A new dominance relation-based evolutionary algorithm for many-objective optimization

Yuan, Yuan; Xu, Hua; Wang, Bo; Yao, Xin

DOI:
10.1109/TEVC.2015.2420112

Document Version
Publisher's PDF, also known as Version of record

Citation for published version (Harvard):
Yuan, Y, Xu, H, Wang, B & Yao, X 2016, 'A new dominance relation-based evolutionary algorithm for many-objective optimization', *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 1, pp. 16-37. https://doi.org/10.1109/TEVC.2015.2420112

A New Dominance Relation-Based Evolutionary Algorithm for Many-Objective Optimization

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

***Abstract*—Many-objective optimization has posed a great challenge to the classical Pareto dominance-based multiobjective evolutionary algorithms (MOEAs). In this paper, an evolutionary algorithm based on a new dominance relation is proposed for many-objective optimization. The proposed evolutionary algorithm aims to enhance the convergence of the recently suggested nondominated sorting genetic algorithm III by exploiting the fitness evaluation scheme in the MOEA based on decomposition, but still inherit the strength of the former in diversity maintenance. In the proposed algorithm, the nondominated sorting scheme based on the introduced new dominance relation is employed to rank solutions in the environmental selection phase, ensuring both convergence and diversity. The proposed algorithm is evaluated on a number of well-known benchmark problems having 3–15 objectives and compared against eight state-of-the-art algorithms. The extensive experimental results show that the proposed algorithm can work well on almost all the test functions considered in this paper, and it is compared favorably with the other many-objective optimizers. Additionally, a parametric study is provided to investigate the influence of a key parameter in the proposed algorithm.**

***Index Terms*—Convergence, diversity, dominance relation, many-objective optimization, nondominated sorting.**

I. Introduction

RECENTLY, many-objective optimization, typically referring to the optimization of problems having four or more objectives, has attracted increasing attention in evolutionary multiobjective optimization (EMO) community [1], [2]. The boom of the research on evolutionary many-objective optimization is mainly inspired from two aspects. On the one hand, the optimization problems involving a high number of objectives indeed appear widely in many real-world applications, e.g., control system design [3], [4], industrial scheduling [5], [6], and software engineering [7], [8]. Hence, the practitioners are in need of an effective optimizer to solve these problems at hand. On the other hand, the popular Pareto dominance-based multiobjective evolutionary algorithms (MOEAs), such as nondominated sorting genetic algorithm II (NSGA-II) [9], strength Pareto evolutionary algorithm 2 (SPEA2) [10], and Pareto envelope-based selection algorithm II (PESA-II) [11], have encountered great difficulties in many-objective optimization, although they have shown excellent performance on problems with two or three objectives. The primary reason is that almost all the solutions in the population become nondominated with the number of objectives increasing, which would lead to the severe loss of Pareto-based selection pressure toward the Pareto front (PF). This difficulty has been pointed out both analytically and experimentally in the early studies [12]–[14] on evolutionary many-objective optimization.

To overcome the drawback of Pareto dominance-based MOEAs, some efforts have been made in this paper. In summary, the developing techniques can be roughly classified into the following three types.

1) *Adoption of New Preference Relations:* Since the Pareto-dominance relation scales poorly in many-objective optimization, it is natural to use other preference relations, including modified Pareto dominance and different ranking schemes, so as to produce fine selection pressure toward PF. Up to now, many alternative preference relations, such as favor relation [15], ϵ dominance [16], [17], fuzzy Pareto dominance [18], [19], preference order ranking [20], and so on [21]–[25], have been proposed.
2) *Adoption of New Diversity Promotion Mechanisms:* In many-objective optimization, the Pareto dominance could not provide sufficient selection pressure to make progress in a given population, so the diversity selection mechanism begins to play a key role in such cases. This phenomenon is so called active diversity promotion [26]. Some experimental observations [1], [2], [26], [27] have indicated that it has the potential detrimental effect on the convergence of MOEAs because most of the existing diversity criteria, e.g., crowding distance [9], tend to favor dominance resistant solutions [12] (i.e., the solutions with high performance in at least one of the objectives, but with especially poor performance in the rest of the objectives). Consequently, the final obtained solutions

Manuscript received May 9, 2014; revised September 6, 2014 and December 14, 2014; accepted February 23, 2015. Date of publication April 6, 2015; date of current version January 27, 2016. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2012CB316301, in part by the National Natural Science Foundation of China under Grants 61175110, 61305079 and 61329302, in part by the National S&T Major Projects of China under Grant 2011ZX02101-004, in part by the National Banking Information Technology Risk Management Projects of China, and in part by the China Scholarship Council. The work of Xin Yao was also supported by EPSRC under Grant EP/J017515/1 and a Royal Society Wolfson Research Merit Award.

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

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

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

Digital Object Identifier 10.1109/TEVC.2015.2420112

may present a good diversity over the objective space but with the poor proximity to the global PF. To weaken or avoid such effect, new mechanisms promoting the diversity are thus needed. In contrast to the first type, there has not been much research in this respect yet. Adra and Fleming [28] presented two mechanisms for managing diversity and investigated their impact on the overall convergence in many-objective optimization. Deb and Jain [29] proposed an improved NSGA-II procedure, i.e., NSGA-III, which replaces the crowding distance operator in NSGA-II with a clustering operator aided by a set of well-distributed reference points. Li *et al.* [30] developed a general modification of the diversity criterion for Pareto dominance-based MOEAs, i.e., the shift-based density estimation (SDE) strategy, covering both distribution and convergence information of individuals.

3) Adoption of two separate archives for convergence and diversity: This kind of approach separates the nondominated solutions into two archives, which promote convergence and diversity during the evolutionary process, respectively. The Two-Archive algorithm [31] is the first MOEA based on this idea. Recently, Wang *et al*. [32] suggested an improved one for many-objective optimization.

It is worth noting that, unlike Pareto dominance-based MOEAs, decomposition- and indicator-based MOEAs, have been found to be very promising in many-objective optimization, although they also have their own drawbacks [25]. The former decomposes a problem with multiple objectives into a set of single-objective subproblems through aggregation functions, and then solves these subproblems simultaneously by evolving a population of solutions. MOEA based on decomposition (MOEA/D) [33], [34] is the most typical implementation of this class. So far, lots of studies based on MOEA/D have been conducted from various aspects, e.g., combing it with the swarm intelligence [35], [36], hybridizing it with local search [37], [38], and incorporating self-adaptation mechanisms [39]–[41].

The indicator-based approach employs a single performance indicator to optimize a desired property of evolutionary population. Among the current indicators available, the hypervolume [42] is probably the most popular one that is ever used in multiobjective search. This is mainly due to its good theoretical properties. It has been indicated that maximizing the hypervolume indicator is equivalent to finding the PF [43]. Nowadays, some hypervolume-based MOEAs have been well-established, such as indicator-based evolutionary algorithm [44], S metric selection evolutionary algorithm [45], and multiobjective covariance matrix adaptation evolution strategy [46]. Nevertheless, the computation cost of the hypevolume grows exponentially with the number of objectives [47], generally inhibiting the use of these algorithms for problems having more than five objectives [48]. To relieve this issue, researchers have been trying to calculate exactly the hypervolume in more efficient ways [49]–[51]. But these ways are still not efficient enough to satisfy the requirement of hypervolume-based MOEAs when solving many-objective problems. Bader and Zitzler [52] suggested the hypervolume estimation (HypE) algorithm for multiobjective optimization, where the exact hypervolume values are approximated by using Monte Carlo simulation. Since the exact calculation of hypervolume is avoided in HypE, it renders the hypervolume-based search possible in high-dimensional objective space to some extent. Very recently, several other performance indicators, such as $R2$ [53]–[55] and the additive approximation [56], [57], have also shown potentials in guiding the many-objective search.

The reference point-based MOEAs are another class of approaches that deserve to be highlighted in many-objective optimization. Slightly different from decomposition-based MOEAs, they perform the predefined multiple targeted search by means of multiple predefined reference points instead of multiple search directions, which can effectively alleviate several difficulties in handling many objectives [29]. Figueira *et al.* [58] proposed a multiple reference point approach that can be divided into two consecutive phases. The first phase is the preparation phase and it is devoted to estimate the bounds of the PF, generate multiple reference points, and design a version of the solver for each reference point. The second phase is the running phase and it launches a solver for every reference point in every processor. Moen *et al.* [59] presented a taxi-cab surface evolutionary algorithm, where the Manhattan distance is adopted as the basis for generating the attraction points and as the single metric for selecting solutions for the next generation. Some other studies with this regard can be referred in [29] and [60]–[62]. In particular, the recently proposed NSGA-III also falls into this class.

Despite a number of recent achievements in this field, the research on evolutionary many-objective optimization is far from being fully explored. For the algorithm design, the existing state-of-the-art MOEAs specially for many-objective problems are still not powerful enough [23], [63], and the need for more effective algorithms is pressing. Moreover, although various evolutionary many-objective approaches have been proposed, there are few comparative studies of different methods available to date. In [63], eight evolutionary algorithms were compared for many-objective optimization, but the comparison was restricted to problems with only two kinds of objective space dimensions, and the latest algorithms, such as NSGA-III [29], were not investigated in their work.

Our main contribution is twofold. First of all, a simple but effective θ dominance-based evolutionary algorithm (θ-DEA), is proposed for many-objective optimization. This algorithm is motivated by the strength and weakness of two recently suggested many-objective optimizers (NSGA-III and MOEA/D). NSGA-III emphasizes population members that are Pareto nondominated but are close to the reference line of each reference point. Nevertheless, when the number of objectives is high, the Pareto-dominance relied on by NSGA-III lacks enough selection pressure to pull the population toward PF, therefore, NSGA-III indeed stresses diversity more than convergence in such cases. MOEA/D implicitly maintains the diversity via the diverse weight vectors, and it could generally approach the PF very well by means of the aggregation

function-based selection operator, even in high-dimensional objective space. However, in MOEA/D, whether a new solution replaces an old solution or not is completely determined by their aggregation function values. In many-objective optimization, such replacement may lead to the severe loss of diversity in MOEA/D. The major reason is that, in high-dimensional objective space, it is highly possible that a solution achieves a good aggregation function value but is far away from the corresponding weight vector. Thus, MOEA/D is at high risk of missing some search regions if aggregation function values are emphasized too much. The issue on the loss of diversity in MOEA/D has been experimentally observed in several recent studies on many-objective optimization [23], [29], [62], [63].

We aim to improve the convergence of NSGA-III in many-objective optimization by exploiting the fitness evaluation scheme in MOEA/D, but still inherit the strength of NSGA-III in preserving the diversity. To this end, a new dominance relation, referred to as θ dominance, is introduced in this paper. In θ dominance, solutions are allocated into different clusters represented by well-distributed reference points. Only the solutions within the same cluster have the competitive relationship, where a fitness function similar to penalty-based boundary intersection (PBI) function [33] is carefully defined. When conducting the environmental selection in θ-DEA, the nondominated sorting scheme [64] based on θ dominance not only prefers solutions with better fitness values in each cluster, but also ensures that the selected solutions distribute as evenly as possible between these clusters.

The second contribution of this paper lies in the experimental aspect. We provide an extensive comparison between the proposed θ-DEA with eight state-of-the-art algorithms on 80 instances of 16 test problems taken from two well-known test suites. The results indicate that θ-DEA is a very promising algorithm for many-objective optimization.

The remainder of this paper is organized as follows. Section II introduces the background knowledge of this paper. The proposed θ-DEA is described in detail in Section III. Section IV presents the test problems, quality indicators, and algorithm settings used for performance comparison. Section V provides the extensive experimental results and discussion. Finally, the conclusion is drawn in Section VI.

II. Preliminaries

In this section, some basic definitions in multiobjective optimization are first given. Then, we will briefly introduce the original MOEA/D and NSGA-III, which are the basis of our proposed algorithm.

A. Basic Definitions

The multiobjective optimization problem (MOP) can be mathematically defined as

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

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^{\mathrm{T}}$ is a n-dimensional decision variable vector from the decision space Ω; $\mathbf{f} : \Omega \rightarrow \Theta \subseteq \mathbb{R}^m$ consists a set of m objective functions, and is a mapping from n-dimensional decision space Ω to m-dimensional objective space Θ.

Definition 1: Given two decision vectors $\mathbf{x}, \mathbf{y} \in \Omega$, $\mathbf{x}$ is said to Pareto dominate $\mathbf{y}$, denoted by $\mathbf{x} \prec \mathbf{y}$, iff $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$, for every $i \in \{1, 2, \ldots, m\}$, and $f_j(\mathbf{x}) < f_j(\mathbf{y})$, for at least one index $j \in \{1, 2, \ldots, m\}$.

Definition 2: A decision vector $\mathbf{x}^* \in \Omega$ is Pareto optimal iff there is no $\mathbf{x} \in \Omega$ such that $\mathbf{x} \prec \mathbf{x}^*$.

Definition 3: The Pareto set (PS) is defined as

$$\mathrm{PS} = \{\mathbf{x} \in \Omega | \mathbf{x} \text{ is Pareto optimal}\}. \tag{2}$$

Definition 4: The PF is defined as

$$\mathrm{PF} = \left\{\mathbf{f}(\mathbf{x}) \in \mathbb{R}^m | \mathbf{x} \in \mathrm{PS}\right\}. \tag{3}$$

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

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

The goal of MOEAs is to move the nondominated objective vectors toward PF (convergence), and also generate a good distribution of these vectors over the PF (diversity).

B. MOEA/D

The key idea of MOEA/D is to decompose an MOP into a number of single-objective optimization subproblems through aggregation functions. It aims to optimize these subproblems in parallel instead of trying to directly approximate the true PF. This mechanism works since the optimal solution to each subproblem is indeed a Pareto optimal solution to the given MOP. And the collection of these optimal solutions can be viewed as an approximation of the true PF. Generally, three aggregation functions, weighted sum, Chebyshev, and PBI function can well serve the purpose in MOEA/D. Just take the Chebyshev as an example, let $\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_N$ be a set of evenly spread weight vectors, then an MOP can be decomposed into N single-objective subproblems represented as

$$g_j^{\mathrm{te}}\left(\mathbf{x}|\boldsymbol{\lambda}_j, \mathbf{z}^*\right) = \max_{k=1}^{m}\left\{\lambda_{j,k}\left|f_k(\mathbf{x}) - z_k^*\right|\right\} \tag{4}$$

where $j = 1, 2, \ldots, N$ and $\boldsymbol{\lambda}_j = (\lambda_{j,1}, \lambda_{j,2}, \ldots, \lambda_{j,m})^{\mathrm{T}}$.

In MOEA/D, for each vector $\boldsymbol{\lambda}_j$, a set $B(j) = \{j_1, j_2, \ldots, j_T\}$ is computed in the initialization phase, where $\{\boldsymbol{\lambda}_{j_1}, \boldsymbol{\lambda}_{j_2}, \ldots, \boldsymbol{\lambda}_{j_T}\}$ is a set of T closest weight vectors to $\boldsymbol{\lambda}_j$ according to the Euclidean distance and is also called the neighborhood of $\boldsymbol{\lambda}_j$. The neighborhood of the jth subproblem contains all the subproblems with weight vectors from the neighborhood of $\boldsymbol{\lambda}_j$. At each generation of MOEA/D, a population of N solutions $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ are maintained, where $\mathbf{x}_j$ is the current solution to the jth subproblem. One of the feature of MOEA/D is that the mating restriction is adopted in the reproduction phase. When producing the jth offspring, two indexes k and l are randomly selected from $B(j)$, and a new solution $\mathbf{y}$ is generated from $\mathbf{x}_k$ and $\mathbf{x}_l$ by using genetic operators. The local replacement is another feature of MOEA/D. That is once

$\mathbf{y}$ is obtained, it will be compared with each neighboring solution $\mathbf{x}_u, u \in B(j)$, and $\mathbf{x}_u$ is to be replaced by $\mathbf{y}$, if and only if $g^{\text{te}}(\mathbf{y}|\boldsymbol{\lambda}_u, \mathbf{z}^*) < g^{\text{te}}(\mathbf{x}_u|\boldsymbol{\lambda}_u, \mathbf{z}^*)$. After all the N offsprings are produced in this manner one after another, a new population is formed, and the above procedure is repeated until the stopping criterion is met. Further details of MOEA/D can be referred in [33] and [34].

C. NSGA-III

The basic framework of NSGA-III remains similar to the established NSGA-II [9] with significant changes in its selection mechanism. The main procedure of NSGA-III can be briefly described below.

NSGA-III starts with the definition of a set of reference points. Then an initial population with N members is randomly generated, where N is the population size. The next steps are iterated until the termination criterion is satisfied. At the tth generation, the current parent population P_t is used to produce an offspring population Q_t by using random selection, simulated binary crossover (SBX) operator and polynomial mutation [65]. The size of P_t and Q_t are both N. Thereafter, the two populations P_t and Q_t are merged together to form a new population $R_t = P_t \cup Q_t$ (of size $2N$). To choose the best N members from R_t for the next generation, the nondominated sorting based on Pareto dominance is first used, which classifies R_t into different nondomination levels (F_1, F_2, and so on). Then, a new population S_t is constructed by filling members of different nondomination levels one at a time, starting from F_1, until the size of S_t is equal to N or for the first time becomes greater than N. Let us suppose that the last level included is the lth level. Hence, the solutions from the level $l+1$ onward are simply rejected. Members in $S_t \setminus F_l$ are already chosen for P_{t+1}, and the remaining population slots are chosen from F_l such that a desired diversity is maintained in the population. In the original NSGA-II, the solutions in F_l with the largest crowding distance values are selected. However, the crowding distance measure does not perform well on many-objective problems [66]. Thus, NSGA-III uses a new selection mechanism that conducts a more systematic analysis of members in S_t with respect to the supplied reference points.

