An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition

Ke Li, *Student Member, IEEE*, Kalyanmoy Deb, *Fellow, IEEE*, Qingfu Zhang, *Senior Member, IEEE*, and Sam Kwong, *Fellow, IEEE*

***Abstract*—Achieving balance between convergence and diversity is a key issue in evolutionary multiobjective optimization. Most existing methodologies, which have demonstrated their niche on various practical problems involving two and three objectives, face significant challenges in many-objective optimization. This paper suggests a unified paradigm, which combines dominance- and decomposition-based approaches, for many-objective optimization. Our major purpose is to exploit the merits of both dominance- and decomposition-based approaches to balance the convergence and diversity of the evolutionary process. The performance of our proposed method is validated and compared with four state-of-the-art algorithms on a number of unconstrained benchmark problems with up to 15 objectives. Empirical results fully demonstrate the superiority of our proposed method on all considered test instances. In addition, we extend this method to solve constrained problems having a large number of objectives. Compared to two other recently proposed constrained optimizers, our proposed method shows highly competitive performance on all the constrained optimization problems.**

***Index Terms*—Constrained optimization, decomposition, evolutionary computation, many-objective optimization, Pareto optimality, steady state.**

I. Introduction

A MULTIOBJECTIVE optimization problem (MOP) can be stated as

$$\begin{aligned} \text{minimize} \quad & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_m(\mathbf{x}))^T \\ \text{subject to} \quad & g_j(\mathbf{x}) \geq 0, \quad j = 1, \cdots, J \\ & h_k(\mathbf{x}) = 0, \quad k = 1, \cdots, K \\ & \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where J and K are the numbers of inequality and equality constraints, respectively. $\Omega = \prod_{i=1}^{n} [a_i, b_i] \subseteq \mathbb{R}^n$ is the decision (variable) space, $\mathbf{x} = (x_1, \ldots, x_n)^T \in \Omega$ is a candidate solution. $\mathbf{F} : \Omega \rightarrow \mathbb{R}^m$ constitutes m conflicting objective functions, and $\mathbb{R}^m$ is called the objective space. The attainable objective set is defined as $\Theta = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in \Omega, g_j(\mathbf{x}) \geq 0, h_k(\mathbf{x}) = 0\}$, for $j \in \{1, \ldots, J\}$ and $k \in \{1, \ldots, K\}$.

$\mathbf{x}^1$ is said to dominate $\mathbf{x}^2$ (denoted as $\mathbf{x}^1 \preceq \mathbf{x}^2$) if and only if $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$ for every $i \in \{1, \ldots, m\}$ and $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$ for at least one index $j \in \{1, \ldots, m\}$. A solution $\mathbf{x}^*$ is Pareto-optimal to (1) if there is no other solution $\mathbf{x} \in \Omega$ such that $\mathbf{x} \preceq \mathbf{x}^*$. $\mathbf{F}(\mathbf{x}^*)$ is then called a Pareto-optimal (objective) vector. In other words, any improvement of a Pareto-optimal vector in one objective must lead to a deterioration in at least one other objective. The set of all Pareto-optimal solutions is called the Pareto-optimal set (PS). Accordingly, the set of all Pareto-optimal vectors, EF $= \{\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m | \mathbf{x} \in PS\}$, is called the efficient front (EF) [1].

Since the early 1990s, much effort has been devoted to developing evolutionary multiobjective optimization (EMO) algorithms for problems with two and three objectives [2]–[9]. However, many real-world applications, such as water distribution systems [10], automotive engine calibration problems [11], and land use management problems [12], often involve four or more objectives. Therefore, it is not surprising that handling a large number of objectives, also known as many-objective optimization, has been one of the major research topics in the EMO community during recent years.

Without any further information from a decision maker, EMO algorithms are usually designed to meet two common but often conflicting goals: minimizing the distances between solutions and the EF (i.e., convergence) and maximizing the spread of solutions along the EF (i.e., diversity). Balancing convergence and diversity becomes much more difficult in many-objective optimization. Generally speaking, challenges brought by a large number of objectives include the following six aspects. First and foremost, with the increasing number of objectives, almost all solutions in a population

Manuscript received June 12, 2014; revised October 3, 2014; accepted November 4, 2014. Date of publication November 21, 2014; date of current version September 29, 2015. This work was supported in part by the National Natural Science Foundation of China under Grants 61473241 and 61272289, and in part by the City University of Hong Kong Shenzhen Research Institute, Shenzhen, China, and the Hong Kong Research Grant Council General Research Fund under Grant 9042038 (CityU 11205314).

K. Li was with the Department of Computer Science, City University of Hong Kong, Hong Kong. He is now with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA.

K. Deb is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824 USA.

Q. Zhang is with the Department of Computer Science, City University of Hong Kong, Hong Kong, with the City University of Hong Kong Shenzhen Research Institute, Shenzhen 5180057, China, and also with the School of Electronic Engineering and Computer Science, University of Essex, Colchester CO4 3SQ, U.K. (e-mail: qingfu.zhang@cityu.edu.hk; qzhang@essex.ac.uk).

S. Kwong is with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the City University of Hong Kong Shenzhen Research Institute, Shenzhen 5180057, China (e-mail: cssamk@cityu.edu.hk).

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.2014.2373386

become nondominated with one another [13]. This aspect severely deteriorates the selection pressure toward the EF and considerably slows down the evolutionary process, since most elite-preserving mechanisms of EMO algorithms employ Pareto dominance as a major selection criteria. Second, with the increase of the objective space in size, the conflict between convergence and diversity becomes aggravated [14]. Since most of the current diversity management operators (see [15], crowding distance [16], and kth nearest distance [17]) prefer selecting the dominance resistant solutions [14], they cannot strengthen the selection pressure toward the EF, and may even impede the evolutionary process to a certain extent. Third, due to the computational efficiency consideration, the population size used in EMO algorithms cannot be arbitrarily large. However, in a high-dimensional objective space, limited number of solutions are likely to be far away from each other. This might lead to the inefficiency for offspring generation, since the reproduction operation usually produces an offspring far away from its parents in a high-dimensional space. Next, it is well-known that the computational complexity for calculating some performance metrics, such as hypervolume [18], grows exponentially with the increasing number of objectives [19]. Finally, the last two challenges are the representation and visualization of trade-off surface. Although these two issues might not directly affect the evolutionary process, they cause severe difficulties for decision making.

Confronted by the mentioned difficulties, the performance of most current EMO algorithms, which have already demonstrated their capabilities for two- and three-objective problems, deteriorate significantly when more than three objectives are involved [20]. The Pareto-based EMO approach (see [16], [17], [21]), whose basic idea is to compare solutions according to Pareto dominance relation and density estimation, is the first to bear the brunt. Due to the first challenge mentioned above, the large proportion of nondominated solutions makes the primary selection criterion, i.e., Pareto dominance relation, fail to distinguish solutions. Instead, also known as active diversity promotion [14], the secondary selection criterion, i.e., density estimation, takes control of both mating and environmental selection. However, according to the aforementioned second challenge, the active diversity promotion leads to a detrimental effect on the convergence toward the EF [13], [22], [23], due to the presence of dominance resistant solutions [24]. During the last decades, indicator-based EMO approach (see [25]–[27]) was regarded as a promising method for many-objective optimization. Different from the Pareto-based approach, it integrates the convergence and diversity into a single indicator, such as hypervolume [18], to guide the selection process. This characteristic waives the first two challenges mentioned above. But unfortunately, the exponentially increased computation cost of hypervolume calculation [19], as the mentioned in the fourth challenge, severely impedes further developments of indicator-based EMO approaches for many-objective optimization. Although some efforts have been made to remedy the computational issue, (see [28]–[30]), it is still far from widely applicable in practice.

In general, there are five viable ways to alleviate the challenges posed in evolutionary many-objective optimization. The first and most straightforward one is the development of new dominance relations that can increase the selection pressure toward the EF. A large amount of studies, in evolutionary many-objective optimization, have been done along this direction, such as ϵ-dominance [31], dominance area control [32], grid-dominance [33], preference order ranking [34], k-optimality [35], and fuzzy-based Pareto optimality [36]. In addition, some nonPareto-based approaches, such as average rank [37], L-optimality [38], and rank-dominance [39], have also demonstrated their abilities for handling problems with a large number of objectives.

Another avenue is the decomposition-based method, which decomposes a MOP, in question, into a set of subproblems and optimizes them in a collaborative manner. Note that the decomposition concept is so general that either aggregation functions or simpler MOPs [40] can be used to form subproblems. Since the weight vectors of these subproblems are widely spread, the obtained solutions are expected to have a wide distribution over the EF. MOEA/D, proposed in [41], is a representative of this sort. In MOEA/D, each solution is associated with a subproblem, and each subproblem is optimized by using information from its neighborhoods. During the past few years, MOEA/D, as a major framework to design EMO algorithms, has spawned a large amount of research works, e.g., introducing adaptive mechanism in reproduction [42], hybridizing with local search [43], and incorporating stable matching in selection [44]. However, the effectiveness and usefulness of these algorithms for many-objective optimization is yet to be validated. Most existing studies of MOEA/D in many-objective scenario mainly concentrate on investigations of its search behavior (see [45]–[49]). It is worth noting that the cellular multiobjective genetic algorithm (C-MOGA), proposed in [50], can also be regarded as an early version of MOEA/D. It specifies several directions for genetic search and uses the same neighborhood concept for mating restriction. C-MOGA differs from MOEA/D in selecting solutions for offspring reproduction and updating the internal population, and it has to insert solutions from its external population to its internal population at each generation for dealing with nonconvex EFs [41]. Furthermore, the recently proposed NSGA-III [49] also employs a decomposition-based idea to maintain population diversity, while the convergence is still controlled by Pareto dominance.

The third way is to rescue the loss of selection pressure by improving the diversity management mechanism. Although it sounds quite intuitive, surprisingly, not much work have been done along this direction. In [51], a diversity management operator is proposed to control the activation and deactivation of the crowding measure in NSGA-II [16]. In [22], a significant improvement on the convergence performance has been witnessed by a simple modification on the assignment of crowding distance in NSGA-II. In [52], a shift-based density estimation strategy is proposed to assign high density values to the poorly converged solutions by putting them into crowded areas. To a certain extent, the essence of the niche-preservation operation of NSGA-III can also be regarded as an improved diversity management scheme that remedies the loss of selection pressure caused by the mushrooming of nondominated solutions in a high-dimensional objective space.

The last two feasible remedies are multicriteria decision making-based EMO methodologies and objective reduction. The former one concerns finding a preferred subset of solutions from the whole PS (see [53]–[55]). Therefore, the aforementioned difficulties can be alleviated due to the shrunk search space. Based on the assumption of the existence of redundant objectives in a many-objective optimization problem, the latter one considers employing some dimensionality reduction techniques, such as principal component analysis [56], to identify the embedded EF in the ambient objective space. As a consequence, the classic EMO methodologies, proposed for two- and three-objective cases, can be readily applicable for this reduced objective space.

As highlighted in [57], research on evolutionary many-objective optimization is still in its infancy. Great improvements are still needed before EMO algorithms can be considered as an effective tool for many-objective optimization as in two- and three-objective cases. Recent studies in [58] have shown that the representatives of Pareto- and decomposition-based EMO approaches, i.e., NSGA-II and MOEA/D, respectively, are suitable for different problems. This observation greatly motivates us to exploit the merits from both approaches for further improvements and wider applicabilities. In this paper, we propose a unified paradigm, termed MOEA/DD, which combines the dominance- and decomposition-based approaches, to tackle the first three challenges posed in many-objective optimization. The major contributions of this paper are summarized as follows.

1) We present a systematic approach to generate widely spread weight vectors in a high-dimensional objective space. Each weight vector not only defines a subproblem, but also specifies a unique subregion in the objective space.
2) To tackle the aforementioned challenge for diversity management in a high-dimensional objective space, the density of the population is estimated by the local niche count of a subregion.
3) Consider the aforementioned third challenge, a mating restriction scheme is proposed to make most use of the mating parents chosen from neighboring subregions.
4) Inherit the merit of steady-state selection scheme of MOEA/D, each time, only one offspring is considered for updating the population.
5) To tackle the first challenge mentioned, the update of population is conducted in a hierarchical manner, which depends on Pareto dominance, local density estimation, and scalarization functions, sequentially. Moreover, in order to further improve the population diversity, a second chance is provided to the worst solution in the last nondomination level, in case it is associated with an isolated subregion.
6) The proposed MOEA/DD is further extended for constrained optimization problems having a large number of objectives.

The remainder of this paper is organized as follows. Section II provides some background knowledge of this paper. Section III is devoted to the description of our proposed algorithm for many-objective optimization. Experimental settings are provided in Section IV, and comprehensive experiments are conducted and analyzed in Section V. Section VI presents an extension of MOEA/DD for handling constraints in a high-dimensional objective space. Finally, Section VII concludes this paper and threads some future research issues.

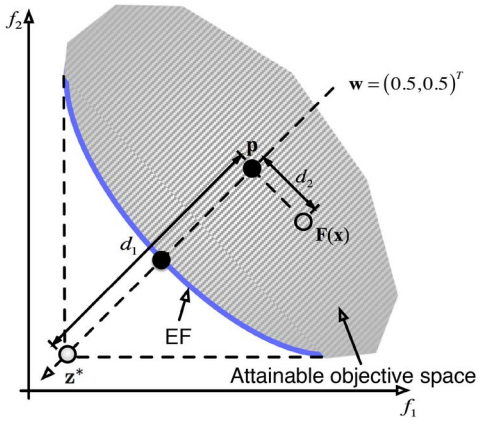

Fig. 1. Illustration of PBI approach.

II. Preliminaries

In this section, we first introduce some basic knowledge about the decomposition method used in MOEA/DD. Then, we briefly introduce the general mechanisms of MOEA/D and NSGA-III, which are related to this paper.

A. Decomposition Methods

In principle, any approach in classic multiobjective optimization [1] can be applied in our algorithm to decompose a MOP, in question, into a set of scalar optimization subproblems. Among them, the most popular ones are weighted sum, weighted Tchebycheff, and boundary intersection approaches [1]. In this paper, we use the penalty-based boundary intersection (PBI) approach [41], due to its promising performance for many-objective optimization reported in [49]. This approach is a variant of the normal-boundary intersection method [59], where equality constraint is handled by a penalty function. More formally, the optimization problem of PBI approach is defined as

$$\begin{aligned} &\text{minimize} \quad g^{\text{pbi}}(\mathbf{x}|\mathbf{w},\mathbf{z}^*) = d_1 + \theta d_2 \\ &\text{subject to} \quad \mathbf{x} \in \Omega \end{aligned} \tag{2}$$

where

$$d_1 = \frac{\left\|(\mathbf{F}(\mathbf{x}) - \mathbf{z}^*)^T \mathbf{w}\right\|}{\|\mathbf{w}\|} \tag{3}$$

$$d_2 = \left\|\mathbf{F}(\mathbf{x}) - \left(\mathbf{z}^* + d_1 \frac{\mathbf{w}}{\|\mathbf{w}\|}\right)\right\| \tag{4}$$

$\mathbf{z}^* = (z_1^*, \ldots, z_m^*)^T$ is the ideal objective vector with $z_i^* < \min_{\mathbf{x}\in\Omega} f_i(\mathbf{x})$, $i \in \{1, \ldots, m\}$, $\theta \geq 0$ is a user-defined penalty parameter. Fig. 1 presents a simple example to illustrate d_1 and d_2 of a solution $\mathbf{x}$ with regard to a weight vector $\mathbf{w} = (0.5, 0.5)^T$. In the PBI approach, d_1 is used to evaluate the convergence of $\mathbf{x}$ toward the EF and d_2 is a kind of measure for population diversity. By adding the value of d_2 multiplied by θ to d_1, $g^{\text{pbi}}(\mathbf{x}|\mathbf{w},\mathbf{z}^*)$ plays as a composite measure of $\mathbf{x}$ for both convergence and diversity. The balance

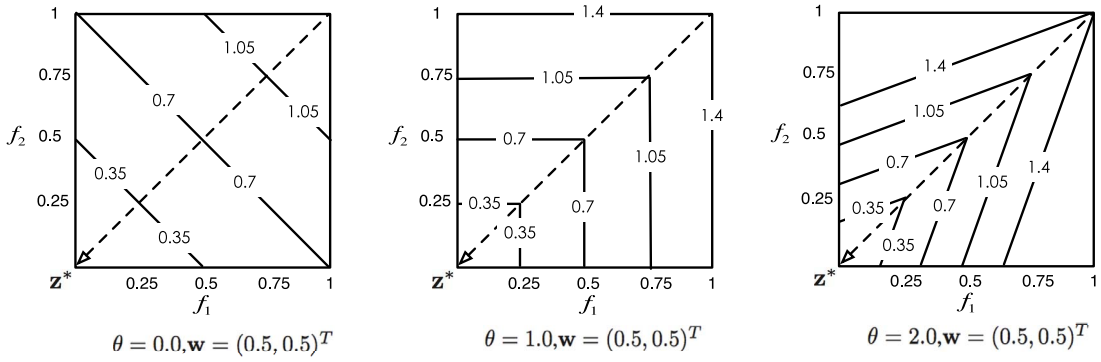

Fig. 2. Illustration of contours of PBI function with different settings of θ and $\mathbf{w}$.

between d_1 and d_2 is controlled by the parameter θ, and the goal of PBI approach is to push $\mathbf{F}(\mathbf{x})$ as low as possible so that it can reach the boundary of Θ.

