>	0.351672*	0.404385*	0.394502*	0.002024 <sup>‡</sup>	0.394373*	0.366298*	0.391238*
	5	1.224940	0.054759	1.211052 <sup>‡</sup>	1.220319	1.023841 <sup>‡</sup>	0.496700 <sup>‡</sup>	1.274197	1.158907	<b>1.294771</b>
	8	1.927300	1.981563	1.777357 <sup>‡</sup>	1.731227 <sup>‡</sup>	1.820582 <sup>‡</sup>	1.831474 <sup>‡</sup>	1.760750 <sup>‡</sup>	1.601322 <sup>‡</sup>	<b>1.994729</b>
	10	2.451253	2.484752	2.388250 <sup>‡</sup>	2.377550 <sup>‡</sup>	2.304133 <sup>‡</sup>	2.405593*	2.326901 <sup>‡</sup>	2.304258 <sup>‡</sup>	<b>2.503337</b>
	13	3.089054	3.343837	2.844142 <sup>‡</sup>	2.609725 <sup>‡</sup>	3.035628 <sup>‡</sup>	3.350566	3.100067*	2.359130 <sup>‡</sup>	<b>3.370061</b>
DTLZ4	2	0.420128	0.420128	0.378778	0.409783	<b>0.420128</b>	0.420005 <sup>‡</sup>	0.420128 <sup>‡</sup>	0.347491 <sup>‡</sup>	0.295993 <sup>‡</sup>
	5	<b>1.308070</b>	1.307969	1.300444 <sup>‡</sup>	1.304319	1.307972 <sup>‡</sup>	1.302482 <sup>‡</sup>	1.305789 <sup>‡</sup>	1.306873 <sup>‡</sup>	1.302604 <sup>‡</sup>
	8	1.990879	1.992482	1.987187*	1.980209 <sup>‡</sup>	1.985272 <sup>‡</sup>	1.980485 <sup>‡</sup>	1.990668	<b>2.013161</b>	1.998266
	10	2.519895	<b>2.521704</b>	2.518763 <sup>‡</sup>	2.517569 <sup>‡</sup>	2.519728*	2.514036 <sup>‡</sup>	2.521234*	2.519544*	2.501111 <sup>‡</sup>
	13	<b>3.388281</b>	3.388239	3.371633 <sup>‡</sup>	3.340614 <sup>‡</sup>	3.384074 <sup>‡</sup>	3.384292 <sup>‡</sup>	3.386687	3.359210 <sup>‡</sup>	3.376911 <sup>‡</sup>
DTLZ7	2	0.491624	0.543805	<b>0.543950</b>	0.543948	0.495961*	0.495626*	0.543947	0.534730*	0.542912*
	5	0.548057	0.554015	0.549165*	0.548522*	0.535319 <sup>‡</sup>	0.486780 <sup>‡</sup>	0.552953	<b>0.584698</b>	0.560808
	8	0.412756	0.454187	0.390231 <sup>‡</sup>	0.409169*	0.469486	0.460563	0.426755*	<b>0.618048</b>	0.491885
	10	0.458421	0.391554	0.459661	0.460771	0.316704 <sup>‡</sup>	0.457378	0.369927 <sup>‡</sup>	<b>0.674957</b>	0.451083
	13	0.467171	0.510667	0.451700 <sup>‡</sup>	0.452945 <sup>‡</sup>	0.501658*	0.482824*	0.470118*	<b>0.612042</b>	0.464647*

<sup>‡</sup> means that the result is significantly outperformed by MOEA/D-DU.

\* means that the result is significantly outperformed by EFR-RR.

<sup>‡</sup> means that the result is significantly outperformed by both MOEA/D-DU and EFR-RR.

“B” (“W”) means that the number of instances on which the results of Alg<sub>1</sub> are significantly better (worse) than those of Alg<sub>2</sub>, and “E” indicates the number of instances where there exists no statistical significance between the results of Alg<sub>1</sub> and Alg<sub>2</sub>.

From the above results, we can obtain the following observations for the proposed MOEA/D-DU and EFR-RR.

- 1) MOEA/D-DU is clearly superior to MOEA/D. Indeed, MOEA/D-DU significantly outperforms MOEA/D in 52 out of 70 instances, whereas MOEA/D only wins in eight instances and half of them are 2-objective ones. This strongly indicates that the proposed updating strategy in MOEA/D-DU is more effective than the original one in MOEA/D for solving MaOPs.
- 2) MOEA/D-DU also shows certain advantages over the other three MOEA/D variants, i.e., MOEA/D-STM, MOEA/D-GR and I-DBEA. Specifically, MOEA/D-DU significantly outperforms both MOEA/D-STM and MOEA/D-GR on vast majority of instances. Compared with I-DBEA, MOEA/D-DU is generally significantly better on 2, 5, and 8-objective instances, and it remains very competitive on 10-objective instances. But on 13-objective instances, I-DBEA generally shows higher performance. These results verify that the selection mechanism in MOEA/D-DU is comparable with or even better than the other alternative ones in state-of-the-art MOEA/D variants.

- 3) EFR-RR achieves an obvious improvement over EFR in many-objective scenario. Among 56 many-objective ( $m > 3$ ) instances, EFR-RR performs significantly better in 46 out of them, whereas EFR is significantly better in only 5 of them. For 2-objective instances, there exists no obvious performance difference between EFR-RR and EFR. These results clearly demonstrate the effectiveness of ranking restriction scheme used in EFR-RR on MaOPs.
- 4) MOEA/D-DU and EFR-RR are both particularly competitive to two nondecomposition-based MOEAs designed specially for many-objective optimization, i.e., GrEA and SDE. Each of the four algorithms performs best on some specific problems.