To achieve this, objective values and supplied reference points are first normalized so that they have an identical range. After normalization, the ideal point of the set S_t is the zero vector. Thereafter, the perpendicular distance between a member in S_t and each of the reference lines (joining the ideal point with a reference point) is calculated. Each member in S_t is then associated with a reference point having the minimum perpendicular distance. Next, the niche count ρ_j for the jth reference point, defined as the number of members in $S_t \setminus F_l$ that are associated with the jth reference point, is computed for further processing. Now, a niche-preservation operation is executed to select members from F_l, and it works as follows. First, the reference point set $J_{\min} = \{j : \text{argmin}_j \rho_j\}$ having the minimum ρ_j value is identified. In case of $|J_{\min}| > 1$, one $\bar{j} \in J_{\min}$ is randomly chosen. If the level F_l does not have any member associated with the $\bar{j}$th reference point, the reference point is excluded from further consideration for the current generation, meanwhile, $J_{\min}$ is recomputed and $\bar{j}$ is reselected. Otherwise, the value of $\rho_{\bar{j}}$ is further considered. If $\rho_{\bar{j}} = 0$, we choose the one having the shortest perpendicular distance to the $\bar{j}$th reference line among members associated with the $\bar{j}$th reference point in F_l, and add it to P_{t+1}. The count of $\rho_{\bar{j}}$ is then increased by one. In the event of $\rho_{\bar{j}} \geq 1$, a randomly chosen member from level F_l that is associated with the $\bar{j}$th reference point is added to P_{t+1}, and the count of $\rho_{\bar{j}}$ also needs to be increased by one. The above niche operation is repeated until the remaining population slots of P_{t+1} are filled. For more details of NSGA-III, please refer to [29].

III. Proposed Algorithm: θ-DEA

A. Overview

The framework of the proposed θ-DEA is described in Algorithm 1. First, a set of N reference points are generated, which can be denoted as $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_1, \boldsymbol{\lambda}_2, \ldots, \boldsymbol{\lambda}_N\}$. For a m-objective problem, $\boldsymbol{\lambda}_j$ ($j \in \{1, 2, \ldots, N\}$) is a m-dimensional vector represented by $\boldsymbol{\lambda}_j = (\lambda_{j,1}, \lambda_{j,2}, \ldots, \lambda_{j,m})^{\text{T}}$, where $\lambda_{j,k} \geq 0, k = 1, 2, \ldots, m$, and $\sum_{k=1}^{m} \lambda_{j,k} = 1$. Next, the initial population P_0 with N members is randomly produced. The ideal point $\mathbf{z}^*$ is initialized in step 3. Since it is often very time consuming to compute exact z_i^*, it is indeed estimated by the minimum value found so far for objective f_i, and is updated during the search. The nadir point $\mathbf{z}^{\text{nad}}$ is initialized in step 4, where z_i^{nad} is assigned to the largest value of f_i found in P_0, and it is updated in the normalization procedure. Steps 6–23 are iterated until the termination criterion is satisfied. In step 7, the offspring population Q_t is produced by using the recombination operator. Then Q_t is combined with the current population P_t, and form a new population R_t. The population $S_t = \cup_{i=1}^{\tau} F_i$, where F_i is the ith Pareto nondomination level of R_t and τ satisfies $\sum_{i=1}^{\tau-1} |F_i| < N$ and $\sum_{i=1}^{\tau} |F_i| \geq N$. In fact, for problems having high number of objectives, S_t is almost always equal to F_1 since there is a large fraction of Pareto nondominated solutions in the population. In step 11, the normalization procedure is executed to S_t assisted by $\mathbf{z}^*$ and $\mathbf{z}^{\text{nad}}$. After normalization, the clustering operator is used to split the members in S_t into a set of N clusters $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$, where the cluster C_j is represented by the reference point $\boldsymbol{\lambda}_j$. Then, the nondominated sorting based on θ dominance (not Pareto dominance) is employed to classify S_t into different θ-nondomination levels (F_1', F_2', and so on). θ dominance, which is the key concept in θ-DEA, would be introduced later. Once θ-nondominated sorting has been finished, the remaining steps fill the population slots in P_{t+1} using one level at a time, starting from F_1'. Different from both NSGA-II and NSGA-III, we just randomly select solutions in the last accepted level F_l in θ-DEA, because θ dominance has stressed both convergence and diversity. Certainly, some strategies to enhance the diversity could also be used as well in step 20. In the following sections, the important procedures of θ-DEA are to be described in detail.

B. Reference Points Generation

To promote diversity in the obtained solutions, Das and Dennis's [67] systematic approach is adopted to generate structured reference points in θ-DEA. The same

Algorithm 1 Framework of the Proposed θ-DEA

$\mathbf{\Lambda} \leftarrow$ GenerateReferencePoints()
$P_0 \leftarrow$ InitializePopulation()
$\mathbf{z}^* \leftarrow$ InitializeIdealPoint()
$\mathbf{z}^{\text{nad}} \leftarrow$ InitializeNadirPoint()
$t \leftarrow 0$
while the termination criterion is not met **do**
  $Q_t \leftarrow$ CreateOffspringPopulation(P_t)
  $R_t \leftarrow P_t \cup Q_t$
  $S_t \leftarrow$ GetParetoNondominationLevels(R_t)
  UpdateIdealPoint(S_t)
  Normalize(S_t, $\mathbf{z}^*$, $\mathbf{z}^{\text{nad}}$)
  $\mathcal{C} \leftarrow$ Clustering(S_t, $\mathbf{\Lambda}$)
  $\{F'_1, F'_2, \ldots\} \leftarrow \theta$-Nondominated-sort($S_t$, $\mathcal{C}$)
  $P_{t+1} \leftarrow \emptyset$
  $i \leftarrow 1$
  while $|P_{t+1}| + |F'_i| < N$ **do**
    $P_{t+1} \leftarrow P_{t+1} \cup F'_i$
    $i \leftarrow i + 1$
  end while
  RandomSort(F'_i)
  $P_{t+1} \leftarrow P_{t+1} \cup F'_i[1 : (N - |P_{t+1}|)]$
  $t \leftarrow t + 1$
end while

mechanism is also used in MOEA/D, NSGA-III, and some earlier algorithms [68]. The number of reference points produced in this way depends on the dimension of objective space m and another positive integer H. Let us consider

$$\sum_{i=1}^{m} x_i = H, \quad x_i \in \mathbb{N}, \quad i = 1, 2, \ldots, m. \tag{5}$$

The number of solutions to (5) can be calculated as

$$N = \binom{H + m - 1}{m - 1}. \tag{6}$$

Suppose that $(x_{j,1}, x_{j,2}, \ldots, x_{j,m})^{\text{T}}$ is the jth solution. Then, the reference point $\mathbf{\lambda}_j$ is obtained, where

$$\lambda_{j,k} = \frac{x_{j,k}}{H}, \quad k = 1, 2, \ldots, m. \tag{7}$$

Geometrically, $\mathbf{\lambda}_1, \mathbf{\lambda}_2, \ldots, \mathbf{\lambda}_N$ are all located at the hyperplane $\sum_{i=1}^{m} f_i = 1$, and H is the divisions considered along each objective axis.

Note that, if $H < m$, no intermediate reference point is created by this approach. However, when m is relatively larger, e.g., $m = 8$, $H \geq m$ would lead to a huge number of reference points, and hence a huge population size. To address this issue, in the proposed θ-DEA, we use two-layered reference points with small values of H as suggested in [29]. Suppose the divisions of boundary and inner layers is H_1 and H_2, respectively, then the population size

$$N = \binom{H_1 + m - 1}{m - 1} + \binom{H_2 + m - 1}{m - 1}. \tag{8}$$

Fig. 1 illustrates the distribution of two-layered reference points using a three-objective problem with $H_1 = 2$ and $H_2 = 1$.

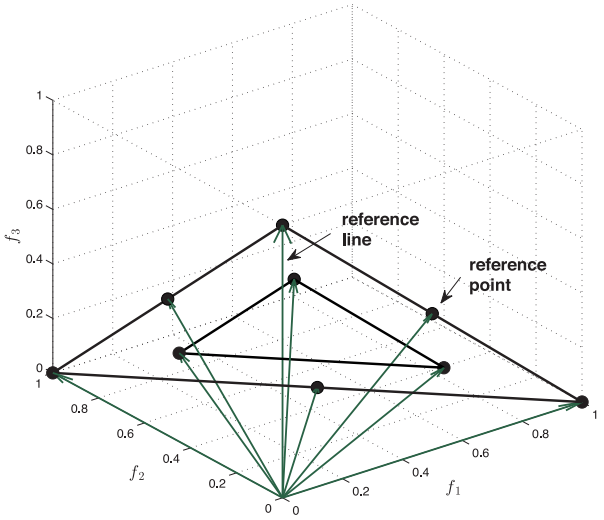

Fig. 1. Distribution of two-layered reference points in the 3-D objective space. Six points on the boundary layer ($H_1 = 2$) and three points on the inner layer ($H_2 = 1$).

C. Recombination Operator

The recombination operator may be ineffective in many-objective optimization. This is mainly because, in high-dimensional objective space, there is a greater probability that solutions widely distant from each other will be selected to recombine and generate poorer performance offspring solutions, known as lethals [26], [28]. There are normally two ways to address this issue. One is to use the mating restriction scheme [69], in which two neighboring solutions are involved in the recombination operator, such as in MOEA/D. The other is to use a special recombination scheme (e.g., SBX operator with a large distribution index) [29], where near-parent solutions are emphasized, such as in NSGA-III.

θ-DEA employs the latter since it has a similar algorithm structure to NSGA-III. When performing the recombination, two parent solutions are randomly selected from the current population P_t, then the child solution is created by using the SBX operator with a large distribution index and polynomial mutation.

D. Adaptive Normalization

The normalization procedure is incorporated into θ-DEA for solving problems having the PF whose objective values may be disparately scaled. In normalization, the objective $f_i(\mathbf{x})$, $i = 1, 2, \ldots, m$, can be replaced by

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

As mentioned before, z_i^* can be estimated by the best value found so far for objective f_i. However, the estimation of z_i^{nad} is a much more difficult task since it requires information about the whole PF [70], [71]. The procedure of estimating z_i^{nad} in θ-DEA is similar to that in NSGA-III but different in the identification of extreme points.

First, in the population to be normalized, i.e., S_t, the extreme point $\mathbf{e}_j$ in the objective axis f_j is identified by finding the solution $\mathbf{x} \in S_t$ that minimizes the following achievement

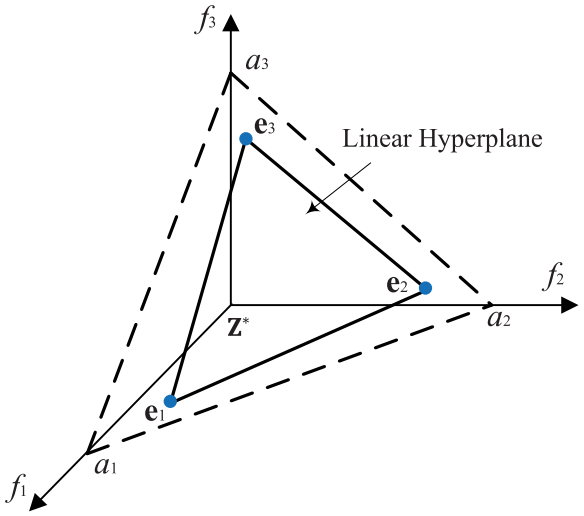

Fig. 2. Illustration of constructing linear hyperplane by extreme points and computing intercepts in 3-D objective space.

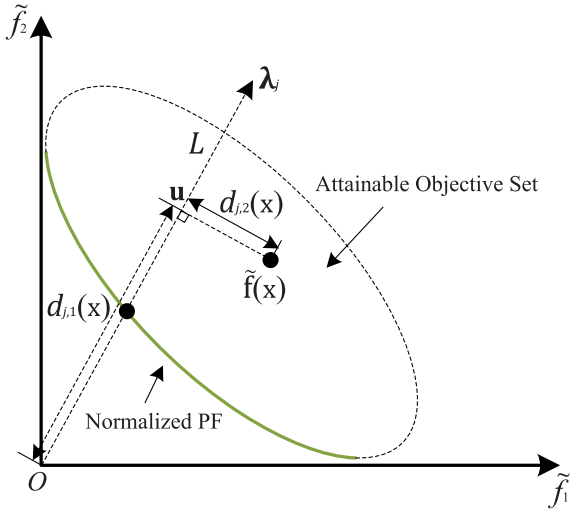

Fig. 3. Illustration of distances $d_{j,1}(\mathbf{x})$ and $d_{j,2}(\mathbf{x})$.

scalarizing function:

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

In (10), $\mathbf{w}_j = (w_{j,1}, w_{j,2}, \ldots, w_{j,m})^{\mathrm{T}}$ is the axis direction of the objective axis f_j and satisfies that if $i \neq j$ $w_{j,i} = 0$, else $w_{j,i} = 1$; for $w_{j,i} = 0$, we replace it with a small number 10^{-6}; z_i^{nad} is the ith dimension of nadir point estimated in the previous one generation. The extreme point $\mathbf{e}_j$ is eventually assigned the objective vector of the found solution $\mathbf{x}$, i.e., $\mathbf{e}_j = \mathbf{f}(\mathbf{x})$.

After all the m objective axes have been considered, we can obtain m extreme points $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m$. The m extreme points are then used to construct a m-dimensional linear hyperplane. Let $a_1, a_2, \ldots, a_m$ denote the intercepts of the hyperplane with the directions $(1, z_2^*, \ldots, z_m^*)^{\mathrm{T}}$, $(z_1^*, 1, \ldots, z_m^*)^{\mathrm{T}}, \ldots,$ $(z_1^*, \ldots, z_{m-1}^*, 1)^{\mathrm{T}}$, respectively. Supposing the matrix $\mathbf{E} = (\mathbf{e}_1 - \mathbf{z}^*, \mathbf{e}_2 - \mathbf{z}^*, \ldots, \mathbf{e}_m - \mathbf{z}^*)^{\mathrm{T}}$ and $\mathbf{u} = (1, 1, \ldots, 1)^{\mathrm{T}}$, the intercepts can be computed by

$$\begin{pmatrix} (a_1 - z_1^*)^{-1} \\ (a_2 - z_2^*)^{-1} \\ \cdots \\ (a_m - z_m^*)^{-1} \end{pmatrix} = \mathbf{E}^{-1}\mathbf{u}. \tag{11}$$

Thereafter, the value of z_i^{nad} is updated as a_i, where $i = 1, 2, \ldots, m$, and the population S_t can be normalized using (9). In Fig. 2, we demonstrate the hyperplane construction and the intercepts formation in 3-D objective space.

Note that, if the rank of matrix $\mathbf{E}$ is less than m, the m extreme points will fail to constitute a m-dimensional hyperplane. And even with the hyperplane built, it is also likely to obtain no intercepts in certain directions or some intercepts a_i do not satisfy $a_i > z_i^*$. In all the above cases, z_i^{nad} is assigned to the largest value of f_i in the nondominated solutions of S_t, for each $i \in \{1, 2, \ldots, m\}$.

E. Clustering Operator

In θ-DEA, the clustering operator is applied to population S_t at each generation. The clustering works in the normalized objective space, where the ideal point is the origin. Suppose that $\tilde{\mathbf{f}}(\mathbf{x}) = (\tilde{f}_1(\mathbf{x}), \tilde{f}_2(\mathbf{x}), \ldots, \tilde{f}_m(\mathbf{x}))^{\mathrm{T}}$ is the normalized objective vector for the solution $\mathbf{x}$, L is a line passing through the origin with the direction $\boldsymbol{\lambda}_j$, and $\mathbf{u}$ is the projection of $\tilde{\mathbf{f}}(\mathbf{x})$ on L. Let $d_{j,1}(\mathbf{x})$ be the distance between the origin and $\mathbf{u}$, and $d_{j,2}(\mathbf{x})$ be the perpendicular distance between $\tilde{\mathbf{f}}(\mathbf{x})$ and L. They can be computed, respectively, as

$$d_{j,1}(\mathbf{x}) = \left\| \tilde{\mathbf{f}}(\mathbf{x})^{\mathrm{T}} \boldsymbol{\lambda}_j \right\| / \left\| \boldsymbol{\lambda}_j \right\| \tag{12}$$

$$d_{j,2}(\mathbf{x}) = \left\| \tilde{\mathbf{f}}(\mathbf{x}) - d_{j,1}(\mathbf{x})(\boldsymbol{\lambda}_j / \left\| \boldsymbol{\lambda}_j \right\|) \right\|. \tag{13}$$

In Fig. 3, the distances $d_{j,1}(\mathbf{x})$ and $d_{j,2}(\mathbf{x})$ are illustrated in the 2-D objective space.

