A Scalar Projection and Angle-Based Evolutionary Algorithm for Many-Objective Optimization Problems

Yuren Zhou, Yi Xiang, Zefeng Chen, Jun He, *Senior Member, IEEE*, and Jiahai Wang, *Member, IEEE*

Abstract—In decomposition-based multiobjective evolutionary algorithms, the setting of search directions (or weight vectors), and the choice of reference points (i.e., the ideal point or the nadir point) in scalarizing functions, are of great importance to the performance of the algorithms. This paper proposes a new decomposition-based many-objective optimizer by simultaneously using adaptive search directions and two reference points. For each parent, binary search directions are constructed by using its objective vector and the two reference points. Each individual is simultaneously evaluated on two fitness functions—which are motivated by scalar projections—that are deduced to be the differences between two penalty-based boundary intersection (PBI) functions, and two inverted PBI functions, respectively. Solutions with the best value on each fitness function are emphasized. Moreover, an angle-based elimination procedure is adopted to select diversified solutions for the next generation. The use of adaptive search directions aims at effectively handling problems with irregular Pareto-optimal fronts, and the philosophy of using the ideal and nadir points simultaneously is to take advantages of the complementary effects of the two points when handling problems with either concave or convex fronts. The performance of the proposed algorithm is compared with seven state-of-the-art multi-/many-objective evolutionary algorithms on 32 test problems with up to 15 objectives. It is shown by the experimental results that the proposed algorithm is flexible when handling problems with different types of Pareto-optimal fronts, obtaining promising results regarding both the quality of the returned solution set and the efficiency of the new algorithm.

Index Terms—Dynamic decomposition, evolutionary algorithms, many-objective optimization, reference points.

I. INTRODUCTION

MULTI-OBJECTIVE optimization problems (MOPs) with at least four conflicting objectives are known as many-objective optimization problems (MaOPs) [1], [2]. Due to extensive existences of MaOPs in real-world applications, such as automotive engine calibration [3], water resource system planning [4], car controller optimization [5], and optimal product selection from software product lines [6], they have recently drawn steady attention in the evolutionary multiobjective optimization community. A number of many-objective evolutionary algorithms have been specially designed to handle MaOPs [7]–[15]

In this paper, the following unconstrained MOP (or MaOP) is considered:

$$\text{Minimize } \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T$$
$$\text{subject to: } \mathbf{x} \in \Omega \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is the decision vector, and n denotes the number of decision variables. In MOP (1), $\Omega \subseteq \mathbb{R}^n$ is called the decision space; $\mathbf{F}(\mathbf{x}) \in \mathbb{R}^m$, denoting the objective vector of $\mathbf{x}$, consists of m objective functions $f_i(\mathbf{x}), i = 1, 2, \ldots, m$. Due to the nature of conflict among all the objectives, there is no single optimal solution available for an MOP, but a set of solutions representing tradeoffs among different objectives. For solutions $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{x}$ is said to Pareto dominate $\mathbf{y}$ if and only if $f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ holds for every $1 \leq i \leq m$ and there exists at least one $j \in \{1, 2, \ldots, m\}$ such that $f_j(\mathbf{x}) < f_j(\mathbf{y})$. If neither $\mathbf{x}$ Pareto dominates $\mathbf{y}$ nor $\mathbf{y}$ Pareto dominates $\mathbf{x}$, then they are Pareto nondominated with each other. A solution $\mathbf{x}^*$ is Pareto-optimal if there is no other solution $\mathbf{x} \in \Omega$ such that $\mathbf{x}$ Pareto dominates $\mathbf{x}^*$. The $\mathbf{F}(\mathbf{x}^*)$ is then called the Pareto-optimal (objective) vector. All the Pareto-optimal solutions constitute the Pareto-optimal set. Accordingly, the set of all the Pareto-optimal vectors is called the Pareto-optimal front (PF) [9].

Being simple, flexible, free from derivatives and being able to approximate the true PF with multiple solutions in a single run [16], [17], multiobjective evolutionary algorithms

Y. Zhou is with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China, and also with the Engineering Research Institute, Guangzhou College of South China University of Technology, Guangzhou 510800, China (e-mail: zhouyuren@mail.sysu.edu.cn).

Y. Xiang is with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China, and also with the Collaborative Innovation Center of High Performance Computing, Sun Yat-sen University, Guangzhou 510006, China (e-mail: gzhuxiang_yi@163.com).

Z. Chen and J. Wang are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China (e-mail: chzfeng@mail2.sysu.edu.cn; wangjiah@mail.sysu.edu.cn).

J. He is with the School of Science and Technology, Nottingham Trent University, Nottingham NG11 8NS, U.K. (e-mail: jun.he@ieee.org).