Based on the above extensive results, we further introduce the performance score [33] to quantify the overall performance of compared algorithms, which makes it easier to gain some insights into the behaviors of these algorithms. For a problem instance, suppose there are  $l$  algorithms Alg<sub>1</sub>, Alg<sub>2</sub>, ..., Alg<sub>l</sub> employed in comparison, let  $\delta_{i,j}$  be 1, if Alg<sub>j</sub> significantly outperforms Alg<sub>i</sub> in terms of HV metric, and 0 otherwise. Then, for each algorithm Alg<sub>i</sub>, the performance score  $P(\text{Alg}_i)$  is determined as

$$P(\text{Alg}_i) = \sum_{\substack{j=1 \\ j \neq i}}^l \delta_{i,j}. \quad (13)$$

TABLE IX  
PERFORMANCE COMPARISON ON NORMALIZED WFG PROBLEMS WITH RESPECT TO THE AVERAGE HV VALUES.  
THE BEST AVERAGE HV VALUE FOR EACH INSTANCE IS HIGHLIGHTED IN BOLDFACE

Problem	$m$	MOEA/D -DU	EFR -RR	MOEA/D	MOEA/D -STM	MOEA/D -GR	I-DBEA	EFR	GrEA	SDE
WFG1	2	0.562230	0.581095	0.593810	<b>0.641743</b>	0.568671	0.426905 <sup>‡</sup>	0.573331	0.623764	0.561651*
	5	1.001123	1.242156	1.306573	<b>1.349242</b>	0.465755 <sup>‡</sup>	0.882835 <sup>‡</sup>	1.236781	1.136911*	1.212965*
	8	1.410532	1.562607	1.766691	<b>1.935880</b>	1.482641*	1.711815	1.782913	1.631993	1.898297
	10	1.666437	2.178243	<b>2.442026</b>	2.439172	1.292318 <sup>‡</sup>	2.416167	2.396391	2.156655	2.399104
	13	3.167303	3.125288	3.125818 <sup>‡</sup>	2.975475 <sup>‡</sup>	3.024708 <sup>‡</sup>	<b>3.179508</b>	3.156495	3.158010 <sup>‡</sup>	3.170219
WFG2	2	0.740281	0.758307	0.761351	0.760855	0.742650*	0.713250 <sup>‡</sup>	0.762530	0.758330	<b>0.762608</b>
	5	1.600169	<b>1.601199</b>	1.571654 <sup>‡</sup>	1.589303 <sup>‡</sup>	1.464050 <sup>‡</sup>	1.584469 <sup>‡</sup>	1.546560 <sup>‡</sup>	1.565527 <sup>‡</sup>	1.581520 <sup>‡</sup>
	8	<b>2.139066</b>	2.135469	2.119648 <sup>‡</sup>	2.121209 <sup>‡</sup>	2.107631 <sup>‡</sup>	2.108573 <sup>‡</sup>	2.116394 <sup>‡</sup>	2.111040 <sup>‡</sup>	2.118015 <sup>‡</sup>
	10	<b>2.591736</b>	2.589398	2.578580 <sup>‡</sup>	2.583832 <sup>‡</sup>	2.589241 <sup>‡</sup>	2.573491 <sup>‡</sup>	2.576869 <sup>‡</sup>	2.562607 <sup>‡</sup>	2.573589 <sup>‡</sup>
	13	3.431186	<b>3.442014</b>	3.406848 <sup>‡</sup>	3.386077 <sup>‡</sup>	3.430181*	3.428283*	3.414637 <sup>‡</sup>	3.414581 <sup>‡</sup>	3.431470*
WFG3	2	0.700664	0.699118	0.700160	0.700704	0.700284	0.677166 <sup>‡</sup>	0.699375 <sup>‡</sup>	0.682771 <sup>‡</sup>	<b>0.700997</b>
	5	<b>1.050757</b>	1.034773	0.867282 <sup>‡</sup>	0.866017 <sup>‡</sup>	0.906913 <sup>‡</sup>	1.037868 <sup>‡</sup>	0.852188 <sup>‡</sup>	1.045229 <sup>‡</sup>	0.994442 <sup>‡</sup>
	8	1.280380	1.204763	1.222794 <sup>‡</sup>	1.227956 <sup>‡</sup>	1.136276 <sup>‡</sup>	1.244489 <sup>‡</sup>	1.213499 <sup>‡</sup>	<b>1.416407</b>	1.247331 <sup>‡</sup>
	10	1.532371	1.460729	1.458075 <sup>‡</sup>	1.460352 <sup>‡</sup>	1.489880 <sup>‡</sup>	1.528830	1.454140 <sup>‡</sup>	<b>1.726470</b>	1.518089
	13	1.920702	1.816198	1.671833 <sup>‡</sup>	1.642120 <sup>‡</sup>	1.736590 <sup>‡</sup>	1.922486	1.754964 <sup>‡</sup>	<b>2.284531</b>	1.925885
WFG4	2	0.417000	<b>0.418377</b>	0.416908*	0.417200*	0.417183*	0.405476 <sup>‡</sup>	0.418244	0.406336 <sup>‡</sup>	0.418255*
	5	1.285940	<b>1.287692</b>	0.920686 <sup>‡</sup>	0.962506 <sup>‡</sup>	0.792782 <sup>‡</sup>	1.254008 <sup>‡</sup>	0.961821 <sup>‡</sup>	1.253821 <sup>‡</sup>	1.247417 <sup>‡</sup>
	8	1.934635	<b>1.965264</b>	1.679865 <sup>‡</sup>	1.659225 <sup>‡</sup>	1.715845 <sup>‡</sup>	1.860717 <sup>‡</sup>	1.724350 <sup>‡</sup>	1.930146*	1.905486 <sup>‡</sup>
	10	2.468681	<b>2.491699</b>	2.231219 <sup>‡</sup>	2.247558 <sup>‡</sup>	2.369648 <sup>‡</sup>	2.435737 <sup>‡</sup>	2.194736 <sup>‡</sup>	2.405230 <sup>‡</sup>	2.347108 <sup>‡</sup>
	13	3.008914	3.337246	2.900558 <sup>‡</sup>	2.777046 <sup>‡</sup>	3.030456*	<b>3.339312</b>	3.085706*	3.119742*	3.247644*