Fig. 2 presents contours of the PBI function in three 2-D cases, with $\theta = 0.0$, $\theta = 1.0$, and $\theta = 2.0$, respectively, for weight vector $\mathbf{w} = (0.5, 0.5)^T$. From this figure, we observe that different settings of θ lead to distinct search behaviors of the PBI approach. In particular, the contours of PBI approach are the same as those of weighted sum approach when $\theta = 0.0$. As reported in [48], the performance of MOEA/D with PBI ($\theta = 0.1$) is similar to MOEA/D with weighted sum on multiobjective knapsack problems. When $\theta = 1.0$, the contours of PBI are the same as those of the weighted Tchebycheff. Moreover, the contours in this case have the same shape as a dominating region of a point. This implies that MOEA/D with PBI ($\theta = 1.0$) should have a similar search behavior as MOEA/D with weighted Tchebycheff. In this paper, we set $\theta = 5.0$ for empirical studies, in view of its reportedly promising results on tackling various continuous MOPs in [41] and [49]. It is worth noting that MOEA/D with weighted sum and PBI with $\theta = 0.1$ outperform MOEA/D with weighted Tchebycheff and PBI with $\theta = 5.0$ on multiobjective knapsack problems [48].

B. MOEA/D

As a representative of the decomposition-based method, the basic idea of MOEA/D is to decompose a MOP into a number of single objective optimization subproblems through aggregation functions and optimizes them simultaneously. Since the optimal solution of each subproblem is proved to be Pareto-optimal to the MOP in question, the collection of optimal solutions can be treated as a good EF approximation. There are two major features of MOEA/D: one is local mating, the other is local replacement. In particular, local mating means that the mating parents are usually selected from some neighboring weight vectors. Then, an offspring solution is generated by applying crossover and mutation over these selected parents. Local replacement means that an offspring is usually compared with solutions among some neighboring weight vectors. However, as discussed in [60], with a certain probability, it is helpful for the population diversity to conduct mating and replacement from all weight vectors. A parent solution can be replaced by an offspring only when it has a better aggregation function value.

Algorithm 1: General Framework of MOEA/DD

Output: population P

```
[P, W, E] ←INITIALIZATION() ;        // P is the parent population, W is the weight vector set and E is the neighborhood index set
while termination criterion is not fulfilled do
    for i ← 1 to N do
        P̄ ←MATING_SELECTION(E(i), P);
        S ←VARIATION(P̄);
        foreach x^c ∈ S do // x^c is an offspring
            P ←UPDATE_POPULATION(P, x^c)
        end
    end
end
return P
```

C. NSGA-III

It is a recently proposed algorithm for many-objective optimization. Similar to the idea of weight vector generation in MOEA/D, NSGA-III specifies a set of reference points that evenly spread over the objective space. In each generation, the objective function values of each solution is normalized to [0, 1]. Each solution is associated with a reference point based on its perpendicular distance to the reference line. After the generation of a population of offspring solutions, they are combined with their parents to form a hybrid population. Then, a nondominated sorting procedure is applied to divide the hybrid population into several nondomination levels. Solutions in the first level have the highest priority to be selected as the next parents, so on and so forth. Solutions in the last acceptable level are selected based on a niche-preservation operator, where solutions associated with a less crowded reference line have a larger chance to be selected.

III. Proposed Algorithm: MOEA/DD

A. Framework of Proposed Algorithm

Algorithm 1 presents the general framework of MOEA/DD. First, the initialization procedure generates N initial solutions and N weight vectors. Within the main while-loop, in case the termination criteria are not met, for each weight vector, the

Algorithm 2: Initialization Procedure (INITIALIZATION)

Output: parent population P, weight vector set W, neighborhood set of weight vectors E

1 Generate an initial parent population $P = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ by random sampling from Ω;
2 **if** $m < 7$ **then**
3 Use Das and Dennis's method to generate $N = \binom{H+m-1}{m-1}$ weight vectors $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$;
4 **else**
5 Use Das and Dennis's method to generate $N_1 = \binom{H_1+m-1}{m-1}$ weight vectors $B = \{\mathbf{b}^1, \ldots, \mathbf{b}^{N_1}\}$ and $N_2 = \binom{H_2+m-1}{m-1}$ weight vectors $I = \{\mathbf{i}^1, \ldots, \mathbf{i}^{N_2}\}$, where $N_1 + N_2 = N$;
6 **for** $k \leftarrow 1$ **to** N_2 **do**
7 **for** $j \leftarrow 1$ **to** m **do**
8 $i_j^k = \frac{1-\tau}{m} + \tau \times i_j^k$;
9 **end**
10 **end**
11 $W_{m\times N} \leftarrow B_{m\times N_1} \cup I_{m\times N_2}$;
12 **end**
13 **for** $i \leftarrow 1$ **to** N **do**
14 $E(i) = \{i_1, \ldots, i_T\}$ where $\mathbf{w}^{i_1}, \ldots, \mathbf{w}^{i_T}$ are the T closest weight vectors to $\mathbf{w}^i$;
15 **end**
16 Use the nondominated sorting method to divide P into several nondomination levels $F_1, F_2, \ldots, F_l$;
17 Associate each member in P with a unique subregion;
18 **return** P, W, E

mating selection procedure chooses some parents for offspring generation. Then, the offspring is used to update the parent population according to some elite-preserving mechanism. In the following paragraphs, the implementation details of each component in MOEA/DD will be explained step-by-step.

B. Initialization Procedure

The initialization procedure of MOEA/DD (line 1 of Algorithm 1), whose pseudo-code is given in Algorithm 2, contains three main aspects: the initialization of parent population P, the identification of nondomination level structure of P, the generation of weight vectors, and the assignment of neighborhood. To be specific, the initial parent population P is randomly sampled from Ω via a uniform distribution. A set of weight vectors $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$ is generated by a systematic approach developed from Das and Dennis's method [59]. In this approach, weight vectors are sampled from a unit simplex. $N = \binom{H+m-1}{m-1}$ points, with a uniform spacing $\delta = 1/H$, where $H > 0$ is the number of divisions considered along each objective coordinate, are sampled on the simplex for any number of objectives. Fig. 3 gives a simple example to illustrate the Das and Dennis's weight vector generation method.

As discussed in [49], in order to have intermediate weight vectors within the simplex, we should set $H \geq m$. However, in a high-dimensional objective space, we will have a large

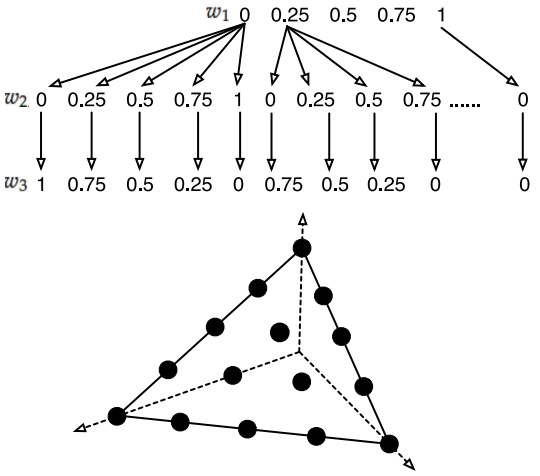

Fig. 3. Structured weight vector $\mathbf{w} = (w_1, w_2, w_3)^T$ generation process with $\delta = 0.25$, i.e., $H = 4$ in 3-D space [59]. $\binom{4+3-1}{3-1} = 15$ weight vectors are sampled from an unit simplex.

amount of weight vectors even if $H = m$ (e.g., for an seven-objective case, $H = 7$ will results in $\binom{7+7-1}{7-1} = 1716$ weight vectors). This obviously aggravates the computational burden of an EMO algorithm. On the other hand, if we simply remedy this issue by lowering H (e.g., set $H < m$), it will make all weight vectors sparsely lie along the boundary of the simplex. This is apparently harmful to the population diversity. To avoid such situation, we present a two-layer weight vector generation method. At first, the sets of weight vectors in the boundary and inside layers (denoted as $B = \{\mathbf{b}^1, \ldots, \mathbf{b}^{N_1}\}$ and $I = \{\mathbf{i}^1, \ldots, \mathbf{i}^{N_2}\}$, respectively, where $N_1 + N_2 = N$) are initialized according to the Das and Dennis's method, with different H settings. Then, the coordinates of weight vectors in the inside layer are shrunk by a coordinate transformation. Specifically, as for a weight vector in the inside layer $\mathbf{i}^k = (i_1^k, \ldots, i_m^k)^T$, $k \in \{1, \ldots, N_2\}$, its jth component is reevaluated as

$$i_j^k = \frac{1-\tau}{m} + \tau \times i_j^k \tag{5}$$

where $j \in \{1, \ldots, m\}$ and $\tau \in [0, 1]$ is a shrinkage factor (here we set $\tau = 0.5$ without loss of generality). At last, B and I are combined to form the final weight vector set W. Fig. 4 presents a simple example to illustrate our two-layer weight vector generation method.

Each weight vector $\mathbf{w}^i = (w_1^i, \cdots, w_m^i)^T$, $i \in \{1, \ldots, N\}$, specifies a unique subregion, denoted as Φ^i, in the objective space, where Φ^i is defined as

$$\Phi^i = \left\{\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m | \langle \mathbf{F}(\mathbf{x}), \mathbf{w}^i \rangle \leq \langle \mathbf{F}(\mathbf{x}), \mathbf{w}^j \rangle\right\} \tag{6}$$

where $j \in \{1, \ldots, N\}$, $\mathbf{x} \in \Omega$ and $\langle \mathbf{F}(\mathbf{x}), \mathbf{w}^j \rangle$ is the acute angle between $\mathbf{F}(\mathbf{x})$ and $\mathbf{w}^j$. It is worth noting that a recently proposed EMO algorithm MOEA/D-M2M [40] also employs a similar method to divide the objective space. However, its idea is in essence different from our method. To be specific, in MOEA/DD, the assignment of subregions is to facilitate the local density estimation which will be illustrated in detail later; while, in MOEA/D-M2M, different weight vectors are

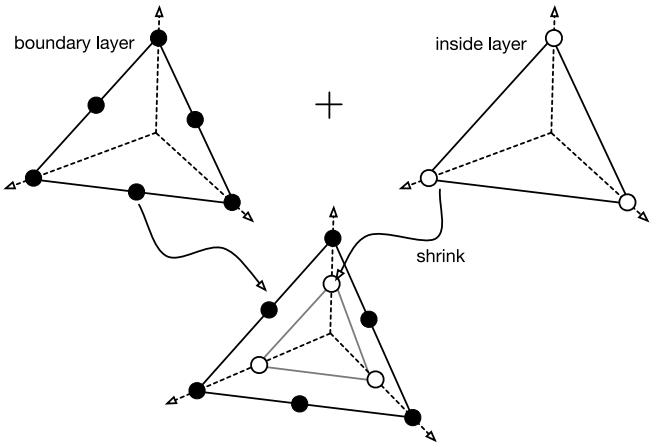

Fig. 4. Example our two-layer weight vector generation method, with $H_1 = 2$ for the boundary layer and $H_2 = 1$ for the inside layer.

Algorithm 3: Mating Selection (MATING_SELECTION)

Input: neighborhood set of the current weight vector $E(i)$, parent population P
Output: mating parent set $\overline{P}$

```
1  if rnd < δ then
2      Randomly choose k indices from E(i);
3      if no solution in the selected subregions then
4          Randomly choose k solutions from P to form P̄;
5      else
6          Randomly choose k solutions from the selected
           subregions to form P̄;
7      end
8  else
9      Randomly choose k solutions from P to form P̄;
10 end
11 return P̄
```

used to specify different subpopulations for approximating a small segment of the EF.

For each weight vector $\mathbf{w}^i$, $i \in \{1, \ldots, N\}$, its neighborhood consists of T closest weight vectors evaluated by Euclidean distances. Afterwards, solutions in P are divided into different nondomination levels (i.e., $F_1, \ldots, F_l$, where $l \leq N$) by using the fast nondominated sorting method suggested in [16]. At last, each member of P is initially associated with a unique subregion in a random manner.

C. Reproduction Procedure

Reproduction procedure (lines 4 and 5 of Algorithm 1), is to generate offspring solutions to update the parent population. It contains two steps: 1) mating selection, which chooses some mating parents for offspring generation and 2) variation operation, which generates new candidate solutions based on those selected mating parents.

Considering the third challenge posed in Section I, it is desirable that the mating parents should be chosen from a neighborhood as much as possible. In MOEA/DD, each solution is associated with a subregion, uniquely specified by a weight vector; and each weight vector (or subregion) has been assigned with a neighborhood based on the Euclidean distance. Thus, for the current weight vector, we can easily choose neighboring solutions from its neighboring subregions. In case no associated solution exists in the selected subregions, mating parents are randomly chosen from the whole population. Moreover, in order to enhance the exploration ability [60], we also allow the mating parents to be selected from the whole population with a low probability $1 - \delta$, where $\delta \in [0, 1]$. The pseudo-code of the mating selection procedure is given in Algorithm 3.

As for variation operation, in principle, any genetic operator can serve this purpose. In this paper, we use the simulated binary crossover (SBX) [61] and polynomial mutation [62] as in [49].

D. Update Procedure

After the generation of offspring solutions, we use them to update the parent population P. The pseudo-code of this update procedure is presented in Algorithm 4. It is worth noting that we only consider one offspring each time. That is to say, multiple rounds of this update procedure will be implemented if more than one offspring solution have been generated. First and foremost, we identify the associated subregion of the offspring solution $\mathbf{x}^c$ (line 1 of Algorithm 4). Then, $\mathbf{x}^c$ is combined with P to form a hybrid population P' (line 2 of Algorithm 4). Afterwards, we need to identify the nondomination level structure of P'. In traditional steady-state methodologies (see [63]–[65]), the nondomination level structure of a population is usually obtained by conducting nondominated sorting from scratch. However, as discussed in [66], the nondomination level structure of P has already been obtained in the previous generation. Thus, it always happens that not all nondomination levels need to be changed, when introducing a new member to P. For example, in Fig. 5(a), no solution needs to change its nondomination level; while in Fig. 5(b), only some solutions change their nondomination levels. Therefore, applying the nondominated sorting over P' from scratch, each time, obviously incurs many unnecessary dominance comparisons. It will be even more time-consuming when the number of objectives and the population size become large. In view of this consideration, we use the efficient method[1] proposed in [66] to update the nondomination level structure of P'. Analogously, after the update procedure, where an inferior solution is eliminated from P', we also use the method suggested in [66] to update the nondomination level structure of the newly formed P.

Generally speaking, we might have the one of the following two scenarios when updating P.

1) There is Only One Nondomination Level (i.e., $l = 1$): In this case, all members in P' are nondominated from each other. Therefore, we have to seek other measures, such as density estimation and scalarization function, to distinguish solutions. Since each solution is associated with a subregion, we can estimate the density (or niche count) of a subregion by counting the number of solutions associated with it. As

[1]Due to the page limit, this nondomination level update method is not illustrated in this paper, interested readers are recommended to [66] for details.

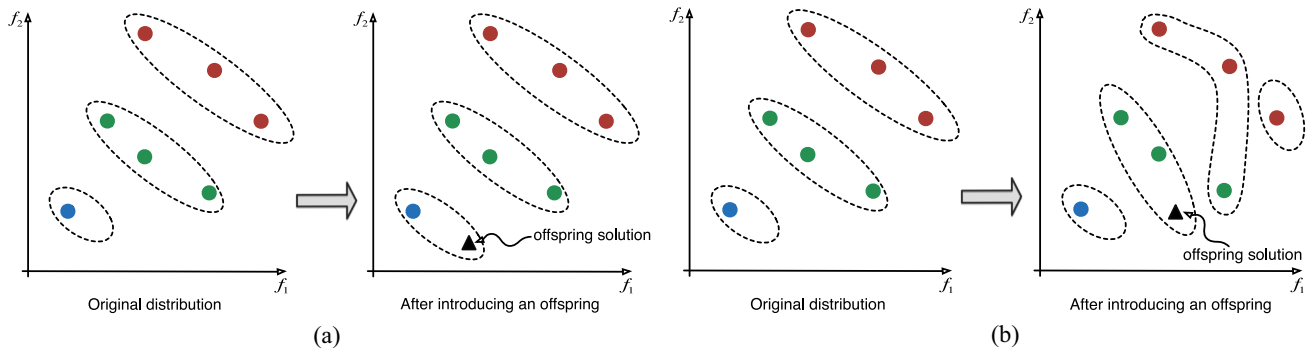

Fig. 5. Illustration of updating nondomination level structure when introducing a new offspring solution. (a) Nondomination level structure keep unchanged. (b) Nondomination level structure has been changed.

shown in Algorithm 5, we identify the most crowded subregion (denoted as Φ^h) that has the largest niche count. In case more than one subregion has the same largest niche count, we choose the one with the largest sum of PBI values

$$h = \underset{i \in S}{\operatorname{argmax}} \sum_{\mathbf{x} \in \Phi^i} g^{\text{pbi}}\left(\mathbf{x}|\mathbf{w}^i, \mathbf{z}^*\right) \tag{7}$$

where S is the set of subregion indices that have the same largest niche count. Finally, we identify the worst solution (denoted as $\mathbf{x}'$) in Φ^h that has the largest PBI value

$$\mathbf{x}' = \underset{\mathbf{x} \in \Phi^h}{\operatorname{argmax}}\, g^{\text{pbi}}\left(\mathbf{x}|\mathbf{w}^h, \mathbf{z}^*\right). \tag{8}$$

Thereafter, $\mathbf{x}'$ is eliminated from P'.