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

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

Digital Object Identifier 10.1109/TCYB.2018.2819360

(MOEAs) have achieved great successes when optimizing MOPs with mostly two or three objectives [18]–[22]. Since the output of MOEAs for an MOP is a set of Pareto nondominated solutions, Pareto dominance naturally becomes a feasible criterion for selecting individuals during the evolutionary process [23]. In differentiating between individuals for 2- or 3-objective MOPs, Pareto dominance is popular and effective. However, the performance of this criterion degenerates greatly on MaOPs mainly due to the fact that the number of Pareto nondominated solutions increases rapidly with the number of objectives [2], [7]. As a natural consequence, Pareto-based MOEAs, such as the nondominated sorting genetic algorithm (NSGA-II) [19] and the improved strength Pareto evolutionary algorithm [18], may suffer from great performance deterioration because of insufficient selection pressures toward the true PF. Although non-Pareto-based MOEAs, such as the decomposition-based (or aggregation-based) and indicator-based approaches, do not suffer from ineffectiveness in distinguishing individuals (because they do not rely on the Pareto-dominance to push the population toward the true PF), they may need to face the problem of diversity maintenance especially for problems with an irregular PF [24]. For decomposition-based approaches, such as the MOEA based on decomposition (MOEA/D) [21], one key point is the setting of weight vectors, which have significant influences on the distribution of a population [25], [26]. In indicator-based approaches, the population is guided by using an indicator, such as the hypervolume (HV) [27] and R2 indicator [28], which can simultaneously evaluate convergence and diversity [29], [30]. However, according to [29], in the calculation of HV, the choice of the reference point is a crucial issue. The HV may prefer the knee points and the boundary of the PF if the reference point is set improperly, which may make the final solutions obtained by HV-based MOEAs distributed not widely along the whole front [23], [31]. Similarly, R2-based approaches, if not designed properly, may also suffer from the loss of diversity. For example, the many-objective meta-heuristic based on the R2 indicator (MOMBI) [32] was experimentally demonstrated to be ineffective in maintaining a set of diversified solutions for some MaOPs [33]. Another shortcoming of indicator-based (especially HV-based) approaches is the high computational cost [34], [35], which seriously restricts their applications to MaOPs.

Decomposition-based MOEAs are very popular when handling both MOPs and MaOPs. In these algorithms, two issues are of great importance to the performance of the algorithms. One is the settings of weight vectors. According to a latest study [24], the performance of decomposition-based algorithms strongly depends on the shapes of the PFs. For problems with an irregular (i.e., discontinued, degenerated, etc.) PF, such as DTLZ5-7 and WFG3, decomposition-based algorithms with fixed weight vectors may suffer from performance degeneration as some weight vectors may have no intersection with the PF [24], or many subproblems can only find the solutions on the boundary of the PF [36]. Therefore, to deal with problems with irregular PFs, decomposition-based algorithms need to dynamically adjust weight vectors so as to adapt the distribution of search directions to the shape of the

PF [24], [37]–[39]. The other is the choice of the reference points in the scalarizing functions. The ideal point was widely used in most of the decomposition-based approaches, such as MOEA/D [21], MOEA/DD [9], NSGA-III [8], and reference vector guided evolutionary algorithm (RVEA) [40]. As explained in [36] and [41], this may be problematic sometimes. For example, for problems with convex PFs, scalarizing functions using the ideal point may pull most solutions toward the central region of the PF [36], [41]–[43]. It was demonstrated recently in [36] that the simultaneous use of both ideal and nadir points is a feasible way to improve the performance of decomposition-based algorithms for problems with both convex and concave PFs.

Given the above facts, this paper proposes a new decomposition-based many-objective optimizer which uses two reference points (i.e., both ideal and nadir points) and adaptive search directions. In the new algorithm, each solution is evaluated on two fitness functions which consider the ideal and nadir points, respectively. These fitness functions are defined based on the scalar projection and the perpendicular distance from the objective vector to a search direction. In addition, binary search directions are considered for each solution in the current population within which solutions are selected one by one according to the angle information. Since the proposed algorithm mainly adopts two basic concepts, i.e., scalar projection and angle, we use PAEA to name the new proposal. In PAEA, as discussed previously, the simultaneous use of two reference points aims at handling both convex and concave PFs, while adaptively adjusted search directions are designed for problems with irregular PFs. Main innovations of PAEA are summarized as follows.

1) The simultaneous use of two reference points. In PAEA, the search is guided by pulling the current solutions toward the ideal point, and pushing them away from the nadir point simultaneously. Since the effect of the use of the ideal point is complementary with that of the nadir point, PAEA is expected to be effective when handling both convex and concave PFs.