WFG5	2	0.378917	<b>0.379074</b>	0.373096 <sup>‡</sup>	0.378518*	0.378374	0.374261 <sup>‡</sup>	0.379034	0.366330 <sup>‡</sup>	0.377682 <sup>‡</sup>
	5	<b>1.216121</b>	1.207217	1.173121 <sup>‡</sup>	1.175929 <sup>‡</sup>	0.798748 <sup>‡</sup>	1.208029 <sup>‡</sup>	1.186556 <sup>‡</sup>	1.213000 <sup>‡</sup>	1.210585 <sup>‡</sup>
	8	1.809546	1.853939	1.523779 <sup>‡</sup>	1.499358 <sup>‡</sup>	1.610531 <sup>‡</sup>	1.811091*	1.572135 <sup>‡</sup>	<b>1.868017</b>	1.840307*
	10	2.288982	2.338790	2.044104 <sup>‡</sup>	2.111802 <sup>‡</sup>	2.110555 <sup>‡</sup>	2.327699*	1.988422 <sup>‡</sup>	<b>2.338894</b>	2.285479*
	13	2.685390	3.077929	2.629138 <sup>‡</sup>	2.599395 <sup>‡</sup>	2.629734 <sup>‡</sup>	<b>3.135248</b>	2.802878*	2.945444*	3.118274
WFG6	2	0.386121	0.387136	0.388269	0.388159	0.386992	0.371930 <sup>‡</sup>	0.386763	0.375526 <sup>‡</sup>	<b>0.388339</b>
	5	1.203532	1.202838	0.932134 <sup>‡</sup>	0.989178 <sup>‡</sup>	0.745472 <sup>‡</sup>	1.198173 <sup>‡</sup>	0.814322 <sup>‡</sup>	<b>1.209398</b>	1.204154
	8	1.792582	1.845513	1.524041 <sup>‡</sup>	1.514136 <sup>‡</sup>	1.632452 <sup>‡</sup>	1.788382*	1.568782 <sup>‡</sup>	<b>1.860831</b>	1.823996*
	10	2.264165	<b>2.318441</b>	1.973321 <sup>‡</sup>	2.019177 <sup>‡</sup>	2.106371 <sup>‡</sup>	2.288335*	1.942445 <sup>‡</sup>	2.291735*	2.281144*
	13	2.694571	3.028384	2.632777 <sup>‡</sup>	2.577972 <sup>‡</sup>	2.703796*	<b>3.057529</b>	2.797318*	2.947601*	3.048374
WFG7	2	0.419219	0.419127	0.419264	0.419271	0.419270	0.403176 <sup>‡</sup>	0.419119 <sup>‡</sup>	0.406740 <sup>‡</sup>	<b>0.419480</b>
	5	1.278119	1.280956	1.119326 <sup>‡</sup>	1.134063 <sup>‡</sup>	0.843455 <sup>‡</sup>	1.269955 <sup>‡</sup>	1.032286 <sup>‡</sup>	<b>1.294253</b>	1.280318
	8	1.926683	1.987161	1.657222 <sup>‡</sup>	1.640169 <sup>‡</sup>	1.741279 <sup>‡</sup>	1.917133 <sup>‡</sup>	1.704069 <sup>‡</sup>	<b>1.998229</b>	1.953074*
	10	2.458886	<b>2.509162</b>	2.329332 <sup>‡</sup>	2.319595 <sup>‡</sup>	2.315842 <sup>‡</sup>	2.475375*	2.162850 <sup>‡</sup>	2.491561*	2.421732 <sup>‡</sup>
	13	3.080436	3.350532	2.792831 <sup>‡</sup>	2.760968 <sup>‡</sup>	2.918742 <sup>‡</sup>	<b>3.353342</b>	3.106133*	3.198467*	3.320231*
WFG8	2	0.380438	0.379809	0.379589	0.380673	0.380463	0.354848 <sup>‡</sup>	0.380610	0.374610 <sup>‡</sup>	<b>0.381698</b>
	5	1.174365	<b>1.179315</b>	0.620310 <sup>‡</sup>	0.606449 <sup>‡</sup>	0.687016 <sup>‡</sup>	1.165600 <sup>‡</sup>	0.614472 <sup>‡</sup>	1.160200 <sup>‡</sup>	1.153417 <sup>‡</sup>
	8	1.753599	1.739425	1.426568 <sup>‡</sup>	1.416873 <sup>‡</sup>	1.545540 <sup>‡</sup>	1.686752 <sup>‡</sup>	1.443098 <sup>‡</sup>	<b>1.760000</b>	1.682100 <sup>‡</sup>
	10	2.227371	2.192016	1.770687 <sup>‡</sup>	1.781845 <sup>‡</sup>	1.925330 <sup>‡</sup>	2.228087	1.789877 <sup>‡</sup>	<b>2.235939</b>	2.079279 <sup>‡</sup>
	13	2.668338	3.070900	2.465452 <sup>‡</sup>	2.451922 <sup>‡</sup>	2.766640*	<b>3.073486</b>	2.535248 <sup>‡</sup>	2.990715*	3.036104*
WFG9	2	0.395282	0.393623	0.400667	0.387581*	0.388456	0.378018 <sup>‡</sup>	0.407289	0.382390 <sup>‡</sup>	<b>0.409947</b>
	5	<b>1.237722</b>	1.223673	0.658931 <sup>‡</sup>	0.674282 <sup>‡</sup>	0.677244 <sup>‡</sup>	1.176343 <sup>‡</sup>	0.789826 <sup>‡</sup>	1.226696 <sup>‡</sup>	1.228754 <sup>‡</sup>
	8	1.819899	1.830079	1.628043 <sup>‡</sup>	1.623609 <sup>‡</sup>	1.639324 <sup>‡</sup>	1.744089 <sup>‡</sup>	1.682517 <sup>‡</sup>	<b>1.870276</b>	1.848786
	10	<b>2.341604</b>	2.314426	2.062388 <sup>‡</sup>	2.074169 <sup>‡</sup>	2.134863 <sup>‡</sup>	2.293407 <sup>‡</sup>	2.096066 <sup>‡</sup>	2.327689 <sup>‡</sup>	2.296196 <sup>‡</sup>
	13	2.871460	3.003053	2.722066 <sup>‡</sup>	2.711396 <sup>‡</sup>	2.842311 <sup>‡</sup>	3.045041	2.811748*	2.941058*	<b>3.082909</b>

<sup>‡</sup> means that the result is significantly outperformed by MOEA/D-DU.