2) There Are More Than One Nondomination Levels (i.e., $l > 1$): Since only one solution needs to be eliminated from P', we start the decision process from the last nondomination level F_l. There exists the following two cases.

1) $|F_l| = 1$, i.e., F_l contains only one solution (denoted as $\mathbf{x}^l$). First of all, we investigate the density of the subregion (denoted as Φ^l) associated with $\mathbf{x}^l$.
   a) If more than one solution (including $\mathbf{x}^l$) have been associated with Φ^l, $\mathbf{x}^l$ is eliminated from P' (line 10 of Algorithm 4). This is because $\mathbf{x}^l$ belongs to F_l and Φ^l contains some other better solutions in term of convergence. Therefore, $\mathbf{x}^l$ cannot provide any further useful information. Fig. 6 presents an example to illustrate this issue.
   b) Otherwise, Φ^l is regarded as an isolated subregion, which means that Φ^l might be an unexploited area in the objective space. In this case, $\mathbf{x}^l$ is very important for population diversity and it should survive to the next round without reservation. Instead, we identify the most crowded subregion Φ^h [tie is broken according to (7)]. Then, we find out the solutions (denoted as a set R) in Φ^h that belong to the current worst nondomination level. Finally, the worst solution $\mathbf{x}' \in R$, which has the largest PBI value [according to (9)], is eliminated from P' (lines 12 and 13 of Algorithm 4). Fig. 7 presents an example to illustrate this issue

$$\mathbf{x}' = \underset{\mathbf{x} \in R}{\operatorname{argmax}}\, g^{\text{pbi}}\left(\mathbf{x}|\mathbf{w}^h, \mathbf{z}^*\right). \tag{9}$$

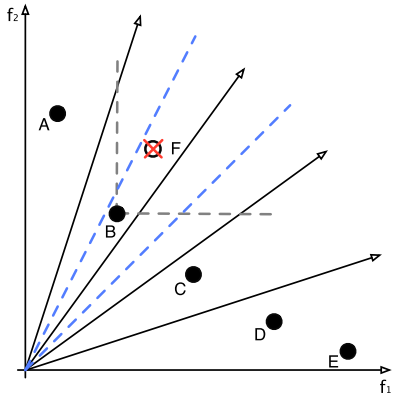

Fig. 6. F will be eliminated as it belongs to the last nondomination level and there is another better solution B associated with F's subregion.

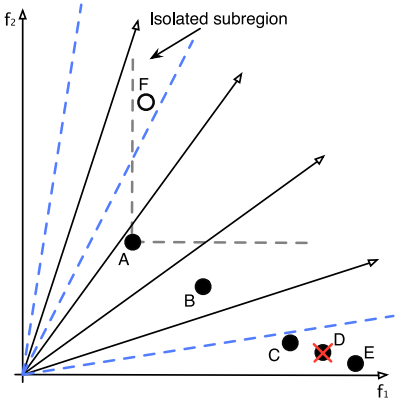

Fig. 7. Although F belongs to the worst nondomination level, it is associated with an isolated subregion. This indicates that F is important for population diversity and it should be preserved without reservation. Instead, we eliminate the worst solution D associated with the most crowded subregion.

2) $|F_l| > 1$, i.e., F_l contains more than one solution. We at first identify the most crowded subregion Φ^h [tie is broken according to (7)] associated with those solutions in F_l (line 16 of Algorithm 4).
   a) If more than one solution is associated with Φ^h, we eliminate the worst solution $\mathbf{x}' \in \Phi^h$, which owns the largest PBI value, according to (8), from P' (lines 18 and 19 of Algorithm 4).

Algorithm 4: Update Procedure (UPDATE_POPULATION)

Input: parent population P, offspring solution $\mathbf{x}^c$
Output: parent population P

1 Find the subregion associated with $\mathbf{x}^c$ according to (6);
2 $P' \leftarrow P \bigcup \{\mathbf{x}^c\}$;
3 Use the method suggested in [66] to update the nondomination level structure of P';
4 **if** $l = 1$ **then** // All solutions in P' are nondominated from each other
5 $\quad \mathbf{x}' \leftarrow$LOCATE_WORST($P'$);
6 $\quad P \leftarrow P' \setminus \{\mathbf{x}'\}$;
7 **else**
8 $\quad$ **if** $|F_l| = 1$ **then** // F_l has only one solution $\mathbf{x}^l$
9 $\quad\quad$ **if** $|\Phi^l| > 1$ **then** // Φ^l is the associated subregion of $\mathbf{x}^l$
10 $\quad\quad\quad P \leftarrow P' \setminus \{\mathbf{x}^l\}$;
11 $\quad\quad$ **else** // $|\Phi^l| = 1$
12 $\quad\quad\quad \mathbf{x}' \leftarrow$LOCATE_WORST($P'$);
13 $\quad\quad\quad P \leftarrow P' \setminus \{\mathbf{x}'\}$;
14 $\quad\quad$ **end**
15 $\quad$ **else**
16 $\quad\quad$ Identify the most crowded subregion Φ^h associated with those solutions in F_l;
17 $\quad\quad$ **if** $|\Phi^h| > 1$ **then**
18 $\quad\quad\quad$ Find the worst solution $\mathbf{x}' = \text{argmax}_{\mathbf{x}\in\Phi^h} g^{\text{pbi}}(\mathbf{x}|\mathbf{w}^h, \mathbf{z}^*)$;
19 $\quad\quad\quad P \leftarrow P' \setminus \{\mathbf{x}'\}$;
20 $\quad\quad$ **else** // $|\Phi^h| = 1$
21 $\quad\quad\quad \mathbf{x}' \leftarrow$LOCATE_WORST($P'$);
22 $\quad\quad\quad P \leftarrow P' \setminus \{\mathbf{x}'\}$;
23 $\quad\quad$ **end**
24 $\quad$ **end**
25 **end**
26 Use the method suggested in [66] to update the nondomination level structure of P;
27 **return** P

Algorithm 5: Find the Worst Solution (LOCATE_WORST)

Input: hybrid population P'
Output: the worst solution $\mathbf{x}'$

1 Identify the most crowded subregion Φ^h in P', tie is broken as $h = \text{argmax}_{i\in S} \sum_{\mathbf{x}\in\Phi^i} g^{\text{pbi}}(\mathbf{x}|\mathbf{w}^i, \mathbf{z}^*)$;
2 In Φ^h, find the solution set R that belong to the worst nondomination level;
3 Find the worst solution $\mathbf{x}' = \text{argmax}_{\mathbf{x}\in R} g^{\text{pbi}}(\mathbf{x}|\mathbf{w}^h, \mathbf{z}^*)$;
4 **return** $\mathbf{x}'$

b) Otherwise, if the niche count of Φ^h is one, it means that every member in F_l is associated with an isolated subregion. As discussed before, such solutions should be preserved for the next round without reservation. Similar to the operations done in case 1) b), we eliminate the worst solution $\mathbf{x}'$ from P' (lines 21 and 22 of Algorithm 4).

After eliminating an inferior solution from P', we apply the method suggested in [66] to update the nondomination level structure of the newly formed P.

E. Discussion

After describing the technical details of MOEA/DD, this section discusses the similarities and differences of MOEA/DD, MOEA/D, and NSGA-III.

1) *Similarities Between MOEA/DD and MOEA/D:*
   a) Both of them employ a set of weight vectors to guide the selection procedure.
   b) Both of them rely on a neighborhood concept.
   c) Both of them apply the scalarization function to measure the fitness value of a solution.
2) *Similarities Between MOEA/DD and NSGA-III:*
   a) Both of them employ a set of weight vectors (or reference points) to guide the selection procedure.
   b) In both algorithms, each solution is associated with a weight vector (or reference point).
   c) Both of them divide the population into several nondomination levels according to the Pareto dominance relation.
3) *Differences Between MOEA/DD and MOEA/D:*
   a) MOEA/D abandons the Pareto dominance concept in selection, while the quality of a solution is fully determined by a predefined scalarization function. As discussed in [58], NSGA-II and MOEA/D are, respectively, suitable for different kinds of problems. This observation motivates the combination of Pareto dominance and decomposition approaches in MOEA/DD.
   b) In MOEA/DD, each weight vector not only defines a subproblem that can evaluate the fitness value of a solution, but also specifies a subregion that can be used to estimate the local density of a population.
   c) As discussed in [44], the update/selection procedure of MOEA/D is a one-sided selection, where only subproblems have the rights to select their preferred solutions. In contrast, by associating with a subregion, each solution in MOEA/DD also has the right to select its preferred subproblem (i.e., subregion). After that, a subproblem can only choose its preferred solutions from its associated ones.
   d) In MOEA/DD, since a solution in the last nondomination level will be preserved without reservation if it is associated with an isolated subregion, the convergence speed of MOEA/DD might be slower than that of MOEA/D. However, this mechanism benefits the diversity preservation, which is very important for many-objective optimization.
   e) In MOEA/DD, the neighborhood concept is only used for mating restriction, while it is used for both mating and update procedures in MOEA/D.
4) *Differences Between MOEA/DD and NSGA-III:*
   a) Although MOEA/DD divides the population into several nondomination levels, its selection procedure does not fully obey the decision made by

TABLE I
CHARACTERISTICS OF TEST INSTANCES

Test instance	Characteristics
DTLZ1	linear, multi-modal
DTLZ2	concave
DTLZ3	concave, multi-modal
DTLZ4	concave, biased
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

Pareto dominance relation. In particular, a solution, associated with an isolated subregion, will survives to the next iteration even if it belongs to the last nondomination level. In contrast, considering the example discussed in Fig. 7, point F cannot survive to the next generation in NSGA-III. However, the elimination of F obviously results in a severe loss of population diversity.

b) In MOEA/DD, each weight vector not only specify a unique subregion in the objective space, but also defines a subproblem which can be used to evaluate the fitness value of a solution.

c) MOEA/DD uses a steady-state selection scheme, while the selection procedure of NSGA-III is a generational scheme.

IV. EXPERIMENTAL SETUP

This section devotes to the experimental design for investigating the performance of MOEA/DD. At first, we describe the benchmark problems used in our empirical studies. Then, we introduce the performance metrics used for evaluating the performance of an EMO algorithm. Afterwards, we briefly describe the EMO algorithms used for validating our proposed MOEA/DD. Finally, we provide the general parameter settings in our empirical studies.

A. Benchmark Problems

DTLZ1 to DTLZ4 from the DTLZ test suite [67] and WFG1 to WFG9 from WFG test suite [68] are chosen for our empirical studies. For each DTLZ instance, the number of objectives varies from 3 to 15, i.e., $m \in \{3, 5, 8, 10, 15\}$. For each WFG instance, the number of objectives is set as $m \in \{3, 5, 8, 10\}$. According to the recommendations in [67], the number of decision variables is set as $n = m + r - 1$ for DTLZ test instances, where $r = 5$ for DTLZ1 and $r = 10$ for DTLZ2, DTLZ3 and DTLZ4. As suggested in [68], the number of decision variables is set as $n = k + l$, where the position-related variable $k = 2 \times (m - 1)$ and the distance-related variable $l = 20$ for WFG test instances. The characteristics of all test instances are summarized in Table I.

B. Performance Metrics

In our empirical studies, we consider the following two widely used performance metrics. Both of them can simultaneously measure the convergence and diversity of obtained solutions.

1) Inverted Generational Distance (IGD) Metric [69]: Let P^* be a set of points uniformly sampled over the true EF, and S be the set of solutions obtained by an EMO algorithm. The IGD value of S is computed as

$$IGD(S, P^*) = \frac{\sum_{\mathbf{x}^* \in P^*} dist(\mathbf{x}^*, S)}{|P^*|} \tag{10}$$

where $dist(\mathbf{x}^*, S)$ is the Euclidean distance between a point $\mathbf{x}^* \in P^*$ and its nearest neighbor in S, and $|P^*|$ is the cardinality of P^*. The lower is the IGD value, the better is the quality of S for approximating the whole EF.

Since the analytical forms of the exact Pareto-optimal surfaces of DTLZ1 to DTLZ4 are known *a priori*, and a set of evenly spread weight vectors has been supplied in MOEA/DD, we can exactly locate the intersecting points of weight vectors and the Pareto-optimal surface. And these intersecting points are the targeted Pareto-optimal points, which finally constitute P^*, on the true EF. To be specific, as for DTLZ1, the objective functions of a Pareto-optimal solution $\mathbf{x}^*$ satisfy

$$\sum_{i=1}^{m} f_i(\mathbf{x}^*) = 0.5. \tag{11}$$

Given a line connecting the origin and a weight vector $\mathbf{w} = (w_1, \cdots, w_m)^T$, its intersecting point with the Pareto-optimal surface satisfies

$$\frac{f_i(\mathbf{x}^*)}{w_i} = c \tag{12}$$

where $i \in \{1, \ldots, m\}$ and $c > 0$ is a constant. Put (12) into (11), we have

$$c = 0.5 \times \frac{1}{\sum_{i=1}^{m} w_i}. \tag{13}$$

Finally, we have

$$f_i(\mathbf{x}^*) = 0.5 \times \frac{w_i}{\sum_{j=1}^{m} w_j} \tag{14}$$

where $i \in \{1, \ldots, m\}$.

As for DTLZ2, DTLZ3, and DTLZ4, the objective functions of a Pareto-optimal solution $\mathbf{x}^*$ satisfy

$$\sum_{i=1}^{m} f_i^2(\mathbf{x}^*) = 1. \tag{15}$$

Given a line connecting the origin and a weight vector $\mathbf{w} = (w_1, \ldots, w_m)^T$, its intersecting point with the Pareto-optimal surface satisfies (12). Put (12) into (15), we have

$$c = \frac{1}{\sqrt{\sum_{i=1}^{m} w_i^2}}. \tag{16}$$

Finally, we have

$$f_i(\mathbf{x}^*) = \frac{w_i}{\sqrt{\sum_{j=1}^{m} w_j^2}} \tag{17}$$

where $i \in \{1, \ldots, m\}$.

According to (14) and (17), we can obtain the corresponding targeted Pareto-optimal point for each weight vector. Fig. 8

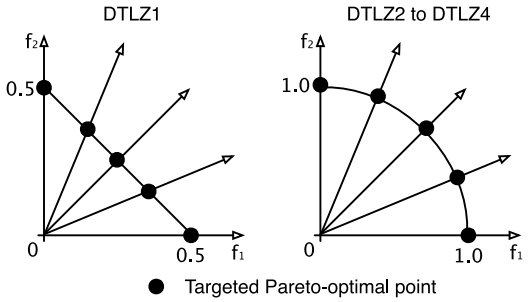

Fig. 8. Simple illustration of finding five targeted Pareto-optimal points on the EFs of DTLZ1 and DTLZ2 to DTLZ4.

TABLE II
SETTINGS OF REFERENCE POINTS

Test instance	Reference point
DTLZ1	$(1.0, \cdots, 1.0)^T$
DTLZ2 to DTLZ4	$(2.0, \cdots, 2.0)^T$
WFG1 to WFG9	$(3.0, \cdots, 2.0 \times m + 1.0)^T$

gives an intuitive illustration of finding five targeted Pareto-optimal points on the EFs of DTLZ1 and DTLZ2 to DTLZ4, respectively, in a 2-D space.

2) Hypervolume (HV) Metric [18]: Let $\mathbf{z}^r = (z_1^r, \ldots, z_m^r)^T$ be a reference point in the objective space that is dominated by all Pareto-optimal objective vectors. HV metric measures the size of the objective space dominated by the solutions in S and bounded by $\mathbf{z}^r$

$$\mathrm{HV}(S) = \mathrm{VOL}\left(\bigcup_{\mathbf{x}\in S}\left[f_1(\mathbf{x}), z_1^r\right] \times \ldots \left[f_m(\mathbf{x}), z_m^r\right]\right) \quad (18)$$

where $\mathrm{VOL}(\cdot)$ indicates the Lebesgue measure. The larger is the HV value, the better is the quality of S for approximating the whole EF.

In our empirical studies, reference points are set according to Table II. For 3- to 10-objective problems, we adopt the recently proposed WFG algorithm [29] to calculate the exact HV, whereas a Monte Carlo sampling [28] is applied to approximate HV in 15-objective cases.[2] Note that solutions, dominated by a reference point, are discarded for HV calculation. The presented HV values in this paper are all normalized to [0, 1] by dividing $z = \Pi_{i=1}^m z_i^r$.