2) Adaptive multiple search directions. Each solution $\mathbf{x}_i$ in the current population defines binary search directions: one is the direction from $\mathbf{F}(\mathbf{x}_i)$ to the ideal point, while the other is the direction from the nadir point to $\mathbf{F}(\mathbf{x}_i)$. Moreover, solutions in the current population are dynamically selected from previous parent and child solutions according to the angle information. Therefore, the search directions are adaptively adjusted according to the distribution of current solutions.

3) The simultaneous evaluations of each solution on two fitness functions. Based on two reference points and two search directions for each parent individual, a child solution (or a neighboring solution) is simultaneously evaluated on two fitness functions which are deduced to be the differences between two penalty-based boundary intersection (PBI) functions [21], and two inverted PBI (IPBI) functions [41], respectively. Solutions with the best value on each fitness function are emphasized.

The rest of this paper is organized as follows. Section II summarizes related works in the field. Section III presents

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ZHOU *et al.*: SCALAR PROJECTION AND ANGLE-BASED EVOLUTIONARY ALGORITHM FOR MaOPs
3

details of our proposed PAEA, followed by the experimental study in Section IV. The discussions on the experimental results are given in Section V. Finally, Section VI concludes this paper and lists some research directions for future studies.

II. RELATED WORKS

To effectively handle MaOPs or complicated MOPs, many works have been done to improve the performance of Pareto-, decomposition-, and indicator-based algorithms.

For Pareto-based algorithms, many relaxed dominance relations have been proposed to increase the selection pressure, such as ϵ-dominance [44], grid dominance [10], and θ-dominance [45]. In addition, some customized diversity-based approaches [46], [47] have been injected into these algorithms to improve their performance. In indicator-based algorithms, HV and R2 have been widely used because these performance indicators can simultaneously evaluate convergence and diversity. Bader and Zitzler [11] proposed an algorithm named HypE, where the Monte Carlo simulation was used to approximate exact HV values. Hence, the efficiency of the algorithm has been improved significantly [48]. By improving the diversity of MOMBI [32], Gómez and Coello [33] suggested an improved algorithm MOMBI2 whose overall performance was demonstrated to be improved when solving MaOPs. Finally, the Two_Arch2 [49] can be seen as a hybrid many-objective algorithm, where both indicator-based and Pareto-based selection principles were used.

Decomposition-based algorithms were very popular when handling both MaOPs or complicated MOPs. By combining dominance- and decomposition-based approaches, Li *et al.* [9] proposed the MOEA/DD, where the convergence is addressed by the Pareto-dominance relation and scalarizing functions, and the diversity is maintained by a set of uniformly distributed weight vectors. To handle MaOPs more effectively, Deb and Jain [8] improved the NSGA-II algorithm by replacing the original crowding distance operator with a novel clustering operator, and by supplying a set of well-distributed reference lines to keep diversity among solutions. This leads to NSGA-III which was shown to be effective for MaOPs. Later, Yuan *et al.* [45] proposed the θ-DEA which enhanced the convergence of NSGA-III by exploiting the fitness evaluation scheme in decomposition-based MOEAs. In θ-DEA, each solution is assigned to its nearest reference line in the same manner as NSGA-III. The PBI function is used to rank solutions assigned to a same reference line. Similar to NSGA-III, the θ-DEA requires a set of reference lines for diversity maintenance. Cheng *et al.* [40] proposed an RVEA for MaOPs. In the proposed algorithm, a scalarization approach, named angle penalized distance, is used to balance convergence and diversity of solutions in a high-dimensional objective space.

The above decomposition-based algorithms, i.e., MOEA/DD, NSGA-III, θ-DEA, and RVEA, need to predefine a set of weight vectors or reference lines for diversity maintenance. However, on the one hand, how to set the weight vectors/reference points in a high dimensional objective space is still an open question [2]. For many-objective optimization, systematic approaches either generate a huge number of points in the unit simplex [50], or produce points distributed mainly on two layers in the hyper-plane [8], [9]. On the other hand, according to the studies in [24], the performance of decomposition-based algorithms strongly depends on the shapes of PFs, and they are particularly effective if the shape of the distribution of weight vectors/reference lines is the same as or similar to the shape of the problems' PFs. However, these algorithms with systematically generated weight vectors show severe performance deterioration on problems with irregular (i.e., discontinued, degenerated, and convex) fronts, because the shapes of the distribution of weight vectors are inconsistent with those of the problems' PFs. Therefore, it is of necessity to develop more flexible algorithms.

