A Many-Objective Evolutionary Algorithm With Enhanced Mating and Environmental Selections

Jixiang Cheng, Gary G. Yen, *Fellow, IEEE*, and Gexiang Zhang, *Member, IEEE*

***Abstract*—Multiobjective evolutionary algorithms have become prevalent and efficient approaches for solving multiobjective optimization problems. However, their performances deteriorate severely when handling many-objective optimization problems (MaOPs) due to the loss of selection pressure to drive the search toward the Pareto front and the ineffective design in diversity maintenance mechanism. This paper proposes a many-objective evolutionary algorithm (MaOEA) based on directional diversity (DD) and favorable convergence (FC). The main features are the enhancement of two selection schemes to facilitate both convergence and diversity. In the algorithm, a mating selection based on FC is applied to strengthen selection pressure while an environmental selection based on DD and FC is designed to balance diversity and convergence. The proposed algorithm is tested on 64 instances of 16 MaOPs with diverse characteristics and compared with seven state-of-the-art algorithms. Experimental results show that the proposed MaOEA performs competitively with respect to chosen state-of-the-art designs.**

***Index Terms*—Directional diversity (DD), favorable convergence (FC), many-objective evolutionary algorithm (MaOEA), many-objective optimization problem (MaOP).**

I. Introduction

DUE TO the population-based heuristic that allows searching for a Pareto approximate set in a single run, multiobjective evolutionary algorithms (MOEAs) have become prevalent and efficient approaches for solving multiobjective optimization problems (MOPs) with two or three objectives. Over the last two decades, many MOEAs have been proposed and most of them, including nondominated sorting genetic algorithm II (NSGA-II) [1] and strength Pareto evolutionary algorithm 2 (SPEA2) [2], are based on the concept of Pareto dominance. However, the performances of these MOEAs deteriorate severely when tackling MOPs with more than three objectives which are commonly referred to as many-objective optimization problems (MaOPs) [3].

Manuscript received September 9, 2014; revised January 10, 2015; accepted March 25, 2015. Date of publication April 21, 2015; date of current version July 28, 2015. This work was supported in part by the National Natural Science Foundation of China under Grant 61170016 and Grant 61373047, in part by the Program for New Century Excellent Talents in University under Grant NCET-11-0715, in part by Southwest Jiaotong University (SWJTU) under Project SWJTU12CX008, and in part by the 2014 Doctoral Innovation Funds of SWJTU.

J. Cheng and G. Zhang are with the School of Electrical Engineering, Southwest Jiaotong University, Chengdu 60031, China (e-mail: chengjixiang0106@126.com; zhgxdylan@126.com).

G. G. Yen is with the School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74075 USA (e-mail: gyen@okstate.edu).

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

Digital Object Identifier 10.1109/TEVC.2015.2424921

The main reason that MOEAs lose search capabilities in solving MaOPs is largely due to the loss of selection pressure to drive the search toward the Pareto front and the ineffective design of diversity maintenance scheme to balance diversity and convergence. As the number of objectives increases, the proportion of nondominated solutions in the population grows dramatically [4], which inadvertently makes Pareto dominance fail to provide sufficient selection pressure. Meanwhile, the diversity maintenance mechanism becomes the sole driving force in the environmental selection, which could lead to excessive diversity but insufficient convergence. Nowadays, many-objective evolutionary algorithms (MaOEAs) is an active research topic and various strategies have been proposed to improve the performance of MOEAs for MaOPs [5], [6].

An intuitive way to tackle the challenges caused by the curse of dimensionality is to reduce the number of objectives by using various techniques, such as principal component analysis [7], feature selection [8], and Pareto corner search [9]. However, the objective reduction unavoidably is accompanied by information loss from the original problems and whether MaOPs can be simplified to the MOPs that common MOEAs can handle depends on the problems' features.

To enhance the selective pressure, several relaxed forms of Pareto dominance relation have been proposed in the literature, such as α-dominance [10], ϵ-dominance [11], controlling dominance area of solutions [12], fuzzy Pareto dominance [4], and grid dominance [13]. The main drawback of these relaxed forms of Pareto dominance is that they often involve one or more parameters and these problem-dependent parameters are often chosen heuristically. Another strategy is the ranking composition, which first extracts the separated rank of every solution for each objective and then uses these ranks with an aggregation function to assign a fitness value for each solution. Some examples are global detriment [14], favor relation [15], L-optimality [16], and p-optimality [17]. This kind of methods can well increase the selection pressure; however, due to the lack of diversity maintenance mechanism, the population usually converges into a subregion of the Pareto front. Recently, the preference information is utilized to enhance selective pressure of MaOEAs, such as preference-inspired coevolutionary algorithm based on goals (PICEA-g) [18] and knee point driven evolutionary algorithm [19].

Indicator- and aggregation-based algorithms are two popular alternatives for MaOEAs in recent years. The

1089-778X © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

indicator-based approaches utilize various performance indicators like ϵ-indicator [20], hypervolume (HV) indicator [21], R2 indicator [22], and Δ_p indicator [23] to measure the quality of solutions. Among these indicators, the HV indicator is the most promising one because of its strict monotonic characteristic with regard to Pareto dominance. However, the HV prefers convex regions to concave ones [24] and its computational complexity is quite huge. The aggregation-based algorithms transform the original problem into a set of scalar functions that are simultaneously optimized in a single run. Multiple single objective Pareto sampling (MSOPS) [25] and multiobjective evolutionary algorithm based on decomposition (MOEA/D) [26] are two well-regarded examples. Particularly, MOEA/D has become a promising algorithm because of its good convergence performance and low computational complexity. However, to obtain a good distribution performance, the issue of weight vector generation should be properly addressed especially for problems with irregular Pareto fronts [27].

Balancing convergence and diversity is much more complicated in MaOPs than in MOPs. When applying Pareto dominance-based MOEAs to MaOPs, Pareto dominance criterion could not distinguish individuals well. As a result, the diversity maintenance-based selection plays a leading rule in the selection for survival, causing a phenomenon termed active diversity promotion [28]. Such a phenomenon is detrimental to the evolution of population because it favors dominance resistant solutions (DRSs), where DRSs refer to the nondominated solutions with poor values in at least one objective but with nearly optimal values in other objectives [10]. As the number of DRSs increases, Pareto based algorithms will fail in finding solutions near Pareto front, but produce solutions spreading over the objective space. Hence, making a good tradeoff between convergence and diversity becomes a challenging task when improving Pareto dominance-based MOEAs for MaOPs. At present, while most works address this problem by developing new individual comparison methods, i.e., modifying Pareto dominance concept or introducing new individual ranking schemes, only a few works deal with this problem from the aspect of diversity maintenance scheme.

This paper proposes an MaOEA based on directional diversity (DD) and favorable convergence (FC), in short named MaOEA-DDFC. In the algorithm, an FC indicator and a DD indicator are introduced to, respectively, measure the convergence and the diversity performances of an individual. In mating selection stage, the FC indicator combined with Pareto dominance criterion is applied to construct a mating pool in order to generate offspring with good convergence performance. For environmental selection, a diversity maintenance mechanism considering both diversity and convergence information of an individual is designed to balance diversity and convergence of the algorithm.

The rest of this paper is organized as follows. In Section II, an overview of related works that inspire the motivation of the proposed algorithm is given. Section III is devoted to the description of the proposed MaOEA-DDFC algorithm. Experimental results are presented and discussed in Section IV. Finally, the conclusion is drawn in Section V.

II. Related Works and Motivation

As how to facilitate the convergence performance of individuals and how to maintain a diversified population are two important ingredients of our MaOEA-DDFC algorithm, we first briefly review the main works on two related issues, i.e., the aggregation-based MaOEAs and the diversity maintenance mechanisms in Pareto-based MaOEAs. Afterwards, the motivation for this paper is presented.

A. Aggregation-Based MaOEAs

The basic aggregation method is the single objective optimizer, which multiplies the objective values with weights and accumulates them into a scalar value. Multiple runs of the single objective optimizer are applied in order to obtain a set of solutions, each corresponding to one weight vector. In [29], repeated single objective (RSO) was presented as a benchmark for other MaOEAs, where the weighted min–max function was used to aggregate objectives. As the total function evaluation time had to be divided among all weight vectors and the information in each run was not fully exploited, RSO could not succeed in reaching the Pareto front. However, according to [30], RSO still displays a good convergence performance and outperforms NSGA-II and SPEA2 on MaOPs with five and six objectives due to the property of the min–max function.

MSOPS [25] uses an aggregation function and a set of target vectors to assign scores for each solution. The scores are then ranked twice, resulting in a lexicographical order of the individuals. In this paper, two aggregation functions were discussed, i.e., weighted min–max function and a combination of weighted min–max with vector angle distance scaling. The weighted min–max focuses on the distance to the origin while vector angle distance scaling favors solutions whose position vectors have small intersecting angles with the target vectors. However, the distribution of solutions highly depends on the choice of target vectors. In MSOPS-II [31], a scheme using current population to generate target vectors was proposed and showed a significant performance enhancement.

MOEA/D [26] is one of the most popular algorithms for both MOPs and MaOPs. The algorithm decomposes the original problem into a number of scalar subproblems using uniformly distributed weight vectors and all subproblems are solved simultaneously. Each subproblem is optimized by only using information from its several neighboring subproblems, where the neighborhood is defined according to the Euclidean distance between their aggregation weight vectors. In this paper, three aggregation methods were discussed, i.e., weighted sum function, weighted Chebyshev function, and penalty-based boundary intersection. MOEA/D has demonstrated its superiority on both MOPs [32] and MaOPs [33]. In addition to the neighborhood structure, the weight vectors play an important role in the diversity performance. Recently, some works focus on developing new weight vector generation methods to improve MOEA/D performance [27], [34], [35].

The aggregation-based algorithms exhibit good convergence capability and computational efficiency. However, the main weakness lies in the difficulty of determining appropriate weight vectors. Although some works proposed different methods to generate uniformly distributed weight vectors like simplex-lattice design, simplex-centroid design, and uniform design [35], they unfortunately do not produce uniformly distributed solutions for most aggregation methods, such as Chebyshev function. Meanwhile, due to the characteristics of these weight vector generation schemes, most weight vectors have at least one component equal to zero, which makes the solutions searched by these weight vectors locate at the boundary of the front, rather than scatter over the front. Thus, for a circumstance that diversity is not excessively demanded, the aggregation based algorithm is a good choice.

B. Diversity Maintenance in Pareto-Based MaOEAs

A few works have been devoted to modify the diversity maintenance mechanism. Two kinds of modifications can be observed, that is, weakening diversity information and incorporating convergence information.

For the former one, Koppen and Yoshida [36] proposed four distance assignment schemes to replace crowding distance in NSGA-II based on measuring the highest degree to which a solution is nearly Pareto-dominated by others. Wagner *et al.* [30] proposed to assign the crowding distance of boundary solutions a value of zero in NSGA-II to weaken the effects of extreme solutions. Adra and Fleming [37] introduced a diversity management scheme to adaptively adjust the diversity requirement during the evolution process. When an excessive diversity is observed, which is indicated by a spread indicator, the crowding distance is neglected as a discriminatory criterion at individual survival stage. However, the approach requires a prior knowledge of the problem to calculate the spread indicator, which may hinder its applications. In NSGA-III [38], the diversity maintenance is aided by a number of well distributed reference points. Before the survival stage, each individual in the combined population is associated with a reference point according to its perpendicular distance to the reference line. When surviving some individuals from the critical front, the ones associated with the reference points whose niche counts are small have better chances to be selected. Thus, the diversity is limited through the niche count of reference point.

For the latter case, Li *et al.* [39] developed a shift-based density estimation (SDE) scheme to preserve the diversity while not losing the convergence of the population. The main idea is that when estimating the density of an individual, the positions of other individuals are first shifted according to a convergence comparison on each objective. As a result, poorly converged individuals will have high-density values. The scheme has shown significant performance improvements of three MOEAs on MaOPs and the combination (SPEA2 + SDE) of SPEA2 with SDE has been demonstrated to outperform several MaOEAs.