For the clustering operator, only $d_{j,2}$ will be considered, $d_{j,1}$ will be involved later in the definition of θ dominance. We assign a solution $\mathbf{x}$ to the cluster C_j with the minimum $d_{j,2}(\mathbf{x})$ value. The details of the clustering process are shown in Algorithm 2.

Algorithm 2 Clustering $(S_t, \boldsymbol{\Lambda})$

```
{C_1, C_2, ..., C_N} ← {∅, ∅, ..., ∅}
for each solution x in S_t do
    n ← 1
    min ← d_{1,2}(x)
    for j ← 2 to N do
        if d_{j,2}(x) < min then
            min ← d_{j,2}(x)
            n ← j
        end if
    end for
    C_n ← C_n ∪ {x}
end for
```

F. θ Dominance

The proposed θ dominance is defined on population S_t with the supply of a set of reference points $\boldsymbol{\Lambda}$. And each solution in S_t is associated with a cluster among a set of clusters $\mathcal{C}$

by the clustering operator. Let $\mathcal{F}_j(\mathbf{x}) = d_{j,1}(\mathbf{x}) + \theta d_{j,2}(\mathbf{x})$, $j \in \{1, 2, \ldots, N\}$, where θ is a predefined penalty parameter. The form of $\mathcal{F}_j(x)$ is the same as that of the PBI function [33]. But here, the distances $d_{j,1}$ and $d_{j,2}$ are both computed in the normalized objective space. Generally, $d_{j,2}(\mathbf{x}) = 0$ ensures that $\mathbf{f}(\mathbf{x})$ is always in L, resulting in perfect diversity, and smaller $d_{j,1}(\mathbf{x})$ value under the condition $d_{j,2}(\mathbf{x}) = 0$ means better convergence. With the definition of $\mathcal{F}_j$ and $\mathcal{C}$, several concepts related to θ dominance can be defined as follows.

Definition 7: Given two solutions $\mathbf{x}, \mathbf{y} \in S_t$, $\mathbf{x}$ is said to θ-dominate $\mathbf{y}$, denoted by $\mathbf{x} \prec_\theta \mathbf{y}$, iff $\mathbf{x} \in C_j$, $\mathbf{y} \in C_j$, and $\mathcal{F}_j(\mathbf{x}) < \mathcal{F}_j(\mathbf{y})$, where $j \in \{1, 2, \ldots, N\}$.

Definition 8: A solution $\mathbf{x}^* \in S_t$ is θ-optimal iff there is no other solution $\mathbf{x} \in S_t$ such that $\mathbf{x} \prec_\theta \mathbf{x}^*$.

Definition 9: All solutions that are θ-optimal in S_t form the θ-optimal set (θ-OS), and the corresponding mappings of θ-OS in the objective space form the θ-optimal front.

Based on the definition of θ dominance, we have the following three properties, which respectively illustrate that the relation $\prec_\theta$ is irreflexive, asymmetric, and transitive.

Property 1: If a solution $\mathbf{x} \in S_t$, then $\mathbf{x} \not\prec_\theta \mathbf{x}$.

Proof: Suppose $\mathbf{x} \prec_\theta \mathbf{x}$, then $\exists j \in \{1, 2, \ldots, N\}$, $\mathbf{x} \in C_j$, and $\mathcal{F}_j(\mathbf{x}) < \mathcal{F}_j(\mathbf{x})$. However, $\mathcal{F}_j(\mathbf{x}) = \mathcal{F}_j(\mathbf{x})$. Thus, $\mathbf{x} \not\prec_\theta \mathbf{x}$. ■

Property 2: If two solutions $\mathbf{x}, \mathbf{y} \in S_t$ satisfy $\mathbf{x} \prec_\theta \mathbf{y}$, then $\mathbf{y} \not\prec_\theta \mathbf{x}$.

Proof: Suppose $\mathbf{y} \prec_\theta \mathbf{x}$, then $\exists j \in \{1, 2, \ldots, N\}$, $\mathbf{y} \in C_j$, $\mathbf{x} \in C_j$, and $\mathcal{F}_j(\mathbf{y}) < \mathcal{F}_j(\mathbf{x})$. However, according to $\mathbf{x} \prec_\theta \mathbf{y}$, $\mathcal{F}_j(\mathbf{x}) < \mathcal{F}_j(\mathbf{y})$. So, the supposition is invalid, and the proposition is true. ■

Property 3: If three solutions $\mathbf{x}, \mathbf{y}, \mathbf{z} \in S_t$ satisfy $\mathbf{x} \prec_\theta \mathbf{y}$ and $\mathbf{y} \prec_\theta \mathbf{z}$, then $\mathbf{x} \prec_\theta \mathbf{z}$.

Proof: By $\mathbf{x} \prec_\theta \mathbf{y}$, we have that $\exists j \in \{1, 2, \ldots, N\}$, $\mathbf{x} \in C_j$, $\mathbf{y} \in C_j$, and $\mathcal{F}_j(\mathbf{x}) < \mathcal{F}_j(\mathbf{y})$. Then according to $\mathbf{y} \prec_\theta \mathbf{z}$, we have that $\mathbf{z} \in C_j$ and $\mathcal{F}_j(\mathbf{y}) < \mathcal{F}_j(\mathbf{z})$. Overall, $\mathbf{x} \in C_j$, $\mathbf{z} \in C_j$, and $\mathcal{F}_j(\mathbf{x}) < \mathcal{F}_j(\mathbf{z})$. Thus, $\mathbf{x} \prec_\theta \mathbf{z}$. ■

Due to the above three properties, the θ dominance defines a strict partial order on S_t. Hence, the fast nondominated sorting approach [9] can be immediately adopted in θ-dominance sense, and the population S_t would be partitioned into different θ-nondomination levels.

Note that, there is no competitive relationship between clusters in θ dominance, and thus it can indeed use different θ values in different clusters. To explore this characteristic, we do a bit more work on the proposed θ-DEA. In the normalized objective space, if $\boldsymbol{\lambda}_j$ is the axis direction, then we assign a large θ value ($\theta = 10^6$ is used) in the cluster C_j, otherwise assign a normal θ value. This is just to cooperate with the normalization procedure presented in Section III-D, and if normalization is disabled, it may be unnecessary. The large θ values in the clusters represented by axis directions would make θ-DEA more likely to capture the nadir point in high-dimensional objective space, and thus conduct a more stable normalization.

G. Computational Complexity of θ-DEA

The computational complexity of θ-DEA in one generation is dominated by the clustering operator that is described in Algorithm 2 under the general condition. In Algorithm 2, N distances $d_{1,2}(\mathbf{x}), d_{2,2}(\mathbf{x}), \ldots, d_{N,2}(\mathbf{x})$ need to be calculated for each solution $\mathbf{x}$ in S_t, and each distance is computed in $O(m)$ computations. Thus, totally $O(mN|S_t|)$ computations are required. Since $|S_t| \leq 2N$, the overall worst complexity of one generation of θ-DEA is approximately $O(mN^2)$.

TABLE I
SCALING FACTORS FOR SDTLZ1 AND SDTLZ2 PROBLEMS

No. of Objectives (m)	Scaling Factor	
	SDTLZ1	SDTLZ2
3	10^i	10^i
5	10^i	10^i
8	3^i	3^i
10	2^i	3^i
15	1.2^i	2^i

IV. EXPERIMENTAL DESIGN

This section is devoted to the experimental design for investigating the performance of the proposed θ-DEA. First, the test problems and the quality indicators used in our experiments are given. Then, we briefly introduce eight state-of-the-art algorithms that are employed for comparison. Finally, the experimental settings adopted in this paper are provided.

A. Test Problems

As a basis for the comparisons, two well-known test suites for many-objective optimization, Deb–Thiele–Laumanns–Zitzler (DTLZ) [72] and Walking Fish Group (WFG) [73], are involved in the experiments. To compute the quality indicators reliably, we only consider DTLZ1–4 and DTLZ7 problems for DTLZ test suite, since the nature of DTLZ5 and DTLZ6s PFs is unclear beyond three objectives [73]. Moreover, we also use two scaled test problems, i.e., scaled DTLZ1 and DTLZ2 problems [29], which are the modifications of DTLZ1 and DTLZ2 problems, respectively. To illustrate, if the scaling factor is 10^i, the objectives f_1, f_2, and f_3 for three-objective scaled DTLZ1 problem are multiplied by 10^0, 10^1, and 10^2, respectively. In our experiments, we call scaled DTLZ1 and DTLZ2 problems SDTLZ1 and SDTLZ2 for short, respectively.

All these problems can be scaled to any number of objectives and decision variables. We consider the number of objectives $m \in \{3, 5, 8, 10, 15\}$. For DTLZ1–4, DTLZ7, SDTLZ1, and SDTLZ2 problems, the total number of decision variables is given by $n = m + k - 1$, unless otherwise specified, k is set to 5 for DTLZ1 and SDTLZ1, 10 for DTLZ2–4 and SDTLZ2, and 20 for DLTZ7 as recommended in [29] and [72]. As for all WFG problems, unless otherwise stated, the number of decision variables is set to 24 and the position-related parameter is set to $m - 1$ according to [54] and [73]. The scaling factors for SDTLZ1 and SDTZL2 problems with different number of objectives are shown in Table I.

These test problems have a variety of characteristics, such as having linear, mixed (convex/concave), multimodal, disconnected, degenerate, and disparately scaled PFs, which

TABLE II
FEATURES OF THE TEST PROBLEMS

Problem	Features
DTLZ1	Linear, Multi-modal
DTLZ2	Concave
DTLZ3	Concave, Multi-modal
DTLZ4	Concave, Biased
SDTLZ1	Linear, Multi-modal, Scaled
SDTLZ2	Concave, Scaled
DTLZ7	Mixed, Disconnected, Multi-modal, Scaled
WFG1	Mixed, Biased, Scaled
WFG2	Convex, Disconnected, Multi-modal, Non-separable, Scaled
WFG3	Linear, Degenerate, Non-separable, Scaled
WFG4	Concave, Multi-modal, Scaled
WFG5	Concave, Deceptive, Scaled
WFG6	Concave, Non-separable, Scaled
WFG7	Concave, Biased, Scaled
WFG8	Concave, Biased, Non-separable, Scaled
WFG9	Concave, Biased, Multi-modal, Deceptive, Non-separable, Scaled

challenge different abilities of an algorithm. In Table II, we summarize the main features of all the adopted test problems.

B. Quality Indicators

The quality indicators are needed to evaluate the performance of the concerned algorithms. In the EMO literature, the inverted generational distance (IGD) [74] is one of the most widely used indicators, which could provide a combined information about convergence and diversity of a solution set. The calculation of IGD requires a set of uniformly distributed points along the known PF. Surely, the true PFs of most benchmark problems are known, and it is relatively easy to uniformly sample the points on the 2- or 3-D PF. However, for a high-dimensional PF, how to make sure the sampled points are uniformly distributed and how many points are enough for representing the true PF, are themselves difficult questions. In fact, numerous studies [23], [30], [63] relating to many-objective optimization, where IGD is used as the quality indicator, failed to indicate how they sampled those points along the PF.

Recently, Deb and Jain [29] suggested a way to compute IGD for MOEAs in which the reference points or reference directions are supplied, e.g., MOEA/D and NSGA-III. This way works as follows. For each reference direction $\boldsymbol{\lambda}_j$, $j = 1, 2, \ldots, N$, we can exactly locate its targeted point $\mathbf{v}_j$ on the known PF in the normalized objective space. All N targeted points constitute the set $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_N\}$. For any algorithm, let $\mathbf{A}$ be the set of final nondominated points obtained in the objective space. Then IGD is computed as

$$\mathrm{IGD}(\mathbf{A}, \mathbf{V}) = \frac{1}{|\mathbf{V}|} \sum_{i=1}^{|\mathbf{V}|} \min_{\mathbf{f} \in \mathbf{A}} d(\mathbf{v}_i, \mathbf{f}) \tag{14}$$

where $d(\mathbf{v}_i, \mathbf{f})$ is the Euclidean distance between the points $\mathbf{v}_i$ and $\mathbf{f}$. Note that, for the scaled problems, the objective values in the set $\mathbf{A}$ should be first normalized using the ideal and nadir points of the exact PF before computing IGD. The set $\mathbf{A}$ with smaller IGD values is better.

It makes sense to compute IGD using the above method for reference point/direction-based MOEAs. This is mainly because that the many-objective optimization task for these algorithms can be seen as finding the Pareto-optimal points close to the supplied reference points to some extent. Since the proposed θ-DEA is also based on reference points, we will evaluate and compare its performance using IGD defined by (14) in the experiments.

However, such IGD is not applicable to MOEAs without using reference points/directions, e.g., HypE [52] and SDE [30]. For these algorithms, the many-objective optimization task is to search for sparsely distributed Pareto-optimal points over the entire PF [75]. In this scenario, another popular indicator, i.e., hypervolume [42], is adopted to evaluate the performance. The hypervolume is strict Pareto-compliant [74], whose nice theoretical qualities make it a rather fair indicator. Let $\mathbf{A}$ be the set of final nondominated points obtained in the objective space by an algorithm, and $\mathbf{r} = (r_1, r_2, \ldots, r_m)^{\mathrm{T}}$ be a reference point in the objective space which is dominated by any point in the set $\mathbf{A}$. Then the hypervolume indicator value of $\mathbf{A}$ with regard to $\mathbf{r}$ is the volume of the region dominated by $\mathbf{A}$ and bounded by $\mathbf{r}$, and can be described as

$$\mathrm{HV}(\mathbf{A}, \mathbf{r}) = \mathrm{volume}\left(\bigcup_{\mathbf{f} \in \mathbf{A}} [f_1, r_1] \times \ldots [f_m, r_m]\right). \tag{15}$$

HV can measure both convergence and diversity of a solution set in a sense. Given a reference point $\mathbf{r}$, larger HV value means better quality. In the calculation of HV, the choice of reference point is a crucial issue. It has been found that choosing r_i that is slightly larger than z_i^{nad} is suitable since the balance between convergence and diversity of the solution set is well emphasized [76], [77]. In our experiments, we set $\mathbf{r}$ to $1.1\mathbf{z}^{\mathrm{nad}}$, where $\mathbf{z}^{\mathrm{nad}}$ can be analytically obtained for all the adopted test problems. Following the practice in [2] and [22], the points that do not dominate reference point are discarded for the HV calculation. Considering the problems that have PFs with differently scaled objective values, we first normalize the objective values of points in $\mathbf{A}$ and the reference point $\mathbf{r}$ using $\mathbf{z}^{\mathrm{nad}}$ and $\mathbf{z}^*$ ($\mathbf{z}^*$ is $\mathbf{0}$ for all the adopted test problems) prior to computing HV by (15). Thus, the presented HV value for a m-objective instance in the experiments is between 0 and $1.1^m - V_m$, where V_m is the hypervolume of the region enclosed by the exact normalized PF and the coordinate axes. In addition, for problems with no more than ten objectives, we calculate HV exactly using the recently proposed WFG algorithm [50]. As for problems having 15 objectives, we approximate the HV by the Monte Carlo simulation proposed in [52], and 10 000 000 sampling points are used to ensure the accuracy.

C. Other Algorithms in Comparison

To verify the proposed θ-DEA, the following eight state-of-the-art algorithms are considered as the peer algorithms.

1) *Grid based evolutionary algorithm (GrEA) [23]:* It exploits the potential of a grid to strengthen the selection pressure toward the PF while maintaining an extensive and uniform distribution of solutions. To this end, two

concepts (i.e., grid dominance and grid difference), three grid-based criteria, (i.e., grid ranking, grid crowding distance, and grid coordinate point distance), and a fitness adjustment strategy are incorporated into GrEA.

2) *Preference ordering genetic algorithm (POGA) [20]:* It uses the preference order-based approach as an optimality criterion in the ranking stage of MOEAs. By exploiting the definition of efficiency of order in the subsets of objectives, a ranking procedure is employed within the framework of NSGA-II, which exerts the higher selection pressure over objective spaces of different dimensionality compared with the traditional Pareto dominance-based ranking scheme.
3) *NSGA-III [29]:* It is a reference point-based evolutionary algorithm for many-objective optimization, whose framework is still similar to the original NSGA-II. But unlike NSGA-II, the maintenance of diversity among population members is aided by providing and adaptively updating a number of well-distributed reference points. Overall, NSGA-III emphasizes population members which are nondominated yet close to a set of supplied reference points.
4) *SDE [30]:* It is a general modification of the density estimation strategies that could make the Pareto dominance-based MOEAs suitable for many-objective optimization. Its basic idea is that, given the preference of density estimators for solutions in sparse regions, the solutions with poor convergence are put into crowded regions by SDE, so that they will be assigned a high density value and then be eliminated easily during the evolutionary process. SDE is simple in implementation and can be applied to any specific density estimator with negligible computational cost and no additional parameters. In this paper, the version that integrates SDE into SPEA2 (SPEA2+SDE) is used, since it shows the best overall performance among all the three considered versions (NSGA-II+SDE, SPEA2+SDE, and PESA-II+SDE) in [30].
5) *MOEA/D [33]:* It is a representative algorithm that belongs to the decomposition-based approach. In [34], a new version of MOEA/D (MOEA/D-DE) based on differential evolution (DE) [78] was proposed to deal with problems with complicated PSs. In this paper, the original MOEA/D with the PBI function is selected, since it has been reported in [29] that MOEA/D-DE shows poor performance on many-objective problems and PBI is more suitable for solving problems having a high-dimensional objective space.
6) *Decomposition-based multiobjective particle swarm optimizer (dMOPSO) [35]:* It is an MOEA that extends the particle swarm optimization [79] technique to the decomposition-based multiobjective approach. It updates the position of each particle using a set of solutions considered as the global best according to the decomposition-based approach. And it is mainly characterized by the use of a memory reinitialization process aiming to provide diversity to the swarm. Similar to MOEA/D, the PBI function is also chosen in dMOPSO.