\* means that the result is significantly outperformed by EFR-RR.

<sup>‡</sup> means that the result is significantly outperformed by both MOEA/D-DU and EFR-RR.

This value means how many other algorithms are significantly better than  $Alg_i$  on the considered problem instance. So, the smaller the value, the better the algorithm. Fig. 5 summarizes the average performance score for different number of objectives and different test problems, respectively.

From Fig. 5(a), both MOEA/D-DU and EFR-RR rank very well on MaOPs. For 2-objective problems, MOEA/D-DU is overall slightly better than MOEA/D, whereas EFR-RR is

overall slightly worse than EFR. It is interesting to observe that, GrEA and I-DBEA, which show competitive performance on MaOPs, do not perform well on 2-objective problems. Conversely, MOEA/D, MOEA/D-STM, and MOEA/D-GR, which perform satisfactorily on 2-objective problems, scale poorly on MaOPs. Moreover, I-DBEA presents an interesting search behavior, i.e., it shows more competitiveness on problems having higher number of objectives.



TABLE X  
SUMMARY OF THE SIGNIFICANCE TEST OF HV BETWEEN THE PROPOSED ALGORITHMS AND THE OTHER ALGORITHMS

		MOEA/D -DU	EFR- RR	MOEA/D	MOEA/D -STM	MOEA/D -GR	I-DBEA	EFR	GrEA	SDE
MOEA/D-DU vs.	B	—	21	52	50	46	40	43	32	26
	W	—	41	8	8	6	21	15	34	32
	E	—	8	10	12	18	9	12	4	12
EFR-RR vs.	B	41	—	52	52	54	45	48	41	39
	W	21	—	11	11	8	16	7	24	27
	E	8	—	7	7	8	9	15	5	5

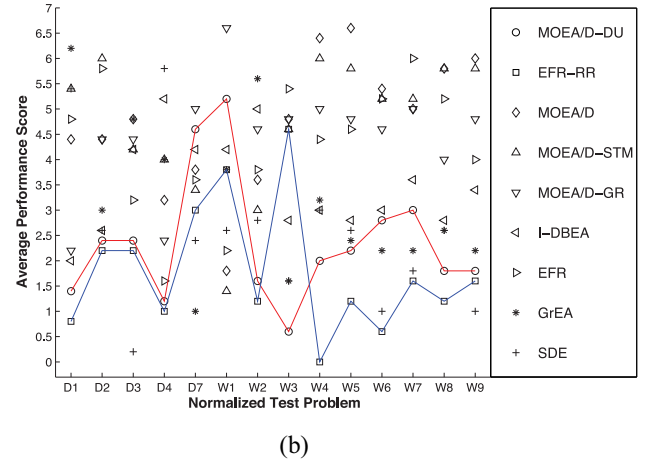
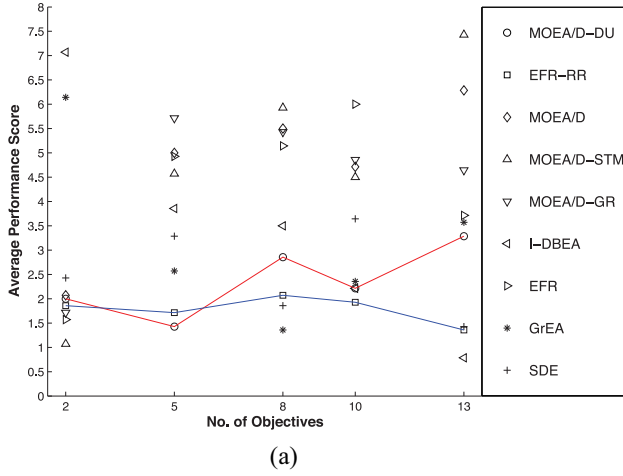


Fig. 5. Average performance score over all (a) normalized test problems for different number of objectives and (b) dimensions for different normalized test problems, namely DTLZ(Dx) and WFG(Wx). The smaller the score, the better the PF approximation in terms of HV metric. The values of proposed MOEA/D-DU and EFR-RR are connected by a solid line to easier assess the score.

TABLE XI  
RANKING IN THE AVERAGE PERFORMANCE SCORE OVER ALL  
NORMALIZED TEST INSTANCES FOR THE NINE ALGORITHMS.  
THE SMALLER THE SCORE, THE BETTER THE OVERALL  
PERFORMANCE IN TERMS OF HV METRIC

Algorithm	Score	Rank
MOEA/D-DU	2.36	2
EFR-RR	1.79	1
MOEA/D	4.71	9
MOEA/D-STM	4.70	8
MOEA/D-GR	4.47	7
I-DBEA	3.49	5
EFR	4.27	6
GrEA	3.20	4
SDE	2.53	3

From Fig. 5(b), EFR-RR achieves the best overall performance in 10 out of total 14 test problems, and it takes the second place on WFG9 problem. But EFR-RR does not show such excellent performance on WFG1 and WFG3. MOEA/D-DU performs overall best on WFG3, and it ranks second in eight instances, all following EFR-RR. However, MOEA/D-DU may struggle on DTLZ7 and WFG1, where it is only overall better than MOEA/D-GR.

Table XI presents the average performance score over all 70 problem instances for nine concerned algorithms, and the overall rank is given to each algorithm according to the score.