In NSGA-III, multiple reference points (lines) were applied to limit excessive diversity. Similar to aggregation-based methods, NSGA-III also encounters a difficulty of generating a limited number of well distributed reference points in a high-dimensional space. For example, when solving an MaOP with an irregular Pareto front, i.e., disconnected Pareto front or Pareto front only concentrated in a small region, some reference points (lines) will be useless and clustering phenomenon will appear [40]. Furthermore, NSGA-III requires a large computational budget to ensure convergence. The SDE scheme directly considers both diversity and convergence information in the individual survival stage and shows a good performance, which paves a new way to increase the convergence of Pareto based algorithms for MaOPs without losing diversity.

C. Motivation

To improve the performance of Pareto-based MOEAs for MaOPs, two important factors should be considered, i.e., mating selection and environmental selection (or diversity maintenance mechanism). For the former one, individuals with better convergence performance should have better chances to generate offspring. For the later one, as most individuals are nondominated solutions, the original diversity information should be weakened or other convergence information should be considered in order to survive solutions with good diversity as well as good convergence performances. Thus, how to measure the convergence and diversity degrees of an individual, and how to integrate the convergence information into diversity maintenance mechanism are two key design issues.

Although the diversity performance is subjected to the distribution of weight vectors, aggregation-based algorithms have shown good convergence performance and computational efficiency due to the use of aggregation functions. Thus, using aggregation function with an appropriate weight vector is a simple way to measure the convergence degree of an individual toward the Pareto front. A concept of favorable weight was proposed in [41], where each individual was assigned a weight vector that makes the individual superior over the others as much as possible with respect to an aggregation function. Based on this concept, we define an FC measurement based on Chebyshev function and favorable weight to measure the convergence degree of an individual. Meanwhile, the diversity of solutions in an MOP or MaOP can be understood in two ways. The first is that the solutions should distribute well along its approximated front, which is commonly used in MOEAs. The second is that the lines connecting the solutions and the origin should point at diverse directions. Although some algorithms have used predefined reference points as reference directions to control diversity (e.g., NSGA-III), generating appropriate reference points for MaOPs, especially those with irregular Pareto fronts, is not an easy task. To circumvent the drawback caused by reference points, we can project individuals onto a hyperplane toward the origin and use projection lines to represent the directions of solutions. Then the intersection points between the projection lines and the hyperplane can be used to measure the DD of solutions. The intersection points distributing well on the hyperplane also implies a good diversity of the solutions. As the convergence and diversity degrees of an individual are separately measured, the convergence can be incorporated into the diversity maintenance scheme in a tournament-like manner. Based on these considerations, we propose the MaOEA-DDFC algorithm, expounded in the next section.

Algorithm 1 Main Loop of MaOEA-DDFC

Input: Population size N

Randomly generate a population P
while termination condition is not satisfied **do**
    $P' \leftarrow Mating\_selection(P)$
    $P'' \leftarrow Variation(P')$
    $P \leftarrow Environmental\_selection(P \cup P'')$
end while
return P

III. PROPOSED ALGORITHM

A. Framework of the Proposed Algorithm

The main loop of MaOEA-DDFC is given by Algorithm 1. Firstly, a population, P, with N individuals is randomly sampled in the search space to form an initial population. Then, mating selection is implemented to select some promising solutions for variation. Finally, environmental selection is performed to survive N individuals from the union of the current population and their offspring. The basic procedure of the proposed algorithm is similar to most MOEAs, but with improved mating and environmental selections.

B. Mating Selection

Mating selection, aiming at selecting better-fit individuals for variation, plays an important role in the convergence performance of an algorithm. Commonly, individuals with promising convergence performance should have better chances to produce offspring. However, as the Pareto dominance criterion fails to discriminate the convergence degrees of individuals in most circumstances, another criterion should be used. In our algorithm, an FC function, based on Chebyshev function and favorable weight, is applied to measure the convergence performance of an individual, defined as

$$\mathrm{FC}(\boldsymbol{u}) = \max_{1 \le j \le M} \left\{ w_{u,j} \left| f_j(\boldsymbol{u}) - z_j^* \right| \right\} \tag{1}$$

where M is the number of objectives; $\boldsymbol{z}^* = (z_1^*, \ldots, z_M^*)$ is the ideal point of the problem; and $\boldsymbol{w_u} = (w_{u,1}, \ldots, w_{u,M})$ is the favorable weight for $\boldsymbol{u}$. A small $\mathrm{FC}(\boldsymbol{u})$ value implies a better convergence performance. Usually, $\boldsymbol{z}^*$ is unavailable and we use $\boldsymbol{z}^{\min} = (z_1^{\min}, \ldots, z_M^{\min})$ instead, where $z_j^{\min} = \min_{\boldsymbol{u} \in P}\{f_j(\boldsymbol{u})\}$ and it is updated each time a smaller value is found. The favorable weight $\boldsymbol{w_u}$ with respect to Chebyshev function can be formulated as [42]

$$w_{u,j} = \begin{cases} 1, & \text{if } f_j(\boldsymbol{u}) = z_j^* \\ 0, & \text{if } f_j(\boldsymbol{u}) \neq z_j^* \\ & \text{but } \exists m, f_m(\boldsymbol{u}) = z_m^* \\ \frac{1}{f_j(\boldsymbol{u}) - z_j^*} \left[\sum_{m=1}^{M} \frac{1}{f_m(\boldsymbol{u}) - z_m^*} \right]^{-1}, & \text{otherwise.} \end{cases} \tag{2}$$

The favorable weight has some good attributes according to [41]. If $\boldsymbol{u}$ is an efficient solution, no other solution can be closer to the ideal point than $\boldsymbol{u}$ with respect to the favorable weight of $\boldsymbol{u}$. Also, it makes an individual superior over the others as much as possible within this weight. Therefore, the FC value reflects the best convergence degree of an individual. Fig. 1 gives the contours based on Chebyshev distance using favorable weight of a solution $\boldsymbol{u}$.

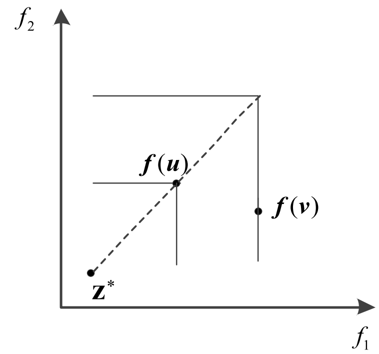


Fig. 1. Illustration of contours based on Chebyshev function from the ideal point with a favorable weight of $\boldsymbol{u}$.

Algorithm 2 Mating Selection

Input: Population P, mating pool size N

$P' \leftarrow \emptyset$
while $|P'| < N$ **do**
    randomly select two individuals $\boldsymbol{u}, \boldsymbol{v}$ from P
    if $\boldsymbol{u} \prec \boldsymbol{v}$ **then**
        $P' \leftarrow P' \cup \{\boldsymbol{u}\}$
    else if $\boldsymbol{v} \prec \boldsymbol{u}$ **then**
        $P' \leftarrow P' \cup \{\boldsymbol{v}\}$
    else if $\mathrm{FC}(\boldsymbol{u}) < \mathrm{FC}(\boldsymbol{v})$ **then**
        $P' \leftarrow P' \cup \{\boldsymbol{u}\}$
    else if $\mathrm{FC}(\boldsymbol{v}) < \mathrm{FC}(\boldsymbol{u})$ **then**
        $P' \leftarrow P' \cup \{\boldsymbol{v}\}$
    else if $rand(0, 1) \le 0.5$ **then**
        $P' \leftarrow P' \cup \{\boldsymbol{u}\}$
    else
        $P' \leftarrow P' \cup \{\boldsymbol{v}\}$
    end if
end while
return P'

The procedure of mating selection based on FC function is given by Algorithm 2. Here, a binary tournament selection is applied, although any other selection scheme could be used as well. At each selection step, two individuals are randomly picked up from the population. The one that Pareto dominates the other is allowed to enter the mating pool. If two individuals are nondominated with respect to each other, the one with a smaller FC value is preferred. When FC function still cannot distinguish between the two solutions, a random one is chosen.

C. Environmental Selection

Environmental selection aims at surviving N individuals from the union of the current population and their offspring to create a new population for the next generation. The surviving solutions should be well distributed and have good convergence performance. However, for an MaOP, classic Pareto dominance-based ranking scheme scarcely works to guarantee convergence and the diversity maintenance mechanism

Algorithm 3 Environmental Selection

Input: Population size N, combined population Q with size of $2N$

1: $(\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_L) \leftarrow$ *Pareto_nondominated_sort*(Q)
2: $P \leftarrow \emptyset$, $i \leftarrow 1$
3: **while** $|P \cup \mathcal{F}_i| < N$ **do**
4: $\quad P \leftarrow P \cup \mathcal{F}_i$, $i \leftarrow i+1$
5: **end while**
6: **if** $|P \cup \mathcal{F}_i| = N$ **then**
7: $\quad P \leftarrow P \cup \mathcal{F}_i$
8: **else**
9: $\quad (\tilde{P}, \tilde{\mathcal{F}}_i) \leftarrow$ *projection*$(P, \mathcal{F}_i)$
10: $\quad$ **while** $|P| < N$ **do**
11: $\quad\quad (R, \tilde{R}) \leftarrow$ *direction_based_selection* $(\mathcal{F}_i, \tilde{P}, \tilde{\mathcal{F}}_i)$
12: $\quad\quad (r, \tilde{r}) \leftarrow$ *convergence_based_selection* $(R, \tilde{R})$
13: $\quad\quad P \leftarrow P \cup \{r\}$, $\tilde{P} \leftarrow \tilde{P} \cup \{\tilde{r}\}$
14: $\quad\quad \mathcal{F}_i \leftarrow \mathcal{F}_i \backslash \{r\}$, $\tilde{\mathcal{F}}_i \leftarrow \tilde{\mathcal{F}}_i \backslash \{\tilde{r}\}$,
15: $\quad$ **end while**
16: **end if**
17: **return** P

plays the leading role in the selection, which results in a well distributed but poorly-converged population. To address this deficiency, other convergence information should be considered in the selection. Furthermore, as an effective diversity scheme, the surviving solutions should point at diverse directions. We define directional density function to measure the diversity degree of an individual and then combine it with FC to perform environmental selection.

The procedure of environmental selection is shown in Algorithm 3. The union of the current population and their offspring is first sorted into different fronts $(\mathcal{F}_1, \mathcal{F}_2, \ldots,)$. based on the principle of nondominated sorting [1] (line 1). Then, the fronts are moved one by one into population P from the lowest level to the highest level until a front $\mathcal{F}_i$ is encountered (lines 2–7), where $\mathcal{F}_i$ is the critical front that cannot be survived completely. In fact, for MaOPs, i usually equals to 1 given all individuals in the combined population are nondominated with respect to each other. To further select some individuals from $\mathcal{F}_i$, the union $P \cup \mathcal{F}_i$ is projected onto a hyperplane (line 9) in order to estimate the directional density. Thereafter, we select individuals from $\mathcal{F}_i$ one by one. Specifically, at each time surviving an individual, we first select L candidates from $\mathcal{F}_i$ according to their directional densities (line 11). When the size of $\mathcal{F}_i$ is less than L, all the members in $\mathcal{F}_i$ are considered as candidates. Afterwards, we select an individual from the L candidates to enter the population P according to their convergence performances (line 12). Finally, information is updated in order to survive the next individual (lines 13 and 14). According to the procedure, we can see that the proposed environmental selection takes the diversity and convergence information into account in a tournament-like manner, where the parameter L adjusts the degree of convergence information utilized. The procedures of projection for estimating directional density, direction-based selection for selecting candidate individuals, and convergence-based selection for choosing one individual are explained as follows.