TABLE III
SUMMARY OF THE ALGORITHMS EMPLOYED IN COMPARISON

Algorithm Name	Algorithm Category	Reference
GrEA	C1	Yang *et al.* (2013) [23]
POGA	C1	Pierro *et al.* (2007) [20]
NSGA-III	C2, C5	Deb and Himanshu (2014) [29]
SDE	C2	Li *et al.* (2014) [30]
MOEA/D	C3	Zhang and Li (2007) [33]
dMOPSO	C3	Martínez and Coello (2011) [35]
HypE	C4	Bader and Zitzler (2011) [52]
MOMBI	C4	Gómez and Coello (2013) [54]

C1: Adoption of new preference relations;
C2: Adoption of new diversity promotion mechanisms;
C3: Decomposition-based approach;
C4: Indicator-based approach;
C5: Reference-point based approach.

7) *HypE [52]:* It is a hypervolume-based evolutionary algorithm for many-objective optimization, which adopts Monte Carlo simulation to approximate the exact hypervolume values. Its core idea is that only the rankings of the solutions induced by the hypervolume indicator are important, while the actual indicator values are not. HypE makes a tradeoff between the accuracy of the estimates and the available computing resources, making the hypervolume-based search more easily applied to many-objective problems.
8) *Many-objective metaheuristic based on the R2 indicator (MOMBI) [54]:* It is a many-objective metaheuristic based on the $R2$ indicator. To use $R2$ in the selection mechanism, a nondominated scheme based on the adopted utility functions is designed for that purpose. Its main idea is that the solutions which achieve the best values on any of the chosen utility functions are given the first rank. Such solutions are then removed and a second rank will be identified in the same manner. The process will continue until all the solutions have been ranked.

These eight algorithms have covered most categories of techniques mentioned in Section I for many-objective optimization. Table III summarizes the list of selected algorithms for comparison. All the concerned algorithms, including the proposed θ-DEA are implemented in the jMetal framework [80], and run on an Intel 2.83 GHz Xeon processor with 15.9 Gb of RAM. For all the algorithms except SDE, the obtained final population is used for computing quality indicators, whereas for SDE, the final archive is used.

D. Experimental Settings

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

1) *Number of Runs:* Each algorithm is run 20 times independently for each test instance.
2) *Termination Criterion:* The termination criterion of an algorithm for each run is specified in the form of the maximum number of generations (MaxGen). Since the used test problems are of varying computational complexity, we use different MaxGen for different problems.

TABLE IV
SETTING OF THE POPULATION SIZE

No.of Objectives (m)	Divisions (H)	Population Size (N)
3	12	91
5	6	210
8	3, 2	156
10	3, 2	275
15	2, 1	135

3) *Significance Test:* To test the difference for statistical significance in some cases, the Wilcoxon signed-rank test [81] at a 5% significance level is carried out on the assessment results obtained by two competing algorithms.

As for the parameter settings, several common settings for algorithms are first given as follows.

1) *Population Size:* The setting of population size N for NSGA-III, MOEA/D, and θ-DEA cannot be arbitrarily specified, where N is controlled by a parameter H [see (6)]. Moreover, as presented in Section III-B, we use two-layered reference points for problems having 8, 10, and 15 objectives to create intermediate reference points. For the other algorithms, the population size can be set to any positive integer, but for ensuring a fair comparison, the same population size is adopted. Table IV lists the population sizes used in this paper for the problem with different number of objectives. Noting that, POGA, HypE, MOMBI, and θ-DEA have similar framework to that of NSGA-III, we slightly adjust the population size of these algorithms to the multiple of four just as in the original NSGA-III study [29], i.e., 92, 212, 276, and 136 for 3-, 5-, 10-, and 15-objective problems.
2) *Penalty Parameter θ:* Since PBI function is involved in MOEA/D, dMOPSO, and θ-DEA, the penalty parameter θ needs to be set for them. In this paper, θ is just set to 5 for both MOEA/D and dMOPSO as suggested in [33], whereas some studies [75], [82] have indicated that the good specification of θ for MOEA/D may depend on the problem to be solved and its number of objectives. For the proposed θ-DEA, θ is also set to 5, but we will investigate the influence of θ on the performance of the proposed θ-DEA in Section V-C.
3) *Parameters for Crossover and Mutation:* The SBX and polynomial mutation [65] are used in all the considered algorithms except dMOPSO. For GrEA, POGA, SDE, MOEA/D, HypE, MOMBI, the parameter values for crossover and mutation are presented in Table V. As for NSGA-III and θ-DEA, the settings are only a bit different according to [29], where η_c is set to 30.

Besides the parameters mentioned above, GrEA, SDE, MOEA/D, dMOPSO, and HypE have their specific parameters. These parameters are set mainly according to the suggestions given by their developers, which are shown below.

1) *Parameter Setting in GrEA:* The grid division (div) needs to be set. Since the population size and the termination criterion are quite different from the original GrEA study [23], we adjust div according to the guidelines provided in [23] for each problem instance as shown in Table VI, aiming to well balance the convergence and diversity.
2) *Parameter Setting in SDE:* The archive size is just set as the same as the population size.
3) *Parameter Setting in MOEA/D:* The neighborhood size T is set to 20.
4) *Parameter Setting in dMOPSO:* The age threshold T_a is set to 2.
5) *Parameter Setting in HypE:* The bound of the reference point is set to 200, and the number of sampling points M is set to 10 000.

TABLE V
PARAMETER SETTINGS FOR CROSSOVER AND MUTATION

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

TABLE VI
SETTING OF GRID DIVISION IN GREA

Problem	No.of Objectives (m)	Grid Division (div)
DTLZ1	3, 5, 8, 10, 15	14, 17, 12, 17, 28
DTLZ2	3, 5, 8, 10, 15	15, 12, 12, 12, 12
DTLZ3	3, 5, 8, 10, 15	17, 19, 19, 19, 33
DTLZ4	3, 5, 8, 10, 15	16, 11, 12, 17, 18
DTLZ7	3, 5, 8, 10, 15	16, 12, 9, 11, 10
SDTLZ1	3, 5, 8, 10, 15	16, 16, 11, 17, 29
SDTLZ2	3, 5, 8, 10, 15	16, 10, 12, 15, 12
WFG1	3, 5, 8, 10, 15	6, 14, 13, 13, 13
WFG2	3, 5, 8, 10, 15	18, 16, 15, 19, 17
WFG3	3, 5, 8, 10, 15	16, 14, 12, 14, 16
WFG4	3, 5, 8, 10, 15	10, 13, 10, 11, 11
WFG5	3, 5, 8, 10, 15	10, 11, 12, 14, 14
WFG6	3, 5, 8, 10, 15	10, 12, 12, 14, 15
WFG7	3, 5, 8, 10, 15	10, 11, 12, 14, 14
WFG8	3, 5, 8, 10, 15	10, 12, 10, 14, 11
WFG9	3, 5, 8, 10, 15	9, 12, 12, 15, 13

V. EXPERIMENTAL RESULTS

In this section, the performance of θ-DEA is to be validated according to the experimental design described in Section IV. Our experiments can be divided into three parts. The first one is to compare θ-DEA with the other two MOEAs with reference points/directions, i.e., NSGA-III and MOEA/D. The aim is to demonstrate the superiority of θ-DEA in achieving the desired convergence and diversity as a reference point-based algorithm. The second one is to compare θ-DEA with various types of many-objective techniques. The aim is to show the great ability of θ-DEA in searching for the sparsely distributed nondominated points over the entire PF as a general many-objective optimizer. The third one is to investigate the influence of parameter θ on the performance

of the proposed algorithm. Moreover, we will also make some further comments on our experimental results in Section V-D.

A. Comparison With NSGA-III and MOEA/D

In this section, IGD is used to evaluate the algorithms. Since all the experimental settings including the way to compute IGD are consistent with those in the original NSGA-III study [29], we compare the IGD results of θ-DEA with those of NSGA-III and MOEA/D-PBI taken from [29].

First, the normalized test problems, DTLZ1–4, which have an identical range of values for each objective over the PF, are employed for comparison. The DTLZ1 problem has a simple, linear PF ($\sum_{i=1}^{m} f_i = 0.5$) but with $11^5 - 1$ local optima in the search space. The difficulty in this problem is to converge to the hyperplane. The PFs of DTLZ 2–4 problems have the same geometrical shape ($\sum_{i=1}^{m} f_i^2 = 1$), but they are designed to challenge different capacities of an algorithm. The DTLZ2 problem is a relatively easy problem with a spherical PF. The DTLZ3 problem introduces a huge number of local PFs paralleling to the global PF based on DTLZ2, which poses a stiff challenge for algorithms to converge to the global PF. The DTLZ4 problem challenges the ability of an algorithm to maintain the diversity in the objective space by introducing a variable density of solutions along the PF.

Table VII shows the results of θ-DEA and NSGA-III on the four problems, where the best, median, and worst IGD values are reported. From Table VII, θ-DEA performs consistently better than NSGA-III on all the instances except 3, 15-objective DTLZ1 and 3-objective DTLZ2. For 15-objective DTLZ1 and DTLZ3, θ-DEA is unable to get close to the PF in some runs, as evident from the large worst IGD value. We suspect that the reason is that θ-DEA sometimes fails to capture the nadir point in high-dimensional objective space and does a wrong normalization. But for 15-objective DTLZ4, θ-DEA performs very well in all runs, whereas NSGA-III sometimes struggles to maintain good convergence and diversity on DTLZ4 problems having more than five objectives. It is interesting to note that, for three-objective DTLZ4, neither of θ-DEA and NSGA-III could achieve good performance all the time.

MOEA/D-PBI does not incorporate the normalization technique. To compare against it more reasonably, we remove the normalization procedure from θ-DEA, and refer this version as θ-DEA*. Table VIII shows the comparison results between θ-DEA and MOEA/D-PBI. It can be seen that θ-DEA* outperforms MOEA/D-PBI on DTLZ1, DTLZ3, and DTLZ4 problems, whereas MOEA/D-PBI wins on DTLZ2 problem. The advantage of θ-DEA* is particularly obvious on DTLZ4 problem. We also notice that the results of θ-DEA* are generally better than those of θ-DEA shown in Table VII, which indicates that the normalization is not quite necessary for the normalized problems. In addition, unlike θ-DEA, for 15-objective DTLZ1 and DTLZ3, θ-DEA* works well in all runs. This illustrates that the normalization is the bottleneck of the performance of θ-DEA on these two instances to some extent.

TABLE VII
BEST, MEDIAN, AND WORST IGD VALUES FOR θ-DEA AND NSGA-III ON m-OBJECTIVE DTLZ1–4 PROBLEMS. BEST PERFORMANCE IS SHOWN IN BOLD

Problem	m	MaxGen	θ-DEA	NSGA-III
DTLZ1	3	400	5.655E-04	**4.880E-04**
			1.307E-03	1.308E-03
			9.449E-03	**4.880E-03**
	5	600	**4.432E-04**	5.116E-04
			7.328E-04	9.799E-04
			2.138E-03	**1.979E-03**
	8	750	**1.982E-03**	2.044E-03
			2.704E-03	3.979E-03
			4.620E-03	8.721E-03
	10	1000	**2.099E-03**	2.215E-03
			2.448E-03	3.462E-03
			3.935E-03	6.869E-03
	15	1500	**2.442E-03**	2.649E-03
			8.152E-03	**5.063E-03**
			2.236E-01	**1.123E-02**
DTLZ2	3	250	**1.042E-03**	1.262E-03
			1.569E-03	**1.357E-03**
			5.497E-03	**2.114E-03**
	5	350	**2.720E-03**	4.254E-03
			3.252E-03	4.982E-03
			5.333E-03	5.862E-03
	8	500	**7.786E-03**	1.371E-02
			8.990E-03	1.571E-02
			1.140E-02	1.811E-02
	10	750	**7.558E-03**	1.350E-02
			8.809E-03	1.528E-02
			1.020E-02	1.697E-02
	15	1000	**8.819E-03**	1.360E-02
			1.133E-02	1.726E-02
			1.484E-02	2.114E-02
DTLZ3	3	1000	1.343E-03	**9.751E-04**
			3.541E-03	4.007E-03
			5.528E-03	6.665E-03
	5	1000	**1.982E-03**	3.086E-03
			4.272E-03	5.960E-03
			1.911E-02	**1.196E-02**
	8	1000	**8.769E-03**	1.244E-02
			1.535E-02	2.375E-02
			3.826E-02	9.649E-02
	10	1500	**5.970E-03**	8.849E-03
			7.244E-03	1.188E-02
			2.323E-02	**2.083E-02**
	15	2000	**9.834E-03**	1.401E-02
			1.917E-02	2.145E-02
			6.210E-01	**4.195E-02**
DTLZ4	3	600	**1.866E-04**	2.915E-04
			2.506E-04	5.970E-04
			5.320E-01	**4.286E-01**
	5	1000	**2.616E-04**	9.849E-04
			3.790E-04	1.255E-03
			4.114E-04	1.721E-03
	8	1250	**2.780E-03**	5.079E-03
			3.098E-03	7.054E-03
			3.569E-03	6.051E-01
	10	2000	**2.746E-03**	5.694E-03
			3.341E-03	6.337E-03
			3.914E-03	1.076E-01
	15	3000	**4.143E-03**	7.110E-03
			5.904E-03	3.431E-01
			7.680E-03	1.073E+00

Next, two WFG problems, WFG6 and WFG7, are used to test the performance of θ-DEA, NSGA-III, and MOEA/D-PBI. The two problems have the same PF shape, which is part of

TABLE VIII
BEST, MEDIAN, AND WORST IGD VALUES FOR θ-DEA* AND MOEA/D-PBI ON m-OBJECTIVE DTLZ1–4 PROBLEMS. BEST PERFORMANCE IS SHOWN IN BOLD

Problem	m	MaxGen	θ-DEA*	MOEA/D-PBI
DTLZ1	3	400	**3.006E-04**	4.095E-04
			9.511E-04	1.495E-03
			2.718E-03	4.743E-03
	5	600	3.612E-04	**3.179E-04**
			4.259E-04	6.372E-04
			5.797E-04	1.635E-03
	8	750	**1.869E-03**	3.914E-03
			2.061E-03	6.106E-03
			2.337E-03	8.537E-03
	10	1000	**1.999E-03**	3.872E-03
			2.268E-03	5.073E-03
			2.425E-03	6.130E-03
	15	1500	**2.884E-03**	1.236E-02
			3.504E-03	1.431E-02
			3.992E-03	1.692E-02
DTLZ2	3	250	7.567E-04	**5.432E-04**
			9.736E-04	**6.406E-04**
			1.130E-03	**8.006E-04**
	5	350	1.863E-03	**1.219E-03**
			2.146E-03	**1.437E-03**
			2.288E-03	**1.727E-03**
	8	500	6.120E-03	**3.097E-03**
			6.750E-03	**3.763E-03**
			7.781E-03	**5.198E-03**
	10	750	6.111E-03	**2.474E-03**
			6.546E-03	**2.778E-03**
			7.069E-03	**3.235E-03**
	15	1000	7.269E-03	**5.254E-03**
			8.264E-03	**6.005E-03**
			9.137E-03	9.409E-03
DTLZ3	3	1000	**8.575E-04**	9.773E-04
			3.077E-03	3.426E-03
			5.603E-03	9.113E-03
	5	1000	**8.738E-04**	1.129E-03
			1.971E-03	2.213E-03
			4.340E-03	6.147E-03
	8	1000	6.493E-03	**6.459E-03**
			1.036E-02	1.948E-02
			1.549E-02	1.123E+00
	10	1500	5.074E-03	**2.791E-03**
			6.121E-03	**4.319E-03**
			7.243E-03	1.010E+00
	15	2000	7.892E-03	**4.360E-03**
			9.924E-03	1.664E-02
			1.434E-02	1.260E+00
DTLZ4	3	600	**1.408E-04**	2.929E-01
			1.918E-04	4.280E-01
			5.321E-01	**5.234E-01**
	5	1000	**2.780E-04**	1.080E-01
			3.142E-04	5.787E-01
			3.586E-04	7.348E-01
	8	1250	**2.323E-03**	5.298E-01
			3.172E-03	8.816E-01
			3.635E-03	9.723E-01
	10	2000	**2.715E-03**	3.966E-01
			3.216E-03	9.203E-01
			3.711E-03	1.077E+00
	15	3000	**4.182E-03**	5.890E-01
			5.633E-03	1.133E+00
			6.562E-03	1.249E+00

a hyperbolises with radii $r_i = 2i$, $i = 1, 2, \ldots, m$. WFG6 problem is nonseparable, while WFG7 problem is biased. Table IX presents the comparison results. As can be seen,