TABLE XII  
AVERAGE CPU TIME PER RUN FOR MOEA/D-DU,  
EFR-RR, GrEA, AND SDE

Algorithm	CPU Time (sec)
MOEA/D-DU	8.981
EFR-RR	14.011
GrEA	44.464
SDE	98.801

The proposed EFR-RR and MOEA/D-DD are ranked first and second, respectively, followed by SDE and GrEA. But it is worth pointing out that this is just a comprehensive evaluation of algorithms on all the problem instances considered in this paper. Indeed, no algorithm can beat any of the other algorithms on all possible instances, and some algorithm is more suitable for solving certain kinds of problem, e.g., I-DBEA may show advantages over the others on problems with a very high number of objectives. In addition, the overall excellent performance of GrEA in our experiments is obtained by suitably setting div for each test instance. In this regard, GrEA takes advantage of the other compared algorithms.

Since GrEA and SDE are two competitive many-objective optimizers to MOEA/D-DU and EFR-RR, we further show the average CPU time over total  $70 \times 30 = 2100$  runs for each of the four algorithms in Table XII. Because all the four algorithms are implemented in the same framework, and run on

TABLE XIII  
PERFORMANCE COMPARISON ON SCALED TEST PROBLEMS WITH  
RESPECT TO THE AVERAGE HV VALUES. THE BEST AVERAGE HV  
VALUE FOR EACH INSTANCE IS HIGHLIGHTED IN BOLDFACE

Problem	$m$	MOEA/D-DU	EFR-RR	NSGA-III
DTLZ1	2	<b>0.704179</b>	0.704174	0.701746*
	5	<b>1.576914</b>	1.563334	1.571783†
	8	2.130914	<b>2.137705</b>	2.129752‡
	10	2.591432	<b>2.592426</b>	2.585756‡
	13	3.386799	3.450218	<b>3.450789</b>
DTLZ2	2	<b>0.420127</b>	0.420115	0.419951‡
	5	<b>1.307157</b>	1.306985	1.297456‡
	8	1.952661	<b>1.983739</b>	1.970329*
	10	2.499967	<b>2.506082</b>	2.505281
	13	3.135602	3.349935	<b>3.379670</b>
WFG4	2	0.416934	<b>0.418212</b>	0.417910*
	5	<b>1.287030</b>	1.286825	1.265777‡
	8	1.937218	<b>1.971430</b>	1.879845‡
	10	2.472125	<b>2.499611</b>	2.442636‡
	13	3.039460	<b>3.359619</b>	3.316773*
WFG5	2	0.378437	<b>0.379066</b>	0.378669*
	5	<b>1.216065</b>	1.206244	1.213641†
	8	1.812500	<b>1.858908</b>	1.820270*
	10	2.291171	<b>2.341408</b>	2.327403*
	13	2.792708	3.103004	<b>3.134267</b>
WFG6	2	<b>0.386738</b>	0.386458	0.386413
	5	<b>1.205599</b>	1.204359	1.201132†
	8	1.788193	<b>1.847898</b>	1.803972*
	10	2.262895	<b>2.324245</b>	2.300161*
	13	2.691167	3.057463	<b>3.067444</b>
WFG7	2	<b>0.419193</b>	0.419167	0.418768‡
	5	1.278789	<b>1.282254</b>	1.278558*
	8	1.927311	<b>1.989183</b>	1.920527‡
	10	2.459857	<b>2.511870</b>	2.474063*
	13	3.115259	<b>3.370701</b>	3.347897*
WFG8	2	0.379874	<b>0.379964</b>	0.379504
	5	1.166643	<b>1.178156</b>	1.163539‡
	8	<b>1.774488</b>	1.773226	1.679544‡
	10	2.243801	<b>2.264483</b>	2.214018‡
	13	2.587436	<b>2.993202</b>	2.947889*
WFG9	2	0.395017	<b>0.403498</b>	0.401294*
	5	<b>1.239003</b>	1.229890	1.179849‡
	8	1.834668	<b>1.850715</b>	1.728561‡
	10	<b>2.352997</b>	2.336695	2.274673‡
	13	2.961438	<b>3.095129</b>	2.978443*

† means that the result is significantly outperformed by MOEA/D-DU.

\* means that the result is significantly outperformed by EFR-RR.

‡ means that the result is significantly outperformed by both MOEA/D-DU and EFR-RR.

the same computing environment, the data can give us a rough understanding of how fast each algorithm is in a relatively fair manner. The results indicate that, although GrEA and SDE also achieve decent overall performance, their efficiency cannot match that of MOEA/D-DU and EFR-RR, clearly requiring much more computational effort.

### B. Comparison on Scaled Test Problems

In this section, we incorporate the normalization procedure presented in Section IV-C into the proposed MOEA/D-DU and EFR-RR, and compare them with NSGA-III on the second group of test problems, i.e., scaled test problems.

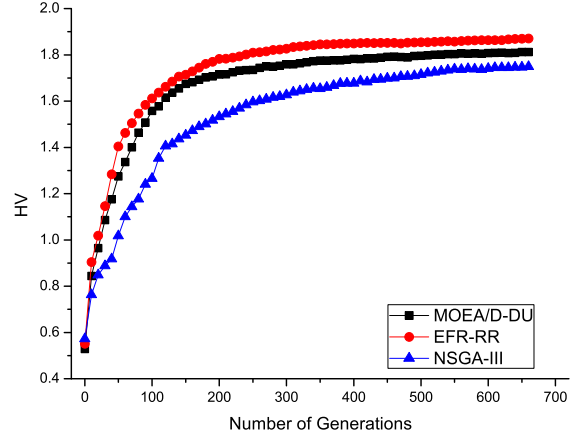


Fig. 6. Evolutionary trajectories of HV over generations for MOEA/D-DU, EFR-RR, and NSGA-III on the 8-objective WFG9 instance.

Table XIII shows the average HV results obtained by MOEA/D-DU, EFR-RR, and NSGA-III. From Table XIII, EFR-RR shows obvious advantage over NSGA-III. To be specific, EFR-RR significantly outperforms NSGA-III on 30 out of 40 instances, whereas NSGA-III is significantly better than EFR-RR only on five instances including 5- and 13-objective DTLZ1, 13-objective DTLZ2, and 5- and 13-objective WFG5. As for MOEA/D-DU, it is generally superior to NSGA-III on 2-, 5-, and 8-objective problems. On 10-objective problems, MOEA/D-DU and NSGA-III have equal shares, and either of them performs significantly better than the other on half of problems. Nevertheless, MOEA/D-DU does not perform as well as NSGA-III on 13-objective problems, where NSGA-III wins in 7 of them.