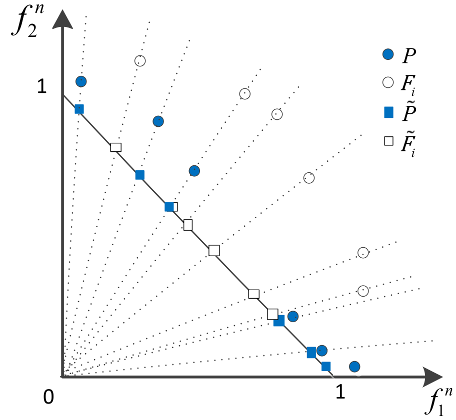


Fig. 2. Illustration of projection.

1) Projection: In most situations, different objectives usually have different ranges. In order to eliminate the effects of different ranges on the directional density estimation, a normalization procedure is performed on the union of survived individual P and critical front $\mathcal{F}_i$. In literature, several normalization methods exist. In our algorithm, we utilize the adaptive normalization method proposed in [38]. First, the ideal point $\boldsymbol{z}^{\min} = (z_1^{\min}, \ldots, z_M^{\min})$ is determined by identifying the minimum value for each objective on the union $P \cup \mathcal{F}_i$. Afterwards, each objective $f_j(\boldsymbol{u})$, $\forall \boldsymbol{u} \in P \cup \mathcal{F}_i$, is translated as $f_j^{'}(\boldsymbol{u})$ by subtracting $z_j^{\min}$ so the ideal point becomes the origin. Subsequently, the extreme point $\boldsymbol{z}_{\boldsymbol{j}}^{\max}$ in the jth objective axis is identified by finding the solution $\boldsymbol{u} \in P \cup \mathcal{F}_i$ that minimizes an achievement scalarizing function (ASF) with weight vector $\boldsymbol{w}^{\boldsymbol{j}}$, that is

$$\boldsymbol{z}_{\boldsymbol{j}}^{\mathbf{max}} = \underset{\boldsymbol{u} \in P \cup \mathcal{F}_i}{\operatorname{argmin}} \operatorname{ASF}\left(u, w^{\mathrm{j}}\right) = \underset{\boldsymbol{u} \in P \cup \mathcal{F}_i}{\operatorname{argmin}} \left\{ \max_{1 \le m \le M} \left\{ f_m^{'}(\boldsymbol{u}) \big/ w_m^j \right\} \right\} \tag{3}$$

where $\boldsymbol{w}^{\boldsymbol{j}} = (w_1^j, \ldots, w_M^j)$, $w_m^j = 10^{-6}$, $m \ne j$, and $w_j^j = 1$. Then the M extreme points can construct a $M-1$ dimensional linear hyperplane H. The intercept a_j of the jth objective axis and the hyperplane H can be calculated according to analytic geometry. Afterward, the objective value is normalized by

$$f_j^n(\boldsymbol{u}) = \frac{f_j^{'}(\boldsymbol{u})}{a_j - z_j^{\min}}, \quad \forall \boldsymbol{u} \in P \cup \mathcal{F}_i, \; j = 1, \ldots, M. \tag{4}$$

After normalization, the point $\boldsymbol{f}^n(\boldsymbol{u})$ is projected onto unit hyperplane toward the origin, producing a projection point $\tilde{\boldsymbol{u}}$. As the unit hyperplane is used, the vector $\tilde{\boldsymbol{u}}$ can reflect the direction of an individual $\boldsymbol{u}$ in the objective space. We store the projection points of P and $\mathcal{F}_i$ into $\tilde{P}$ and $\tilde{\mathcal{F}}_i$, respectively, and we will use them to estimate directional density of an individual. An illustration of projection is shown in Fig. 2, where the blue circle points are survived individuals, the blank circle points are the individuals on the critical front, and square points are their projection points onto the

Algorithm 4 Projection

Input: Survived solutions P, critical front $\mathcal{F}_i$

Determine idea point: $z_j^{\min} = \min_{\boldsymbol{u} \in P \cup \mathcal{F}_i} f_j(\boldsymbol{u})$, $j = 1, \ldots, M$

Translate objectives: $f_j^{'}(\boldsymbol{u}) = f_j(\boldsymbol{u}) - z_j^{\min}$, $j = 1, \ldots, M, \boldsymbol{u} \in P \cup \mathcal{F}_i$

Calculate extreme points $\boldsymbol{z_j^{\max}}$ by using (3), $j = 1, \ldots, M$

Calculate intercept a_j, $j = 1, \ldots, M$

$\forall \boldsymbol{u} \in P \cup \mathcal{F}_i$, normalize objective values $\boldsymbol{f}(\boldsymbol{u})$ by using (4).

$\forall \boldsymbol{u} \in P \cup \mathcal{F}_i$, project normalized point $\boldsymbol{f}^n(\boldsymbol{u})$ onto unit hyperplane toward the origin. The projection points of P and $\mathcal{F}_i$ are stored in $\tilde{P}$ and $\tilde{\mathcal{F}}_i$, respectively.

Algorithm 5 Direction-Based Selection

Input: Critical front $\mathcal{F}_i$, projection points $\tilde{P}$, $\tilde{\mathcal{F}}_i$, neighborhood size K, candidate set size L

```
1: if P̃ = ∅ then
2:     T = min{K, |F_i|}
3:     ∀u ∈ F_i, calculate the distances of its projection ũ to
       its T nearest points in F̃_i, denoted as d_1^ũ, ..., d_T^ũ
4: else
5:     T = min{K, |P|}
6:     ∀u ∈ F_i, calculate the distances of its projection ũ to
       its T nearest points in P̃, and the distances are denoted
       as d_1^ũ, ..., d_T^ũ
7: end if
8: ∀u ∈ F_i, calculate the directional density of u by using (5)
9: T = min{L, |F_i|}
10: Select T candidate individuals from F_i with T smallest
    directional densities. The selected individuals and their
    projections are stored in R and R̃, respectively.
```

unit-hyperplane, respectively. The procedure of projection is shown in Algorithm 4.

2) Direction-Based Selection: The direction-based selection aims at selecting L candidate individuals with good DD from $\mathcal{F}_i$, on which a convergence-based selection will be conducted in the next step. The procedure of this selection is shown in Algorithm 5. As the projection point on the unit hyperplane reflects the direction of an individual, we define a DD function to estimate the directional density of an individual by using projection points and k-nearest neighborhood distance. Specifically, for each individual $\boldsymbol{u} \in \mathcal{F}_i$, we first calculate the distances of projection $\tilde{\boldsymbol{u}}$ to its K nearest points in $\tilde{P}$, denoted as $d_1^{\tilde{\boldsymbol{u}}}, \ldots, d_K^{\tilde{\boldsymbol{u}}}$, then the DD of $\boldsymbol{u}$ is estimated by

$$\mathrm{DD}(\boldsymbol{u}) = \sum_{k=1}^{K} 1 \Big/ d_k^{\tilde{\boldsymbol{u}}}. \tag{5}$$

A small $\mathrm{DD}(\boldsymbol{u})$ value indicates the individual $\boldsymbol{u}$ is located at a sparse direction, and vice versa. After that, we select L individuals from $\mathcal{F}_i$ with L smallest DD values. The candidate individuals and their projection points are stored in R and $\tilde{R}$, respectively. In this step, L solutions with good diversity are selected to compete for survival according to their convergence degrees in the next step. When L equals to 1, the solution in the sparsest region but (maybe) with inferior convergence performance will be survived. As a result, the strategy falls back to the diversity maintenance scheme of the original Pareto-based algorithms.

Algorithm 6 Convergence-Based Selection

Input: Candidate solutions R and their projection points $\tilde{R}$

$\forall \boldsymbol{u} \in R$, calculate favorable weight $\boldsymbol{w_u}$ by using (2)

$\forall \boldsymbol{u} \in R$, calculate favorable convergence performance $\mathrm{FC}(\boldsymbol{u})$ by using (1)

Select a solution r from R with a probability proportional to its convergence performance. Its projection is denoted as $\tilde{r}$.

In the procedure, there are two parameters, K and L, which regulate how much information is used to estimate the directional density and how much convergence information is used to perform environmental selection. In addition, there are three special cases in this procedure. The first is that $\tilde{P}$ may be empty when surviving the first individual, which happens when the critical front $\mathcal{F}_i$ is the first front. For this case, we use the projection points in $\tilde{\mathcal{F}}_i$ to estimate the directional density (line 3). The second is the size of $\mathcal{F}_i$ or P may be smaller than K in some situations. For this case, we use the minimum value between K and $|\mathcal{F}_i|$ or $|P|$ to calculate DD values (lines 2 and 5). The third case happens when the size of $\mathcal{F}_i$ is less than L (line 9). In this case, all individuals in $\mathcal{F}_i$ are selected as candidate individuals without the consideration of diversity information.

3) Convergence-Based Selection: The convergence-based selection aims at selecting one individual with the most promising convergence performance to enter the next generation from L candidates with good DD. Here, we first use the FC function already described in Section III-B to measure the convergence degree of each candidate individual. Then an individual r is chosen with a probability proportional to its FC performance (i.e., inversely proportional to FC value), and its projection point $\tilde{r}$ is also selected to update $\tilde{P}$ and $\tilde{\mathcal{F}}_i$ for surviving the next individual. The reason that we use a probabilistic selection instead of a deterministic selection is to give any individual in R a chance to survive. The procedure of the convergence-based selection is shown in Algorithm 6.

In the environmental selection scheme, parameters K and L play important roles. For parameter K, a small value means a few neighbors are applied to estimate the directional density of an individual, and vice versa. For parameter L, it weighs the relative degree of diversity and convergence information considered in the selection. In particular, when $L = 1$, the scheme always selects an individual in the most sparse direction, regardless of the convergence degree of the individual. When $L \geq |\mathcal{F}_i|$, the scheme survives individuals according to their FC performance without the consideration of their diversity degrees. In the experiment section, we will discuss the effects of the two parameters on algorithm performance.

D. Complexity Analysis

The main complexity of MaOEA-DDFC lies in mating and environmental selections. Here, we consider the complexity of

TABLE I
SETTINGS OF THE TEST PROBLEMS

Problems	M	D	Parameter
DTLZ1	3,5,7,10	$M-1+k$	$k=5$
DTLZ2	3,5,7,10	$M-1+k$	$k=10$
DTLZ3	3,5,7,10	$M-1+k$	$k=10$
DTLZ4	3,5,7,10	$M-1+k$	$k=10$
DTLZ5	3,5,7,10	$M-1+k$	$k=10$
DTLZ6	3,5,7,10	$M-1+k$	$k=10$
DTLZ7	3,5,7,10	$M-1+k$	$k=20$
WFG1	3,5,7,10	$k+l$	$k=M-1, l=20$
WFG2	3,5,7,10	$k+l$	$k=M-1, l=20$
WFG3	3,5,7,10	$k+l$	$k=M-1, l=20$
WFG4	3,5,7,10	$k+l$	$k=M-1, l=20$
WFG5	3,5,7,10	$k+l$	$k=M-1, l=20$
WFG6	3,5,7,10	$k+l$	$k=M-1, l=20$
WFG7	3,5,7,10	$k+l$	$k=M-1, l=20$
WFG8	3,5,7,10	$k+l$	$k=M-1, l=20$
WFG9	3,5,7,10	$k+l$	$k=M-1, l=20$

the worst case, that is, all the individuals in the population are nondominated with one another. The mating selection involves comparing individuals based on Pareto dominance relationship, calculating FC value and comparing individuals based on FC. To construct a mating pool of size N, the total complexity of the above three steps is $O(M \cdot N)$. The environmental selection involves five procedures, i.e., nondominated sorting of the union population, calculating FC of the union population, calculating DD of the union population, surviving one individual, and updating DD of individuals in order to survive the next individual. The former two procedures have complexities of $O(M \cdot N^2)$ and $O(M \cdot N)$, respectively. The main complexity of the latter three procedures lies in recalculating the k-nearest distance of individuals after an individual is survived. However, by using appropriate data structure, we only need to calculate the distance between a pair of projection points and sort the distances only one time for each individual, which has a complexity of $O(N^2 \cdot \log N)$. Hence, the complexity of environmental selection is $O(M \cdot N^2 + N^2 \cdot \log_2 N)$. As a result, the overall complexity of MaOEA-DDFC becomes $O(M \cdot N^2 + N^2 \cdot \log_2 N)$.