It was implied in [24] that there are two ways to improve the performance of decomposition-based algorithms. One is using dynamic weight vectors (or search directions) to adapt the shapes of the PFs. The other is adjusting reference points in scalarizing functions. Actually, there are already some works along the two research directions. To dynamically adjust weight vectors, Qi *et al.* [39] proposed an improved MOEA/D, where an adaptive weight vector adjustment (MOEA/D-AWA) is utilized to deal with MOPs with complex PFs. The weights are adjusted periodically so that the weights of subproblems can be redistributed adaptively. In the adaptive weight adjustment strategy, by introducing an external population, overcrowded subproblems are detected and removed, while new subproblems are added into the real sparse regions. Li and Landa-Silva [51] proposed an improved version of MOEA/D, called EMOSA, which incorporates the simulated annealing algorithm. In EMOSA, the weight vector of each subproblem is adaptively modified at the lowest temperature in order to make the search diversified toward unexplored parts of the PF. Gu *et al.* [52] suggested a dynamic weight design method based on the projection of the current nondominated solutions and an equidistant interpolation. The results indicated that the dynamic weight design method can dramatically improve the performance of MOEA/D. Jiang *et al.* [37] suggested a novel method called Pareto-adaptive weight vectors ($pa\lambda$) to automatically adjust weight vectors according to geometrical characteristics of the PFs. In the adaptive NSGA-III (A-NSGA-III) [38], Jain and Deb used a mechanism to adaptively add and delete reference points, depending on the crowdedness of population members on different parts of the current Pareto nondominated front. To use RVEA to handle irregular PFs, Cheng *et al.* [40] proposed a new reference vector regeneration method based on a "replacement" strategy, and it is more efficient than the "addition-and-deletion" as in A-NSGA-III.

In the scalarizing functions, different reference points have different search behaviors. In general, the ideal and nadir points are suitable for problems having concave and convex PFs, respectively. There are some works on the use of the nadir point or both of the ideal and nadir points in decomposition-based algorithms. Sato [41] proposed an MOEA/D variant with the IPBI function (MOEA/D-IPBI) which evolves solutions from the current nadir point by maximizing the scalarizing function value. In MOEA/D-IPBI, the nadir point is estimated

by finding the worst objective function value for each objective among all the current solutions. To approximate the whole PF of a given problem, Saborido *et al.* [42] proposed the GWASF-GA algorithm, where the fitness function is defined by an achievement scalarizing function (ASF) based on the Tchebycheff distance, in which both the utopian point (a point that is strictly better than the ideal point) and the nadir point are used as the reference points. It was shown that considering two reference points at the same time plays an important role in obtaining a final set of nondominated solutions that approximate the whole PF. Jiang and Yang [43] proposed the MOEA/D-TPN algorithm to handle complex MOPs. In the algorithm, the whole optimization process is divided into two phases. In the first phase, the ideal point is used in the scalarizing (Tchebycheff) function, while the nadir point may be used in the second phase if solutions found in the first phase are more crowded at the intermediate part of the approximated PF than at the boundaries. Recently, Wang *et al.* [36] studied the effect of the reference point setting on the performance of decomposition-based algorithms for problems with either concave or convex PFs. They proposed a new MOEA/D variant, i.e., MOEA/D-MR, where both ideal and nadir points are used. In the algorithm, the whole population is divided into two subpopulations. The first subpopulation uses the ideal point as the reference point, while the second one adopts the nadir point as the reference point. Experimental results on a set of complicated 2- and 3-objective test problems showed that the simultaneous use of two reference points indeed improves the performance of the algorithm.

By simultaneously considering the above two aspects, this paper proposes a new MOEA/D variant, i.e., PAEA, which uses both adaptive search directions and two reference points. The basic idea behind PAEA is using adaptive search directions to handle irregular PFs, and is taking advantages of complementary effects of both reference points so as to simultaneously deal with both concave and convex PFs. The proposed PAEA will be compared with other related algorithms on a large number of test problems whose PFs are either irregular (e.g., DTLZ5-DTLZ7 and WFG1-WFG3), or concave (e.g., DTLZ2-4 and WFG4-9), or convex (e.g., DTLZ2-4^{-1} and WFG4-9^{-1}).

III. PROPOSED PAEA ALGORITHM

In this section, we first give the general framework of the proposed approach, then we present details of main algorithmic components in each section.

A. General Framework of PAEA

The framework of the proposed PAEA is shown in Algorithm 1. In PAEA, apart from the population size N, there are two additional parameters θ and α which are used in the fitness functions and the handling of extreme solutions, respectively. Details of the above two parameters will be given in Sections III-C and III-D, respectively. First, a population P with N individuals is initialized within the whole decision space (line 1 in Algorithm 1). Then, for each individual $\mathbf{x}$ in P, a random solution (denoted by $\mathbf{x}'$, which is different from

---

Algorithm 1 Framework of the Proposed Algorithm (PAEA)

Input:
> N (population size), θ (a parameter used in the fitness functions) and α (a parameter used in handling extreme solutions).