C. EMO Algorithms for Comparisons

We consider four state-of-the-art EMO algorithms, including NSGA-III, MOEA/D, GrEA [70], and HypE [28] for comparisons. Since the general ideas of MOEA/D and NSGA-III have been introduced in Section II-B and II-C, respectively, here we only present the general working principles of GrEA and HypE in the following paragraphs.

1) *GrEA:* It is a many-objective optimizer developed under the framework of NSGA-II, where the fitness assignment, mating selection, and environmental selection have been modified. In particular, to increase the selection pressure of Pareto dominance, grid dominance and grid difference are introduced to distinguish solutions in a grid environment. Moreover, grid ranking, grid crowding distance, and grid coordinate point distance are introduced for fitness assignment.
2) *HypE:* It is an indicator-based EMO algorithm designed specifically for many-objective optimization. In order to alleviate the high computation cost required for exact HV calculation, Monte Carlo simulation is applied to obtain an HV approximation. One other major idea of HypE is to use the rankings of solutions induced by HV values, instead of the actual HV values, in fitness evaluation, mating selection, and environmental selection.

[2]The source code of WFG algorithm is downloaded from http://www.wfg.csse.uwa.edu.au/hypervolume/. The source code of Monte Carlo sampling to approximate HV values can be downloaded from http://www.tik.ee.ethz.ch/sop/download/supplementary/hype/

TABLE III
NUMBER OF WEIGHT VECTORS AND POPULATION SIZE

m	♯ of weight vectors	NSGA-III	MOEA/DD
3	91	92	91
5	210	212	210
8	156 ($H_1 = 3, H_2 = 2$)	156	156
10	275 ($H_1 = 3, H_2 = 2$)	276	275
15	135 ($H_1 = 2, H_2 = 1$)	136	135

The two-layer weight vector generation method is only applied for 8-, 10- and 15-objective problems. H_1 and H_2 is the number of divisions for the boundary layer and inside layer, respectively. NSGA-III, GrEA and HypE use a N setting; while MOEA/DD and MOEA/D use another N setting.

TABLE IV
NUMBER OF GENERATIONS FOR DIFFERENT TEST INSTANCES

Test instance	$m = 3$	$m = 5$	$m = 8$	$m = 10$	$m = 15$
DTLZ1	400	600	750	1,000	1,500
DTLZ2	250	350	500	750	1,000
DTLZ3	1,000	1,000	1,000	1,500	2,000
DTLZ4	600	1,000	1,250	2,000	3,000

D. General Parameter Settings

The five EMO algorithms considered in this paper have several parameters, they are summarized as follows.

1) *Settings for Reproduction Operations:* The crossover probability is $p_c = 1.0$ and its distribution index is $\eta_c = 30$. The mutation probability is $p_m = 1/n$ and its distribution index is $\eta_m = 20$.
2) *Population Size:* The population size N and the number of weight vectors for different number of objectives are summarized in Table III.
3) *Number of Runs and Stopping Condition:* Each algorithm is independently run 20 times on each test instance. The stopping condition of an algorithm is a predefined number of generations, summarized in Table IV.
4) *Penalty Parameter in PBI:* $\theta = 5.0$.
5) *Neighborhood Size:* $T = 20$.
6) *Probability Used to Select in the Neighborhood:* $\delta = 0.9$.
7) *Grid Division (div) in GrEA:* the settings of *div* are summarized in Table V.
8) *Number of Points in Monte Carlo Sampling:* it is set to 10 000 according to [28].

TABLE V
SETTINGS OF GRID DIVISION *div* IN GREA

Test instance	$m = 3$	$m = 5$	$m = 8$	$m = 10$	$m = 15$
DTLZ1	10	10	10	12	12
DTLZ2	10	9	8	8	10
DTLZ3	10	11	10	10	12
DTLZ4	10	9	8	9	10
WFG1	7	14	14	13	14
WFG2	15	16	14	18	20
WFG3	14	14	13	14	15
WFG4	10	11	10	11	12
WFG5	10	11	12	14	14
WFG6	11	12	13	14	15
WFG7	10	11	12	14	16
WFG8	10	12	11	13	14
WFG9	9	12	13	15	15

The suitable div values are chosen by investigating performance of GrEA for $div \in [5, 30]$ with a step size 1.

V. EMPIRICAL RESULTS AND DISCUSSION

A. Performance Comparisons on DTLZ Test Suite

Comparison results of MOEA/DD with other four EMO algorithms in terms of IGD and HV values are presented in Tables VI and VII, respectively. The best metric values are highlighted in bold face with gray background. From these empirical results, it is clear that MOEA/DD is the best optimizer as it wins on almost all comparisons (234 out of 240 for IGD and 235 out 240 for HV). In the following paragraphs, we will explain these results instance by instance.

The EF of DTLZ1 is a linear hyper-plane, where the objective functions of a Pareto-optimal solution $\mathbf{x}^*$ satisfy: $\sum_{i=1}^{m} f_i(\mathbf{x}^*) = 0.5$. The presence of $11^r - 1$ local optima in the search space leads to difficulties for converging to the global EF. From the experimental results, we find that MOEA/DD shows better performance than the other four EMO algorithms in all 3- to 15-objective test instances. NSGA-III obtains better IGD values than MOEA/D in almost all DTLZ1 instances, except the five-objective case. HypE performs worst in all DTLZ1 instances, where it obtains zero HV values in all cases. Fig. 9 shows the parallel coordinates of nondominated fronts obtained by MOEA/DD and the other four EMO algorithms, respectively, for 15-objective DTLZ1 instance. This particular run is associated with the median IGD value. From these five plots, it is observed that the nondominated front obtained by MOEA/DD is promising in both convergence and diversity. Although the nondominated front obtained by NSGA-III is also well converged, the solution distribution is not as good as MOEA/DD. In contrast, the nondominated front obtained by MOEA/D does not converge well on the 11th to 15th objectives. Both GrEA and HypE obtain zero HV values in the 15-objective DTLZ1 instance. This means that their obtained solutions are fully dominated by the reference points. Therefore, their obtained nondominated fronts are far away from the EF.

DTLZ2 is a relatively simple test instance, where the objective functions of a Pareto-optimal solution $\mathbf{x}^*$ need to satisfy: $\sum_{i=1}^{m} f_i^2(\mathbf{x}^*) = 1$. The performance of MOEA/DD and MOEA/D is comparable in this problem. In particular, MOEA/DD shows better results than MOEA/D in 5-, 8-, and 15-objective test instances, while MOEA/D obtains the best IGD values in 3- and 10-objective cases. Moreover, it is worth noting that both MOEA/DD and MOEA/D perform consistently better than NSGA-III in all 3- to 15-objective test instances. Comparing to DTLZ1 instances, the other two EMO algorithms, i.e., GrEA and HypE, obtain much better results this time. Fig. 10 presents the comparisons of the nondominated fronts obtained by these five EMO algorithms, respectively, for 15-objective DTLZ2 instance. It is clear that solutions achieved by MOEA/DD and MOEA/D are similar in terms of convergence and diversity. In contrast, the distribution of solutions achieved by NSGA-III is slightly worse than the other two algorithms. Although the solutions obtained by GrEA and HypE well converge to the EF, their distributions are not satisfied enough.

The EF of DTLZ3 is the same as DTLZ2. But its search space contains $3^r - 1$ local optima, which can make an EMO algorithm get stuck at any local EF before converging to the global EF. The performance of MOEA/DD is significantly better than both NSGA-III and MOEA/D in all 3- to 15-objective test instances. Fig. 11 plots the nondominated fronts obtained by MOEA/DD and the other four EMO algorithms, for 15-objective DTLZ3 instance. It is evident that only the solutions obtained by MOEA/DD converge well to the global EF. In contrast, the solutions obtained by MOEA/D concentrate on several parts of the EF. Similar to the observations in 15-objective DTLZ1, GrEA, and HypE have significant difficulties in converging to the EF (their median HV values are all zero in the 15-objective case).

DTLZ4 also has the identical EF shape as DTLZ2. However, in order to investigate an EMO algorithm's ability to maintain a good distribution of solutions, DTLZ4 introduces a parametric variable mapping to the objective functions of DTLZ2. This modification allows a biased density of points away from $f_m(\mathbf{x}) = 0$. MOEA/D performs significantly worse than MOEA/DD and NSGA-III on this problem. The IGD values obtained by MOEA/D are two or three orders of magnitude larger than those obtained by MOEA/DD and NSGA-III. Nevertheless, the best optimizer is still MOEA/DD, whose IGD values are much smaller than those of NSGA-III. It is also worth noting that the performance of MOEA/DD is rather robust, as the best, median, and worst IGD values obtained by MOEA/DD are in the same order of magnitude. Fig. 12 shows the parallel coordinates of nondominated fronts obtained by these five algorithms. These plots clearly demonstrate that MOEA/DD is able to find a well converged and widely distributed set of points for DTLZ4 instance with 15 objectives. In contrast, both NSGA-III and MOEA/D can only obtain several parts of the true EF. Although the nondominated front obtained by GrEA well converge to the EF, the solution distribution is rather messy. Solutions obtained by HypE seem to concentrate on several extreme points.

B. Performance Comparisons on WFG Test Suite

By introducing a series of composable complexities (such as nonseparability, multimodality, biased parameters, and mixed EF geometries), the WFG test suite poses a significant challenge for algorithms to obtain a well-converged and well-distributed solution set. Table VIII presents the

TABLE VI
BEST, MEDIAN, AND WORST IGD VALUES OBTAINED BY MOEA/DD AND THE OTHER ALGORITHMS ON DTLZ1, DTLZ2, DTLZ3, AND DTLZ4 INSTANCES WITH DIFFERENT NUMBER OF OBJECTIVES. BEST PERFORMANCE IS HIGHLIGHTED IN BOLD FACE WITH GRAY BACKGROUND

	m	MOEA/DD	NSGA-III	MOEA/D	GrEA	HypE		MOEA/DD	NSGA-III	MOEA/D	GrEA	HypE
DTLZ 1	3	**3.191E-4**	4.880E-4	4.095E-4	2.759E-2	1.822E+1	DTLZ 3	**5.690E-4**	9.751E-4	9.773E-4	6.770E-2	1.653E+2
		5.848E-4	1.308E-3	1.495E-3	3.339E-2	1.974E+1		**1.892E-3**	4.007E-3	3.426E-3	7.693E-2	1.700E+2
		6.573E-4	4.880E-3	4.743E-3	1.351E-1	2.158E+1		**6.231E-3**	6.665E-3	9.113E-3	4.474E-1	1.757E+2
	5	**2.635E-4**	5.116E-4	3.179E-4	7.369E-2	1.799E+1		**6.181E-4**	3.086E-3	1.129E-3	5.331E-1	1.826E+2
		2.916E-4	9.799E-4	6.372E-4	3.363E-1	2.141E+1		**1.181E-3**	5.960E-3	2.213E-3	8.295E-1	2.172E+2
		3.109E-4	1.979E-3	1.635E-3	4.937E-1	2.359E+1		**4.736E-3**	1.196E-2	6.147E-3	1.124E+0	2.278E+2
	8	**1.809E-3**	2.044E-3	3.914E-3	1.023E-1	1.030E+1		**3.411E-3**	1.244E-2	6.459E-3	7.518E-1	2.196E+2
		2.589E-3	3.979E-3	6.106E-3	1.195E-1	2.265E+1		**8.079E-3**	2.375E-2	1.948E-2	1.024E+0	2.700E+2
		2.996E-3	8.721E-3	8.537E-3	3.849E-1	2.426E+1		**1.826E-2**	9.649E-2	1.123E+0	1.230E+0	2.949E+2
	10	**1.828E-3**	2.215E-3	3.872E-3	1.176E-1	1.427E+1		**1.689E-3**	8.849E-3	2.791E-3	8.656E-1	1.720E+2
		2.225E-3	3.462E-3	5.073E-3	1.586E-1	1.693E+1		**2.164E-3**	1.188E-2	4.319E-3	1.145E+0	2.893E+2
		2.467E-3	6.869E-3	6.130E-3	5.110E-1	2.034E+1		**3.226E-3**	2.082E-2	1.010E+0	1.265E+0	3.391E+2
	15	**2.867E-3**	2.649E-3	1.236E-2	8.061E-1	1.797E+1		**5.716E-3**	1.401E-2	4.360E-3	9.391E+1	2.358E+2
		4.203E-3	5.063E-3	1.431E-2	2.057E+0	2.519E+1		**7.461E-3**	2.145E-2	1.664E-2	1.983E+2	2.635E+2
		4.699E-3	1.123E-2	1.692E-2	6.307E+1	2.594E+1		**1.138E-2**	4.195E-2	1.260E+0	3.236E+2	3.451E+2
DTLZ 2	3	6.666E-4	1.262E-3	**5.432E-4**	6.884E-2	6.732E-2	DTLZ 4	**1.025E-4**	2.915E-4	2.929E-1	6.869E-2	6.657E-2
		8.073E-4	1.357E-3	**6.406E-4**	7.179E-2	6.910E-2		**1.429E-4**	5.970E-4	4.280E-1	7.234E-2	7.069E-2
		1.243E-3	2.114E-3	**8.006E-4**	7.444E-2	7.104E-2		**1.881E-4**	4.286E-1	5.234E-1	9.400E-1	5.270E-1
	5	**1.128E-3**	4.254E-3	1.219E-3	1.411E-1	2.761E-1		**1.097E-4**	9.849E-4	1.080E-1	1.422E-1	2.603E-1
		1.291E-3	4.982E-3	1.437E-3	1.474E-1	2.868E-1		**1.296E-4**	1.255E-3	5.787E-1	1.462E-1	2.676E-1
		1.424E-3	5.862E-3	1.727E-3	1.558E-1	2.922E-1		**1.532E-4**	1.721E-3	7.348E-1	1.609E-1	5.301E-1
	8	**2.880E-3**	1.371E-2	3.097E-3	3.453E-1	5.475E-1		**5.271E-4**	5.079E-3	5.298E-1	3.229E-1	4.792E-1
		3.291E-3	1.571E-2	3.763E-3	3.731E-1	6.033E-1		**6.699E-4**	7.054E-3	8.816E-1	3.314E-1	4.956E-1
		4.106E-3	1.811E-2	5.198E-3	4.126E-1	6.467E-1		**9.107E-4**	6.051E-1	9.723E-1	3.402E-1	5.387E-1
	10	3.223E-3	1.350E-2	**2.474E-3**	4.107E-1	6.778E-1		**1.291E-3**	5.694E-3	3.966E-1	4.191E-1	6.760E-1
		3.752E-3	1.528E-2	**2.778E-3**	4.514E-1	6.901E-1		**1.615E-3**	6.337E-3	9.203E-1	4.294E-1	6.828E-1
		4.145E-3	1.697E-2	**3.235E-3**	5.161E-1	6.917E-1		**1.931E-3**	1.076E-1	1.077E+0	4.410E-1	6.877E-1
	15	**4.577E-3**	1.360E-2	5.254E-3	5.087E-1	6.237E-1		**1.474E-3**	7.110E-3	5.890E-1	4.975E-1	5.986E-1
		5.863E-3	1.726E-2	6.005E-3	5.289E-1	8.643E-1		**1.881E-3**	3.431E-1	1.133E+0	5.032E-1	6.102E-1
		6.929E-3	2.114E-2	9.409E-3	5.381E-1	3.195E+0		**3.159E-3**	1.073E+0	1.249E+0	5.136E-1	6.126E-1

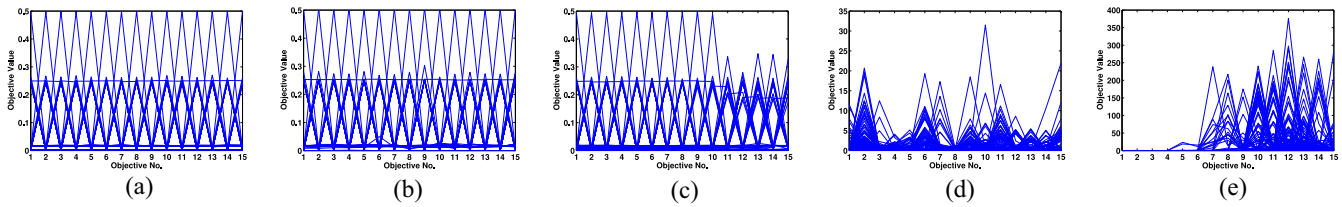

Fig. 9. Parallel coordinates of nondominated fronts obtained by five algorithms on the 15-objective DTLZ1 instance. (a) MOEA/DD. (b) NSGA-III. (c) MOEA/D. (d) GrEA. (e) HypE.

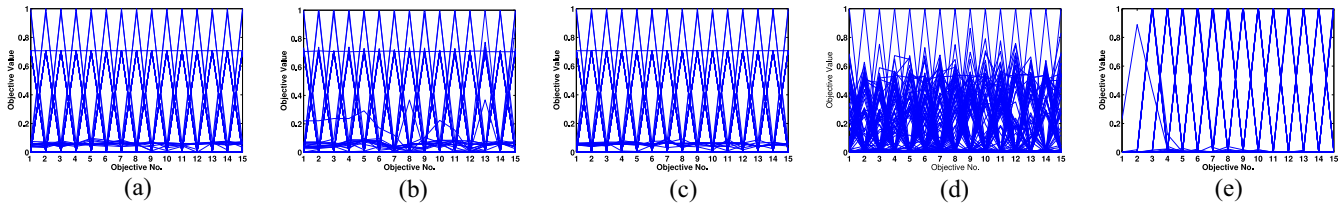

Fig. 10. Parallel coordinates of nondominated fronts obtained by five algorithms on the 15-objective DTLZ2 instance. (a) MOEA/DD. (b) NSGA-III. (c) MOEA/D. (d) GrEA. (e) HypE.

comparison results of MOEA/DD with other three EMO algorithms[3] in terms of HV values. The best metric values are highlighted in bold face with gray background. It is clear that our proposed MOEA/DD shows the best performance in most cases (294 out of 324 comparisons).