TABLE IX
BEST, MEDIAN, AND WORST IGD VALUES FOR θ-DEA, NSGA-III, AND MOEA/D-PBI ON m-OBJECTIVE WFG6 AND WFG7 PROBLEMS. BEST PERFORMANCE IS SHOWN IN BOLD

Problem	m	MaxGen	θ-DEA	NSGA-III	MOEA/D-PBI
WFG6	3	400	2.187E-02	**4.828E-03**	1.015E-02
			2.906E-02	**1.224E-02**	3.522E-02
			3.355E-02	5.486E-02	1.066E-01
	5	750	2.430E-02	**5.065E-03**	8.335E-03
			3.270E-02	**1.965E-02**	4.230E-02
			3.661E-02	4.474E-02	1.058E-01
	8	1500	2.528E-02	**1.009E-02**	1.757E-02
			3.140E-02	**2.922E-02**	5.551E-02
			3.727E-02	7.098E-02	1.156E-01
	10	2000	2.098E-02	1.060E-02	**9.924E-03**
			3.442E-02	**2.491E-02**	4.179E-02
			4.183E-02	6.129E-02	1.195E-01
	15	3000	3.759E-02	**1.368E-03**	1.513E-02
			4.892E-02	**2.877E-02**	6.782E-02
			6.380E-02	6.970E-02	1.637E-01
WFG7	3	400	5.524E-03	**2.789E-03**	1.033E-02
			6.951E-03	**3.692E-03**	1.358E-02
			8.923E-03	**4.787E-03**	1.926E-02
	5	750	**5.770E-03**	8.249E-03	8.780E-03
			6.854E-03	9.111E-03	1.101E-02
			8.100E-03	1.050E-02	1.313E-02
	8	1500	**4.405E-03**	2.452E-02	1.355E-02
			5.603E-03	2.911E-02	1.573E-02
			8.527E-03	6.198E-02	2.626E-02
	10	2000	**7.069E-03**	3.228E-02	1.041E-02
			8.322E-03	4.292E-02	1.218E-02
			9.664E-03	9.071E-02	1.490E-02
	15	3000	8.915E-03	3.457E-02	**7.552E-03**
			1.059E-02	5.450E-02	1.063E-02
			1.236E-02	8.826E-02	2.065E-02

NSGA-III performs best on WFG6 problems and three-objective WFG7, whereas the proposed θ-DEA performs best on WFG7 with more than three objectives. MOEA/D-PBI is outperformed by θ-DEA on both WFG6 and WFG7 problems.

To further investigate the performance of θ-DEA on problems with disparately scaled objective values, SDTLZ1 and SDTLZ2 problems are considered. The comparison results are shown in Table X. For SDTLZ1 problem, the situation is similar to DTLZ1 problem. That is, NSGA-III performs better on 3- and 15-objective instances, whereas θ-DEA performs better on the remaining instances. As for SDTLZ2 problem, θ-DEA clearly outperforms NSGA-III.

Based on the above comparisons, it can be concluded that the proposed θ-DEA can generally maintain a good balance between convergence and diversity assisted by structured reference points. Indeed, through the test on problems with varying features, θ-DEA outperforms NSGA-III and MOEA/D-PBI on most of them in terms of IGD values.

B. Comparison With State-of-the-Art Algorithms

In this section, we compare the proposed θ-DEA with all the eight algorithms mentioned in Section IV-C. The version of θ-DEA without normalization, i.e., θ-DEA*, will also be involved in the comparison. The HV indicator is used to evaluate the concerned algorithms.

TABLE X
BEST, MEDIAN, AND WORST IGD VALUES FOR θ-DEA AND NSGA-III ON SCALED m-OBJECTIVE DTLZ1 AND DTLZ2 PROBLEMS. BEST PERFORMANCE IS SHOWN IN BOLD

Problem	m	MaxGen	θ-DEA	NSGA-III
SDTLZ1	3	400	8.153E-04	**3.853E-04**
			3.039E-03	**1.214E-03**
			1.413E-02	**1.103E-02**
	5	600	**8.507E-04**	1.099E-03
			1.225E-03	2.500E-03
			6.320E-03	3.921E-02
	8	750	**4.043E-03**	4.659E-03
			4.938E-03	1.051E-02
			8.734E-03	1.167E-01
	10	1000	4.100E-03	**3.403E-03**
			4.821E-03	5.577E-03
			6.676E-03	3.617E-02
	15	1500	5.037E-03	**3.450E-03**
			1.078E-02	**6.183E-03**
			4.774E-01	**1.367E-02**
SDTLZ2	3	250	**9.709E-04**	1.347E-03
			1.926E-03	2.069E-03
			7.585E-03	**5.284E-03**
	5	350	**2.755E-03**	1.005E-02
			3.521E-03	2.564E-02
			6.258E-03	8.430E-02
	8	500	**7.790E-03**	1.582E-02
			9.015E-03	1.788E-02
			1.091E-02	2.089E-02
	10	750	**7.576E-03**	2.113E-02
			8.680E-03	3.334E-02
			1.068E-02	2.095E-01
	15	1000	**9.373E-03**	2.165E-02
			1.133E-02	2.531E-02
			1.401E-02	4.450E-02

Table XI presents the average HV results on DTLZ1–4, DTLZ7, SDTLZ1, and SDTLZ2 problems, and Table XII on WFG problems. Table XIII gives a summary of the significance test on HV results between the proposed θ-DEA (θ-DEA*) and the other algorithms. In this table, Alg_1 versus Alg_2, "B" ("W") means that the number of instances on which the results of Alg_1 are significantly better (worse) than those of Alg_2, and "E" indicates the number of instances where there exists no statistical significance between the results of Alg_1 and Alg_2.

To describe the distribution of obtained solutions in high-dimensional objective space, we use 15-objective WFG7 instance as an illustration. Fig. 4 plots the final solutions of four competitive algorithms, i.e., θ-DEA, GrEA, NSGA-III, and SDE, in a single run by parallel coordinates. This particular run is associated with the result closest to the average HV value. It is clear from Fig. 1 that θ-DEA and NSGA-III are able to find a good approximation and coverage of the PF, whereas GrEA and SDE can only converge to a portion of the PF.

To quantify how well each algorithm performs overall, the performance score [52] is introduced to rank the algorithms. For a specific problem instance, suppose there are l algorithms $\text{Alg}_1, \text{Alg}_2, \ldots, \text{Alg}_l$ involved in the comparison, let $\delta_{i,j}$ be 1, if Alg_j is significantly better than Alg_i in terms of HV, and 0 otherwise. Then, for each algorithm Alg_i, the performance score $P(\text{Alg}_i)$ is determined as

$$P(\text{Alg}_i) = \sum_{\substack{j=1 \\ j \neq i}}^{l} \delta_{i,j}. \tag{16}$$

This value reveals how many other algorithms significantly outperform the corresponding algorithm on the considered test instance. So, the smaller the value, the better the algorithm. In Fig. 5, the average performance score is summarized for different number of objectives and different test problems. Fig. 6 shows the average performance score over all 80 problem instances for the selected ten algorithms, and the overall rank of each algorithm according to the score is also given in the corresponding bracket.

Based on the above results, we can obtain some observations for each algorithm. The proposed θ-DEA works well on nearly all the considered instances. In particular, it shows the best overall performance on DTLZ4, SDTLZ1, SDTLZ2, and WFG4–9 problems. Relatively speaking, θ-DEA does not show such outstanding performance on WFG1–3 problems. For WFG2 problem, θ-DEA exhibits an interesting search behavior, it remains competitive on 3-, 5-, 8-, and 10-objective instances, but performs worst on 15-objective instance. θ-DEA* shows the advantage on the normalized test problems, i.e., DTLZ1–4 problems. But it cannot be compared to θ-DEA when handling scaled problems. Indeed, θ-DEA* is significantly outperformed by θ-DEA on 53 out of 60 scaled problem instances, verifying the effectiveness of the normalization procedure in θ-DEA.

GrEA can effectively deal with scaled problems in general, which is verified by its competitive results on DTLZ7, SDTLZ1, SDTLZ2, and most of WFG problem instances. This is mainly because that GrEA divides each dimension of the objective space into the same number of divisions. So, GrEA indeed does the objective normalization implicitly during the evolutionary process. In addition, it is worth mentioning that the performance of GrEA is sensitive to the parameter div, the overall excellent performance of GrEA in our experiments is obtained by suitably setting div for each test instance. In this regard, GrEA takes advantage of the other compared algorithms.

POGA shows overall competitive performance on WFG2 and WFG3 problems, and it even achieves the best performance on ten-objective WFG2 instance and 5-, 8-, and 10-objective WFG3 instances. But it generally cannot obtain very satisfying results on the other problems. It is interesting to note that, for DTLZ1, DTLZ3, and SDTLZ1 problems, POGA obtains zero HV values on 5-, 8-, and 10-objective instances, but nonzero values for 15-objective instance. Since the PFs of the three problems have a huge number of local PFs, it is not surprising that POGA always fails to converge to the PFs and obtains zero HV values. But the reason for nonzero HV values (especially poor) for 15-objective instances is waiting to be explored.

NSGA-III shows the closest overall performance to the proposed θ-DEA, and it can perform very well over a wide range of test problems. There is no significant difference between

TABLE XI
PERFORMANCE COMPARISON OF θ-DEA TO DIFFERENT ALGORITHMS WITH RESPECT TO THE AVERAGE HV VALUES ON DTLZ1–4, DTLZ7, SDTLZ1, AND SDTLZ2 PROBLEMS. THE BEST AVERAGE HV VALUE AMONG THE TEN ALGORITHMS FOR EACH INSTANCE IS HIGHLIGHTED IN BOLD

Problem	m	MaxGen	θ-DEA	θ-DEA*	GrEA	POGA	NSGA-III	SDE	MOEA/D	dMOPSO	HypE	MOMBI
DTLZ1	3	400	1.116137	**1.118329**	1.072987‡	1.083131‡	1.117600	1.097010‡	1.116679*	1.074976‡	1.117483*	1.072413‡
	5	600	1.576983	**1.577892**	1.509905‡	0.000000‡	1.577027‡	1.545051‡	1.577632*	1.482412‡	1.466936‡	1.509007‡
	8	750	2.137924	**2.137998**	2.105894‡	0.000000‡	2.137837*	2.089314‡	2.136337‡	1.824428‡	1.999087‡	2.018629‡
	10	1000	2.592719	2.592696	2.566547‡	0.000000‡	**2.592792**	2.562563‡	2.592233‡	2.317805‡	2.520526‡	2.470400‡
	15	1500	4.131873	4.175713	3.914267‡	0.169237‡	**4.176773**	4.083228‡	4.169859*	3.394256‡	3.702937‡	3.623057‡
DTLZ2	3	250	0.743778	0.744320	0.740172‡	0.698084‡	0.743523*	0.742896‡	0.744137*	0.712523‡	**0.753191**	0.703623‡
	5	350	1.306928	**1.307368**	1.304274‡	0.853065‡	1.303638‡	1.299877‡	1.307343	1.239853‡	0.756079‡	1.142958‡
	8	500	1.977904	1.978469	**1.989406**	0.181198‡	1.969096‡	1.980826	1.978216*	1.816420‡	0.892917‡	1.373654‡
	10	750	2.514259	2.514485	**2.515566**	0.368282‡	2.508717‡	2.505222‡	2.515040	2.428399‡	1.174930‡	1.623827‡
	15	1000	4.137225	4.137766	4.070080‡	0.600316‡	4.133743‡	4.105103‡	**4.137792**	3.931332‡	1.407601‡	1.891898‡
DTLZ3	3	1000	0.736938	0.738977	0.678608‡	0.699556‡	0.737407*	0.739591	0.736044	0.665529	**0.750325**	0.702072‡
	5	1000	1.303987	**1.305846**	1.135837‡	0.000000‡	1.301481‡	1.300661‡	1.303168	1.252229‡	0.740621‡	1.138243‡
	8	1000	1.968943	**1.970805**	1.622438‡	0.000000‡	1.954336‡	1.968342	1.251873‡	1.428208‡	0.881516‡	1.325053‡
	10	1500	2.512662	**2.514027**	2.306975‡	0.000000‡	2.508879‡	2.507127‡	2.406221	2.107556‡	1.175350‡	1.643232‡
	15	2000	3.788788	**4.136332**	3.215646‡	0.311735‡	4.123622*	4.072396*	2.722371*	2.269634‡	1.553939‡	1.901460‡
DTLZ4	3	600	0.729265	0.602951	0.573359†	0.702980†	**0.744634**	0.630875†	0.406020‡	0.677459†	0.549999†	0.691034†
	5	1000	**1.308945**	1.308934	1.307539‡	0.996996‡	1.308698‡	1.289331‡	1.205512‡	1.203429‡	1.014145‡	1.128597‡
	8	1250	1.980779	1.977231	**1.981321**	0.058851‡	1.980236‡	1.966667‡	1.826489‡	1.829561‡	0.925370‡	1.326579‡
	10	2000	2.515436	**2.515468**	2.514960‡	0.080187‡	2.515172‡	2.500043‡	2.423727‡	2.438748‡	1.235517‡	1.453693‡
	15	3000	4.138203	**4.138225**	4.132426‡	1.482621‡	4.138154*	4.109571‡	3.978200‡	3.936754‡	2.056801‡	1.263883‡
DTLZ7	3	1000	0.415753	0.406455	0.417124	0.416076	0.415824	**0.427114**	0.404236†	0.411552†	0.397286‡	0.409400†
	5	1000	0.448398	0.371751	0.521976	0.413607†	0.493987	**0.528809**	0.366661‡	0.365309‡	0.411226†	0.491190
	8	1000	0.506288	0.152916	**0.581912**	0.445509†	0.528482	0.345470†	0.012948‡	0.121468†	0.262019†	0.452561†
	10	1500	0.594612	0.122341	**0.651955**	0.445526†	0.615565	0.173690†	0.001891‡	0.036388‡	0.272124†	0.508263†
	15	2000	0.747688	0.059097	**0.764778**	0.502990†	0.751188	0.042080†	0.000425‡	0.000220‡	0.069042†	0.581946†
SDTLZ1	3	400	**1.117365**	0.724709	1.096990†	1.084446†	1.116023	1.086490†	0.332926‡	0.470223‡	1.100216†	1.075055†
	5	600	**1.577751**	0.722800	1.501184†	0.000000‡	1.575525†	1.491374†	0.556767‡	0.233560‡	0.753613†	1.508512†
	8	750	**2.137936**	1.413738	2.112299†	0.000000‡	2.126815†	1.712786†	1.319874‡	0.810944‡	1.072366‡	2.021830†
	10	1000	**2.592726**	2.158027	2.568992†	0.000000‡	2.587997†	2.285257†	2.068300‡	1.378952‡	1.490181‡	2.465309†
	15	1500	4.164391	4.056134	3.960558‡	0.033188‡	**4.176670**	4.084369†	3.972655‡	3.218729‡	2.988263‡	3.590185‡
SDTLZ2	3	250	0.743746	0.378277	0.740767†	0.697453†	0.743277†	0.706272†	0.210273‡	0.205731‡	**0.753099**	0.703222†
	5	350	**1.306804**	0.405596	1.303628†	0.856364†	1.290520†	0.667578†	0.251787‡	0.246862‡	0.386773‡	1.142126†
	8	500	1.977826	0.729306	**1.990325**	0.231584‡	1.963251†	0.888944†	0.414793‡	0.479764‡	0.441930‡	1.365266†
	10	750	**2.514252**	0.866187	1.851829†	0.265022‡	2.504209†	1.004728†	0.486005‡	0.584936‡	0.544107‡	1.630979†
	15	1000	**4.137314**	1.529079	4.079169†	0.858370‡	4.129375†	1.861987†	0.715960‡	0.466954‡	1.323373‡	1.898389†

$\dagger$ means that the result is significantly outperformed by θ-DEA.
$*$ means that the result is significantly outperformed by θ-DEA*.
$\ddagger$ means that the result is significantly outperformed by both θ-DEA and θ-DEA*.

the results of NSGA-III and θ-DEA in 27 out of 80 instances. It is interesting to find that, for WFG1 problem, NSGA-III performs poorly on 3-, 5-, 8-, and 10-objective instances, but it takes the second place on 15-objective instance.

SDE generally has the medium-high performance on most of the considered problems among the compared algorithms, leading to its fourth rank in the average performance score over all the instances as shown in Fig. 6. It is worth noting, for DTLZ7 problem, SDE performs best on three- and five-objective instances, but its performance scales poorly in instances with a higher number of objectives.