IV. EXPERIMENTAL RESULTS

A. Experimental Design

1) Test Problems: Two test suites, Deb-Thiele-Laumanns-Zitzler (DTLZ) [43] consisting of seven problems, i.e., DTLZi ($i = 1, 2, \ldots, 7$), and Walking Fish Group (WFG) [44] consisting of nine problems, i.e., WFGi ($i = 1, 2, \ldots, 9$), are employed to conduct the experiments. The DTLZ problems have characteristics such as nonconvex, multimodal, disconnected, and nonuniform Pareto fronts; while the WFG problems possess nonseparable, deceptive, mixed shape, and scalable objective features. Therefore, these benchmark problems are challenging to evaluate the performance of MaOEAs. In addition, all the problems can be scaled to any number of objectives (M) and decision variables (D) by setting parameters k or l in their definitions. In this paper, we consider four instances for each problem; hence, there are 64 test instances in total. The settings of the test problems are shown in Table I, following the suggestions in [43] and [44].

TABLE II
NUMBER OF SAMPLE POINTS USED TO CALCULATE IGD VALUE

Problems	M=3	M=5	M=7	M=10
DTLZ1-4	5,151	15,335	27,771	36,350
DTLZ5	5,152	15,336	27,772	36,351
DTLZ6	5,152	15,336	27,772	36,351
DTLZ7	5,476	14,641	46,656	67,087
WFG1	5,041	14,096	50,752	20,195
WFG2	5,305	17,692	27,181	42,454
WFG3-9	5,151	15,335	27,771	36,350

2) Performance Metrics: Two performance metrics, inverted generational distance (IGD) [45] and HV [46] are utilized to quantify the algorithm performance. The smaller the IGD value or the larger the HV value, the better the performance of an algorithm. To calculate IGD, a reference set representing the Pareto front is required. The true Pareto fronts of DTLZ1 and WFG3 are hyperplanes. For these two problems, we use the reference point generation scheme in NSGA-III [38] to sample points on the hyperplane to construct the reference set. The Pareto fronts of DTLZ2, DTLZ3, DTLZ4, and WFG4-9 are hyperspheres; hence the intersection points between the hypersphere and the lines connecting the origin and the sample points on the hyperplane form the reference set. For DTLZ5-7 and WFG1-2 problems whose Pareto fronts are irregular, we first uniformly sample points over each decision space to generate a large solution set and then delete the dominated ones. Due to the characteristics of the above schemes, we could not generate any number of sample points on the true Pareto front. In our implementation, we construct reference sets with sizes close to $(M-2) \times 5000$ for problems with M objectives. Table II lists the exact number of sample points used to calculate IGD value for each test instance. To calculate HV, a reference point is needed. For DTLZ problems, each objective value of the reference point is set at 22. As WFG problems have different ranges for different objectives, we scale the jth objective value by dividing it by $2 \cdot j$. Because the objective values of the approximate solutions may still be out of the range [0, 1] or even far away from the value, we use a relatively large value 11 for each objective of the reference point to calculate HV value. In addition, we use Monte Carlo simulation with 10 000 sampling points to approximate the exact HV value.

3) Peer Algorithms: In order to validate the performance of MaOEA-DDFC algorithm, seven state-of-the-art MaOEAs, i.e., MSOPS [25], ϵ-MOEA [11], MOEA/D [26], HypE [21], PICEA-g [18], NSGA-III [38], and SPEA2 + SDE [39], are considered as peer competitors. MSOPS and MOEA/D are aggregation-based algorithms. ϵ-MOEA is a steady-state algorithm based on ϵ-dominance relation. PICEA-g is a coevolutionary algorithm that uses a family of preferences as a means of comparing solutions. HypE is a HV-based algorithm. NSGA-III is a recent improvement of NSGA-II that uses reference points to implement diversity maintenance. SPEA2 + SDE integrates a SDE scheme into the fitness assignment and archive truncation procedures of SPEA2.

4) Parameter Setting: The performances of MOEA/D, ϵ-MOEA, HypE, PICEA-g, and NSGA-III largely depend

TABLE III

IGD COMPARISONS OF THE EIGHT ALGORITHMS ON DTLZ TEST SUITE. THE BEST RESULT ON EACH TEST INSTANCE IS HIGHLIGHTED IN BOLDFACE. "+," "=," AND "−" DENOTE THAT MAOEA-DDFC PERFORMS SIGNIFICANTLY BETTER THAN, EQUIVALENT TO, AND WORSE THAN THE COMPARED ALGORITHM, RESPECTIVELY

	M	MaOEA-DDFC	MSOPS	ϵ-MOEA	MOEA/D	HypE	PICEA-g	NSGA-III	SPEA2+SDE
DTLZ1	3	2.17e-2(1.72e-3)	6.26e-1(5.59e-1) +	5.81e-2(7.51e-2) +	2.84e-2(2.51e-4) +	1.15e-1(4.25e-2) +	8.72e-2(2.82e-2) +	3.32e-2(1.37e-2) +	**2.04e-2(2.13e-4) -**
	5	9.32e-2(4.47e-2)	6.57e-1(5.58e-1) +	4.04e-1(4.53e-1) +	4.62e-1(5.34e-1) +	1.95e-1(8.03e-2) +	1.86e-1(3.63e-2) +	2.61e-1(2.29e-1) +	**6.05e-2(8.41e-4) -**
	7	5.82e-1(6.54e-1)	6.61e-1(3.53e-1) =	1.23e+0(1.14e+0) +	2.71e+0(2.51e+0) +	3.01e-1(1.91e-1) =	3.17e-1(3.23e-2) =	3.33e-1(2.00e-1) =	**8.23e-2(2.45e-3) -**
	10	3.20e-1(2.95e-1)	1.04e+0(7.86e-1) +	1.06e+0(7.88e-1) +	8.43e-1(6.41e-1) +	3.87e-1(2.13e-1) +	3.61e-1(1.88e-2) +	5.42e-1(3.38e-1) +	**1.08e-1(2.55e-2) -**
DTLZ2	3	5.58e-2(5.04e-4)	7.52e-2(9.17e-4) +	6.58e-2(2.28e-3) +	6.92e-2(2.78e-4) +	2.08e-1(6.58e-2) +	**5.51e-2(7.57e-4) -**	5.77e-2(8.52e-3) +	7.61e-2(3.05e-3) +
	5	2.33e-1(2.34e-3)	2.41e-1(2.06e-3) +	2.97e-1(4.55e-3) +	4.98e-1(1.91e-2) +	6.07e-1(8.10e-2) +	**2.27e-1(1.94e-3) -**	2.40e-1(4.25e-3) +	2.44e-1(3.35e-3) +
	7	4.20e-1(6.09e-3)	**3.66e-1(5.51e-3) -**	4.52e-1(4.16e-3) +	8.58e-1(3.37e-2) +	8.64e-1(4.93e-2) +	4.25e-1(7.29e-2) +	7.82e-1(1.78e-1) +	4.06e-1(6.91e-3) -
	10	5.28e-1(2.66e-2)	**4.90e-1(6.41e-3) -**	6.38e-1(7.45e-3) +	1.05e+0(1.44e-2) +	1.02e+0(3.70e-2) +	7.79e-1(6.02e-2) +	9.87e-1(2.39e-2) +	5.85e-1(7.78e-3) +
DTLZ3	3	2.27e-1(3.45e-1)	8.70e+0(4.94e+0) +	3.90e+0(3.25e+0) +	**1.76e-1(3.94e-1) =**	3.42e-1(2.74e-1) +	7.27e-1(4.75e-1) +	8.07e-1(5.70e-1) +	3.43e-1(4.55e-1) +
	5	4.62e+0(3.12e+0)	7.46e+0(3.25e+0) +	1.80e+1(9.54e+0) +	1.13e+1(8.57e+0) +	1.91e+0(1.18e+0) -	9.47e-1(5.80e-1) -	9.65e+0(5.86e+0) +	**4.46e-1(3.43e-1) -**
	7	1.42e+1(7.05e+0)	8.37e+0(5.01e+0) -	6.27e+1(2.26e+1) +	8.51e+0(6.97e+0) -	2.71e+0(1.63e+0) -	1.21e+0(6.63e-1) -	1.82e+1(1.10e+1) =	**5.17e-1(2.06e-1) -**
	10	1.25e+1(9.77e+0)	1.40e+1(5.96e+0) =	7.33e+1(2.43e+1) +	5.13e+0(3.67e+0) -	3.06e+0(2.11e+0) -	1.41e+0(6.68e-1) -	1.87e+1(1.27e+1) +	**8.62e-1(3.85e-1) -**
DTLZ4	3	**5.61e-2(5.82e-4)**	7.50e-2(9.08e-4) +	7.17e-2(3.75e-3) +	1.32e-1(1.63e-1) +	1.03e-1(1.72e-2) +	5.70e-2(1.27e-3) +	5.63e-2(5.05e-3) +	7.64e-2(2.86e-3) +
	5	2.38e-1(2.47e-3)	2.39e-1(3.15e-3) =	**2.28e-1(1.07e-2) -**	5.32e-1(5.26e-2) +	6.57e-1(8.02e-2) +	2.38e-1(4.65e-3) =	2.71e-1(1.05e-1) =	2.43e-1(3.13e-3) +
	7	4.26e-1(4.81e-3)	**3.76e-1(5.05e-3) -**	4.02e-1(2.09e-2) -	8.31e-1(4.27e-2) +	9.85e-1(5.09e-2) +	4.19e-1(7.87e-3) -	6.21e-1(1.77e-1) =	4.00e-1(7.27e-3) -
	10	5.94e-1(3.88e-2)	5.98e-1(1.93e-2) +	**5.45e-1(1.26e-2) -**	9.92e-1(3.88e-2) +	1.18e+0(5.82e-2) +	6.05e-1(1.40e-2) +	8.97e-1(5.87e-2) +	5.49e-1(9.24e-3) -
DTLZ5	3	**6.10e-3(4.58e-4)**	1.92e-2(2.63e-3) +	1.30e-2(3.71e-4) +	1.19e-2(3.43e-5) +	6.86e-2(3.01e-2) +	7.99e-3(4.29e-3) =	1.98e-2(8.07e-3) +	8.23e-3(6.17e-4) +
	5	7.66e-2(1.26e-2)	8.49e-2(2.32e-2) =	6.40e-2(4.26e-3) -	4.72e-2(2.28e-3) -	5.69e-1(1.32e-1) +	**3.33e-2(7.08e-3) -**	1.16e-1(2.90e-2) +	6.63e-2(1.16e-2) -
	7	1.73e-1(3.31e-2)	1.32e-1(4.00e-2) -	1.15e-1(1.68e-2) -	1.14e-1(7.71e-3) -	4.77e-1(2.28e-1) +	**5.58e-2(9.47e-3) -**	1.24e-1(2.55e-2) -	1.13e-1(2.22e-2) -
	10	2.30e-1(3.69e-2)	1.46e-1(4.23e-2) -	2.79e-1(1.56e-2) +	3.15e-1(1.47e-1) +	4.97e-1(2.21e-1) +	1.40e-1(3.13e-2) -	**1.39e-1(3.71e-2) -**	1.52e-1(3.78e-2) -
DTLZ6	3	1.39e-1(7.72e-3)	1.29e-1(4.61e-3) -	1.14e+0(1.15e-1) +	1.25e-1(1.42e-2) -	1.98e-1(1.48e-2) +	2.01e-1(1.48e-2) +	2.61e-1(2.48e-2) +	**1.13e-1(1.48e-2) -**
	5	**1.66e-1(2.46e-2)**	1.67e-1(2.07e-2) =	5.31e+0(3.42e-1) +	3.81e+0(7.57e-1) +	4.42e+0(4.13e-1) +	4.41e-1(5.37e-2) +	4.34e+0(4.93e-1) +	2.30e-1(1.27e-2) +
	7	**8.51e-2(7.95e-2)**	2.56e-1(3.43e-2) +	6.86e+0(3.92e-1) +	5.87e+0(7.92e-1) +	5.00e+0(3.50e-1) +	6.43e-1(1.04e-1) +	5.81e+0(5.61e-1) +	3.34e-1(1.83e-2) +
	10	**2.68e-1(4.50e-2)**	2.95e-1(1.68e-2) +	6.96e+0(4.00e-1) +	8.08e+0(6.18e-1) +	4.96e+0(4.11e-1) +	1.32e+0(1.74e-1) +	6.34e+0(4.75e-1) +	4.61e-1(3.28e-2) +
DTLZ7	3	1.16e-1(7.81e-2)	3.11e-1(2.11e-2) +	3.23e-1(2.47e-1) =	3.00e-1(1.31e-1) +	3.59e-1(2.83e-1) +	5.96e-1(2.96e-1) +	1.16e-1(5.95e-2) =	**6.78e-2(5.42e-2) -**
	5	4.56e-1(7.83e-2)	1.35e+0(2.30e-1) +	1.08e+0(4.58e-1) +	8.96e-1(1.04e-1) +	1.46e+0(8.91e-1) +	1.98e+0(1.94e-1) +	6.09e-1(3.04e-1) +	**3.36e-1(7.94e-2) -**
	7	8.81e-1(1.35e-1)	1.92e+0(3.43e-1) +	1.96e+0(4.74e-1) +	1.13e+0(1.06e-1) +	2.72e+0(1.07e+0) +	3.86e+0(1.88e-1) +	2.02e+0(1.01e+0) +	**6.35e-1(2.95e-2) -**
	10	1.53e+0(2.74e-1)	2.79e+0(5.82e-1) +	2.58e+0(6.65e-1) +	1.43e+0(1.40e-1) =	3.38e+0(1.68e+0) +	5.47e+0(8.59e-2) +	7.11e+0(1.82e+0) +	**1.00e+0(7.36e-2) -**
# +/=/-		—	16/5/7	22/1/5	21/2/5	24/1/3	16/3/9	21/5/2	10/0/18