WFG1 investigates an EMO algorithm's ability for coping with flat bias and mixed EF geometries (including both

[3]Due the copyright police of NSGA-III, it cannot be used for empirical studies in WFG test suite.

TABLE VII
BEST, MEDIAN, AND WORST HV VALUES OBTAINED BY MOEA/DD AND THE OTHER ALGORITHMS ON DTLZ1, DTLZ2, DTLZ3, AND DTLZ4 INSTANCES WITH DIFFERENT NUMBER OF OBJECTIVES. BEST PERFORMANCE IS HIGHLIGHTED IN BOLD FACE WITH GRAY BACKGROUND

	m	MOEA/DD	NSGA-III	MOEA/D	GrEA	HypE		MOEA/DD	NSGA-III	MOEA/D	GrEA	HypE
DTLZ 1	3	**0.973597**	0.973519	0.973541	0.967404	0.000000	DTLZ 3	**0.926617**	0.926480	0.926598	0.924652	0.000000
		0.973510	0.973217	0.973380	0.964059	0.000000		**0.926346**	0.925805	0.925855	0.922650	0.000000
		0.973278	0.971931	0.972484	0.828008	0.000000		**0.924901**	0.924234	0.923858	0.621155	0.000000
	5	**0.998980**	0.998971	0.998978	0.991451	0.000000		**0.990558**	0.990453	0.990543	0.963021	0.000000
		0.998975	0.998963	0.998969	0.844529	0.000000		**0.990515**	0.990344	0.990444	0.808084	0.000000
		0.998968	0.998673	0.998954	0.500179	0.000000		**0.990349**	0.989510	0.990258	0.499908	0.000000
	8	0.999949	**0.999975**	0.999943	0.999144	0.000000		**0.999343**	0.999300	0.999328	0.953478	0.000000
		0.999919	0.993549	0.999866	0.997992	0.000000		**0.999311**	0.924059	0.999303	0.791184	0.000000
		0.999887	0.966432	0.999549	0.902697	0.000000		**0.999248**	0.904182	0.508355	0.498580	0.000000
	10	**0.999994**	0.999991	0.999983	0.999451	0.000000		**0.999923**	0.999921	0.999922	0.962168	0.000000
		0.999990	0.999985	0.999979	0.998587	0.000000		**0.999922**	0.999918	0.999920	0.735934	0.000000
		0.999974	0.999969	0.999956	0.532348	0.000000		**0.999921**	0.999910	0.999915	0.499676	0.000000
	15	**0.999882**	0.999731	0.999695	0.172492	0.000000		**0.999982**	0.999910	0.999918	0.000000	0.000000
		0.999797	0.999686	0.999542	0.000000	0.000000		**0.999951**	0.999793	0.999792	0.000000	0.000000
		0.999653	0.999574	0.999333	0.000000	0.000000		**0.999915**	0.999780	0.999628	0.000000	0.000000
DTLZ 2	3	**0.926674**	0.926626	0.926666	0.924246	0.925691	DTLZ 4	**0.926731**	0.926659	0.926729	0.924613	0.926351
		0.926653	0.926536	0.926639	0.923994	0.925650		**0.926729**	0.926705	0.926725	0.924094	0.926223
		0.926596	0.926395	**0.926613**	0.923675	0.925531		**0.926725**	0.799572	0.500000	0.500000	0.800459
	5	**0.990535**	0.990459	0.990529	0.990359	0.987889		0.990575	**0.991102**	0.990569	0.990514	0.988150
		0.990527	0.990400	0.990518	0.990214	0.987665		**0.990573**	0.990413	0.990568	0.990409	0.988009
		0.990512	0.990328	0.990511	0.990064	0.987545		**0.990570**	0.990156	0.973811	0.990221	0.987743
	8	0.999346	0.999320	0.999341	**0.999991**	0.997401		**0.999364**	0.999363	0.999363	0.999102	0.997994
		0.999337	0.978936	0.999329	**0.999670**	0.996551		**0.999363**	0.999361	0.998497	0.999039	0.997730
		0.999329	0.919680	0.999307	0.989264	0.995761		**0.998360**	0.994784	0.995753	0.998955	0.997569
	10	**0.999952**	0.999918	0.999922	0.997636	0.998995		**0.999921**	0.999915	0.999918	0.999653	0.999019
		0.999932	0.999916	0.999921	0.996428	0.998934		**0.999920**	0.999910	0.999907	0.999608	0.998934
		0.999921	0.999915	0.999919	0.994729	0.998913		**0.999917**	0.999827	0.999472	0.999547	0.998921
	15	**0.999976**	0.999975	0.999967	0.999524	0.999936		**0.999915**	0.999910	0.999813	0.999561	0.999121
		0.999954	0.999939	0.999951	0.999496	0.996985		**0.999762**	0.999581	0.546405	0.999539	0.999433
		0.999915	0.999887	0.999913	0.998431	0.000000		**0.999680**	0.617313	0.502115	0.999521	0.999110

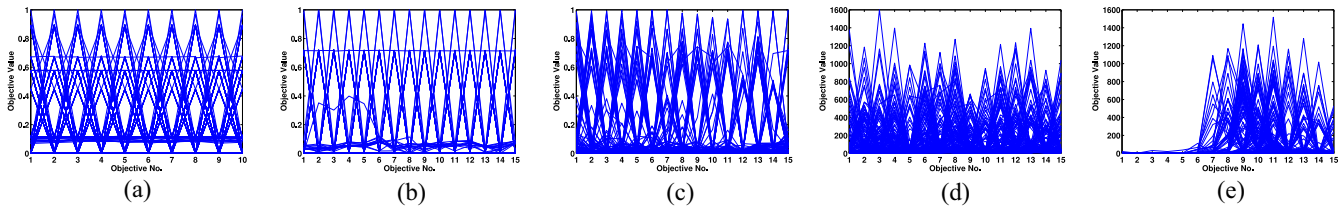

Fig. 11. Parallel coordinates of nondominated fronts obtained by five algorithms on the 15-objective DTLZ3 instance. (a) MOEA/DD. (b) NSGA-III. (c) MOEA/D. (d) GrEA. (e) HypE.

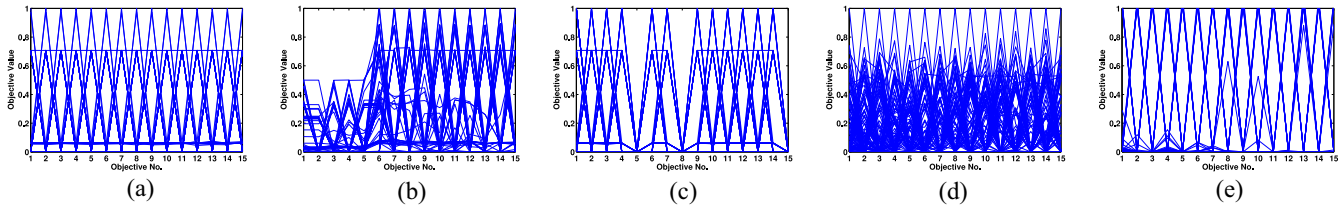

Fig. 12. Parallel coordinates of nondominated fronts obtained by five algorithms on the 15-objective DTLZ4 instance. (a) MOEA/DD. (b) NSGA-III. (c) MOEA/D. (d) GrEA. (e) HypE.

convex and concave). From the empirical results shown in Table VIII, it is clear that MOEA/DD is the best optimizer, where it achieves the best HV values in all three- to ten-objective test instances. MOEA/D performs slightly worse than MOEA/DD, where the differences in their HV values are not significant. In contrast, the performance of GrEA and HypE are not very promising. It is worth noting that HypE, whose performance is not satisfied in DTLZ

TABLE VIII
BEST, MEDIAN, AND WORST HV VALUES OBTAINED BY MOEA/DD AND THE OTHER ALGORITHMS ON WFG1 TO WFG9 INSTANCES WITH DIFFERENT NUMBER OF OBJECTIVES. BEST PERFORMANCE IS HIGHLIGHTED IN BOLD FACE WITH GRAY BACKGROUND

	m	MOEA/DD	MOEA/D	GrEA	HypE		MOEA/DD	MOEA/D	GrEA	HypE		MOEA/DD	MOEA/D	GrEA	HypE
WFG1	3	**0.937694**	0.932609	0.794748	0.844491	WFG2	**0.958287**	0.951685	0.950084	0.880921	WFG3	**0.703664**	0.697968	0.699502	0.688992
		0.933402	0.929839	0.692567	0.796960		**0.952467**	0.803246	0.942908	0.823928		**0.702964**	0.692355	0.672221	0.681211
		0.899253	0.815356	0.627963	0.726153		**0.803397**	0.796567	0.800186	0.792981		**0.701624**	0.679281	0.662046	0.653920
	5	**0.963464**	0.918652	0.876644	0.881727		**0.986572**	0.982796	0.980806	0.964351		0.673031	0.669009	**0.695221**	0.666272
		0.960897	0.915737	0.831814	0.830198		**0.985129**	0.978832	0.976837	0.951230		0.668938	0.662925	**0.684583**	0.662198
		0.959840	0.912213	0.790367	0.811234		**0.980035**	0.807951	0.808125	0.808821		0.662951	0.654729	**0.671553**	0.661010
	8	**0.922284**	0.918252	0.811760	0.782011		**0.981673**	0.963691	0.980012	0.961271		0.598892	0.529698	**0.657744**	0.505013
		0.913024	0.911586	0.681959	0.745122		**0.967265**	0.800333	0.840293	0.790291		0.565609	0.457703	**0.649020**	0.432217
		0.877784	0.808931	0.616006	0.733212		**0.789739**	0.787271	0.778921	0.771029		0.556725	0.439274	**0.638147**	0.301029
	10	**0.926815**	0.922484	0.866298	0.859382		**0.968201**	0.962841	0.964235	0.959821		**0.552713**	0.382068	0.542352	0.364398
		0.919789	0.915715	0.832016	0.819201		**0.965345**	0.957434	0.959740	0.950213		**0.532897**	0.337978	0.513261	0.353298
		0.865689	0.813928	0.757841	0.733623		**0.961400**	0.773474	0.956533	0.876231		**0.504943**	0.262496	0.501210	0.310003
WFG4	3	**0.727060**	0.724682	0.723403	0.720091	WFG5	**0.693665**	0.693135	0.689784	0.683583	WFG6	**0.708910**	0.712840	0.699876	0.690932
		0.726927	0.723945	0.722997	0.719872		**0.693544**	0.687378	0.689177	0.681014		**0.699663**	0.695081	0.693984	0.688372
		0.726700	0.723219	0.722629	0.717377		**0.691173**	0.681305	0.688885	0.677899		**0.689125**	0.684334	0.685599	0.680100
	5	0.876181	0.870868	**0.881161**	0.857632		0.833159	0.829696	**0.836232**	0.820087		0.850531	0.846015	**0.855839**	0.837264
		0.875836	0.862132	**0.879484**	0.850393		0.832710	0.826739	**0.834726**	0.814039		0.838329	0.813844	**0.847137**	0.812938
		0.875517	0.844219	**0.877642**	0.845329		0.830367	0.812225	**0.832212**	0.810098		0.828315	0.754054	**0.840637**	0.803927
	8	**0.920869**	0.784340	0.787287	0.702938		**0.852838**	0.779091	0.838183	0.750398		0.876310	0.692409	**0.912095**	0.683927
		0.910146	0.737386	0.784141	0.659238		**0.846736**	0.753486	0.641973	0.608898		0.863087	0.661156	**0.902638**	0.672932
		0.902710	0.718648	0.679178	0.603272		**0.830338**	0.705938	0.571933	0.430292		0.844535	0.567108	**0.885712**	0.620389
	10	**0.913018**	0.747485	0.896261	0.632747		**0.848321**	0.730990	0.791725	0.719273		0.884394	0.643198	**0.943454**	0.502832
		0.907040	0.712680	0.843257	0.609283		**0.841118**	0.715161	0.725198	0.658736		0.859986	0.582342	**0.927443**	0.459384
		0.888885	0.649713	0.840257	0.537262		**0.829547**	0.673789	0.685882	0.643392		0.832299	0.409210	**0.884145**	0.400239
WFG7	3	**0.727069**	0.725252	0.723229	0.715927	WFG8	**0.672022**	0.671355	0.671845	0.670937	WFG9	**0.707269**	0.688940	0.702489	0.648271
		0.727012	0.724517	0.722843	0.710852		**0.670558**	0.669927	0.669762	0.660923		**0.687401**	0.681725	0.638103	0.629238
		0.726907	0.723449	0.722524	0.708572		**0.668593**	0.664120	0.667948	0.657362		**0.638194**	0.636355	0.636575	0.612387
	5	0.876409	0.859727	**0.884178**	0.863746		**0.818663**	0.808204	0.797496	0.702847		**0.834616**	0.798069	0.823916	0.769394
		0.876297	0.843424	**0.883079**	0.819573		**0.795215**	0.793773	0.792692	0.683752		**0.797185**	0.789998	0.753683	0.749237
		0.874909	0.811292	**0.881305**	0.808876		**0.792900**	0.771763	0.790693	0.582735		**0.764723**	0.727728	0.747315	0.710284
	8	**0.920763**	0.729953	0.918742	0.623850		**0.876929**	0.537772	0.803050	0.350457		0.772671	0.633476	**0.842953**	0.610239
		0.917584	0.708701	0.910023	0.610294		**0.845975**	0.446544	0.799986	0.305873		0.759369	0.604016	**0.831775**	0.589348
		0.906219	0.605900	0.901292	0.592752		**0.730348**	0.347990	0.775434	0.294586		0.689923	0.548119	**0.765730**	0.510284
	10	0.927666	0.706473	**0.937582**	0.629381		**0.896317**	0.508625	0.841704	0.398275		0.717168	0.572925	**0.860676**	0.559382
		0.923441	0.625828	0.902343	0.592752		**0.844036**	0.350409	0.838256	0.298371		**0.717081**	0.546451	0.706632	0.549388
		0.917141	0.596189	0.901477	0.452975		0.715250	0.270931	**0.830394**	0.252731		**0.696061**	0.516309	0.686917	0.482750

test suite, shows relatively competitive performance in WFG1 instances.

The EF of WFG2 is composed of several disconnected convex segments and its variables are nonseparable. MOEA/DD still obtains the best HV values in all three- to ten-objective WFG2 instances. The performance of MOEA/D and GrEA are similar in most cases, where the prior one only wins in the five-objective case.

WFG3 is the connected version of WFG2, where its characteristic is a linear and degenerate EF shape. The performance of these four algorithms are similar in the three- and five-objective cases. MOEA/DD shows the best HV values in the three- and ten-objective cases, while GrEA wins in the five- and eight-objective cases. When the number of objectives becomes large (i.e., $m = 8$ and $m = 10$), the performance of MOEA/D and HypE are much worse than that of MOEA/DD and GrEA.

Although WFG4 to WFG9 share the same EF shape in the objective space, which is a part of a hyper-ellipse with radii $r_i = 2i$, where $i \in \{1, \ldots, m\}$, their characteristics in the decision space are different. Specifically, WFG4 is featured by its multimodality with large "hill sizes." This characteristic can easily cause an algorithm to be trapped in local optima. MOEA/DD shows the best performance in most cases, except the five-objective WFG4 instances. In particular, its superiority becomes more evident when the number of objectives becomes large. Similar to the observations in WFG4 instances, for WFG5, a deceptive problem, the performance of MOEA/D, GrEA and HypE become much worse than MOEA/DD in eight- and ten-objective cases. WFG6 is a nonseparable and reduced problem. GrEA is the best optimizer this time, where it is only outperformed by MOEA/DD in the three-objective case. WFG7 is a separable and uni-modal problem, but with parameter dependency. The performance of four algorithms are not very significantly different in the three- and five-objective cases. However, the superiorities of MOEA/DD and GrEA are evident when the number of objectives becomes large. Both WFG8 and WFG9 have nonseparable property, but the parameter dependency caused by WFG8 is much harder than that in WFG9. MOEA/DD shows the best performance in WFG8, while GrEA only outperforms it in the eight-objective WFG9 instance. Similar to the observations before, the performance of MOEA/D and HypE are acceptable in three- and five-objective cases, but decline a lot in the higher-dimensional cases. From the empirical studies on WFG test suite, we conclude that the promising results obtained by MOEA/DD should be attributed to its advanced technique for balancing convergence and diversity.