MOEA/D achieves good performance on the normalized test problems except DTLZ4, but it cannot produce satisfactory results on almost all the scaled test problems. This is why it ranks poorly in Fig. 6. Nevertheless, from this, we do not claim that MOEA/D is a poor many-objective optimizer, since a naive normalization procedure may even enhance its ability to deal with scaled problems [33]. dMOPSO is a relatively poor algorithm, and it does not perform well even on the normalized problems.

HypE is very competitive on three-objective instances, which is reflected in Fig. 5(a). Indeed, it performs best on 12 out of the 16 three-objective instances. However, it does not show advantage over the other algorithms on problems having more than three objectives except on WFG3, where it performs very well on 8-, 10-, and 15-objective instances. Note that, HypE computes exactly the hypervolume-based fitness values when $m \leq 3$, otherwise it estimates the fitness values using Monte Carlo simulation. Thus, we suspect that its relatively poor performance on problems with high number of objectives is mainly due to its inaccurate fitness estimation. Increasing the number of sampling points may improve the situation, but the computational effort will soon become unacceptable. Although, HypE is a popular many-objective optimizer, our

TABLE XII
PERFORMANCE COMPARISON OF θ-DEA TO DIFFERENT ALGORITHMS WITH RESPECT TO THE AVERAGE HV VALUES ON WFG PROBLEMS. THE BEST AVERAGE HV VALUE AMONG THE TEN ALGORITHMS FOR EACH INSTANCE IS HIGHLIGHTED IN BOLD

Problem	m	MaxGen	θ-DEA	θ-DEA*	GrEA	POGA	NSGA-III	SDE	MOEA/D	dMOPSO	HypE	MOMBI
WFG1	3	400	0.704526	0.697356	0.846287	0.766621	0.669729‡	0.803966	0.657143‡	0.403170‡	**0.976181**	0.885760
	5	750	1.138794	1.236030	1.268898	1.109501‡	0.859552‡	1.354217	1.349888	0.461233‡	0.911020‡	**1.528811**
	8	1500	1.875997	1.905395	1.769013‡	1.745597‡	1.424963‡	1.883743	1.755326‡	0.484046‡	1.536599‡	**2.042375**
	10	2000	2.364268	2.386742	2.365107	2.355532	2.249535‡	2.375338	1.799394‡	0.536340‡	2.268813‡	**2.465608**
	15	3000	4.003682	3.867407	3.811128‡	3.528350‡	4.085931	3.903864†	1.772444‡	0.750153‡	4.028462	**4.122918**
WFG2	3	400	1.227226	1.221941	1.226099	1.208077‡	1.226956	1.224626†	1.111085‡	1.125810‡	**1.244737**	1.192808‡
	5	750	1.597188	1.564818	1.570086†	1.591412†	**1.598410**	1.579667†	1.520168‡	1.478517‡	1.535704‡	1.596152†
	8	1500	2.124411	2.055014	2.102930†	2.136409	2.136525	2.107484†	2.016854‡	1.971067‡	2.084336†	**2.136910**
	10	2000	2.578311	2.491268	2.570389†	**2.591798**	2.588104	2.573254†	2.459026‡	2.406484‡	2.556327†	2.589607
	15	3000	3.467983	3.961793	4.094032	4.163488	**4.173427**	4.143102	3.921513*	3.822155*	4.126212	4.157740
WFG3	3	400	0.814962	0.798550	0.834432	0.829395	0.819758	0.834035	0.757034‡	0.774135‡	**0.847567**	0.835361
	5	750	1.028412	0.999933	1.013341†	**1.064772**	1.013941†	0.956846‡	0.906075‡	0.957250‡	0.977617‡	0.959638‡
	8	1500	1.147203	1.174530	1.263233	**1.427165**	1.221543	1.127832	0.770754‡	1.093482‡	1.351959	1.352012
	10	2000	1.573090	1.359598	1.577058	**1.770022**	1.567908	1.370175†	0.524917‡	1.004506‡	1.720463	1.699805
	15	3000	2.510031	0.904082	2.711461	2.534502	2.510223	2.458935	0.579003‡	0.535783‡	**2.793994**	2.702406
WFG4	3	400	0.729664	0.720486	0.717433‡	0.679313‡	0.728892†	0.720904†	0.685079‡	0.643591‡	**0.750714**	0.687819‡
	5	750	**1.286861**	1.259362	1.271279†	1.096965‡	1.285072†	1.220702‡	1.161435‡	1.074986‡	0.855942‡	1.146019‡
	8	1500	**1.964648**	1.858132	1.933492†	1.448769‡	1.962156	1.784107‡	1.188847‡	1.078243‡	1.137827‡	1.373071‡
	10	2000	**2.504065**	2.232877	2.481299†	1.696742‡	2.502319†	2.292594†	1.432285‡	1.330296‡	1.557451‡	1.683372‡
	15	3000	**4.136892**	2.990774	3.893931†	2.748079‡	4.136393	3.683401†	1.694794‡	0.991603‡	2.551034‡	1.872957‡
WFG5	3	400	0.687005	0.676813	0.669193‡	0.655342‡	0.687220	0.683408†	0.656189‡	0.633971‡	**0.698642**	0.646584‡
	5	750	**1.222746**	1.190345	1.219312†	1.080845‡	1.222480†	1.173486‡	1.120619‡	1.049378‡	0.893813‡	1.061277‡
	8	1500	1.850361	1.727167	**1.862278**	0.936887‡	1.850281	1.711954‡	1.279934‡	0.671722‡	1.183477‡	1.286271‡
	10	2000	2.346521	2.092514	2.335886†	1.138726‡	**2.346581**	2.204977†	1.541144‡	0.303028‡	1.659310‡	1.428448‡
	15	3000	3.833116	2.719208	3.400492†	3.455143†	**3.833242**	3.184516†	1.864379‡	0.089205‡	2.764870†	1.455641‡
WFG6	3	400	0.690060	0.679787	0.677130†	0.640068‡	0.685939	0.685988†	0.654956‡	0.657493‡	**0.708633**	0.646929‡
	5	750	1.223099	1.189960	**1.224094**	1.026995‡	1.219001	1.176927‡	1.041593‡	1.116645‡	0.441287‡	1.071505‡
	8	1500	1.841974	1.727171	**1.858232**	1.266641‡	1.843340	1.695095‡	0.698152‡	1.087488‡	0.475371‡	1.285725‡
	10	2000	**2.333417**	2.011900	2.331650	1.610767‡	2.326666	2.175824†	0.811370‡	1.189267‡	0.627106‡	1.482006‡
	15	3000	**3.723823**	2.338636	3.571653†	2.063536†	3.717982	3.164650†	0.594620‡	1.134651‡	1.123090‡	1.550496‡
WFG7	3	400	0.731157	0.722678	0.721095‡	0.688454‡	0.729030†	0.732513	0.619351‡	0.589746‡	**0.752768**	0.693879‡
	5	750	1.295864	1.263840	**1.298471**	0.997043‡	1.291999†	1.249762‡	1.073783‡	0.992021‡	0.525322‡	1.143173‡
	8	1500	1.973601	1.843617	**1.992683**	1.233529‡	1.971529†	1.786518‡	0.813288‡	0.986483‡	0.521515‡	1.391605‡
	10	2000	**2.508710**	2.280858	2.503361	1.663976‡	2.507511†	2.337352†	0.950840‡	1.154313‡	1.234357‡	1.666138‡
	15	3000	**4.136189**	3.174241	3.829786†	2.538966‡	4.134418†	3.898443†	0.772304‡	1.134836‡	2.267561‡	1.847512‡
WFG8	3	400	0.666959	0.655503	0.656139†	0.613089‡	0.665932	0.662973†	0.633207‡	0.491854‡	**0.686264**	0.631815‡
	5	750	**1.183904**	1.147459	1.173895†	0.973151‡	1.182260†	1.136938‡	0.968246‡	0.836739‡	0.340591‡	0.623923‡
	8	1500	**1.768213**	1.714995	1.733031†	1.295632‡	1.759882†	1.666713‡	0.326124‡	0.195763‡	0.693903‡	1.313675‡
	10	2000	**2.297054**	2.198196	2.252147†	1.684519‡	2.280276†	2.168904†	0.255629‡	0.237683‡	1.014398‡	1.636747‡
	15	3000	**3.854067**	2.756205	3.667292†	2.734447†	3.815520	3.717599†	0.706627‡	0.265396‡	1.596445‡	1.961387‡
WFG9	3	400	0.680306	0.671742	0.688081	0.626639‡	0.670081	0.695933	0.564686‡	0.652229‡	**0.733841**	0.663962‡
	5	750	1.224104	1.159929	**1.238784**	0.318624‡	1.212266†	1.186947†	1.028928‡	1.031762‡	1.053354‡	0.949513‡
	8	1500	1.842840	1.627886	**1.860060**	0.842951‡	1.803989†	1.694549†	0.882226‡	1.017887‡	1.526635‡	1.343829‡
	10	2000	**2.364149**	1.956897	2.343906†	1.641153‡	2.326700†	2.204423†	1.095281‡	1.125591‡	2.044502†	1.688538‡
	15	3000	**3.862664**	2.472073	3.687303†	0.976440‡	3.801860	3.466938†	1.002115‡	0.804192‡	2.591612†	2.188369‡

$\dagger$ means that the result is significantly outperformed by θ-DEA.
$*$ means that the result is significantly outperformed by θ-DEA*.
$\ddagger$ means that the result is significantly outperformed by both θ-DEA and θ-DEA*.

experimental results indicate that it is not compared favorably with some newly proposed many-objective algorithms. The similar observation can also be found in several recent studies [23], [30], [63].

MOMBI performs well on most of WFG1–3 problem instances, especially on WFG1, where it attains the best results on 5-, 8-, 10-, and 15-objective instances. But it does not behave quite well on the other problems.

Since, we have allocated more computational efforts for problems having higher number of objectives, it is not possible to analyze performance scalability. In fact, it is not clear from our experimental results that the performance of each algorithm decays as the number of objectives is increased. For the proposed θ-DEA, it performs as well even on 15-objective instances, which can be seen from the results on the problems whose PFs have regular geometrical shapes, i.e., DTLZ1–4, SDTLZ1, SDTLZ2, and WFG4–9 problems. The average HV values of θ-DEA on these instances are close to $1.1^{15} \approx 4.177$, indicating that it can achieve a good approximation of PF even in high-dimensional objective space.

TABLE XIII
SUMMARY OF THE SIGNIFICANCE TEST BETWEEN θ-DEA (θ-DEA*) AND THE OTHER ALGORITHM

		θ-DEA	θ-DEA*	GrEA	POGA	NSGA-III	SDE	MOEA/D	dMOPSO	HypE	MOMBI
θ-DEA vs.	B	–	55	52	69	39	64	66	78	62	67
	W	–	17	22	9	14	10	8	2	17	13
	E	–	8	6	2	27	6	6	0	1	0
θ-DEA* vs.	B	17	–	22	57	20	27	72	76	51	52
	W	55	–	53	19	58	41	3	1	26	27
	E	8	–	5	4	2	12	5	3	3	1

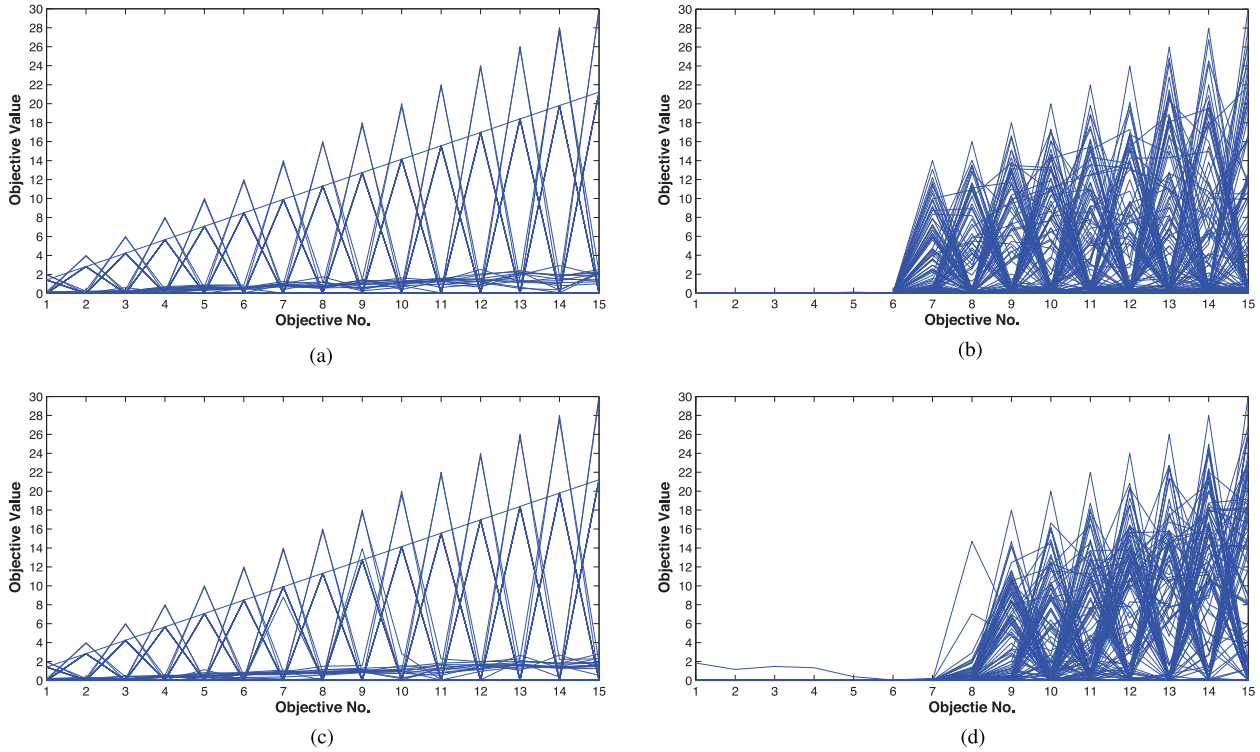

Fig. 4. Final solution set of the (a) θ-DEA, (b) GrEA, (c) NSGA-III, and (d) SDE on the 15-objective WFG7 instance, shown by parallel coordinates.

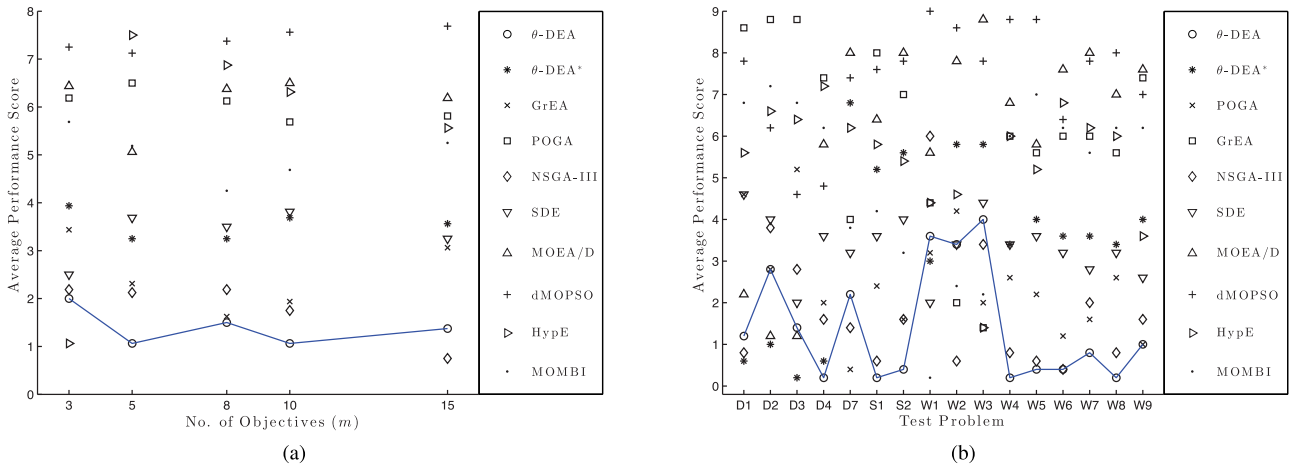

Fig. 5. (a) Average performance score over all test problems for different number of objectives. (b) Average performance score over all objective dimensions for different test problems, namely DTLZ (Dx), SDTLZ (Sx), and WFG (Wx). The smaller the score, the better the PF approximation in terms of HV. The values of θ-DEA are connected by a solid line to easier assess the score.