upon the chosen weight vectors, the ϵ value, the number of sample points to approximate HV value, the number of goals, and the chosen reference points, respectively. In our experiments, we use exactly the same settings for compared algorithms as in their original papers, including the way to generate weight vectors and reference points for MOEA/D and NSGA-III. The maximum number of function evaluations is set to 30 000 for all algorithms as the stopping criterion. The population size for general algorithms (including MaOEA-DDFC, MSOPS, HypE, and SPEA2 + SDE) is set to 100. In MOEA/D, the population size is equal to the number of weight vectors. Due to the combinatorial nature of uniformly distributed weight vectors, the population size cannot be arbitrarily specified. Here, we use the closest integer to 100 among the possible options as population size, that is, 105, 126, 84, and 55 for three-, five-, seven-, and ten-objective problems, respectively. NSGA-III shares the same issue, where the population size is equal to the number of reference points. Although a double-layer reference point generation method is applied, the reference points still cannot be arbitrarily set. Here, we use the same method to generate reference points whose size is close to 100, i.e., 105, 85, 91, and 65 for three-, five-, seven-, and ten-objective problems, respectively.

The other parameters are set as follows. In ϵ-MOEA, the ϵ value for each test instance is set through a fine-tuned procedure so that the size of the approximated solutions is around 100. In MOEA/D, Chebyshev function is applied and the neighborhood size is set to 10. In HypE, the number of sample points to approximate HV is set to 10 000. In PICEA-g, the number of goals is set to $M \times 100$. In SPEA2 + SDE, the archive size is set to 100. For MaOEA-DDFC, the parameters K and L are, respectively, set to 5 and 3 according to a parameter sensitivity analysis. In addition, the simulated binary crossover [47] and polynomial mutation [48] are adopted variation operators. The crossover rate p_c, mutation rate p_m, as distribution indexes η_c and η_m for crossover and mutation are set to 1.0, $1/D$, 20, and 20 (except that NSGA-III uses a value of 30 for η_c), respectively.

To draw statistically meaningful observations, 30 independent runs for each algorithm on each test instance are performed. All the algorithms are executed on a notebook PC with 2.4 GHz CPU, 8 GB RAM, and Windows 7 OS.

B. Performance Comparison

1) Comparison in Terms of Performance Metric Values: Tables III and IV list the mean and standard deviation (in parentheses) of IGD values of the eight algorithms on DTLZ and WFG problem suites, respectively, where the best result based on the mean values for each test instance is highlighted in boldface. To investigate whether MaOEA-DDFC is superior to other algorithms in a statistical sense, Wilcoxon's rank sum test is performed between MaOEA-DDFC and each of the competing algorithms over each test instance at a 0.05 significance level. The test result is given at the end of each cell, represented by one of the symbols $+$, $=$, or $-$, where $+$, $=$, and $-$ indicate that MaOEA-DDFC performs significantly better than, equivalent to, and worse than the algorithm in the corresponding column, respectively. Meanwhile, the last row of Tables III and IV summarizes the number of test instances where MaOEA-DDFC is significantly superior to, equivalent to, and inferior to its competitors.

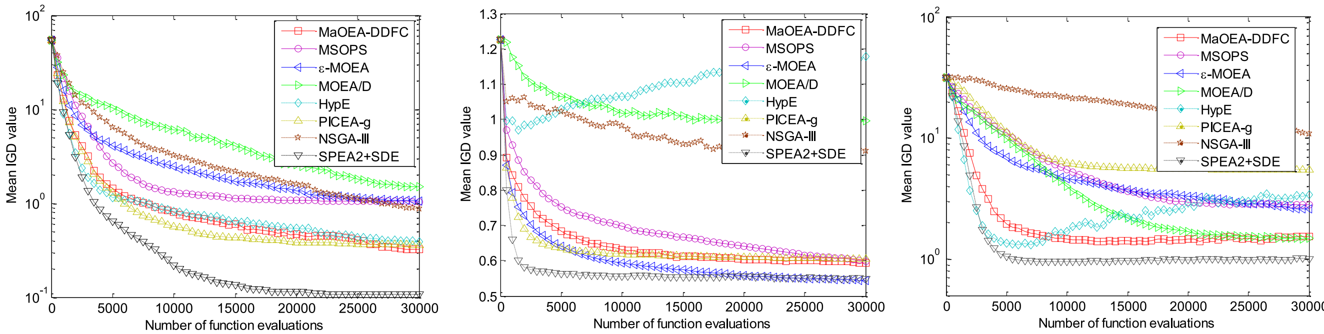
TABLE IV
IGD COMPARISONS OF THE EIGHT ALGORITHMS ON WFG TEST SUITE. THE BEST RESULT ON EACH TEST INSTANCE IS HIGHLIGHTED IN BOLDFACE. +, =, AND − DENOTE THAT MAOEA-DDFC PERFORMS SIGNIFICANTLY BETTER THAN, EQUIVALENT TO, AND WORSE THAN THE COMPARED ALGORITHM, RESPECTIVELY