Fig. 6 plots the evolutionary trajectories of HV with generation numbers for MOEA/D-DU, EFR-RR and NSGA-III on 8-objective WFG9 instance. From Fig. 6, although all the three algorithms can stably increase HV with elapse of generation numbers, MOEA/D-DU and EFR-RR converge to the PF faster than NSGA-III and finally achieve higher HV values.

From the above results, it can be concluded that either MOEA/D-DU or EFR-RR with an elaborate normalization procedure is at least competitive to or even better than NSGA-III for solving scaled problems.

### C. Further Discussion

In this section, we would further discuss several issues concerning with the experimental studies.

The first issue is about the setting of population size in our experiments. Intuitively, exponentially more points are needed to represent a tradeoff of higher dimensional PF. Thus, a large population size is required for MOEAs to approximate the entire PF. However, just as mentioned in [26], it is difficult for a decision maker to comprehend and select a preferred solution in this scenario. Another drawback is that a large population size would make MOEAs computationally inefficient, and sometimes may be impractical. For example, if we use a population size of 100 in 2-objective case, then a population size of  $10^8$  could be used in the 8-objective case, in the premise that

we keep the same dense distribution. Clearly, it is generally unaffordable and impractical for a MOEA to evolve a population having such a huge number of solutions. So, as illustrated in [56], one of the practical search strategies in many-objective optimization is to globally search for sparsely distributed non-dominated solutions over the entire PF. In our experiments, the setting of population size is shown in Table IV, and the similar settings can be seen in most existing studies on many-objective optimization (see [14], [20], [25], [26], [48]).

The second issue is why MOEA/D-DU and EFR-RR do not exhibit clear superiority over their predecessors on 2-objective problems, although they have shown much better performance on many-objective ones. We suspect there are two possible reasons. First, just as mentioned above, we can only in fact use a population with the sparse distribution of solutions (and weight vectors) in a high-dimensional objective space, and a particular region may be associated with only one solution or one weight vector. So, if the aggregation function value is emphasized too much, the solution is very likely to deviate far from the corresponding region and thus to deteriorate the diversity preservation. In a 2-D objective space, just a normal population size, e.g., 100, can achieve a very dense distribution of solutions, so it is much less likely to fail to capture some regions of PF in this case even for MOEA/D or EFR. Second, there are more degrees of freedom in a higher dimensional objective space, so it is much more likely to produce the solutions that achieve a good aggregation value but far from the weight vector. Hence, the distance-based updating strategy in MOEA/D-DU and raking restriction scheme in EFR-RR would be more helpful to maintain the desired distribution of solutions in the evolutionary process.

The third issue is why two similar proposals to MOEA/D-DU, i.e., MOEA/D-STM and MOEA/D-GR, fail to show performance on par with MOEA/D-DU on MaOPs. For MOEA/D-STM, it uses a matching algorithm to coordinate the preferences of subproblems and solutions. Thus, it indeed implicitly gives equal importance to the two kinds of preferences. Nevertheless, it may be more appropriate to have a bias toward the preference of solutions in many-objective optimization since it is much more possible to lose track of some regions in high-dimensional objective space using sparsely distributed solutions. This is also experimentally verified by the parametric study of  $K$  in MOEA/D-DU and EFR-RR, where a small  $K$  value is recommended. As for MOEA/D-GR, when handling MaOPs, it is not very reasonable to find a subproblem for a new solution using the aggregation functions of subproblems. This is because, in high-dimensional objective space, the sparsely distributed weight vectors are far from each other, and thus the optimal values to each of subproblems (aggregation functions) are rather different, leading to the incomparability of the absolute values for different aggregation functions. Another potential drawback of MOEA/D-GR is that to use the neighborhood of subproblems as the replacement neighborhood may be questionable for many-objective optimization. It can also be attributed to the sparsity of weight vectors in the objective space, which limits the information sharing between subproblems.

The last issue is about the setting of parameter  $K$  in MOEA/D-DU and EFR-RR. In our experiments, we just use the constant  $K$  value for all the test instances when comparing the two proposed algorithms with the others. However, just as shown in Section VI-A, the most desirable  $K$  depends on the problem instance to be solved. Indeed,  $K$  plays an important role in controlling the balance between convergence and diversity. If a large  $K$  is used, the aggregation function value is more emphasized, thus the convergence is encouraged. In contrast, if a small  $K$  is used,  $d_{j,2}(\mathbf{x})$  is more stressed, and thus the diversity is promoted. For a specific problem instance, there is supposed to be an optimal balance between the two. This is the meaning of the existence of  $K$  in MOEA/D-DU and EFR-RR.

## VIII. CONCLUSION

In this paper, MOEA/D-DU and EFR-RR are proposed for many-objective optimization, which enhance two decomposition-based MOEAs, i.e., MOEA/D and EFR, respectively. The basic idea is to explicitly maintain the distribution of solutions during the evolutionary process by exploiting the perpendicular distance from a solution to a weight vector in the objective space. Specifically, in MOEA/D-DU, the updating procedure first determines  $K$  closest weight vectors to the newly generated solution in the objective space in terms of the perpendicular distance, then the new solution can only try replacing these solutions corresponding to the  $K$  weight vectors by comparing the aggregation function values. In EFR-RR, we argue that each solution involves in the ranking on a part of aggregation functions could be better than on all, where the selection of these aggregation functions is also aided by the perpendicular distance.