Next, we would like to briefly investigate what will happen if we set a larger number of decision variables (n) for test problems, although this is not the focus of this paper. As an illustration, we select a normalized problem DTLZ2 and a scaled problem WFG7. For DTLZ2, k is reset to 98, thus $n = m + 97$. For WFG7, n is reset to 100, and the

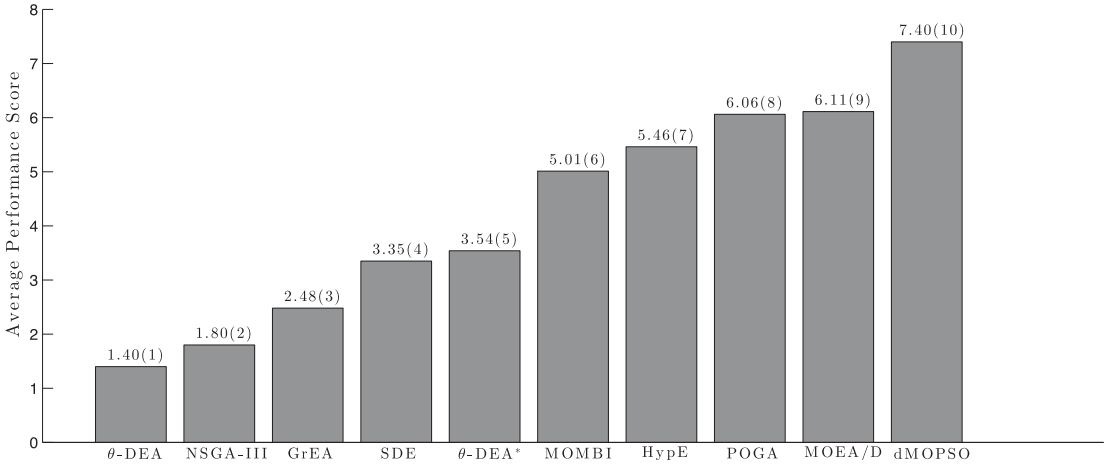

Fig. 6. Ranking in the average performance score over all test problem instances for the selected ten algorithms. The smaller the score, the better the overall performance in terms of HV.

TABLE XIV
AVERAGE HV VALUES FOR θ-DEA, NSGA-III, AND SDE ON LARGE-SCALE DTLZ2 AND WFG7 PROBLEMS. BEST PERFORMANCE IS SHOWN IN BOLD

Problem	m	MaxGen	θ-DEA	NSGA-III	SDE
DTLZ2	3	250	0.605706	0.576772†	**0.681272**
	5	350	1.194953	1.033459†	**1.270122**
	8	500	1.819000	1.193003†	**1.955069**
	10	750	2.474082	2.228702†	**2.492587**
	15	1000	**4.100986**	3.382835†	4.070196†
WFG7	3	400	0.664099	0.656173†	**0.711816**
	5	750	**1.240632**	1.225887†	1.224485†
	8	1500	**1.916811**	1.903908†	1.716448†
	10	2000	**2.472800**	2.462653†	2.232295†
	15	3000	**4.107807**	4.105074†	3.574938†

† means that the result is significantly outperformed by θ-DEA.

position-related parameter is still set to $m-1$. We run three competitive algorithms θ-DEA, NSGA-III, and SDE on the two problems, and the algorithm parameters and the termination criterion are kept unchanged. Table XIV shows the average HV results. It can be seen that all the three algorithms obtain smaller HV values than on the original problem instances, which indicates that larger n would pose greater difficulty to all of them. However, the comparison situation changes to some extent. For example, θ-DEA performs significantly better than SDE on the original 3-, 5-, and 10-objective DTLZ2 instances, but it is significantly outperformed by SDE on these instances with larger n. It seems that, for DTLZ2 problem, SDE is less influenced by the increasing of n than θ-DEA. In addition, it is worth mentioning that a cooperative coevolution technique [83] has been developed specially for solving MOPs with large number of decision variables.

In the end, we intend to gain some insights based on the extensive experimental results provided in this section. First, it should be pointed out that the performance of an algorithm not only depends on its ability to cope with specific problem features but also depends on its ability to handle high number of objectives (e.g., θ-DEA on WFG2 and SDE on DTLZ7). Second, for certain problems, the increasing of the number of objectives would exert different levels of impact on different algorithms (e.g., NSGA-III on WFG1). Third, θ-DEA, NSGA-III, and GrEA show strong competitiveness on most of the problem instances considered in this paper. θ-DEA may have difficulty in handling problems with high-dimensional convex disconnected PFs (e.g., 15-objective WFG2); NSGA-III may struggle on relatively low-dimensional biased problems having mixed PFs (e.g., 3-, 5-, 8-, and 10-objective WFG1); GrEA may not be good at solving problems with a huge number of local PFs (e.g., DTLZ1 and DTLZ3). The other concerned algorithms show advantage on specific problem instances, and their main characteristics can be summarized as follows.

1) θ-DEA* performs very well on normalized test problems (e.g., DTLZ1–4).
2) POGA is effective on relatively high-dimensional problems having disconnected or degenerated PFs (e.g., 8-, 10-, and 15-objective WFG2, and 5-, 8-, and 10-objective WFG3).
3) SDE is not bad on majority of problems, and it achieves superior performance on low-dimensional problems having disconnected and mixed PFs (e.g., three- and five-objective DTLZ7).
4) MOEA/D generally shows good performance on normalized test problems (e.g., DTLZ1–3), but it cannot effectively deal with problems with strong bias (e.g., DTLZ4).
5) HypE shows superior performance in solving three-objective problems, and it also does well in high-dimensional problems with linear and degenerated PFs (e.g., 8-, 10-, and 15-objective WFG3).
6) MOMBI is better at tackling PFs with irregular geometrical shapes (e.g., WFG1–3), and it performs particularly well on biased problems with mixed PFs (e.g., WFG1).

C. Influence of Parameter θ

In this section, we investigate the effect of parameter θ on the performance of the proposed algorithm. The variation of θ would influence the normalization process presented in Section III-D, and hence influence the search

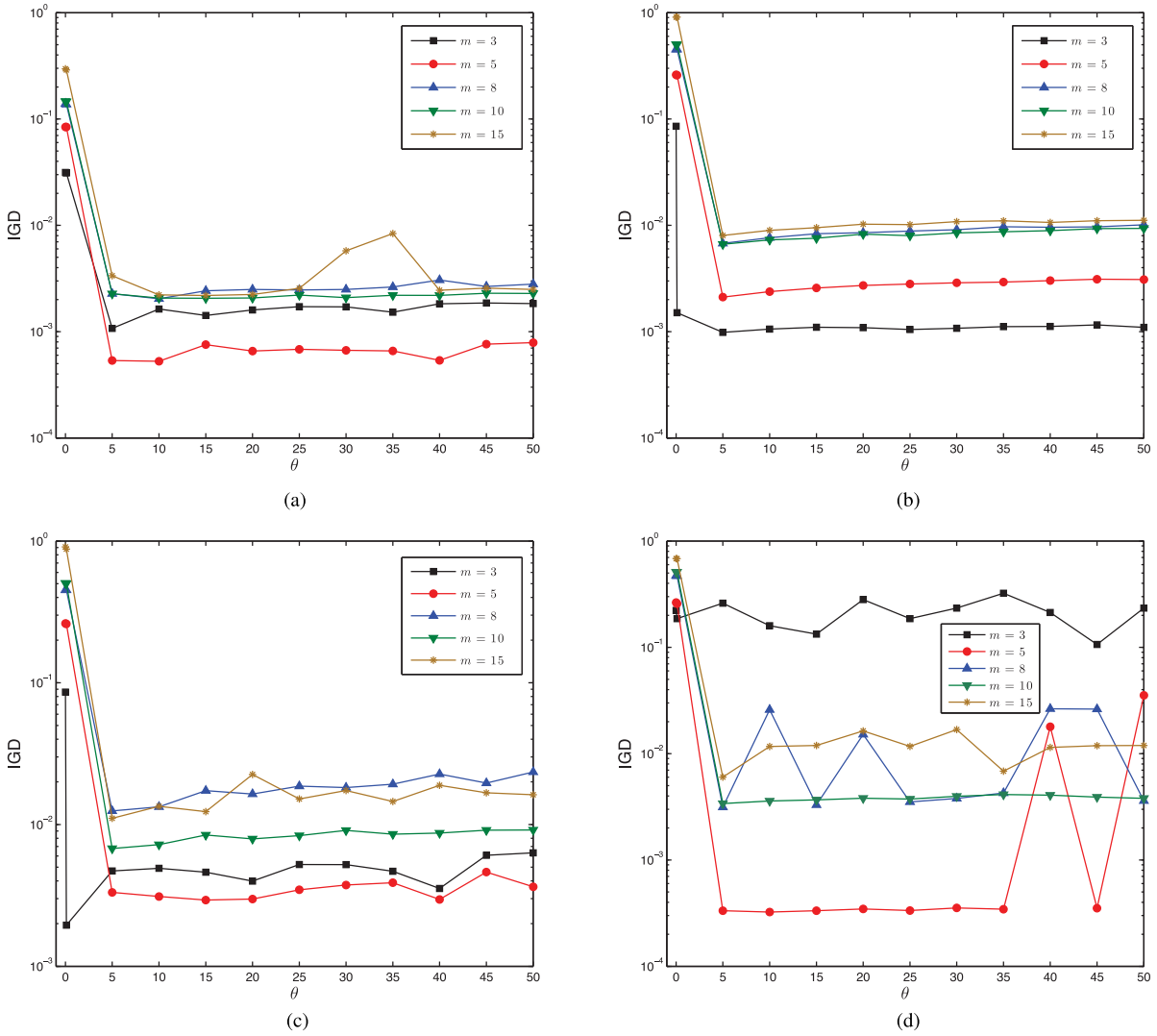

Fig. 7. Examination of the influence of θ on IGD of θ-DEA* for DTLZ1–4 problems with varying number of objectives m. The figures show the average IGD of 20 independent runs each. (a) DTLZ1. (b) DTLZ2. (c) DTLZ3. (d) DTLZ4.

behavior of θ-DEA. To observe the pure effect of θ, we hypothesize that the ideal and nadir points are known *a priori* and the accurate normalization can be done. Therefore, here we decide to show the influence of θ by running θ-DEA* on the normalized test problems, i.e., DTLZ1–4. Similar observation can be obtained from the other test cases.

Figs. 7 and 8 present how the performance of θ-DEA* varies with the change of θ on DTLZ1–4 problems in terms of average IGD and average HV, respectively. We vary θ between 0 and 50 with a step size of 5, and $\theta = 0.1$ is also examined. Based on the two figures, we obtain the following observations.

1) $\theta = 0$ almost always leads to the worst performance.
2) $\theta = 0.1$ is also usually a bad choice, but it seems to be the most suitable setting for three-objective DTLZ3.
3) The most appropriate setting of θ depends on the problem instance to be solved.
4) The performance of θ-DEA* on most of the problem instances is robust over a wide range of θ values, which is beneficial to the practical use of the proposed algorithm.

Note that, for eight-objective DTLZ4, the IGD values fluctuate with the increase in θ, but the corresponding HV values remain stable. The way we compute IGD in our experiments makes IGD a more sensitive indicator than HV. For example, it may lead to much larger IGD even if there is only one reference point that cannot be associated well with the found solutions, but the corresponding HV may only get slightly smaller in this case.

It is also interesting to understand how well θ-DEA* performs when θ tends to positive infinity. In this case, only the closeness to the corresponding reference points for each solution is emphasized in the environmental selection phase, which makes the search behavior of θ-DEA* more like NSGA-III. To investigate this extreme case, we set θ to a large value 10^6. Table XV compares the average IGD and average HV values of θ-DEA* ($\theta = 5$) and θ-DEA* ($\theta = 10^6$). It can be seen clearly that $\theta = 5$ generally achieves a better balance between convergence and diversity than $\theta = 10^6$ by means of reference points.

D. Further Discussion

In this section, we further discuss three issues. The first is about the quality indicators adopted in our experiments;

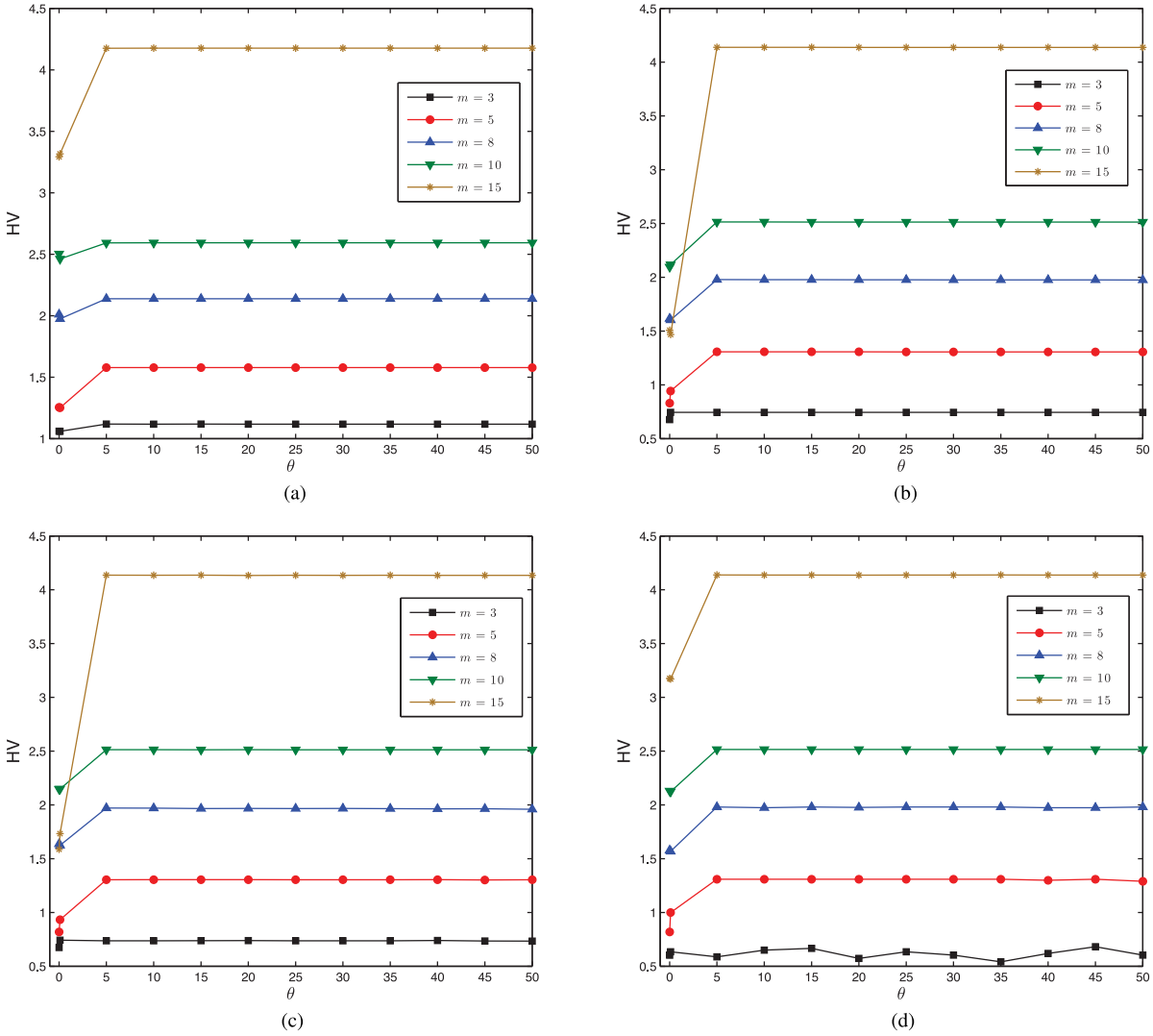

Fig. 8. Examination of the influence of θ on HV of θ-DEA* for DTLZ1–4 problems with varying number of objectives m. The figures show the average HV of 20 independent runs each. (a) DTLZ1. (b) DTLZ2. (c) DTLZ3. (d) DTLZ4.

the second is about the computational effort of the concerned algorithms; and the third is about the comparison of our experimental findings to the existing ones in this paper.

In our experiments, we adopt two quality indicators for the purpose of comparison, i.e., IGD and HV, both of which can provide a combined information about convergence and diversity of a solution set. IGD used in this paper only applies to MOEAs based on reference points/directions, whereas HV applies to all MOEAs. We find that IGD and HV can usually result in consistent conclusions, but there exist a few exceptions. For example, in Table XV, θ-DEA* ($\theta = 5$) significantly outperform θ-DEA* ($\theta = 10^6$) on 8-, 10-, and 15-objective DTLZ1 in terms of IGD, but the HV results show that θ-DEA* ($\theta = 10^6$) performs significantly better. The main reason is that the task to find Pareto-optimal points close to the supplied reference points is not completely equivalent to the task to maximize the hypervolume, which strongly depends on the distribution of the reference points. But it is indeed desirable to evaluate reference points/directions based MOEAs using IGD indicator, since IGD can better measure the "closeness" between the outcome and goal for these algorithms. In addition, IGD can also be used in the scenario that the user is interested in only a preferred part of PF [29], where only a few representative reference points are used. It is worth pointing out that if we use more uniformly spread reference points in θ-DEA, it would be beneficial for the task to search for sparsely distributed Pareto-optimal points over the entire PF, and thus may enable θ-DEA to achieve better HV results.