Output: The final population.
1: $P \leftarrow initialization(N)$ // Generate an initial population with N individuals.
2: **while** *the termination criterion is not fulfilled* **do**
3:     $Q \leftarrow variation(P)$ // Generate $2N$ offspring solutions by using genetic operators
4:     $P' \leftarrow environmentalSelection(P, Q)$ // Maintain a diversified population with N individuals
5:     $P \leftarrow P'$
6: **end while**
7: **return** P

Algorithm 2 $P' \leftarrow environmentalSelection(P, Q)$
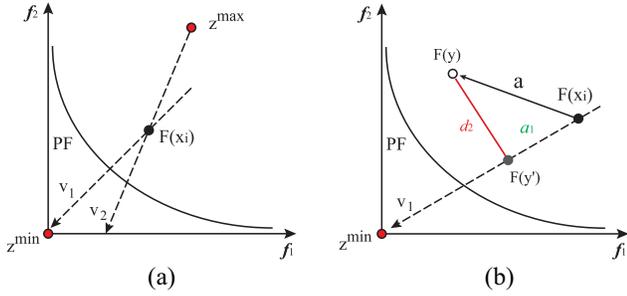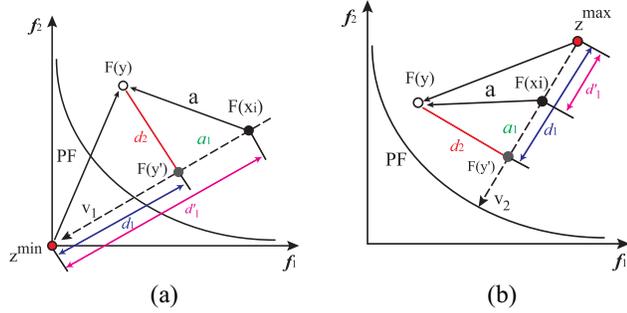
Input: P, Q
Output: The new population P'
1: Normalize members in $P \cup Q$ using the method in [8]
2: $S \leftarrow binaryDirectionsSelection(P, Q)$
3: $P' \leftarrow elimination(S)$ // Select N solutions by eliminating individuals from S one by one
4: **return** P'

$\mathbf{x}$) is selected from the whole population. By applying genetic operators (crossover and mutation) to $\mathbf{x}$ and $\mathbf{x}'$, we can get two offspring of $\mathbf{x}$, i.e., $\mathbf{y}_1$ and $\mathbf{y}_2$. All the offspring are stored in the population Q (line 3 in Algorithm 1). Since each parent generates two offspring at a time, this will consume evaluations as twice as the population size at each generation. Finally, the environmental selection is adopted to select N diversified individuals from both P and Q (line 4 in Algorithm 1). The above procedures are repeated until the termination criterion is fulfilled. In the following sections, we will describe algorithmic components in more details.

B. Environmental Selection

The pseudo-code of the environmental selection is given in Algorithm 2. Since the optimization problems may have different ranges for each objective, the population $P \cup Q$ is recommended to be normalized. In PAEA, we adopt the same method as in NSGA-III [8] to adaptively normalize $P \cup Q$ (line 1 in Algorithm 2). The advantage of the normalization of $P \cup Q$ is that it considers the normalization of both parent and offspring individuals at the same time. Details of this normalization technique can be found in [8]. Hereafter, when objective values of a solution are mentioned, we always refer to the normalized ones.

Now, we consider the objective space. After normalization, the ideal point $\mathbf{z}^{\min}$ becomes a zero vector, and the nadir point $\mathbf{z}^{\max} = (z_1^{\max}, z_2^{\max}, \ldots, z_m^{\max})^T$ can be constructed by finding the maximum value for each objective. For each individual $\mathbf{x}_i$ in P, we consider two search directions $\mathbf{v}_1$ and $\mathbf{v}_2$ along which $\mathbf{F}(\mathbf{x}_i)$ approaches to the PF. As shown in Fig. 1(a),

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ZHOU *et al.*: SCALAR PROJECTION AND ANGLE-BASED EVOLUTIONARY ALGORITHM FOR MaOPs

5

Fig. 1.   Illustrations of (a) two search directions and (b) *scalar projection* a_1.

Fig. 2.   Illustrations of the calculation of (a) $g_1(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min})$ and (b) $g_2(\mathbf{y}|\mathbf{v}_2, \mathbf{z}^{\max})$.