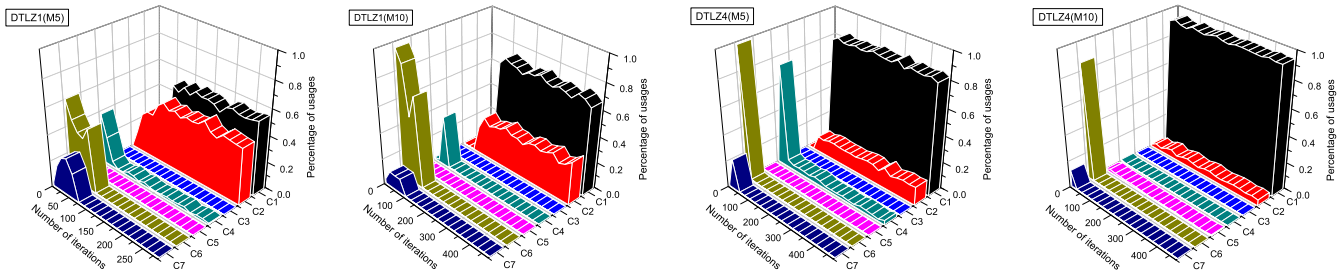

Fig. 13. Plots of average percentage of usages for each case on DTLZ1 and DTLZ4 with five and ten objectives, respectively.

TABLE IX
DIFFERENT CASES IN UPDATE PROCEDURE

Index	Description
Case1 (C1)	$l = 1$
Case2 (C2)	$\|F_l\| = 1$, $\mathbf{x}^c \in F_l$ and $\|\Phi^c\| > 1$
Case3 (C3)	$\|F_l\| = 1$, $\mathbf{x}^c \in F_l$ and $\|\Phi^c\| = 1$
Case4 (C4)	$\|F_l\| = 1$, $\mathbf{x}^l \in F_l$, $\mathbf{x}^l \neq \mathbf{x}^c$ and $\|\Phi^l\| > 1$
Case5 (C5)	$\|F_l\| = 1$, $\mathbf{x}^l \in F_l$, $\mathbf{x}^l \neq \mathbf{x}^c$ and $\|\Phi^l\| = 1$
Case6 (C6)	$\|F_l\| > 1$ and $\|\Phi^h\| = 1$
Case7 (C7)	$\|F_l\| > 1$ and $\|\Phi^h\| > 1$

$\mathbf{x}^c$: the offspring solution
$\mathbf{x}^l$: solution in the last non-domination level
Φ^c: the subregion associated with $\mathbf{x}^c$
Φ^l: the subregion associated with $\mathbf{x}^l$
Φ^h: the subregion with the largest niche count

C. Investigations of Different Scenarios in Update Procedure

As discussed in Section III-D, the update procedure of MOEA/DD is implemented in a hierarchical manner. It carefully considers all possible cases (Table IX presents a brief description for each of them) that might happen when introducing an offspring solution into the population. In this section, without loss of generality, we empirically investigate the usages of different cases on four selected test instances (DTLZ1 and DTLZ4 with five and ten objectives). The parameters are set the same as suggested in Section IV-D, and 20 independent runs have been conducted for each test instance. The average percentage of usages for each case on each test instance is plotted in Fig. 13, separately. From these empirical results, we have the following three observations.

1) In all test instances, the average percentage of usages of C1 is the largest among all seven cases, and it grows even larger with the increase of number of objectives (the average percentage of usages of C1 in ten-objective instance is larger than that in five-objective instance). This can be explained by the fact that almost all solutions in the population become nondominated when the number of objectives increases [13].
2) The second frequent case is C2, where the offspring solution is discarded immediately. This implies that the reproduction operators, i.e., SBX and polynomial mutation, used in MOEA/DD have some difficulties in reproducing competitive offspring solutions when handling problems with a large number of objectives.
3) Both C3 and C5 have hardly been utilized during the whole evolutionary process. C4, C6, and C7 have been utilized in the early stage of evolution, and then their percentages of usages drop down to zero later on.

Algorithm 6: Variant of Update Procedure

Input: parent population P, offspring solution $\mathbf{x}^c$
Output: parent population P

```
Find the subregion associated with x^c according to (6);
P' ← P ⋃ {x^c};
Update the nondomination level structure of P' according to the method suggested in [66];
if l = 1 then
    x' ← LOCATE_WORST(P');
    P ← P' \ {x'};
else
    if |F_l| = 1 then
        P ← P' \ {x^l};
    else
        Identify the most crowded subregion Φ^h associated with those solutions in F_l;
        Find the worst solution x' = argmax_{x∈Φ^h} g^pbi(x|w^h, z*);
        P ← P' \ {x'};
    end
end
Update the nondomination level structure of P according to the method suggested in [66];
return P
```

D. Further Investigations of Update Procedure

As discussed in Section III-D, to preserve the population diversity, we give the worst solution (denoted as $\mathbf{x}'$) in the last nondomination level a second chance for survival, in case it is associated with an isolated subregion. In order to validate the importance of this idea, we design a counter example, which is in compliance with the principle of Pareto dominance to eliminate $\mathbf{x}'$ immediately, for comparison. The pseudo-code of this update procedure variant is given in Algorithm 6.

We use Algorithm 6 to replace line 7 in Algorithm 1, and denote the resulted algorithm as v-MOEA/DD. The performance of v-MOEA/DD is compared with MOEA/DD on DTLZ1 to DTLZ4 test instances with 3–15 objectives. The parameters are set the same as Section IV-D, and 20 independent runs have been conducted for each test instance. From the empirical results shown in Table X, we observe that the performance of MOEA/DD is better than v-MOEA/DD in most cases. It is worth noting that this update procedure variant also has a diversity preservation mechanism, i.e.,

TABLE X
BEST, MEDIAN, AND WORST IGD VALUES OBTAINED BY MOEA/DD AND v-MOEA/DD ON DTLZ1, DTLZ2, DTLZ3, AND DTLZ4 INSTANCES WITH DIFFERENT NUMBER OF OBJECTIVES. BEST PERFORMANCE IS HIGHLIGHTED IN BOLD FACE WITH GRAY BACKGROUND

m	DTLZ1		DTLZ2		DTLZ3		DTLZ4	
	MOEA/DD	v-MOEA/DD	MOEA/DD	v-MOEA/DD	MOEA/DD	v-MOEA/DD	MOEA/DD	v-MOEA/DD
3	**3.197E-4**	6.045E-4	6.666E-4	**6.195E-4**	**5.690E-4**	7.382E-4	**1.025E-4**	1.156E-4
	6.516E-4	1.396E-3	**8.073E-4**	8.319E-4	**1.892E-3**	3.458E-3	**1.429E-4**	1.535E-4
	1.515E-3	4.845E-3	**1.243E-3**	1.821E-3	**6.231E-3**	7.481E-3	**1.881E-4**	5.321E-1
5	2.632E-4	**2.343E-4**	1.128E-3	**1.111E-3**	**6.181E-4**	6.685E-4	**1.097E-4**	1.287E-4
	3.115E-4	3.248E-4	1.291E-3	**1.289E-3**	**1.181E-3**	1.862E-3	**1.296E-4**	1.371E-4
	5.182E-4	1.133E-3	**1.424E-3**	1.453E-3	4.736E-3	**4.311E-3**	**1.532E-4**	3.519E-1
8	**1.808E-3**	2.492E-3	2.880E-3	**2.796E-3**	3.411E-3	**2.945E-3**	**5.271E-4**	5.625E-4
	3.007E-3	3.913E-3	**3.291E-3**	3.503E-3	**8.079E-3**	8.808E-3	**6.699E-4**	7.200E-4
	4.310E-3	4.955E-3	4.106E-3	**3.920E-3**	1.826E-2	**1.233E-2**	**9.107E-4**	2.303E-1
10	1.834E-3	**1.805E-3**	**3.223E-3**	3.287E-3	**1.689E-3**	2.794E-3	**1.291E-3**	1.481E-3
	2.472E-3	2.737E-2	**3.752E-3**	3.990E-3	**2.164E-3**	3.577E-3	**1.615E-3**	1.688E-3
	4.026E-3	4.321E-3	**4.145E-3**	4.446E-3	**3.226E-3**	5.084E-3	**1.931E-3**	1.973E-3
15	**2.356E-3**	3.171E-3	**4.577E-3**	4.815E-3	5.716E-3	**5.715E-3**	**1.474E-3**	1.650E-3
	4.703E-3	4.911E-3	**5.863E-3**	6.652E-3	**7.461E-3**	8.499E-3	**1.881E-3**	2.228E-3
	7.933E-3	9.340E-3	**6.929E-3**	8.365E-3	**1.138E-2**	1.726E-2	**3.159E-3**	1.062E-1

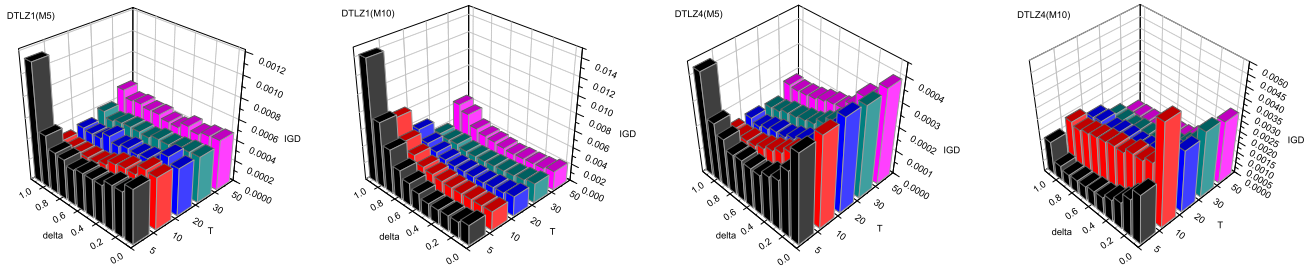

Fig. 14. Median IGD values found by MOEA/DD with 50 different combinations of T and δ on DTLZ1 and DTLZ4 with five and ten objectives, respectively.

LOCATE_WORST(P'), which removes the worst solution in the most crowded subregion. However, the original update procedure introduced in Algorithm 4 has a stronger intention for diversity preservation, which even violates the principle of Pareto dominance. According to the investigations in Section V-C, C3 to C7 have not been frequently utilized during the evolutionary process, this explains the observation that the better IGD values obtained by MOEA/DD are not very significant. Nevertheless, the occasional utilizations of C3 and C7 do contribute a lot to rescue the population diversity, especially for DTLZ4 instances with a biased density.

E. Parameter Sensitivity Studies

There are two major parameters in MOEA/DD: the neighborhood size T and the probability δ of selecting mating parents from neighboring subregions. To study how MOEA/DD is sensitive to these two parameters, we have tried to cover a wide range of values for each parameter. Five values are considered for T: 5, 10, 20, 30, and 50; 11 values are considered for δ, ranging from 0.0 to 1.0 with a step size 0.1. We have taken DTLZ1 and DTLZ4 instances with five and ten objectives, respectively, to compare the performance of all 55 different parameter configurations. Twenty independent runs have been conducted for each configuration on each test instance.

Fig. 14 shows the median IGD values obtained by these 55 different configurations on each selected test instance. From these four plots, one can observe that different configurations can lead to different performances on distinct test instances. Specifically, the mating restriction suggested in Section III-C is blocked when $\delta = 0.0$, which means that all mating parents are randomly chosen from the entire population. We find that this configuration is not a good choice, especially for DTLZ4 instance. This phenomenon is also reflected by the unsatisfied results obtained when setting a large neighborhood size T, say $T = 50$. Moreover, the parameter combination ($T = 5, \delta = 1.0$) is always the worst in all selected test instances. This is because too small T and too large δ settings make the reproduction operation excessively exploit local areas, thus it might lose some important information about the population. In general, it is better to choose T between 10 and 20, and δ between 0.6 and 0.9.

VI. HANDLING CONSTRAINTS

After demonstrating the superiority of MOEA/DD for solving unconstrained (i.e., with box constraints alone) many-objective optimization problems, this section extends MOEA/DD (denoted as C-MOEA/DD) to solve constrained many-objective optimization problems.

In case of the presence of infeasible solutions, to give more emphasis on feasible solutions than the infeasible ones, some modifications are suggested to the update and reproduction procedures of MOEA/DD, while the other parts keep

Algorithm 7: Update Procedure With Constraint Handling

Input: parent population P, offspring solution $\mathbf{x}^c$
Output: parent population P

1 Find the subregion associated with $\mathbf{x}^c$ according to (6);
2 $P' \leftarrow P \bigcup \{\mathbf{x}^c\}$;
3 $S \leftarrow \emptyset$; // infeasible solutions archive
4 **foreach** $\mathbf{x} \in P'$ **do**
5 **if** $\mathrm{CV}(\mathbf{x}) > 0$ **then**
6 $S \leftarrow S \bigcup \{\mathbf{x}\}$;
7 **end**
8 **end**
9 **if** $|S| = 0$ **then**
10 UPDATE_POPULATION($P, \mathbf{x}^c$);
11 **else**
12 Sort S in descending order according to CV;
13 $flag \leftarrow 0$;
14 **for** $i \leftarrow 1$ **to** $|S|$ **do**
15 **if** *$S(i)$'s subregion is not isolated* **then**
16 $flag \leftarrow 1$, $\mathbf{x}' \leftarrow S(i)$;
17 **break**;
18 **end**
19 **end**
20 **if** $flag = 0$ **then**
21 $\mathbf{x}' \leftarrow S(1)$;
22 **end**
23 $P \leftarrow P' \setminus \{\mathbf{x}'\}$;
24 **end**
25 **return** P

Algorithm 8: Binary Tournament Selection Procedure

Input: candidate solutions $\mathbf{x}^1$ and $\mathbf{x}^2$
Output: mating parent $\mathbf{p}$

1 **if** $\mathrm{CV}(\mathbf{x}^1) < \mathrm{CV}(\mathbf{x}^2)$ **then**
2 $\mathbf{p} \leftarrow \mathbf{x}^1$;
3 **else if** $\mathrm{CV}(\mathbf{x}^1) > \mathrm{CV}(\mathbf{x}^2)$ **then**
4 $\mathbf{p} \leftarrow \mathbf{x}^2$;
5 **else**
6 $\mathbf{p} \leftarrow$ RANDOM_PICK($\mathbf{x}^1, \mathbf{x}^2$);
7 **end**
8 **return** $\mathbf{p}$

untouched. It is worth noting that these modifications do not introduce any further parameters to MOEA/DD. Moreover, if all population members are feasible or an unconstrained problem is supplied, the modified procedures reduce to their unconstrained versions. In the following paragraphs, we illustrate these modifications one by one.

A. Modifications on the Update Procedure

As suggested in [71], the constraint violation value of a solution $\mathbf{x}$, denoted as $\mathrm{CV}(\mathbf{x})$, is calculated by the following form:

$$\mathrm{CV}(\mathbf{x}) = \sum_{j=1}^{J} \langle g_j(\mathbf{x}) \rangle + \sum_{k=1}^{K} |h_k(\mathbf{x})| \tag{19}$$

where the bracket operator $\langle \alpha \rangle$ returns the absolute value of α if $\alpha < 0$, and returns 0 otherwise. It is obvious that the smaller is the $\mathrm{CV}(\mathbf{x})$, the better is the quality of $\mathbf{x}$, and the CV of a feasible solution is always 0.

The pseudo-code of this modified update procedure is given in Algorithm 7. First and foremost, we identify the subregion associated with $\mathbf{x}^c$ and combine $\mathbf{x}^c$ with the parent population P to form a hybrid population P' (lines 1 and 2 of Algorithm 7). If every member in P' is feasible, we use the unconstrained update procedure, introduced in Section III-D, to update P (line 10 of Algorithm 7). Otherwise, the feasible solutions will survive to the next round without reservation, while the survival of infeasible ones depends on both CVs and niching scenarios. As discussed in Section III-D, the solution, associated with an isolated subregion, is important for population diversity. And such solution will survive in the update procedure without reservation, even if it is inferior in terms of convergence (i.e., it belongs to the current worst nondomination level). Inheriting this idea, we give the infeasible solution, associated with an isolated subregion, a second chance to survive. Specifically, we will at first identify the solution in P' that has the largest CV. If this solution is not associated with an isolated subregion, it will be treated as the current worst solution (denoted as $\mathbf{x}'$); otherwise, for the sake of population diversity, it will be preserved for later consideration. Instead, we will find the solution in P' that has the second largest CV. Depending on whether this solution is associated with an isolated subregion or not, we decide whether it is treated as $\mathbf{x}'$ or not, so on and so forth (lines 12–19 of Algorithm 7). It is worth noting that if every infeasible solution in P' is associated with an isolated subregion, the one with the largest CV will be treated as $\mathbf{x}'$ (lines 20–22 of Algorithm 7). At last, $\mathbf{x}'$ is eliminated from P' (line 23 of Algorithm 7).

B. Modifications on the Reproduction Procedure

In order to handle the third challenge posed in Section I, the selection of mating parents is restricted to some neighboring subregions as in Section III-C. However, due to the presence of infeasible solutions, there is an additional requirement for emphasizing a feasible solution over an infeasible solution and small CV solution over a large CV solution.