Among the concerned algorithms, we find that MOEA/D and dMOPSO require the least computational effort. θ-DEA, MOMBI, and NSGA-III are also efficient enough to cope with the problem instances under the parameter specifications in this paper. GrEA, POGA, and SDE are generally much more computation expensive than the above mentioned algorithms, whose computational time increases sharply with increasing number of objectives. We would like to specially mention HypE. In the original HypE study [52], the authors used a small population size, i.e., 50, and claimed that HypE is a fast algorithm. However, we find that the computation time of HypE would increase severely not only with the increase in the

TABLE XV
PERFORMANCE COMPARISON BETWEEN θ-DEA* ($\theta = 5$) AND θ-DEA* ($\theta = 10^6$) IN TERMS OF AVERAGE IGD AND HV VALUES. THE PERFORMANCE THAT IS SIGNIFICANTLY BETTER THAN THE OTHER IS SHOWN IN BOLD

Problem	m	MaxGen	IGD		HV	
			θ-DEA* ($\theta = 5$)	θ-DEA* ($\theta = 10^6$)	θ-DEA* ($\theta = 5$)	θ-DEA* ($\theta = 10^6$)
DTLZ1	3	400	**1.250E-03**	1.876E-03	1.118329	1.117384
	5	600	**5.163E-04**	9.613E-04	**1.577892**	1.577693
	8	750	**2.172E-03**	3.422E-03	2.137998	**2.138139**
	10	1000	**2.235E-03**	3.282E-03	2.592696	**2.592832**
	15	1500	**3.626E-03**	6.688E-03	4.175713	**4.176786**
DTLZ2	3	250	**9.735E-04**	1.573E-03	**0.744320**	0.743924
	5	350	**2.135E-03**	4.461E-03	**1.307368**	1.304332
	8	500	**6.902E-03**	1.549E-02	**1.978469**	1.969621
	10	750	**6.725E-03**	1.576E-02	**2.514485**	2.508654
	15	1000	**8.670E-03**	1.993E-02	**4.137766**	4.132136
DTLZ3	3	1000	3.519E-03	5.202E-03	0.738977	0.736557
	5	1000	**2.565E-03**	4.723E-03	**1.305846**	1.302989
	8	1000	**1.290E-02**	3.094E-02	**1.970805**	1.954208
	10	1500	**6.361E-03**	1.541E-02	**2.514027**	2.508178
	15	2000	**1.102E-02**	9.510E-02	**4.136332**	3.989831
DTLZ4	3	600	2.396E-01	2.282E-01	0.602951	0.603710
	5	1000	**3.134E-04**	1.809E-02	**1.308934**	1.299265
	8	1250	**1.443E-02**	1.527E-02	**1.977231**	1.976809
	10	2000	**3.375E-03**	4.251E-03	**2.515468**	2.515248
	15	3000	**5.729E-03**	2.206E-02	**4.138225**	4.135672

number of objectives but also with the increase in population size. Hence, under the parameter settings in our experiments, HypE is indeed the most time-consuming algorithm. The similar observation can be referred in [75], where the computation time of HypE is reported. It should be mentioned that, not just HypE, the efficiency of the other considered algorithms is also influenced by the population size. However, HypE seems to be more affected than the others based on our experimental observations.

Although there have been a number of experimental studies on the comparison of many-objective optimizers, the parameter settings, the adopted algorithm variants (e.g., MOEA/D has several variants), the termination criterion, and the problem settings (e.g., the number of objectives and the number of decision variables) usually vary from study to study, making a rigorous comparison between the experimental findings impractical. However, this paper represents one of the most comprehensive experimental comparisons in the literature so far. A number of different research issues have been investigated, as presented in Section V.

VI. CONCLUSION

In this paper, we have presented a new many-objective evolutionary algorithm, called θ-DEA, whose environmental selection mechanism is based on θ dominance. Given the complementary advantages of NSGA-III and MOEA/D, θ-DEA is expected to enhance the convergence ability of NSGA-III in high-dimensional objective space by utilizing the aggregation function-based fitness evaluation scheme in MOEA/D, thereby achieving a better compromise between convergence and diversity in many-objective optimization. To achieve the goal, a new dominance relation, θ dominance, is introduced into the proposed algorithm, managing to emphasize both convergence and diversity.

In the experimental investigation of θ-DEA, we have shown the evidence of the robustness of the algorithm with respect to the key parameter θ. We have experimentally verified that the θ-DEA, in general, performs better than NSGA-III and MOEA/D in searching for the Pareto-optimal points close to the supplied reference points as a reference point-based algorithm. It has also been found that the embedded normalization procedure enables θ-DEA to handle scaled problems more effectively.

To demonstrate the strong competitiveness, we have made an extensive experimental comparison of θ-DEA with eight state-of-the-art algorithms that belong to five different categories of technologies. A number of well-known benchmark problems are chosen to challenge different abilities of the algorithms. The comparison results reveal that the proposed θ-DEA works well on almost all the problem instances considered in this paper, and it is compared favorably with state-of-the-art many-objective optimizers. However, we have also observed that none of the algorithms is capable of beating any of the other algorithms on all the instances, which indicates that a careful choice of algorithms is sill needed at present when solving a many-objective problem at hand.

In the future, we will have a deeper insight into the search behavior of θ-DEA, so as to further improve its performance. It is also necessary to evaluate θ-DEA further by comparing it to the two most recent many-objective optimization algorithms [25], [32], so that the strength and weakness of different algorithms can be better understood. It would also be interesting to extend our θ-DEA to solve constrained many-objective problems by incorporating constraint handling techniques [84]. Moreover, we would apply θ-DEA to real-world problems in order to further verify its effectiveness.

APPENDIX

The source code of the proposed θ-DEA is available online: http://learn.tsinghua.edu.cn:8080/2012310563/ManyEAs.rar

REFERENCES

[1] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 2419–2426.
[2] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Proc. Evol. Multi-Criter. Optim.*, Matsushima, Japan, 2007, pp. 742–756.
[3] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *Proc. Evol. Multi-Criter. Optim.*, Guanajuato, Mexico, 2005, pp. 14–32.
[4] J. G. Herrero, A. Berlanga, and J. M. M. Lopez, "Effective evolutionary algorithms for many-specifications attainment: Application to air traffic control tracking filters," *IEEE Trans. Evol. Comput.*, vol. 13, no. 1, pp. 151–168, Feb. 2009.
[5] A. Sülflow, N. Drechsler, and R. Drechsler, "Robust multi-objective optimization in high dimensional spaces," in *Proc. Evol. Multi-Criter. Optim.*, Matsushima, Japan, 2007, pp. 715–726.
[6] Y. Yuan and H. Xu, "Multiobjective flexible job shop scheduling using memetic algorithms," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 1, pp. 336–353, Jan. 2015.
[7] K. Praditwong, M. Harman, and X. Yao, "Software module clustering as a multi-objective search problem," *IEEE Trans. Softw. Eng.*, vol. 37, no. 2, pp. 264–282, Mar./Apr. 2011.

[8] A. S. Sayyad, T. Menzies, and H. Ammar, "On the value of user preferences in search-based software engineering: A case study in software product lines," in *Proc. 35th Int. Conf. Softw. Eng.*, San Francisco, CA, USA, 2013, pp. 492–501.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[10] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," in *Proc. Evol. Methods Design Optim. Control Appl. Ind. Prob.*, Athens, Greece, 2002, pp. 95–100.

[11] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proc. 3rd Annu. Conf. Genet. Evol. Comput.*, San Francisco, CA, USA, 2001, pp. 283–290.

[12] 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.*, vol. 2. Seoul, Korea, 2001, pp. 957–962.

[13] V. Khare, X. Yao, and K. Deb, "Performance scaling of multi-objective evolutionary algorithms," in *Proc. Evol. Multi-Criter. Optim.*, Faro, Portugal, 2003, pp. 376–390.

[14] R. C. Purshouse and P. J. Fleming, "Evolutionary many-objective optimisation: An exploratory analysis," in *Proc. IEEE Congr. Evol. Comput.*, vol. 3. Canberra, ACT, Australia, 2003, pp. 2066–2073.

[15] N. Drechsler, R. Drechsler, and B. Becker, "Multi-objective optimisation based on relation favour," in *Proc. Evol. Multi-Criter. Optim.*, Zurich, Switzerland, 2001, pp. 154–166.

[16] 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.

[17] 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.

[18] 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.

[19] M. Köppen, R. Vicente-Garcia, and B. Nickolay, "Fuzzy-Pareto-dominance and its application in evolutionary multi-objective optimization," in *Proc. Evol. Multi-Criter. Optim.*, Guanajuato, Mexico, 2005, pp. 399–412.

[20] 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.

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

[22] 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.

[23] 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.

[24] A. López, C. A. C. Coello, A. Oyama, and K. Fujii, "An alternative preference relation to deal with many-objective optimization problems," in *Proc. Evol. Multi-Criter. Optim.*, Sheffield, U.K., 2013, pp. 291–306.

[25] P. C. Roy, M. M. Islam, K. Murase, and X. Yao, "Evolutionary path control strategy for solving many-objective optimization problem," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 702–715, 2015.

[26] 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.

[27] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Proc. Evol. Multi-Criter. Optim.*, Matsushima, Japan, 2007, pp. 727–741.

[28] 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.

[29] 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.

[30] 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.

[31] K. Praditwong and X. Yao, "A new multi-objective evolutionary optimisation algorithm: The two-archive algorithm," in *Proc. Int. Conf. Comput. Intell. Security*, 2006, pp. 286–291.

[32] H. Wang, L. Jiao, and X. Yao, "An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, in press.

[33] 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.

[34] 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.

[35] S. Z. Martínez and C. A. C. Coello, "A multi-objective particle swarm optimizer based on decomposition," in *Proc. 13th Annu. Conf. Genet. Evol. Comput.*, Dubin, Ireland, 2011, pp. 69–76.

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

[37] 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.

[38] S. Z. Martinez and C. A. C. Coello, "A direct local search mechanism for decomposition-based multi-objective evolutionary algorithms," in *Proc. IEEE Congr. Evol. Comput.*, Brisbane, QLD, Australia, 2012, pp. 1–8.

[39] H. Ishibuchi, Y. Sakane, N. Tsukamoto, and Y. Nojima, "Adaptation of scalarizing functions in MOEA/D: An adaptive scalarizing function-based multiobjective evolutionary algorithm," in *Proc. Evol. Multi-Criter. Optim.*, Nantes, France, 2009, pp. 438–452.

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

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

[42] 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.

[43] M. Fleischer, "The measure of Pareto optima applications to multi-objective metaheuristics," in *Proc. Evol. Multi-Criter. Optim.*, Faro, Portugal, 2003, pp. 519–533.

[44] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Prob. Solv. Nat.*, Birmingham, U.K., 2004, pp. 832–842.

[45] 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.

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

[47] K. Bringmann and T. Friedrich, "Don't be greedy when calculating hypervolume contributions," in *Proc. 10th ACM SIGEVO Workshop Found. Genet. Algorithms*, Orlando, FL, USA, 2009, pp. 103–112.

[48] C. A. R. Villalobos and C. A. C. Coello, "A new multi-objective evolutionary algorithm based on a performance assessment indicator," in *Proc. 14th Annu. Conf. Genet. Evol. Comput.*, Philadelphia, PA, USA, 2012, pp. 505–512.

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

[50] 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.

[51] L. M. Russo and A. P. Francisco, "Quick hypervolume," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 481–502, Aug. 2014.

[52] 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.

[53] D. Brockhoff, T. Wagner, and H. Trautmann, "On the properties of the R2 indicator," in *Proc. 14th Annu. Conf. Genet. Evol. Comput.*, Philadelphia, PA, USA, 2012, pp. 465–472.

[54] R. H. Gomez and C. A. C. Coello, "MOMBI: A new metaheuristic for many-objective optimization based on the R2 indicator," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 2488–2495.

[55] H. Trautmann, T. Wagner, and D. Brockhoff, "R2-EMOA: Focused multiobjective search using R2-indicator-based selection," in *Learning and Intelligent Optimization*. Berlin, Germany: Springer, 2013, pp. 70–74.

[56] K. Bringmann, T. Friedrich, F. Neumann, and M. Wagner, "Approximation-guided evolutionary multi-objective optimization," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, 2011, pp. 1198–1203.
[57] M. Wagner and F. Neumann, "A fast approximation-guided evolutionary multi-objective algorithm," in *Proc. 15th Annu. Conf. Genet. Evol. Comput. Conf.*, Amsterdam, The Netherlands, 2013, pp. 687–694.
[58] J. R. Figueira, A. Liefooghe, E.-G. Talbi, and A. P. Wierzbicki, "A parallel multiple reference point approach for multi-objective optimization," *Eur. J. Oper. Res.*, vol. 205, no. 2, pp. 390–400, 2010.
[59] H. J. Moen, N. B. Hansen, H. Hovland, and J. Tørresen, "Many-objective optimization using taxi-cab surface evolutionary algorithm," in *Proc. Evol. Multi-Criter. Optim.*, Sheffield, U.K., 2013, pp. 128–142.
[60] A. Zhou, Q. Zhang, Y. Jin, and B. Sendhoff, "Combination of EDA and DE for continuous biobjective optimization," in *Proc. IEEE Congr. Evol. Comput.*, Hong Kong, 2008, pp. 1447–1454.
[61] A. Zhou, Q. Zhang, and G. Zhang, "Approximation model guided selection for evolutionary multiobjective optimization," in *Proc. Evol. Multi-Criter. Optim.*, Sheffield, U.K., 2013, pp. 398–412.
[62] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, Aug. 2013.
[63] M. Li, S. Yang, X. Liu, and R. Shen, "A comparative study on evolutionary algorithms for many-objective optimization," in *Proc. Evol. Multi-Criter. Optim.*, Sheffield, U.K., 2013, pp. 261–275.
[64] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Addison-Wesley Longman, 1989.
[65] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Syst.*, vol. 9, no. 2, pp. 115–148, 1995.
[66] S. Kukkonen and K. Deb, "Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, Vancouver, BC, Canada, 2006, pp. 1179–1186.
[67] 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.
[68] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multi-objective genetic algorithms," in *Proc. Evol. Multi-Criter. Optim.*, Zurich, Switzerland, 2001, pp. 82–95.
[69] K. Deb and D. E. Goldberg, "An investigation of niche and species formation in genetic function optimization," in *Proc. 3rd Int. Conf. Genet. Algorithms*, Fairfax, VA, USA, 1989, pp. 42–50.
[70] K. Deb and K. Miettinen, "A review of nadir point estimation procedures using evolutionary approaches: A tale of dimensionality reduction," in *Proc. Multiple Criter. Decis. Making Conf.*, Auckland, New Zealand, 2009, pp. 1–14.
[71] 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.
[72] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable multi-objective optimization test problems," Inst. Commun. Inf. Technol., ETH Zurich, Zurich, Switzerland, TIK Tech. Rep 112, 2001.
[73] 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.
[74] 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.
[75] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multi-objective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, Apr. 2015.
[76] A. Auger, J. Bader, D. Brockhoff, and E. Zitzler, "Theory of the hypervolume indicator: Optimal μ-distributions and the choice of the reference point," in *Proc. 10th ACM SIGEVO Workshop Found. Genet. Algorithms*, Orlando, FL, USA, 2009, pp. 87–102.
[77] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space," in *Proc. Int. Conf. Parallel Prob. Solv. Nat.*, Krakow, Poland, 2010, pp. 91–100.
[78] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[79] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4. Perth, WA, Australia, 1995, pp. 1942–1948.
[80] J. J. Durillo and A. J. Nebro, "jMetal: A java framework for multi-objective optimization," *Adv. Eng. Softw.*, vol. 42, no. 10, pp. 760–771, 2011.
[81] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics*, vol. 1, no. 6, pp. 80–83, 1945.
[82] H. Ishibuchi, N. Akedo, and Y. Nojima, "A study on the specification of a scalarizing function in MOEA/D for many-objective knapsack problems," in *Learning and Intelligent Optimization*. Berlin, Germany: Springer, 2013, pp. 231–246.
[83] L. M. Antonio and C. A. C. Coello, "Use of cooperative coevolution for solving large scale multiobjective optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, Cancún, Mexico, 2013, pp. 2758–2765.
[84] E. Mezura-Montes and C. A. C. Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm Evol. Comput.*, vol. 1, no. 4, pp. 173–194, 2011.

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

From 2014 to 2015 he was a visiting Ph.D. student with the Centre of Excellence for Research in Computational Intelligence and Applications, University of Birmingham, Birmingham, U.K. His research interests include evolutionary multiobjective optimization, statistical machine learning, and their applications.

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

He is an Associate Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include data mining, intelligent information processing, and advanced process controllers for IC manufacturing equipments. He has published over 50 academic papers, and holds ten invention patents on advanced controllers and the copyrights to six software systems.

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

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

His research interests include evolutionary computation and machine learning.

Xin Yao (F'03) is a Chair (Professor) of Computer Science and the Director of the Centre of Excellence for Research in Computational Intelligence and Applications, University of Birmingham, U.K. His research interests include evolutionary computation and ensemble learning. He has published over 400 refereed publications in international journals and conferences.

Prof. Yao received the 2001 IEEE Donald G. Fink Prize Paper Award, the 2010 IEEE Transactions on Evolutionary Computation Outstanding Paper Award, the 2010 BT Gordon Radley Award for Best Author of Innovation (Finalist), the 2011 IEEE Transactions on Neural Networks Outstanding Paper Award, the prestigious Royal Society Wolfson Research Merit Award in 2012, the IEEE CIS Evolutionary Computation Pioneer Award in 2013, and several other best paper awards. He was the Editor-in-Chief of IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION from 2005 to 2008. He is the President of the IEEE Computational Intelligence Society from 2014 to 2015.