$\mathbf{v}_1 = \mathbf{z}^{\min} - \mathbf{F}(\mathbf{x}_i)$ and $\mathbf{v}_2 = \mathbf{F}(\mathbf{x}_i) - \mathbf{z}^{\max}$. Along these directions, a solution set S is constructed by selecting promising solutions from both P and Q with the help of a *binary directions selection* procedure (line 2 in Algorithm 2).

As will be shown in Section III-C, the number of solutions in S is $2N$. Therefore, some technique is needed to prune the population S so as to retain exactly N solutions. In PAEA, the *elimination* function (line 3 in Algorithm 2) is designed for this purpose, where effective techniques are developed to handle extreme solutions and to eliminate solutions one by one. More details of this procedure will be given in Section III-D.

C. Binary Directions Selection

As its name suggests, the binary directions selection chooses elite individuals in two directions in terms of the fitness value of an individual. A crucial issue here is how to measure the quality of individuals. Ideally, the fitness of an individual should reflect information concerning both convergence and diversity. As shown in Fig. 1(b), $\mathbf{a}$ is a vector starting from $\mathbf{F}(\mathbf{x}_i)$ and ending up with $\mathbf{F}(\mathbf{y})$, where $\mathbf{y}$ is one of $\mathbf{x}_i$'s two children. On the one hand, a_1, given by $\|\mathbf{a}\| \cos\langle \mathbf{a}, \mathbf{v}_1 \rangle$, where $\langle \mathbf{a}, \mathbf{v}_1 \rangle$ denotes the angle between $\mathbf{a}$ and $\mathbf{v}_1$, is called the *scalar projection* of $\mathbf{a}$ onto the direction $\mathbf{v}_1$. The larger the a_1 is, the closer the $\mathbf{F}(\mathbf{y})$ approaches to the PF. Inversely, if we consider the negative value of a_1, i.e., $-a_1$, then a smaller value is preferable. Thus, $-a_1$ can be used as a measurement of convergence for an individual. On the other hand, since the diversity of the parent population is well kept by using an angle-based strategy (which will be described in Section III-D), we expect the offspring with better convergence is close to its parent. To this end, the perpendicular distance d_2 from $\mathbf{F}(\mathbf{y})$ to $\mathbf{v}_1$ can be used as a diversity measurement.

Algorithm 3 $S \leftarrow binaryDirectionsSelection\ (P,\ Q)$
Input: P, Q
Output: S
1: $S \leftarrow \emptyset$
2: **for** each $\mathbf{x}_i \in P$ **do**
3:   Calculate fitness values of $\mathbf{x}_i$ and its two child solutions $\mathbf{y}_1$ and $\mathbf{y}_2$ by Eqs. (3) and (4)
4:   Two solutions $\mathbf{y}_1'$ and $\mathbf{y}_2'$ with the smallest values on each fitness function [Eqs. (3) and (4)] are selected from the set $\{\mathbf{x}_i, \mathbf{y}_1, \mathbf{y}_2\}$
5:   **if** $\mathbf{y}_1' = \mathbf{y}_2'$ **then**
6:     Select the solution with the second best value on either g_1 or g_2, and use it to replace $\mathbf{y}_2'$ // g_1 or g_2 is chosen randomly with a probability 0.5
7:   **end if**
8:   $S \leftarrow S \cup \{\mathbf{y}_1', \mathbf{y}_2'\}$ // Add $\mathbf{y}_1'$ and $\mathbf{y}_2'$ into S
9: **end for**
10: **return** S

Therefore, for an individual $\mathbf{y}$, its convergence and diversity (in the direction of $\mathbf{v}_1$) are measured by $-a_1$ and d_2, respectively. Thus, the fitness of $\mathbf{y}$ can be defined as

$$g_1\left(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min}\right) = (-a_1) + \theta d_2 \qquad (2)$$

where $\theta > 0$ is a control parameter that is used to keep a balance between convergence and diversity. Actually, (2) can be represented by two PBI functions as $g^{\text{pbi}}(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min}) - g^{\text{pbi}}(\mathbf{x}_i|\mathbf{v}_1, \mathbf{z}^{\min})$, where $g^{\text{pbi}}(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min})$ and $g^{\text{pbi}}(\mathbf{x}_i|\mathbf{v}_1, \mathbf{z}^{\min})$ are the fitness value of $\mathbf{y}$ and $\mathbf{x}_i$ in the direction $\mathbf{v}_1$ according to the PBI decomposition approach [21]. As shown in Fig. 2(a), d_1 is the distance between $\mathbf{z}^{\min}$ and $\mathbf{F}(\mathbf{y}')$, i.e., the projection of $\mathbf{F}(\mathbf{y})$ on $\mathbf{v}_1$; d_2 is the distance between $\mathbf{F}(\mathbf{y})$ and $\mathbf{F}(\mathbf{y}')$; and d_1' is the distance between $\mathbf{z}^{\min}$ and $\mathbf{F}(\mathbf{x}_i)$.