For this purpose, we randomly select two solutions, denoted as $\mathbf{x}^1$ and $\mathbf{x}^2$, from some specified neighboring subregions. And the better one is chosen as the mating parent. In particular, since the CV of a feasible solution is always 0, we just choose the one with a smaller CV. If both $\mathbf{x}^1$ and $\mathbf{x}^2$ have the same CV, we choose one at random. The pseudo-code of this binary tournament selection procedure is given in Algorithm 8. In order to select other mating parents, similarly, another pair of solutions are randomly chosen from some neighboring subregions and the above tournament selection is applied to choose the better one as the second mating parent, so on and so forth. Afterwards, variation operation is applied on the selected mating parents to reproduce new offspring solutions.

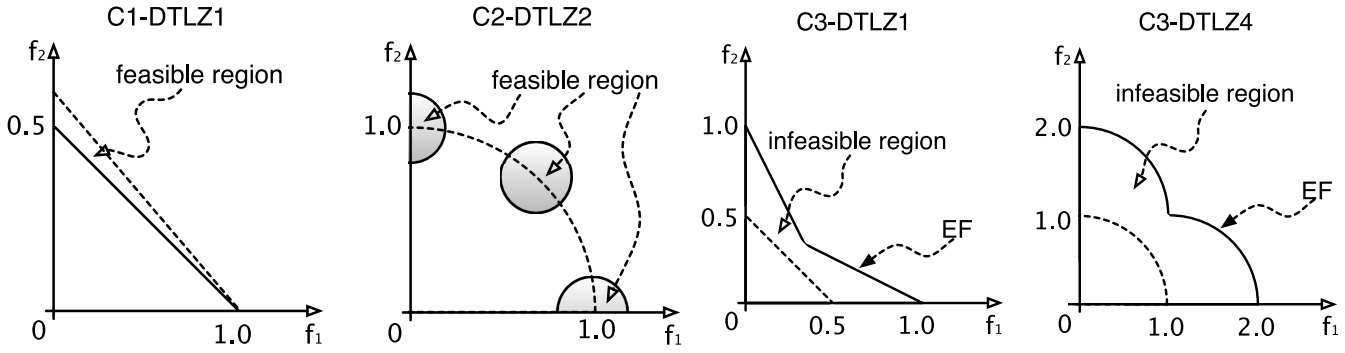

Fig. 15. Illustration of Constraint Surfaces of C1-DTLZ1, C2-DTLZ2, C3-DTLZ1, and C3-DTLZ4 in 2-D Case.

C. Experimental Setup

This section presents the experimental design for the performance investigations of C-MOEA/DD on many-objective optimization problems with various constraints. At first, we introduce the constrained test instances used in our empirical studies. Then, we describe the performance metrics, after which we briefly introduce the EMO algorithms used for comparisons. Finally, we provide the general parameter settings.

1) Constrained Test Instances: Four constrained test instances suggested in [71] (C1-DTLZ1, C2-DTLZ2, C3-DTLZ1, and C3-DTLZ4, where Ci, $i \in \{1, 2, 3\}$, indicates the Type-i constraints) are chosen for our empirical studies. For each test instance, the objective functions, the number of objectives and decision variables are set the same as its unconstrained version. In the following paragraph, we briefly introduce the constraints incurred by each test instance.

1) *C1-DTLZ1:* This is a Type-1 constrained problem, where the original EF is still optimal, but the feasible search space is compressed to a small part that is close to the EF. The constraint is formulated as

$$c(\mathbf{x}) = 1 - \frac{f_m(\mathbf{x})}{0.6} - \sum_{i=1}^{m-1} \frac{f_i(\mathbf{x})}{0.5} \geq 0. \tag{20}$$

2) *C2-DTLZ2:* This is a Type-2 constrained problem, where only the region that lies inside each of the $m+1$ hyperspheres of radius r is feasible. It is with the following constraint:

$$c(\mathbf{x}) = -\min\left\{\min_{i=1}^{m}\left[(f_i(\mathbf{x})-1)^2 + \sum_{j=1,j\neq i}^{m} f_j^2(\mathbf{x}) - r^2\right], \times\left[\sum_{i=1}^{m}\left(f_i(\mathbf{x}) - 1/\sqrt{m}\right)^2 - r^2\right]\right\} \geq 0 \tag{21}$$

where $r = 0.4$, for $m = 3$ and 0.5, otherwise.

3) *C3-DTLZ1 and C3-DTLZ4:* They are two Type-3 constrained problems, which involve multiple constraints. The EF of the original unconstrained problem is not optimal any longer, rather portions of the added constraint surfaces constitute the EF. C3-DTLZ1 instance is modified from DTLZ1 by adding the following m linear constraints:

$$c_i(\mathbf{x}) = \sum_{j=1,j\neq i}^{m} f_j(\mathbf{x}) + \frac{f_i(\mathbf{x})}{0.5} - 1 \geq 0 \tag{22}$$

where $i \in \{1, \ldots, m\}$. Similarly, C3-DTLZ4 is modified by adding the following m quadratic constraints:

$$c_i(\mathbf{x}) = \frac{f_i^2(\mathbf{x})}{4} + \sum_{j=1,j\neq i}^{m} f_j^2(\mathbf{x}) - 1 \geq 0 \tag{23}$$

where $i \in \{1, \ldots, m\}$.

Fig. 15 illustrates the constraint surfaces of C1-DTLZ1, C2-DTLZ2, C3-DTLZ1, and C3-DTLZ4 in a 2-D case.

2) Performance Metric: We still choose IGD as the performance metric to evaluate the quality of the obtained solution set. However, due to the introduction of particular constraints, the targeted Pareto-optimal points are different from the corresponding unconstrained version for C2-DTLZ2, C3-DTLZ1, and C3-DTLZ4. Specifically, as for C2-DTLZ2, we at first sample a set of Pareto-optimal points based on the method introduced in Section IV-B for DTLZ2. Then, only those satisfying the constraint introduced by (21) are kept to form P^*.

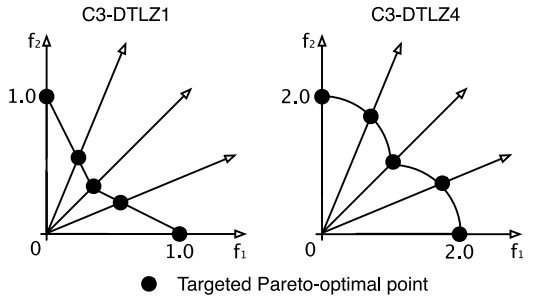

Fig. 16. Illustration of finding five targeted Pareto-optimal points on the EFs of C3-DTLZ1 and C3-DTLZ4.

TABLE XI
NUMBER OF GENERATIONS FOR DIFFERENT TEST INSTANCES

Test instance	$m = 3$	$m = 5$	$m = 8$	$m = 10$	$m = 15$
C1-DTLZ1	500	600	800	1,000	1,500
C2-DTLZ2	250	350	500	750	1,000
C3-DTLZ1	750	1,250	2,000	3,000	4,000
C3-DTLZ4	750	1,250	2,000	3,000	4,000

Algorithm 9: Variant of Update Procedure With Constraint Handling

Input: parent population P, offspring solution $\mathbf{x}^c$
Output: parent population P

Find the subregion associated with $\mathbf{x}^c$ according to (6);
$P' \leftarrow P \bigcup \{\mathbf{x}^c\}$;
$S \leftarrow \emptyset$; // infeasible solutions archive
foreach $\mathbf{x} \in P'$ **do**
  if $\mathrm{CV}(\mathbf{x}) > 0$ **then**
    $S \leftarrow S \bigcup \{\mathbf{x}\}$;
  end
end
if $|S| = 0$ **then**
  UPDATE_POPULATION($P, \mathbf{x}^c$);
else
  Sort S in descending order according to CV;
  $\mathbf{x}' \leftarrow S(1)$;
  $P \leftarrow P' \setminus \{\mathbf{x}'\}$;
end
return P

As for C3-DTLZ1 and C3-DTLZ4, we use the similar routines described in Section IV-B to locate the intersecting points of weight vectors and constraint surfaces. Specifically, for C3-DTLZ1 or C3-DTLZ4, a solution $\mathbf{x}^*$, whose objective vector lies on its corresponding constraint surface, should make (22) or (23) equal to zero. Considering the jth constraint, $j \in \{1, \ldots, m\}$, for a weight vector $\mathbf{w} = (w_1, \ldots, w_m)^T$, $\mathbf{x}^*$ satisfies

$$\frac{f_i(\mathbf{x}^*)}{w_i} = t_j \tag{24}$$

where $i \in \{1, \ldots, m\}$ and $t_j > 0$. Put (24) into (22), we have

$$t_j = \frac{1}{2 \times w_j + \sum_{i=1, i \neq j}^{m} w_i}. \tag{25}$$

Similarly, put (24) into (23), we have

$$t_j = \frac{1}{\sqrt{\sum_{i=1, i \neq j}^{m} w_i^2 + w_j^2/4}}. \tag{26}$$

Finally, the targeted Pareto-optimal point of $\mathbf{x}^*$ on the constraint surface, denoted as $\mathbf{F}(\mathbf{x}^*) = (f_1(\mathbf{x}^*), \ldots, f_m(\mathbf{x}^*))^T$, can be calculated as

$$f_i(\mathbf{x}^*) = \max_{j \in \{1, \ldots, m\}} w_i \times t_j \tag{27}$$

where $i \in \{1, \ldots, m\}$. Fig. 16 presents an intuitive illustration of finding five targeted Pareto-optimal points on the EFs of C3-DTLZ1 and C3-DTLZ4, respectively, in a 2-D space.

3) EMO Algorithms in Comparisons: We consider the two constrained optimizers suggested in [71] for comparisons.

1) *C-NSGA-III:* The differences between C-NSGA-III and NSGA-III lie in two parts: one is the elitist selection operator, where a constraint-domination principle is adopted; and the other is the mating selection, where a binary tournament selection procedure is applied to emphasize feasible solutions and infeasible solutions with small CVs.

TABLE XII
BEST, MEDIAN, AND WORST IGD VALUES OBTAINED BY C-MOEA/DD, C-NSGA-III, AND C-MOEA/D ON C1-DTLZ1, C2-DTLZ2, C3-DTLZ1, AND C3-DTLZ4 INSTANCES WITH DIFFERENT NUMBER OF OBJECTIVES. BEST PERFORMANCE IS HIGHLIGHTED IN BOLD FACE WITH GRAY BACKGROUND

problem	m	C-MOEA/DD	C-NSGA-III	C-MOEA/D
C1-DTLZ1	3	**4.239E-4**	1.229E-3	9.624E-4
		1.182E-3	4.932E-3	6.360E-3
		5.263E-3	2.256E-2	2.276E-2
	5	**6.091E-4**	2.380E-3	1.998E-3
		1.939E-3	4.347E-3	3.960E-3
		9.282E-3	1.024E-2	9.597E-3
	8	**2.952E-3**	4.843E-3	3.442E-3
		6.395E-3	1.361E-2	9.150E-3
		1.528E-2	4.140E-2	3.514E-2
	10	**2.564E-3**	3.042E-3	5.042E-3
		4.511E-3	6.358E-3	7.960E-3
		1.004E-2	2.762E-2	1.536E-2
	15	**4.731E-3**	4.994E-3	8.088E-3
		1.341E-2	**1.041E-2**	1.595E-2
		3.307E-2	2.930E-2	**2.893E-2**
C2-DTLZ2	3	6.623E-4	1.581E-3	**3.844E-4**
		8.708E-4	2.578E-3	**5.526E-4**
		6.688E-3	6.733E-3	7.014E-1
	5	1.179E-3	2.762E-3	**5.404E-4**
		1.533E-3	3.873E-3	**7.304E-4**
		1.975E-3	7.596E-3	3.343E-2
	8	**1.875E-3**	1.404E-2	2.926E-3
		3.258E-3	2.352E-2	4.975E-3
		8.997E-2	8.662E-1	1.131
	10	**9.388E-4**	1.978E-2	9.661E-4
		1.233E-3	2.694E-2	1.491E-3
		2.327E-3	3.491E-2	8.774E-1
	15	2.589E-1	3.117E-2	**1.668E-2**
		3.996E-1	3.544E-2	**2.129E-2**
		1.063	**9.343E-1**	1.204
C3-DLTZ1	3	**1.018E-3**	5.221E-3	1.461E-3
		2.944E-3	9.120E-3	4.323E-3
		1.815E-2	2.058E-2	3.284E-2
	5	**4.647E-4**	1.130E-2	5.482E-4
		1.847E-3	1.964E-2	1.115E-2
		2.163E-3	4.745E-2	1.713E-2
	8	**2.527E-3**	1.243E-2	5.878E-2
		6.907E-3	2.104E-2	7.817E-2
		7.166E-3	8.196E-2	1.159E-1
	10	**2.698E-3**	8.450E-3	6.053E-2
		4.646E-3	1.509E-2	8.968E-2
		4.841E-3	3.753E-2	1.104E-1
	15	**2.232E-3**	4.042E-3	2.222E-1
		2.473E-3	1.064E-2	3.769E-1
		2.743E-3	2.055E-1	4.091E-1
C3-DLTZ4	3	**3.944E-3**	1.862E-2	5.372E-3
		4.968E-3	2.456E-2	4.948E-1
		6.716E-3	5.586E-1	8.320E-1
	5	**6.071E-3**	3.247E-2	6.610E-3
		7.230E-3	3.854E-2	2.195E-1
		1.175E-2	3.466E-2	8.761E-1
	8	**1.116E-2**	5.558E-2	1.503E-1
		1.386E-2	2.646E-1	8.171E-1
		1.974E-2	8.886E-1	1.322
	10	**1.069E-2**	4.247E-2	6.414E-2
		1.269E-2	5.927E-2	4.450E-1
		1.379E-2	9.092E-1	1.234
	15	**2.212E-2**	1.134E-1	1.126
		2.602E-2	9.325E-1	1.454
		3.171E-2	1.424	1.645

TABLE XIII
BEST, MEDIAN, AND WORST IGD VALUES OBTAINED BY C-MOEA/DD AND *v*C-MOEA/DD ON C1-DTLZ1, C2-DTLZ2, C3-DTLZ1, AND C3-DTLZ4 INSTANCES WITH DIFFERENT NUMBER OF OBJECTIVES. BEST PERFORMANCE IS HIGHLIGHTED IN BOLD FACE WITH GRAY BACKGROUND

m	C1-DTLZ1		C2-DTLZ2		C3-DTLZ1		C3-DTLZ4	
	C-MOEA/DD	*v*C-MOEA/DD	C-MOEA/DD	*v*C-MOEA/DD	C-MOEA/DD	*v*C-MOEA/DD	C-MOEA/DD	*v*C-MOEA/DD
3	**4.239E-4**	5.209E-4	**6.623E-4**	6.987E-4	2.018E-3	**1.913E-3**	**3.944E-3**	4.346E-3
	1.181E-3	2.126E-3	**8.531E-4**	9.232E-4	2.899E-3	**2.801E-3**	**4.968E-3**	5.859E-3
	5.263E-3	9.327E-3	6.688E-3	**2.560E-3**	**1.815E-2**	4.818E-2	**6.716E-3**	9.470E-3
5	**6.091E-4**	1.009E-3	**1.179E-3**	1.214E-3	**6.647E-4**	7.289E-4	**6.071E-3**	6.955E-3
	1.869E-3	3.191E-3	**1.523E-3**	1.553E-3	**1.828E-3**	1.964E-3	**7.230E-3**	7.916E-3
	9.282E-3	6.098E-3	**1.975E-3**	2.172E-3	**2.163E-3**	2.242E-3	**1.175E-2**	1.605E-2
8	**2.952E-3**	3.868E-3	**1.875E-3**	1.942E-3	**2.527E-3**	3.322E-3	**1.116E-2**	1.427E-2
	6.044E-3	6.954E-3	**3.233E-3**	3.308E-3	**6.893E-3**	7.873E-3	**1.386E-2**	1.935E-2
	1.528E-2	1.875E-2	**8.997E-1**	9.026E-1	**7.166E-3**	8.018E-3	**1.974E-2**	2.645E-2
10	**2.564E-3**	3.079E-3	**9.388E-4**	9.550E-4	**2.698E-3**	3.181E-3	**1.069E-2**	1.229E-2
	4.427E-3	5.565E-3	**1.227E-3**	1.256E-3	**4.644E-3**	5.695E-3	**1.269E-2**	1.644E-2
	1.004E-2	1.374E-2	2.327E-3	**2.160E-3**	**4.841E-3**	4.936E-3	**1.379E-2**	1.945E-2
15	**4.731E-3**	9.268E-3	2.589E-1	**2.361E-1**	**2.232E-3**	2.871E-3	2.212E-2	**2.057E-2**
	1.323E-2	2.407E-2	3.861E-1	**3.791E-1**	**2.470E-3**	2.555E-3	2.602E-2	**2.571E-2**
	3.307E-2	4.427E-1	**1.063**	1.108	**2.743E-3**	2.881E-3	3.171E-2	**3.031E-2**

2) *C-MOEA/D:* This is an extension of the original MOEA/D introduced in Section IV-C. In order to give adequate emphasis for the feasible and small CV solutions, a modification has been made in the update procedure of MOEA/D.