	M	MaOEA-DDFC	MSOPS	ϵ-MOEA	MOEA/D	HypE	PICEA-g	NSGA-III	SPEA2+SDE
WFG1	3	4.85e-1(3.72e-2)	6.11e-1(1.62e-2) +	5.68e-1(2.22e-2) +	**3.29e-1(7.02e-3)** -	6.12e-1(3.61e-2) +	3.76e-1(2.43e-2) -	4.84e-1(3.42e-2) =	3.50e-1(1.33e-2) -
	5	7.30e-1(2.60e-2)	7.14e-1(3.35e-2) -	7.75e-1(2.06e-2) +	5.88e-1(1.09e-1) -	8.60e-1(2.47e-2) +	6.66e-1(1.76e-2) -	6.73e-1(4.83e-2) -	**5.60e-1(1.95e-2)** -
	7	7.96e-1(1.09e-2)	7.82e-1(5.16e-2) =	8.44e-1(3.28e-2) +	6.95e-1(1.06e-1) -	9.36e-1(2.34e-2) +	7.02e-1(7.27e-3) -	7.36e-1(5.63e-2) -	**6.31e-1(2.78e-2)** -
	10	8.18e-1(1.71e-2)	7.97e-1(6.41e-2) =	8.73e-1(3.47e-2) +	7.72e-1(8.77e-2) =	9.45e-1(1.57e-2) +	7.25e-1(8.07e-3) -	7.77e-1(4.33e-2) -	**6.49e-1(3.08e-2)** -
WFG2	3	1.20e-1(9.76e-3)	1.15e-1(6.71e-3) -	1.22e-1(1.05e-2) =	1.27e-1(2.90e-3) +	5.51e-1(1.19e-1) +	**9.27e-2(3.30e-3)** -	1.46e-1(3.32e-2) +	1.24e-1(1.19e-2) +
	5	2.33e-1(3.07e-2)	1.73e-1(2.07e-2) -	2.05e-1(1.46e-2) -	2.59e-1(7.10e-3) +	6.40e-1(1.24e-1) +	**1.19e-1(2.85e-3)** -	1.46e-1(1.12e-2) -	1.93e-1(3.15e-2) -
	7	2.80e-1(4.93e-2)	2.04e-1(1.18e-2) -	2.62e-1(2.89e-2) =	3.28e-1(3.42e-3) +	5.17e-1(1.95e-1) +	**1.29e-1(2.77e-3)** -	1.53e-1(1.22e-2) -	2.64e-1(2.73e-2) =
	10	3.50e-1(8.27e-2)	2.61e-1(1.02e-2) -	3.12e-1(4.50e-2) =	3.39e-1(3.07e-2) =	4.48e-1(1.74e-1) +	**1.86e-1(2.84e-3)** -	2.39e-1(1.64e-2) -	3.66e-1(5.14e-2) +
WFG3	3	2.56e-1(1.33e-3)	2.61e-1(1.77e-3) +	2.55e-1(8.10e-4) -	2.52e-1(1.48e-4) -	4.58e-1(1.11e-1) +	**2.51e-1(4.00e-4)** -	2.70e-1(1.28e-2) +	2.59e-1(2.94e-3) +
	5	3.26e-1(6.95e-3)	3.59e-1(7.52e-3) +	3.34e-1(2.77e-3) =	3.81e-1(1.04e-2) +	5.03e-1(2.67e-2) +	**3.14e-1(7.88e-3)** -	3.45e-1(1.36e-2) +	3.34e-1(8.07e-3) +
	7	3.48e-1(7.70e-3)	4.14e-1(1.94e-2) +	**3.42e-1(2.42e-3)** -	5.09e-1(4.14e-2) +	5.00e-1(3.05e-2) +	3.67e-1(7.55e-3) +	3.78e-1(1.78e-2) +	3.54e-1(4.29e-3) +
	10	**3.72e-1(5.06e-3)**	4.67e-1(2.19e-2) +	3.91e-1(3.84e-3) =	6.48e-1(7.52e-2) +	4.30e-1(2.29e-2) +	4.15e-1(9.35e-3) +	4.14e-1(2.00e-2) +	3.97e-1(3.74e-3) +
WFG4	3	5.63e-2(5.92e-4)	9.55e-2(3.46e-3) +	6.81e-2(1.42e-3) +	9.05e-2(1.62e-3) +	1.74e-1(4.12e-2) +	5.89e-2(7.82e-4) +	**5.60e-2(4.60e-3)** -	7.62e-2(2.49e-3) +
	5	**2.21e-1(1.69e-3)**	2.94e-1(5.58e-3) +	2.56e-1(2.60e-3) +	5.21e-1(1.96e-2) +	5.04e-1(4.51e-2) +	2.32e-1(2.58e-3) +	2.90e-1(8.18e-2) +	2.45e-1(5.63e-3) +
	7	3.95e-1(3.23e-2)	4.26e-1(9.65e-3) +	4.61e-1(4.52e-3) +	7.95e-1(5.68e-2) +	6.67e-1(4.48e-2) +	5.32e-1(4.52e-2) +	5.44e-1(6.00e-2) +	**3.89e-1(1.35e-2)** =
	10	6.51e-1(5.00e-2)	6.18e-1(2.46e-2) -	6.33e-1(1.16e-2) -	1.03e+0(2.00e-2) +	8.33e-1(5.00e-2) +	6.97e-1(3.89e-2) +	7.56e-1(5.53e-2) +	**5.93e-1(1.87e-2)** -
WFG5	3	**5.99e-2(3.63e-4)**	9.35e-2(2.56e-3) +	7.82e-2(2.03e-3) +	9.61e-2(1.87e-3) +	1.95e-1(6.50e-2) +	6.37e-2(9.78e-4) +	7.12e-2(2.03e-2) +	7.99e-2(3.27e-3) +
	5	**2.22e-1(1.80e-3)**	2.81e-1(7.53e-3) +	2.45e-1(3.80e-3) +	5.27e-1(1.45e-2) +	4.57e-1(4.21e-2) +	2.31e-1(2.98e-3) +	2.37e-1(3.96e-2) +	2.48e-1(5.14e-3) +
	7	3.92e-1(4.16e-3)	4.23e-1(7.57e-3) +	4.64e-1(7.61e-3) +	7.77e-1(5.31e-2) +	6.49e-1(3.05e-2) +	**3.85e-1(6.71e-3)** -	5.83e-1(8.17e-2) +	3.94e-1(9.59e-3) =
	10	**5.41e-1(3.62e-3)**	6.37e-1(2.19e-2) +	5.82e-1(5.78e-3) +	1.01e+0(2.82e-2) +	8.03e-1(5.05e-2) +	5.76e-1(1.54e-2) +	7.93e-1(6.00e-2) +	5.82e-1(1.33e-2) +
WFG6	3	**6.27e-2(2.01e-3)**	9.54e-2(2.93e-3) +	8.15e-2(4.59e-3) +	9.61e-2(2.64e-3) +	3.71e-1(7.66e-2) +	6.50e-2(1.59e-3) +	6.90e-2(1.33e-2) =	8.12e-2(2.84e-3) +
	5	**2.25e-1(1.16e-2)**	2.97e-1(7.27e-3) +	2.43e-1(3.99e-3) +	5.47e-1(3.10e-2) +	7.31e-1(3.94e-2) +	2.33e-1(2.49e-3) +	2.49e-1(4.12e-2) +	2.63e-1(1.19e-2) +
	7	4.45e-1(1.49e-1)	4.47e-1(1.15e-2) +	4.50e-1(5.65e-3) +	7.92e-1(4.22e-2) +	9.59e-1(1.85e-1) +	**3.86e-1(1.92e-2)** -	5.77e-1(7.72e-2) +	4.16e-1(1.09e-2) -
	10	9.34e-1(1.97e-1)	6.63e-1(3.29e-2) -	1.11e+0(1.25e-2) +	9.93e-1(4.80e-2) +	1.01e+0(3.72e-2) +	**5.92e-1(1.70e-2)** -	7.64e-1(4.39e-2) -	5.93e-1(1.64e-2) -
WFG7	3	**5.62e-2(9.01e-4)**	9.42e-2(3.69e-3) +	1.08e-1(1.10e-1) +	9.06e-2(9.52e-4) +	6.40e-1(1.91e-1) +	6.05e-2(1.24e-3) +	7.57e-2(1.10e-2) +	7.30e-2(2.92e-3) +
	5	**2.20e-1(1.66e-3)**	2.99e-1(7.39e-3) +	2.49e-1(5.14e-2) +	5.79e-1(3.08e-2) +	8.96e-1(1.09e-1) +	2.31e-1(2.26e-2) +	2.68e-1(8.60e-2) +	2.33e-1(4.07e-3) +
	7	3.91e-1(1.67e-2)	4.37e-1(1.66e-2) +	4.47e-1(5.64e-2) +	8.58e-1(2.64e-2) +	9.80e-1(5.94e-2) +	4.88e-1(1.00e-1) +	5.13e-1(8.98e-2) +	**3.79e-1(3.04e-2)** -
	10	6.78e-1(1.14e-1)	6.76e-1(4.06e-2) =	6.34e-1(6.74e-2) -	9.29e-1(7.02e-2) +	9.88e-1(4.76e-2) +	7.53e-1(5.72e-2) +	7.85e-1(8.44e-2) +	**6.00e-1(2.61e-2)** =
WFG8	3	5.79e-2(4.76e-4)	9.57e-2(3.31e-3) +	6.54e-2(5.67e-4) +	9.71e-2(1.41e-3) +	5.38e-1(4.14e-2) +	6.10e-2(7.51e-4) +	**5.76e-2(1.21e-3)** -	7.93e-2(2.43e-3) +
	5	2.23e-1(1.70e-3)	2.92e-1(6.20e-3) +	2.37e-1(2.10e-3) +	5.37e-1(8.87e-3) +	8.73e-1(7.56e-2) +	**2.22e-1(2.91e-3)** =	2.68e-1(8.33e-2) +	2.69e-1(1.46e-2) +
	7	4.41e-1(7.01e-2)	4.34e-1(1.23e-2) -	5.92e-1(2.88e-3) +	7.75e-1(5.56e-2) +	1.00e+0(7.11e-2) +	**3.82e-1(1.85e-2)** -	6.02e-1(6.72e-2) +	4.21e-1(8.28e-3) -
	10	8.37e-1(9.33e-2)	6.83e-1(2.98e-2) -	8.32e-1(3.78e-3) -	9.85e-1(5.75e-2) +	1.06e+0(5.89e-2) +	6.03e-1(3.19e-2) -	7.74e-1(5.48e-2) -	**6.02e-1(1.41e-2)** -
WFG9	3	**7.19e-2(2.16e-3)**	9.98e-2(2.59e-3) +	1.32e-1(1.40e-1) +	2.29e-1(2.26e-1) +	1.85e-1(2.29e-2) +	7.36e-2(1.08e-3) +	9.26e-2(8.61e-3) +	8.52e-2(1.69e-3) +
	5	3.57e-1(1.28e-1)	3.01e-1(3.29e-2) =	3.36e-1(1.15e-1) =	6.06e-1(7.65e-2) +	6.20e-1(7.00e-2) +	4.78e-1(1.75e-1) +	5.34e-1(1.36e-1) +	**2.80e-1(7.91e-2)** -
	7	6.90e-1(1.02e-1)	5.14e-1(6.99e-2) -	6.28e-1(1.07e-1) -	9.04e-1(1.08e-1) +	8.22e-1(5.33e-2) +	7.55e-1(1.04e-1) +	8.00e-1(1.01e-1) +	**5.08e-1(1.27e-1)** -
	10	8.11e-1(5.62e-2)	7.83e-1(6.06e-2) =	7.66e-1(8.35e-2) -	1.04e+0(4.72e-2) +	9.04e-1(4.50e-2) +	9.24e-1(5.98e-2) +	9.54e-1(6.30e-2) +	**7.14e-1(9.31e-2)** -
# +/=/-		—	21/5/10	23/6/7	30/2/4	36/0/0	20/1/15	24/2/10	18/4/14

From Table III, it can be seen that MaOEA-DDFC obtained five best IGD values out of 28 DTLZ test instances, worse than the value 12 obtained by SPEA2 + SDE while larger than the values obtained by the other algorithms (i.e., MSOPS, ϵ-MOEA MOEA/D, PICEA-g, and NSGA-III got the best result on 3, 2, 1, 4, and 1 instances while HypE did not obtain any best result). The last row of the table implies that MaOEA-DDFC performs significantly better than the six competitors (i.e., MSOPS, ϵ-MOEA, MOEA/D, HypE, PICEA-g, and NSGA-III) over more than half of the test instances, while performing worse than SPEA2 + SDE in 18 instances. The inferior performance of MaOEA-DDFC against SPEA2 + SDE may be due to the fact that SDE strategy considers the convergence and diversity as a whole when evaluating an individual while MaOEA-DDFC separates the two aspects. From Table IV, it can be seen that MaOEA-DDFC obtained the best IGD results on ten cases out of 36 test instances, second to the value of 11 obtained by both PICEA-g and SPEA2 + SDE. Meanwhile, ϵ-MOEA, MOEA/D, and NSGA-III obtained the best IGD values on 1, 1, and 2 test instances. However, according to statistical test, MaOEA-DDFC is significantly superior to MSOPS, ϵ-MOEA, MOEA/D, HypE, PICEA-g, NSGA-III, and SPEA2 + SDE on 21, 23, 30, 36, 20, 24, and 18 test instances, while it is inferior to MSOPS, ϵ-MOEA, MOEA/D, PICEA-g, NSGA-III, and SPEA2 + SDE only on 10, 7, 4, 15, 10, and 14 test instances, respectively.

To observe the evolutionary behaviors of the algorithms, Fig. 3 plots the trajectory of the mean IGD metric value versus the number of function evaluations in each algorithm on three MaOPs with ten objectives, i.e., DTLZ1, DTLZ4, and DTLZ4. According to the figures, MaOEA-DDFC shows a clear advantage over most of its competitors, although its performance is not as good as SPEA2 + SDE. The reason may be due to the population or archive truncation mechanism. In SPEA2 + SDE, the *k*th nearest neighbor method originally proposed in SPEA2 is utilized to estimate the density of individuals.

To further analyze the results in Tables III and IV, the frequencies of ranks in terms of mean IGD values for each algorithm over all 64 test instances are summarized in Table V, where the rank is determined by a lexicographic order in a descending model. For example, the frequency of the rank in the second row and the second column is 18, which implies that MaOEA-DDFC receives the second best mean IGD values (i.e., rank 2) among the eight algorithms over 18 test instances. Generally, a larger frequency value at a smaller rank indicates a better performance for an algorithm. From Table V, it can be seen that the frequency of rank 1 of MaOEA-DDFC is 15, larger than the values of the other algorithms except for SPEA2 + SDE. Meanwhile, the worst rank of MaOEA-DDF is 7 and its frequency is 1, which are smaller than those of others. Also, it can be inferred that the performance of HypE is the worst one. According to the results, we can conclude that



(a) (b) (c)

Fig. 3. Trajectory of the mean IGD value on three MaOPs with ten objectives. (a) DTLZ1. (b) DTLZ4. (c) DTLZ7.

TABLE V
RANK FREQUENCIES OF THE EIGHT ALGORITHMS OVER ALL 64 TEST INSTANCES IN TERMS OF IGD

	#Rank 1	#Rank 2	#Rank 3	#Rank 4	#Rank 5	#Rank 6	#Rank 7	#Rank 8
SPEA2+SDE	23	8	14	5	11	1	2	0
MaOEA-DDFC	15	18	6	9	7	8	1	0
PICEA-g	15	16	10	6	10	3	1	3
MSOPS	3	7	12	7	9	19	4	3
NSGA-III	3	5	6	13	7	21	8	1
ϵ-MOEA	3	3	8	18	11	6	7	8
MOEA/D	2	6	4	4	5	4	24	15
HypE	0	1	4	2	4	2	17	34

MaOEA-DDFC is superior to MSOPS, ϵ-MOEA, MOEA/D, HypE, PICEA-g, and NSGA-III and comparable to SPEA2 + SDE in terms of IGD metric on the considered test instances.

Due to space limitation, we do not list the detailed results in terms of HV (similar to Tables III–V). Instead, we will consider it in the comparison of overall performance.

2) Comparison of Pareto Fronts: To intuitively compare algorithm performance, the parallel coordinates of the approximated Pareto fronts obtained by the eight algorithms on DTLZ1 with ten objectives are shown in Fig. 4. The solutions in each figure are obtained in a particular run given the same initial population. From Fig. 4, it can be observed that MSOPS, ϵ-MOEA, and HypE fail to converge to the true Pareto front of DTLZ1 whose maximum value of each objective is 0.5. The figure also indicates that MOEA/D converges to only a few boundary solutions. The main reason lies in the weight vectors used. Actually, according to the weight vector generation method in MOEA/D, for a high-dimensional space, several components of a weight vector are zero, which makes the approximate solutions distributed on the boundary of the Pareto front. NSGA-III faces a similar problem because of the distribution of reference points. The solutions obtained by PICEA-g are also biased toward to a few objectives. The reason may lie in the evolutionary mechanism of the preferences used to discriminate individuals. Both MaOEA-DDFC and SPEA2 + SDE can converge to the true Pareto front; however, the objective values of SPEA2 + SDE are much smaller than the maximum value of 0.5, which implies that the solutions obtained by SPEA2 + SDE concentrate at the center of the approximated Pareto front while MaOEA-DDFC can maintain a relatively good diversity of solutions extended into the boundary.