In the experimental studies, we have investigated the influence of a key parameter  $K$  on the performance of MOEA/D-DU and EFR-RR, and several suggestions are provided to set  $K$  properly. We have also shown that MOEA/D-DU and EFR-RR are generally superior to their predecessors in balancing the convergence and diversity in many-objective optimization. To demonstrate the competitiveness, the performance of MOEA/D-DU and EFR-RR has been compared to seven state-of-the-art MOEAs. A total of 70 normalized problem instances with up to 13 objectives from DTLZ and WFG test suites are used for comparison. The results indicate that both MOEA/D-DU and EFR-RR show obvious advantage over their predecessors in many-objective optimization and both are superior overall to other peer algorithms. Further more, we incorporated a sophisticated online normalization procedure into MOEA/D-DU and EFR-RR, respectively, and compared them with NSGA-III on scaled test problems. The results verify the effectiveness of MOEA/D-DU and EFR-RR with the normalization procedure in handling different scales of objectives.

In the future, we will study how to dynamically control the parameter  $K$  in MOEA/D-DU and EFR-RR. It would also be interesting to incorporate a local search into the

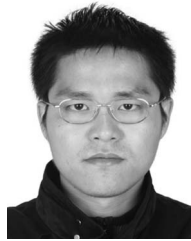


two algorithms so as to achieve faster convergence without loss in diversity. Moreover, it is nontrivial for the two algorithms to better balance the exploration and exploitation in the decision space [64], [65], which may deserve further attention. Finally, further comparative studies are needed to understand the strength and weakness of the two enhanced algorithms in this paper and the two latest MOEAs for MaOPs [28], [66].

## REFERENCES

- [1] 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.
- [2] Y. Yuan and H. Xu, "Multiobjective flexible job shop scheduling using memetic algorithms," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 336–353, Jan. 2015.
- [3] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *Proc. Evol. Multi-Criterion Optim.*, Guanajuato, Mexico, 2005, pp. 14–32.
- [4] J. G. Herrero, A. Berlanga, and J. M. M. Lopez, "Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 151–168, Feb. 2009.
- [5] K. Praditwong, M. Harman, and X. Yao, "Software module clustering as a multi-objective search problem," *IEEE Trans. Softw. Eng.*, vol. 37, no. 2, pp. 264–282, Mar./Apr. 2011.
- [6] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 2419–2426.
- [7] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Proc. Evol. Multi-Criterion Optim.*, Matsushima, Japan, 2007, pp. 742–756.
- [8] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?" in *Proc. IEEE Congr. Evol. Comput.*, Seoul, Korea, 2001, pp. 957–962.
- [9] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Proc. Evol. Multi-Criterion Optim.*, Faro, Portugal, 2003, pp. 376–390.
- [10] 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.
- [11] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proc. Evol. Methods Design Optim. Control*, Athens, Greece, 2002, pp. 95–100.
- [12] P. J. Bentley and J. P. Wakefield, "Finding acceptable solutions in the Pareto-optimal range using multiobjective genetic algorithms," in *Proc. Soft Comput. Eng. Design Manuf.*, 1998, pp. 231–240.
- [13] N. Drechsler, R. Drechsler, and B. Becker, "Multi-objective optimisation based on relation favour," in *Proc. Evol. Multi-Criterion Optim.*, Zurich, Switzerland, 2001, pp. 154–166.
- [14] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, Apr. 2014.
- [15] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," in *Proc. Evol. Multi-Criterion Optim.*, Matsushima, Japan, 2007, pp. 5–20.
- [16] F. di Pierro, S.-T. Khu, and D. A. Savic, "An investigation on preference order ranking scheme for multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 11, no. 1, pp. 17–45, Feb. 2007.
- [17] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, to be published.
- [18] S. Kukkonen and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Singapore, 2007, pp. 3983–3990.
- [19] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1402–1412, Oct. 2008.
- [20] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, Oct. 2013.
- [21] M. Garza-Fabre, G. T. Pulido, and C. A. C. Coello, "Ranking methods for many-objective optimization," in *Proc. Mex. Int. Conf. Artif. Intell.*, Guanajuato, Mexico, 2009, pp. 633–645.
- [22] A. L. Jaimes and C. A. C. Coello, "Study of preference relations in many-objective optimization," in *Proc. 11th Annu. Conf. Genet. Evol. Comput.*, Montreal, QC, Canada, 2009, pp. 611–618.
- [23] M. G. Fabre, G. T. Pulido, and C. A. C. Coello, "Alternative fitness assignment methods for many-objective optimization problems," in *Proc. Artif. Evol.*, Strasbourg, France, 2010, pp. 146–157.
- [24] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 183–195, Apr. 2011.
- [25] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 348–365, Jun. 2014.
- [26] K. Deb and J. Himanshu, "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.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
- [27] K. Praditwong and X. Yao, "A new multi-objective evolutionary optimisation algorithm: The two-archive algorithm," in *Proc. Int. Conf. Comput. Intell. Security*, Guangzhou, China, 2006, pp. 286–291.
- [28] H. Wang, L. Jiao, and X. Yao, "Two\_Arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 524–541, 2015.
- [29] M. Fleischer, "The measure of Pareto optima applications to multi-objective metaheuristics," in *Proc. Evol. Multi-Criterion Optim.*, Faro, Portugal, 2003, pp. 519–533.
- [30] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Prob. Solv. Nat.*, Birmingham, U.K., 2004, pp. 832–842.
- [31] 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.
- [32] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evol. Comput.*, vol. 15, no. 1, pp. 1–28, 2007.
- [33] 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.
- [34] D. Brockhoff, T. Wagner, and H. Trautmann, "On the properties of the R2 indicator," in *Proc. 14th Annu. Conf. Genet. Evol. Comput.*, Philadelphia, PA, USA, 2012, pp. 465–472.
- [35] O. Schutze, X. Esquivel, A. Lara, and C. A. C. Coello, "Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 4, pp. 504–522, Aug. 2012.
- [36] H. Trautmann, T. Wagner, and D. Brockhoff, "R2-EMOA: Focused multiobjective search using R2-indicator-based selection," in *Proc. Learn. Intell. Optim.*, Catania, Italy, 2013, pp. 70–74.
- [37] R. H. Gomez and C. A. C. Coello, "MOMBI: A new metaheuristic for many-objective optimization based on the R2 indicator," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 2488–2495.
- [38] C. A. R. Villalobos and C. A. C. Coello, "A new multi-objective evolutionary algorithm based on a performance assessment indicator," in *Proc. 14th Annu. Conf. Genet. Evol. Comput.*, Philadelphia, PA, USA, 2012, pp. 505–512.
- [39] 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.
- [40] 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.
- [41] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. IEEE Congr. Evol. Comput.*, Canberra, ACT, Australia, 2003, pp. 2678–2684.
- [42] Y. Yuan, H. Xu, and B. Wang, "Evolutionary many-objective optimization using ensemble fitness ranking," in *Proc. 16th Annu. Conf. Genet. Evol. Comput.*, Vancouver, BC, Canada, 2014, pp. 669–676.
- [43] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, Aug. 2013.