4) General Parameter Settings: Almost all parameters are set the same as Section IV-D, except the number of generations which is listed in Table XI.

D. Performance Comparisons With C-NSGA-III and C-MOEA/D

According to the description in Section VI-C1, Type-1 constrained problems introduce difficulties for approximating the global EF. From the empirical results shown in Table XII, it is obvious that C-MOEA/DD performs significantly better than the other two algorithms in almost all C1-DTLZ1 instances, except the case with 15 objectives. The performance of C-NSGA-III and C-MOEA/D is similar in all C1-DTLZ1 instances, but the former one achieves the best medium IGD value for the 15-objective case.

By introducing infeasibility to several parts of the EF, Type-2 constrained problems are designed to test an algorithm's ability for dealing with disconnected EFs. From the IGD values shown in Table XII, we can see that the performance of these three algorithms are similar in most cases, except the 15-objective C2-DTLZ2 instance, where C-MOEA/DD performs significantly worse than the other two algorithms. Moreover, for all 3- to 15-objective C2-DTLZ2 instances, the worst IGD values obtained by C-MOEA/D are much larger than the other algorithms. This indicates that C-MOEA/D is not able to approximate the global EF in all 20 runs.

Rather than finding the original EF of the corresponding unconstrained problem, the optimal front of Type-3 constrained problem is a composite of several constraint surfaces. For C3-DTLZ1 problem, Table XII shows that C-MOEA/DD outperforms C-NSGA-III and C-MOEA/D in all 3- to 15-objective C3-DTLZ1 instances. The performance of C-NSGA-III and C-MOEA/D is similar in most cases, while the former one is slightly better when the number of objectives increases. As for C3-DTLZ4, an additional difficulty is introduced by its biased density of solutions. Similar to the observations for unconstrained DTLZ4, C-MOEA/D suffers from the loss of population diversity, and its obtained IGD values are the worst comparing to the other algorithms. However, it is interesting to note that the best IGD values achieved by C-MOEA/D for three- and five-objective C3-DTLZ4 instances are better than those of C-NSGA-III.

E. Further Investigations of the Modified Update Procedure

As discussed in Section VI-A, to preserve the population diversity, we use the similar idea for unconstrained problems to give the worst solution (denoted as $\mathbf{x}'$), which has the largest CV, a second chance for survival. Similar to the investigations conducted in Section V-D, we also design a counter example to validate this idea. The pseudo-code of this update procedure variant is given in Algorithm 9.

We use Algorithm 9 to replace the update procedure used in C-MOEA/DD, and denote the resulted algorithm as *v*C-MOEA/DD. The performance of *v*C-MOEA/DD is compared with C-MOEA/DD on the benchmark problems introduced in Section VI-C1. The parameters are set the same as Section VI-C4, and 20 independent runs have been conducted for each test instance. Empirical results are presented in Table XIII, from which we find that the performance of C-MOEA/DD is better than *v*C-MOEA/DD in most comparisons. This observation conforms to the conclusion in Section V-D. It supports our basic idea for enhancing the population diversity, regardless of violating the principles of traditional Pareto dominance relation and constraint handling to a certain extent.

VII. CONCLUSION

In this paper, we have suggested a unified paradigm, which combines dominance- and decomposition-based approaches,

for solving many-objective optimization problems. In MOEA/DD, we use a set of weight vectors to specify several subregions in the objective space. Furthermore, each weight vector also defines a subproblem for fitness evaluation. The parent population is updated in a steady-state scheme, where only one offspring solution is considered each time. To avoid unnecessary dominance comparisons, we employ the method proposed in [66] to update the nondomination level structure of the population after introducing an offspring solution. Generally speaking, the worst solution, in term of scalarization function value, belongs to the last nondomination level will be removed from further consideration. However, to further improve the population diversity, MOEA/DD allows such kind of solution survive to the next round in case it is associated with an isolated subregion. The performance of MOEA/DD has been investigated on a set of unconstrained benchmark problems with up to 15 objectives. The empirical results demonstrate that the proposed MOEA/DD is able to find a well-converged and well-distributed set of points for all test instances. In addition to the unconstrained optimization problems, we have further extended MOEA/DD for handling problems with various constraints. Following the idea for unconstrained cases, we give the worst solution, with the largest CV, a second chance for survival in case it is associated with an isolated subregion. Empirical results have extensively demonstrated the superiority of C-MOEA/DD for solving problems with different types of constraints and a large number of objectives.

In the future, it is interesting to investigate the performance of MOEA/DD for a wider range of problems, such as problems with complicated PS shapes (see [40], [72], [73]), combinatorial optimization problems (see [18], [46], [74]), and problems in real-world with a large number of objectives. Moreover, instead of finding the entire PF, it is also interesting to use our proposed MOEA/DD to find the subset of Pareto-optimal solutions preferred by the decision maker. In addition, as discussed in [75], the performance of an EMO algorithm might degenerate with the increase of number of decision variables. Therefore, it is worthwhile to investigate the scalability of MOEA/DD for large-scale problems.

REFERENCES

[1] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer, 1999.

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

[3] C. A. Coello Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. New York, NY, USA: Springer, 2007.

[4] K. Li, J. Zheng, M. Li, C. Zhou, and H. Lv, "A novel algorithm for non-dominated hypervolume-based multiobjective optimization," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, San Antonio, TX, USA, Oct. 2009, pp. 5220–5226.

[5] K. Li, Á. Fialho, and S. Kwong, "Multi-objective differential evolution with adaptive control of parameters and operators," in *Proc. 5th Int. Conf. Learn. Intell. Optim. (LION)*, Rome, Italy, Jan. 2011, pp. 473–487.

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

[7] K. Li, S. Kwong, R. Wang, J. Cao, and I. J. Rudas, "Multi-objective differential evolution with self-navigation," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, Seoul, Korea, Oct. 2012, pp. 508–513.

[8] K. Li, S. Kwong, R. Wang, K.-S. Tang, and K.-F. Man, "Learning paradigm based on jumping genes: A general framework for enhancing exploration in evolutionary multiobjective optimization," *Inf. Sci.*, vol. 226, pp. 1–22, Mar. 2013.

[9] K. Li and S. Kwong, "A general framework for evolutionary multiobjective optimization via manifold learning," *Neurocomputing*, vol. 146, pp. 65–74, Jul. 2014.

[10] G. Fu, Z. Kapelan, J. R. Kasprzyk, and P. Reed, "Optimal design of water distribution systems using many-objective visual analytics," *J. Water Resour. Plan. Manage.*, vol. 139, no. 6, pp. 624–633, 2013.

[11] R. J. Lygoe, M. Cary, and P. J. Fleming, "A real-world application of a many-objective optimisation complexity reduction process," in *Proc. 7th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Sheffield, U.K., 2013, pp. 641–655.

[12] O. Chikumbo, E. D. Goodman, and K. Deb, "Approximating a multi-dimensional Pareto front for a land use management problem: A modified MOEA with an epigenetic silencing metaphor," in *Proc. 2012 IEEE Congr. Evol. Comput. (CEC)*, Brisbane, QLD, Australia, pp. 1–9.

[13] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. 2008 IEEE Congr. Evol. Comput. (CEC)*, Hong Kong, pp. 2419–2426.

[14] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.

[15] J. Horn, N. Nafpliotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization," in *Proc. 1994 Int. Conf. Evol. Comput. (CEC)*, Orlando, FL, USA, pp. 82–87.

[16] 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.

[17] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization," in *Evolutionary Methods for Design, Optimisation, and Control*. Barcelona, Spain: CIMNE, 2002, pp. 95–100.

[18] 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.

[19] R. L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Trans. Evol. Comput.*, vol. 10, no. 1, pp. 29–38, Feb. 2006.

[20] V. R. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Proc. 2nd Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Faro, Portugal, 2003, pp. 376–390.

[21] J. Knowles and D. Corne, "Approximating the nondominated front using the Pareto archived evolution strategy," *Evol. Comput.*, vol. 8, no. 2, pp. 149–172, 2000.

[22] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Proc. 4th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Matsushima, Japan, 2007, pp. 742–756.

[23] S. F. Adra, T. J. Dodd, I. Griffin, and P. J. Fleming, "Convergence acceleration operator for multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 4, pp. 825–847, Aug. 2009.

[24] 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. (CEC)*, Seoul, Korea, 2001, pp. 957–962.

[25] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. 8th Int. Conf. Parallel Probl. Solving Nat. (PPSN)*, Birmingham, U.K., 2004, pp. 832–842.

[26] 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.

[27] C. Igel, N. Hansen, and S. Roth, "Covariance matrix adaptation for multi-objective optimization," *Evol. Comput.*, vol. 15, no. 1, pp. 1–28, 2007.

[28] 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.

[29] R. 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.

[30] K. Bringmann and T. Friedrich, "An efficient algorithm for computing hypervolume contributions," *Evol. Comput.*, vol. 18, no. 3, pp. 383–402, 2010.

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

[32] H. Sato, H. E. Aguirre, and K. Tanaka, "Self-controlling dominance area of solutions in evolutionary many-objective optimization," in *Proc. 8th Int. Conf. Simulate. Evol. Learn. (SEAL)*, Kanpur, India, 2010, pp. 455–465.

[33] M. Li, J. Zheng, R. Shen, K. Li, and Q. Yuan, "A grid-based fitness strategy for evolutionary many-objective optimization," in *Proc. 12th Genet. Evol. Comput. Conf. (GECCO)*, Portland, OR, USA, 2010, pp. 463–470.

[34] 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.

[35] M. Farina and P. Amato, "A fuzzy definition of 'optimality' for many-criteria optimization problems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 3, pp. 315–326, May 2004.

[36] 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.

[37] M. Li, J. Zheng, K. Li, Q. Yuan, and R. Shen, "Enhancing diversity for average ranking method in evolutionary many-objective optimization," in *Proc. 11th Int. Conf. Parallel Probl. Solving Nat. (PPSN)*, Kraków, Poland, 2010, pp. 647–656.

[38] 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.

[39] S. Kukkonen and J. Lampinen, "Ranking-dominance and many-objective optimization," in *Proc. 2007 IEEE Congr. Evol. Comput. (CEC)*, Singapore, pp. 3983–3990.

[40] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 450–455, Jun. 2014.

[41] 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.

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

[43] K. Sindhya, K. Miettinen, and K. Deb, "A hybrid framework for evolutionary multi-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 495–511, Aug. 2013.

[44] 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.

[45] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations," in *Proc. IEEE Int. Conf. Syst. Man Cybern. (SMC)*, San Antonio, TX, USA, 2009, pp. 1758–1763.

[46] 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. 11th Int. Conf. Parallel Probl. Solving Nat. (PPSN)*, Kraków, Poland, 2010, pp. 91–100.

[47] H. Ishibuchi, N. Akedo, and Y. Nojima, "Relation between neighborhood size and MOEA/D performance on many-objective problems," in *Proc. 7th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Sheffield, U.K., 2013, pp. 459–474.

[48] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, to be published.

[49] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.

[50] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Proc. 1st Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Zurich, Switzerland, 2001, pp. 82–95.

[51] 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.

[52] 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.

[53] L. Ben Said, S. Bechikh, and K. Ghedira, "The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 801–818, Oct. 2010.

[54] 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.

[55] K. Deb and A. Kumar, "Interactive evolutionary multi-objective optimization and decision-making using reference direction method," in *Proc. 9th Genet. Evol. Comput. Conf. (GECCO)*, London, U.K., 2007, pp. 781–788.

[56] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 77–99, Feb. 2013.

[57] R. C. Purshouse, C. Jalba, and P. J. Fleming, "Preference-driven co-evolutionary algorithms show promise for many-objective optimisation," in *Proc. 6th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Ouro Preto, Brazil, 2011, pp. 136–150.

[58] G. G. Yen and Z. He, "Performance metric ensemble for multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 131–144, Feb. 2014.

[59] 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.

[60] 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.

[61] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Compl. Syst.*, vol. 9, no. 2, pp. 1–34, 1994.

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

[63] J. J. Durillo, A. J. Nebro, F. Luna, and E. Alba, "On the effect of the steady-state selection scheme in multi-objective genetic algorithms," in *Proc. 5th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Nantes, France, 2009, pp. 183–197.

[64] D. Chafekar, J. Xuan, and K. Rasheed, "Constrained multi-objective optimization using steady state genetic algorithms," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Nantes, France, 2003, pp. 813–824.

[65] M. Yagoubi and M. Schoenauer, "Asynchronous master/slave MOEAs and heterogeneous evaluation costs," in *Proc. Genet. Evol. Comput. Conf. (GECCO)*, Philadelphia, PA, USA, 2012, pp. 1007–1014.

[66] K. Li, K. Deb, Q. Zhang, and S. Kwong, "Efficient non-domination level update approach for steady-state evolutionary multiobjective optimization," Dept. Electr. Comput. Eng., Michigan State Univ., East Lansing, MI, USA, Tech. Rep. COIN No. 2014014, 2014.

[67] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization* (Advanced Information and Knowledge Processing), A. Abraham, L. Jain, and R. Goldberg, Eds. London, U.K.: Springer, 2005, pp. 105–145.

[68] 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.

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

[70] 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.

[71] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

[72] Q. Zhang *et al.*, "Multiobjective optimization test instances for the CEC 2009 special session and competition," School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., and School Electr. Electron. Eng., Nanyang Technol. Univ., Singapore, Tech. Rep. CES-487, 2008.

[73] D. K. Saxena, Q. Zhang, J. A. Duro, and A. Tiwari, "Framework for many-objective test problems with both simple and complicated Pareto-set shapes," in *Proc. 6th Int. Conf. Evol. Multi-Criterion Optim. (EMO)*, Ouro Preto, Brazil, 2011, pp. 197–211.

[74] D. W. Corne and J. D. Knowles, "Techniques for highly multiobjective optimisation: Some nondominated points are better than others," in *Proc. 2007 Genet. Evol. Comput. Conf. (GECCO)*, London, U.K., pp. 773–780.

[75] J. J. Durillo *et al.*, "A study of multiobjective metaheuristics when solving parameter scalable problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 618–635, Aug. 2010.

Ke Li (S'10) received the B.Sc. and M.Sc. degrees in computer science and technology from Xiangtan University, Xiangtan, China, and the Ph.D. degree in computer science from City University of Hong Kong, Hong Kong, in 2007, 2010, and 2014, respectively.

He is a Post-Doctoral Research Associate with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, USA. His research interests include evolutionary multiobjective optimization, surrogate modeling, and machine learning.

Kalyanmoy Deb (F'13) received the bachelor's degree in mechanical engineering from Indian Institute of Technology, Kharagpur, Kharagpur, India, and the master's and Ph.D. degrees from University of Alabama, Tuscaloosa, AL, USA, in 1985, 1989, and 1991, respectively.

He is a Koenig Endowed Chair Professor with the Department of Electrical and Computer Engineering, Michigan State University (MSU), East Lansing, MI, USA. He also holds joint appointments with the Department of Mechanical Engineering and the Department of Computer Science and Engineering at MSU. His research interests include evolutionary optimization and their application in optimization, modeling, and machine learning. He has published over 374 research papers.

Prof. Deb received the Infosys Prize, the World Academy of Sciences Prize in Engineering Sciences, the Cajastur Mamdani Prize, the Distinguished Alumni Award from IIT Kharagpur, the Edgeworth-Pareto Award, the Bhatnagar Prize in Engineering Sciences, and the Bessel Research Award from Germany. He is also an Editorial Board Member of 20 major international journals. He is a fellow of American Society of Mechanical Engineers and three science academies in India.

Qingfu Zhang (M'01–SM'06) received the B.Sc. degree in mathematics from Shanxi University, Taiyuan, China, and the M.Sc. degree in applied mathematics and the Ph.D. degree in information engineering from Xidian University, Xi'an, China, in 1984, 1991, and 1994, respectively.

He is a Professor with the Department of Computer Science, City University of Hong Kong, Hong Kong, a professor with the School of Computer Science and Electronic Engineering, University of Essex, Colchester, U.K., and a Changjiang Visiting Chair Professor with Xidian University, China. He is currently leading the Metaheuristic Optimization Research Group in City University of Hong Kong. His research interests include evolutionary computation, optimization, neural networks, data analysis, and their applications. He holds two patents and has authored several research publications.

Dr. Zhang received the 2010 IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION Outstanding Paper Award. MOEA/D, a multiobjective optimization algorithm developed in his group, won the Unconstrained Multiobjective Optimization Algorithm Competition at the Congress of Evolutionary Computation 2009. He is an Associate Editor of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and the IEEE TRANSACTIONS ON CYBERNETICS. He is also an Editorial Board Member of three other international journals.

Sam Kwong (M'93–SM'04–F'14) received the B.S. and M.S. degrees in electrical engineering from State University of New York at Buffalo, Buffalo, NY, USA, and the University of Waterloo, Waterloo, ON, Canada, and the Ph.D. degree from University of Hagen, Hagen, Germany, in 1983, 1985, and 1996, respectively.

In 1990, he became a Lecturer at the Department of Electronic Engineering, City University of Hong Kong, Hong Kong, where he is currently a Professor and a Head at the Department of Computer Science. His research interests include evolutionary computation and video coding.

Dr. Kwong is an Associate Editor of IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS and *Information Sciences*.