3) Comparison of Overall Performance: In order to measure the overall performance of an algorithm, the average rank [49] is considered. Given a rank set $R = \{r_1, r_2, \ldots, r_{|R|}\}$ of an algorithm, the average rank is calculated by $\mu = 1/R \sum_{r \in R} r$, where $|R|$ is the number of test instances. An algorithm is said to be good if it has a low μ value. Fig. 5 plots the average ranks of the eight algorithms in terms of IGD and HV over all 64 test instances.

According to the figure, MaOEA-DDFC gets the second average rank in terms of IGD, and its value is slightly larger than the value of SPEA2 + SDE. Regarding to HV, the average rank obtained by MaOEA-DDFC is quite smaller than the values achieved by ϵ-MOEA, HypE, NSGA-III, and SPEA2 + SDE, while slightly larger than the values of the rest three algorithms. By simultaneously considering the average ranks in terms of IGD and HV, we can conclude that the overall performance of MaOEA-DDFC is better than or at least equal to the seven competitors.

C. Validation of Selection Schemes

In order to validate the effectiveness of the proposed mating selection and environmental selection schemes, we compare MaOEA-DDFC with the following two variants.

1) *Variant I:* In this variant, the mating selection is implemented only based on Pareto-dominance concept. If two individuals are nondominated with respect to each other during the binary tournament procedure, a random one is selected to enter the mating pool. This variant is to

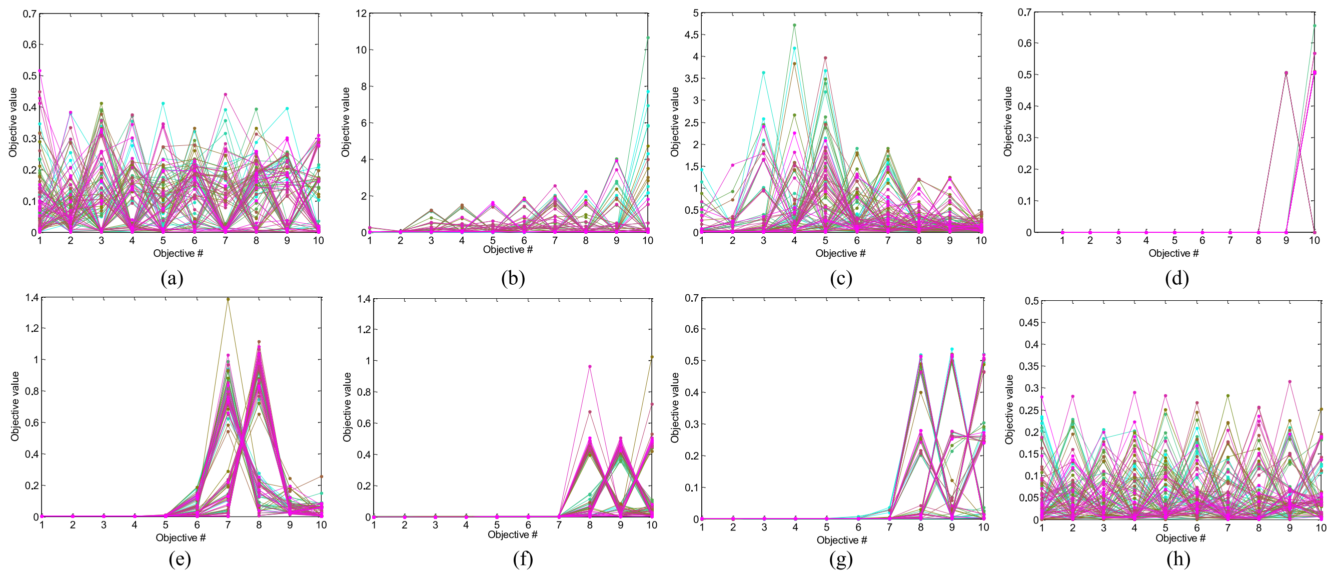


Fig. 4. Parallel coordinates of the solutions obtained by eight algorithms on DTLZ1 with ten objectives. (a) MaOEA-DDFC. (b) MSOPS. (c) ϵ-MOEA. (d) MOEA/D. (e) HypE. (f) PICEA-g. (g) NSGA-III. (h) SPEA2 + SDE.

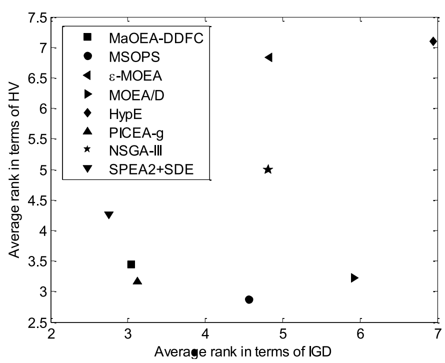


Fig. 5. Plot of average ranks on IGD and HV over 64 test instances.

validate whether FC criterion based mating selection is helpful to improve the convergence performance.

2) *Variant II:* The only difference of this variant and MaOEA-DDFC is that the convergence is not considered in the environmental selection. That is, variant II only uses the Pareto-dominance rank and DD criterion to survive individuals (i.e., $L = 1$). This variant is to investigate whether the proposed diversity maintenance mechanism can maintain a delicate balance between convergence and diversity.

We test the two variants on DTLZ test suite with three-, five-, seven-, and ten-objectives for 30 runs and apply IGD to quantify the performance. The parameter settings of the two variants are the same as in MaOEA-DDFC except that variant II uses a value of 1 for parameter L. The mean and standard deviation (in parenthesis) of IGD values of the two variants as well as MaOEA/DDFC are listed in Table VI, where the best mean value for each test instance is highlighted in boldface. Also, the Wilcoxon's rank sum test is performed to investigate whether there is a significant performance difference between MaOEA-DDFC and its two variants.

Table VI shows that MaOEA-DDFC obtains 14 best mean values among the 28 test instances. Also, the statistical test results indicates that MaOEA-DDFC significantly outperforms the two variants on more than ten cases, while significantly outperformed by the two variants on only a few test instances. Concerning the multimodal problems DTLZ1, DTLZ3, and DTLZ6, the IGD values achieved by MaOEA-DDFC are lower than the values obtained by variant I on most cases. In particular, on DTLZ1 and DTLZ3 problems with seven- and ten-objectives, the IGD values of MaOEA-DDFC are lower than those of variant I in a large degree, which demonstrates that the proposed mating selection scheme indeed improves the convergence performance. Meanwhile, the comparative

TABLE VI
PERFORMANCE COMPARISON OF MAOEA-DDFC AND ITS TWO VARIANTS ON DTLZ TEST SUITE IN TERMS OF IGD

	M	MaOEA-DDFC	Variant I	Variant II
DTLZ1	3	2.17e-2(1.72e-3)	**2.15e-2(1.29e-3)** =	4.02e-2(1.68e-2) +
	5	**9.32e-2(4.47e-2)**	2.48e-1(2.13e-1) +	1.05e-1(5.54e-2) +
	7	5.82e-1(6.54e-1)	1.33e+0(1.04e+0) +	**4.94e-1(3.08e-1)** =
	10	**3.20e-1(2.95e-1)**	2.78e+0(2.49e+0) +	3.38e-1(5.15e-1) =
DTLZ2	3	**5.58e-2(5.04e-4)**	5.58e-2(5.87e-4) =	5.62e-2(6.37e-4) +
	5	**2.33e-1(2.34e-3)**	2.34e-1(2.58e-3) =	2.36e-1(4.70e-3) +
	7	**4.20e-1(6.09e-3)**	4.38e-1(8.82e-3) +	4.36e-1(8.59e-3) +
	10	**5.28e-1(2.66e-2)**	6.59e-1(4.78e-2) +	5.31e-1(3.17e-2) +
DTLZ3	3	**2.27e-1(3.45e-1)**	7.71e-1(7.34e-1) =	4.71e-1(6.18e-1) +
	5	**4.62e+0(3.12e+0)**	9.35e+0(5.45e+0) =	5.53e+0(3.87e+0) =
	7	**1.42e+1(7.05e+0)**	4.26e+1(2.46e+1) =	1.79e+1(1.51e+1) =
	10	**1.25e+1(9.77e+0)**	1.57e+2(5.48e+1) =	1.54e+1(9.79e+0) =
DTLZ4	3	5.61e-2(5.82e-4)	5.63e-2(5.18e-4) =	**5.61e-2(4.57e-4)** =
	5	2.38e-1(2.47e-3)	2.39e-1(2.12e-3) +	**2.36e-1(2.16e-3)** -
	7	**4.26e-1(4.81e-3)**	4.48e-1(3.00e-2) +	4.29e-1(4.76e-3) +
	10	**5.94e-1(3.88e-2)**	6.95e-1(8.77e-2) +	6.21e-1(5.79e-2) +
DTLZ5	3	6.10e-3(4.58e-4)	**5.69e-3(4.68e-4)** -	7.25e-3(1.13e-3) +
	5	7.66e-2(1.26e-2)	1.35e-1(4.56e-2) +	**7.60e-2(1.53e-2)** =
	7	1.73e-1(3.31e-2)	2.85e-1(7.39e-2) +	**1.65e-1(2.54e-2)** =
	10	2.30e-1(3.69e-2)	2.23e+0(1.20e-1) +	**2.26e-1(6.64e-2)** =
DTLZ6	3	1.39e-1(7.72e-3)	**1.38e-1(3.76e-3)** =	1.42e-1(8.86e-3) +
	5	1.66e-1(2.46e-2)	1.75e-1(4.10e-2) =	**1.53e-1(2.22e-2)** -
	7	**8.51e-2(7.95e-2)**	5.68e+0(3.40e-1) +	2.33e+0(2.24e+0) +
	10	2.68e-1(4.50e-2)	6.05e+0(3.10e-1) +	**2.66e-1(3.61e-2)** =
DTLZ7	3	**1.16e-1(7.81e-2)**	1.47e-1(1.27e-1) =	1.24e-1(1.03e-1) =
	5	4.56e-1(7.83e-2)	4.95e-1(2.26e-1) =	**4.52e-1(6.87e-2)** =
	7	8.81e-1(1.35e-1)	**7.57e-1(4.24e-2)** -	8.43e-1(1.36e-1) =
	10	1.53e+0(2.74e-1)	1.41e+0(1.98e-1) =	**1.37e+0(1.51e-1)** -
# +/=/-		—	13/13/2	12/13/3

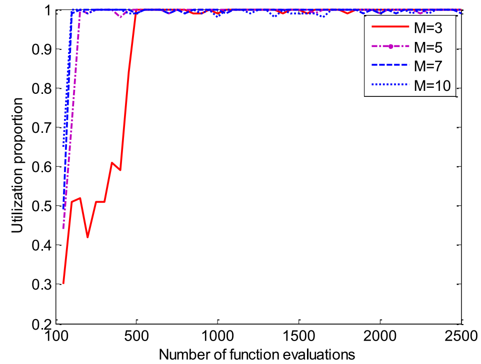


Fig. 6. Proportion of utilizing FC measure to construct mating pool along with the number of generations on DTLZ2 problem.