- [44] 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.
- [45] Y. Qi *et al.*, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.
- [46] 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.
- [47] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective sub-problems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.
- [48] 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.
- [49] Z. Wang, Q. Zhang, M. Gong, and A. Zhou, "A replacement strategy for balancing convergence and diversity in MOEA/D," in *Proc. IEEE Congr. Evol. Comput.*, Beijing, China, 2014, pp. 2132–2139.
- [50] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.
- [51] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.
- [52] K. Deb, K. Miettinen, and S. Chaudhuri, "Toward an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches," *IEEE Trans. Evol. Comput.*, vol. 14, no. 6, pp. 821–841, Dec. 2010.
- [53] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," in *Proc. IEEE Congr. Evol. Comput.*, Honolulu, HI, USA, 2002, pp. 825–830.
- [54] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Trans. Evol. Comput.*, vol. 10, no. 5, pp. 477–506, Oct. 2006.
- [55] 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.
- [56] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, Apr. 2015.
- [57] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space," in *Proc. Int. Conf. Parallel Prob. Solv. Nat.*, Kraków, Poland, 2010, pp. 91–100.
- [58] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the hypervolume indicator: Optimal  $\mu$ -distributions and the choice of the reference point," in *Proc. 10th ACM SIGEVO Workshop Found. Genet. Algorithms*, Orlando, FL, USA, 2009, pp. 87–102.
- [59] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 86–95, Feb. 2012.
- [60] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: Classifications, analyses, and new innovations," Dept. Air Force, Air Force Inst. Technology, Wright-Patterson AFB, OH, USA, Tech. Rep. AFIT/DS/ENG/99-01, 1999.
- [61] M. Li, S. Yang, and X. Liu, "Diversity comparison of Pareto front approximations in many-objective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2568–2584, Dec. 2014.
- [62] J. J. Durillo and A. J. Nebro, "jMetal: A Java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, 2011.
- [63] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.
- [64] K. C. Tan, C. K. Goh, Y. Yang, and T. H. Lee, "Evolving better population distribution and exploration in evolutionary multi-objective optimization," *Eur. J. Oper. Res.*, vol. 171, no. 2, pp. 463–495, 2006.
- [65] K. C. Tan, S. C. Chiam, A. Mamun, and C. K. Goh, "Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization," *Eur. J. Oper. Res.*, vol. 197, no. 2, pp. 701–713, 2009.
- [66] P. C. Roy, M. M. Islam, K. Murase, and X. Yao, "Evolutionary path control strategy for solving many-objective optimization problem," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 702–715, Apr. 2015.



machine learning, and their applications.

**Yuan Yuan** received the B.S. degree in software engineering from Southeast University, Nanjing, China, in 2010. He is currently working toward the Ph.D. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing, China.

From 2014 to 2015, he was a Visiting Ph.D. Student with the Centre of Excellence for Research in Computational Intelligence and Applications, University of Birmingham, Birmingham, U.K. His research interests include evolutionary computation,



papers and holds ten advanced controller invention patents.

**Hua Xu** received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 1998, and the M.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 2000 and 2003, respectively.

He is an Associate Professor with the Department of Computer Science and Technology, Tsinghua University. He is also the copyright owner of six software systems. His research interests include data mining, intelligent information processing, and advanced process controllers for IC manufacturing equipments. He has published over 50 academic

Prof. Xu received the Second Prize from National Science and Technology Progress of China, the First Prize from Beijing Science and Technology, and the Third Prize from Chongqing Science and Technology.



**Bo Wang** received the B.S. degree from Tianjin University, Tianjin, China, in 2011. He is currently working toward the M.S. degree with the Department of Computer Science and Technology, Tsinghua University, Beijing, China.

His research interests include evolutionary computation and machine learning.



**Bo Zhang** received the B.S. degree from the Department of Automatic Control, Tsinghua University, Beijing, China, in 1958.

He is a Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include artificial intelligence, pattern recognition, neural networks, and intelligent control. He has published about 150 papers and three monographs in the above fields.

Prof. Zhang is a fellow of the Chinese Academy of Sciences.



**Xin Yao** (F'03) received the B.S. degree from University of Science and Technology of China (USTC), Hefei, China, in 1982; the M.S. degree from North China Institute of Computing Technology, Beijing, China, in 1985; and the Ph.D. degree from USTC, Hefei, in 1990.

He is a Chair (Professor) of Computer Science and the Director of the Centre of Excellence for Research in Computational Intelligence and Applications, University of Birmingham, Birmingham, U.K. His research interests include

evolutionary computation and ensemble learning. He has published over 400 refereed publications in international journals and conferences.

Prof. Yao received the 2001 IEEE Donald G. Fink Prize Paper Award, the 2010 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award, the 2010 BT Gordon Radley Award for Best Author of Innovation (Finalist), the 2011 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award, the prestigious Royal Society Wolfson Research Merit Award in 2012, the IEEE CIS Evolutionary Computation Pioneer Award in 2013, and several other best paper awards. He was an Editor-in-Chief of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2003 to 2008. He is the President of the IEEE Computational Intelligence Society for the term from 2014 to 2015.