Therefore, according to Fig. 2(a), we have

$$\begin{aligned}
g_1\left(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min}\right) &= -a_1 + \theta d_2 \\
&= \left(d_1 - d_1'\right) + \theta d_2 \\
&= (d_1 + \theta d_2) - \left(d_1' + \theta \cdot 0\right) \\
&= g^{\text{pbi}}\left(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min}\right) - g^{\text{pbi}}\left(\mathbf{x}_i|\mathbf{v}_1, \mathbf{z}^{\min}\right). \quad (3)
\end{aligned}$$

Similarly, the fitness of $\mathbf{y}$, denoted by $g_2(\mathbf{y}|\mathbf{v}_2, \mathbf{z}^{\max})$, can be also calculated in the direction $\mathbf{v}_2$ by using $\mathbf{z}^{\max}$ as a reference point. According to Fig. 2(b), this fitness value is actually the differences between two IPBI functions [41]. According to [41], IPBI function is defined by maximize $g^{\text{ipbi}}(\mathbf{y}|\mathbf{v}_2, \mathbf{z}^{\max}) = d_1 - \theta d_2$. In this paper, we consider the minus version, namely, minimize $g^{\text{ipbi}}(\mathbf{y}|\mathbf{v}_2, \mathbf{z}^{\max}) = -d_1 + \theta d_2$. Therefore, we obtain

$$\begin{aligned}
g_2\left(\mathbf{y}|\mathbf{v}_2, \mathbf{z}^{\max}\right) &= -a_1 + \theta d_2 \\
&= -\left(d_1 - d_1'\right) + \theta d_2 \\
&= (-d_1 + \theta d_2) - \left(-d_1' + \theta \cdot 0\right) \\
&= g^{\text{ipbi}}\left(\mathbf{y}|\mathbf{v}_2, \mathbf{z}^{\max}\right) - g^{\text{ipbi}}\left(\mathbf{x}_i|\mathbf{v}_2, \mathbf{z}^{\max}\right). \quad (4)
\end{aligned}$$

Fig. 3. Contour lines of $g_1(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min})$ with different values of θ, shown in a 2-objective space. (a) $\mathbf{F}(\mathbf{x}_i) = (0.8, 0.2)$. (b) $\mathbf{F}(\mathbf{x}_i) = (0.2, 0.8)$.
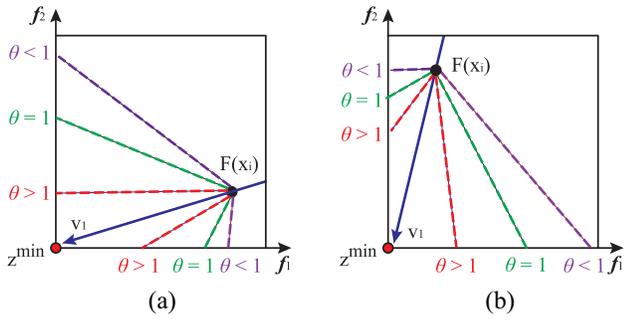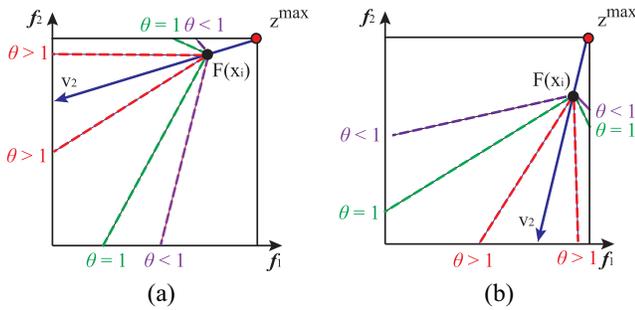

Fig. 4. Contour lines of $g_2(\mathbf{y}|\mathbf{v}_2, \mathbf{z}^{\max})$ with different values of θ, shown in a 2-objective space. (a) $\mathbf{F}(\mathbf{x}_i) = (0.7, 0.9)$. (b) $\mathbf{F}(\mathbf{x}_i) = (0.9, 0.7)$.