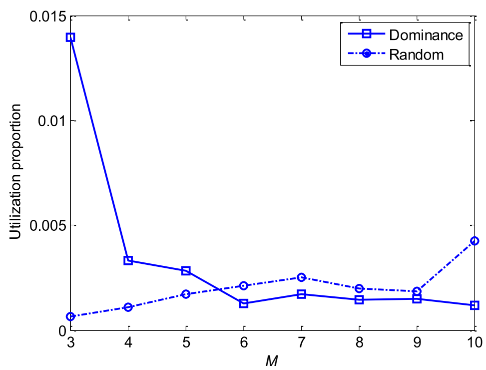


Fig. 7. Proportions of utilizing Pareto dominance and random selection to construct mating pool in the whole evolutionary process on DTLZ2 problem with different number of objectives.

results between MaOEA-DDFC and variant II indicate that the proposed diversity maintenance scheme can maintain a good balance between convergence and diversity.

D. Characteristics of FC Measure

In MaOEA-DDFC, three criteria are employed to construct mating pool, i.e., Pareto dominance relationship, FC measure, and random selection. To investigate the contributions of the three criteria on the mating selection process, the DTLZ2 problem with different number (M) of objectives is adopted to conduct a further experiment. DTLZ2 is chosen because of its simplicity and unimodal features, which enable us to clearly observe algorithm behavior [13]. Fig. 6 illustrates the proportion of utilizing the FC measure to select parents for variation over the number of generations. Because after a few generations, the utilization rate of FC approximates one, we only plot the early 25 generations for clarity. Fig. 7 illustrates the proportions of utilizing Pareto dominance and random selection to construct mating pool in the whole evolutionary process for different number of objectives.

From Fig. 6, several phenomena can be observed. First, the ratio of utilizing FC measure to construct mating pool quickly increases to the value of 1. The reason lies in the fact that in the tournament selection step, the Pareto dominance relationship is usually unable to distinguish the two members randomly selected from the population when the number of dimensions is high, while FC measure is able to differentiate them. Second, there is a slight fluctuation on the FC utilization ratio, which is caused by two factors. On one hand, Pareto dominance relationship discriminates two individuals occasionally. On the other hand, FC measure is still unable to differentiate them and random selection is evoked, which happens rarely. According to Fig. 7, we can learn that among the total number of comparisons in the mating selection step (i.e., 30 000 comparisons in a whole evolutionary stage), Pareto dominance and random selection only occupy very small proportions. Moreover, as the number of objectives increases, the utilization frequency of Pareto dominance decreases. According to Figs. 6 and 7, we can say that the FC criterion offers good discrimination ability in the comparison process.

V. Conclusion

General MOEAs encounter challenges in solving MaOPs due to the loss of selection pressure to drive the population toward the Pareto front and the ineffective design in diversity maintenance mechanism to balance diversity and convergence. To address this issue, this paper develops an MaOEA-DDFC algorithm for MaOPs by exploiting the advantages that an aggregation function with appropriate weight vector can measure the convergence performance of an individual. During the mating selection, FC combined with Pareto dominance relation is applied to select well-converged individuals for generating offspring; therefore, the selection pressure is strengthened. Meanwhile, the DD and FC is combined in a tournament-like manner, so the leading position occupied by diversity is weakened and a good trade-off between diversity and convergence can be reached. Extensive experiments have been conducted to verify the algorithm performance. The results show that the proposed MaOEA-DDFC performs better than or equal to seven state-of-the-art competitors. The effectiveness of the mating and environmental selection schemes and the characteristics of FC measurement are also investigated. Our continuing research will be thoroughly analyzing the performance of MaOEA-DDFC using metrics ensemble approach [50], analyzing the discriminatory power of FC measure on more complex problems, extending the algorithm to solve constrained MaOPs [51], and real-world problems with highly noncommensurate objective dimensions.

References

[1] 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.

[2] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization,” Dept. Electr. Eng., Swiss Fed. Inst. Technol. (ETH), Zurich, Switzerland, TIK Rep. 103, 2001.

[3] M. Farina and P. Amato, “On the optimal solution definition for many-criteria optimization problems,” in *Proc. NAFIPS-FLINT,* New Orleans, LA, USA, 2002, pp. 233–238.

[4] 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.

[5] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, “Evolutionary many-objective optimization: A short review,” in *Proc. IEEE CEC*, Hong Kong, 2008, pp. 2419–2426.

[6] C. von Lucken, B. Baran, and C. Brizuela, “A survey on multi-objective evolutionary algorithms for many-objective problems,” *Comput. Optim. Appl.*, vol. 58, no. 3, pp. 707–756, 2014.

[7] K. Deb and D. Saxena, "Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in *Proc. IEEE CEC*, Vancouver, BC, Canada, 2006, pp. 3352–3360.
[8] A. L. Jaimes, C. C. A. Coello, and D. Chakraborty, "Objective reduction using a feature selection technique," in *Proc. GECCO*, Atlanta, GA, USA, 2008, pp. 673–680.
[9] H. K. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 539–556, Aug. 2011.
[10] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: Does non-dominated really mean near to optimal?" in *Proc. IEEE CEC*, Seoul, Korea, 2001, pp. 957–962.
[11] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ϵ-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evol. Comput.*, vol. 13, no. 4, pp. 501–525, 2005.
[12] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," in *Proc. EMO*, Matsushima, Japan, 2007, pp. 5–20.
[13] 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.
[14] M. Garza-Fabre, G. T. Pulido, and C. A. C. Coello, "Ranking methods for many-objective optimization," in *Proc. MICAI*, Guanajuato, Mexico, 2009, pp. 633–645.
[15] N. Drechsler, R. Drechsler, and B. Becker, "Multi-objective optimisation based on relation favour," in *Proc. Int. Conf. Evol. Multi-Criter. Optim.*, Zurich, Switzerland, 2001, pp. 154–166.
[16] 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.
[17] E. C. Jara, "Multi-objective optimization by using evolutionary algorithms: The *p*-optimality criteria," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 167–179, Apr. 2014.
[18] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 474–494, Aug. 2013.
[19] X. Zhang, Y. Tian, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, to be published.
[20] E. Zitzler and S. Kunzli, "Indicator-based selection in multiobjective search," in *Proc. PPSN*, Birmingham, U.K., 2004, pp. 832–842.
[21] 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.
[22] D. H. Phan and J. Suzuki, "R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization," in *Proc. IEEE CEC*, Cancún, Mexico, 2013, pp. 1836–1845.
[23] C. A. R. Villalobos and C. A. C. Coello, "A new multi-objective evolutionary algorithm based on a performance assessment indicator," in *Proc. GECCO*, Philadelphia, PA, USA, 2012, pp. 505–512.
[24] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Hypervolume-based multiobjective optimization: Theoretical foundations and practical implications," *Theor. Comput. Sci.*, vol. 425, no. 1, pp. 75–103, 2012.
[25] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. IEEE CEC*, vol. 4. Canberra, ACT, Australia, 2003, pp. 2678–2684.
[26] 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.
[27] M. Daneshyari and G. G. Yen, "Cultural-based multiobjective particle swarm optimization," *IEEE Trans. Syst. Man, Cybern. B, Cybern.*, vol. 41, no. 2, pp. 553–567, Apr. 2011.
[28] 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.
[29] E. J. Hughes, "Evolutionary many-objective optimisation: Many once or one many?" in *Proc. IEEE CEC*, Edinburgh, U.K., 2005, pp. 222–227.
[30] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Proc. EMO*, Matsushima, Japan, 2007, pp. 742–756.
[31] E. J. Hughes, "MSOPS-II: A general-purpose many-objective optimiser," in *Proc. IEEE CEC*, Singapore, 2007, pp. 3944–3951.
[32] H. Li and Q. F. 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.
[33] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Effects of using two neighborhood structures on the performance of cellular evolutionary algorithms for many-objective optimization," in *Proc. IEEE CEC*, Trondheim, Norway, 2009, pp. 2508–2515.
[34] X. Ma *et al.*, "MOEA/D with uniform decomposition measurement for many-objective problems," *Soft. Comput.*, vol. 18, no. 12, pp. 2541–2564, 2014.
[35] Y. Y. Tan, Y. C. Jiao, H. Li, and X. K. Wang, "MOEA/D + uniform design: A new version of MOEA/D for optimization problems with many objectives," *Comput. Oper. Res.*, vol. 40, no. 6, pp. 1648–1660, 2013.
[36] M. Koppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Proc. EMO*, Matsushima, Japan, 2007, pp. 727–741.
[37] 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.
[38] 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.
[39] 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.
[40] H. Jain and K. Deb, "An improved adaptive approach for elitist non-dominated sorting genetic algorithm for many-objective optimization," in *Proc. EMO*, Sheffield, U.K., 2013, pp. 307–321.
[41] B. Soylu and M. Koksalan, "A favorable weight-based evolutionary algorithm for multiple criteria problems," *IEEE Trans. Evol. Comput.*, vol. 14, no. 2, pp. 191–205, Apr. 2010.
[42] R. E. Steuer, *Multiple Criteria Optimization: Theory, Computation, and Application*. New York, NY, USA: Wiley, 1986, p. 425.
[43] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary Multiobjective Optimization: Theoretical Advances and Applications*. London, U.K.: Springer, 2005, pp. 105–145.
[44] 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.
[45] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.
[46] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithms—A comparative case study," in *Proc. Parallel Prob. Solving Nat.*, Amsterdam, The Netherlands, 1998, pp. 292–301.
[47] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex. Syst.*, vol. 9, no. 2, pp. 115–148, 1995.
[48] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Comput. Sci. Inf.*, vol. 26, no. 4, pp. 30–45, 1996.
[49] C. K. Chow and S. Y. Yuen, "A multiobjective evolutionary algorithm that diversifies population by its density," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 149–172, Apr. 2012.
[50] G. G. Yen and Z. He, "Performance metrics ensemble for multiobjective evolutionary algorithm," *IEEE Trans. Evol. Comput.*, vol. 18, no. 1, pp. 131–144, Feb. 2014.
[51] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 514–525, Jun. 2009.

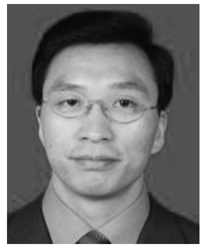
Jixiang Cheng received the B.Sc. and M.Sc. degrees in automation from Southwest Jiaotong University, Chengdu, China, in 2008 and 2011, respectively, where he is currently working toward the Ph.D. degree.

His current research interests include differential evolution, evolutionary multiobjective optimization, membrane computing, and their applications.

Gary G. Yen (S'87–M'88–SM'97–F'09) received the Ph.D. degree in electrical and computer engineering from University of Notre Dame, Notre Dame, IN, USA, in 1992.

He is a Regents Professor with the School of Electrical and Computer Engineering, Oklahoma State University (OSU), Stillwater, OK, USA. His research interest includes intelligent control, computational intelligence, evolutionary multiobjective optimization, conditional health monitoring, signal processing, and their industrial/defense applications.

Prof. Yen received the Regents Distinguished Research Award from OSU in 2009, the 2011 Andrew P. Sage Best Transactions Paper Award from the IEEE Systems, Man and Cybernetics Society, and the 2013 Meritorious Service Award from the IEEE Computational Intelligence Society. He was an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS and *IEEE Control Systems Magazine* from 1994 to 1999, and IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY and IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS from 2000 to 2010. He is an Associate Editor for IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION and IEEE TRANSACTIONS ON CYBERNETICS. He has been the Vice President for the Technical Activities of the IEEE Computational Intelligence Society in 2004 and 2005. He was the Founding Editor-in-Chief of *IEEE Computational Intelligence Magazine* from 2006 to 2009. He was the President of the IEEE Computational Intelligence Society in 2010 and 2011. He was elected as a Distinguished Lecturer for the term 2012–2014.



Gexiang Zhang (M'03) received the B.Sc. and M.Sc. degrees from Southwest University of Science and Technology, Mianyang, China, in 1997 and 2002, respectively, and the Ph.D. degree from Southwest Jiaotong University, Chengdu, China, in 2005.

Since 2005 he has been a Professor with the School of Electrical Engineering, Southwest Jiaotong University, where he leads the Research Group of Nature-Inspired Computation and Smart Grid. His research interests include natural computing, such as membrane computing, evolutionary computation, and DNA computing, bioinformatics, and smart grid. He has published over 100 scientific papers in international journals or conferences.