For the fitness defined by (3) and (4), a small value is preferable. With these fitness assignments, as shown in Algorithm 3, the binary directions selection procedure works as follows. For one parent solution $\mathbf{x}_i$ and its two child solutions $\mathbf{y}_1$ and $\mathbf{y}_2$, based on $g_1(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min})$ and $g_2(\mathbf{y}|\mathbf{v}_2, \mathbf{z}^{\max})$, two solutions with the best (smallest) values on each fitness function are selected (lines 3 and 4 in Algorithm 3). In case that they are identical, $\mathbf{y}_2'$ will be replaced by the solution with the second best value on either g_1 or g_2, which is randomly chosen with a probability 0.5 (lines 5–7 in Algorithm 3). The selected $\mathbf{y}_1'$ and $\mathbf{y}_2'$ are added into the set S (line 8 in Algorithm 3)

Finally, we analyze the search behaviors of PAEA in depth by showing contour lines of both $g_1(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min})$ and $g_2(\mathbf{y}|\mathbf{v}_2, \mathbf{z}^{\max})$, which are given in Figs. 3 and 4, respectively. According to (3), the second part $g^{\text{pbi}}(\mathbf{x}_i|\mathbf{v}_1, \mathbf{z}^{\min})$ is fixed for different $\mathbf{y}$'s, which is actually equal to the distance between $\mathbf{F}(\mathbf{x}_i)$ and $\mathbf{z}^{\min}$. Therefore, the contour lines of $g_1(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min})$ is similar to those of $g^{\text{pbi}}(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min})$. To be more specific, as shown in Fig. 3, the contour lines of $g_1(\mathbf{y}|\mathbf{v}_1, \mathbf{z}^{\min})$ are symmetrical about the search direction $\mathbf{v}_1$, with the angle between the two lines larger than, equal to and smaller than $\pi/2$ for $\theta < 1$, $\theta = 1$ and $\theta > 1$, respectively. Since $g_1(\mathbf{x}_i|\mathbf{v}_1, \mathbf{z}^{\min}) = g^{\text{pbi}}(\mathbf{x}_i|\mathbf{v}_1, \mathbf{z}^{\min}) - g^{\text{pbi}}(\mathbf{x}_i|\mathbf{v}_1, \mathbf{z}^{\min}) = 0$, solutions on the contour line have the same g_1 value (i.e., $g_1 = 0$) as the current solution $\mathbf{x}_i$. In Fig. 3(a), solutions in the region surrounded by the two contour lines of the same θ and the two axes have g_1 value smaller than 0. Thus, solutions in this region are better than $\mathbf{x}_i$. Hence, by using the function g_1, the current population is pulled toward the ideal point $\mathbf{z}^{\min}$ as close as possible. In a similar way, the contour lines of $g_2(\mathbf{y}|\mathbf{v}_2, \mathbf{z}^{\max})$ can be analyzed. As shown in

Algorithm 4 $P' \leftarrow elimination(S)$

Input: S
Output: The new population P'
1: $P' \leftarrow \emptyset$
2: Add m extreme solutions into P' and remove them from S
3: **while** $|P'| + |S| > N$ **do**
4:      Find $\mathbf{x}_r$ and $\mathbf{x}_t$ that have the minimum angle to each other among all the pairs of individuals in S
5:      $\mathbf{x} \leftarrow \arg\max\{\|\mathbf{F}(\mathbf{x}_r)\|, \|\mathbf{F}(\mathbf{x}_t)\|\}$ // Find the worse individual $\mathbf{x}$ in terms of the length of the objective vectors
6:      Remove $\mathbf{x}$ from S
7: **end while**
8: $P' \leftarrow P' \cup S$
9: **return** P'

Fig. 4, the positive effect of the fitness function g_2 is that it pushes the current population toward the PF so as to make solutions in the population away from the nadir point $\mathbf{z}^{\max}$ as far as possible.

Moreover, as shown in Fig. 3(a) and (b), if we use only g_1 (or equally $\mathbf{z}^{\min}$), all the solutions are pulled toward the ideal point, leading to over-crowdedness at middle parts of the PF in case that the problems have convex PFs [note in this case, as shown in Fig. 4(a) and (b), the use of g_2 (or equally $\mathbf{z}^{\max}$) would be helpful for finding more solutions in the boundary]. Similar problem occurs if we use only g_2 for a concave PF [36]. Therefore, the simultaneous use of both g_1 and g_2 is likely to improve the performance of PAEA when approximating both convex and concave PFs. The effect of the binary directions selection will be verified experimentally in Section VI-A in the supplementary material.

D. Elimination Procedure

Since S contains $2N$ solutions, an elimination procedure is needed to prune the population so as to retain exactly N solutions for the next generation. In PAEA, an angle-based procedure is used for this purpose, which is shown in Algorithm 4. The procedure is made up of two main parts: 1) the handling of extreme solutions and 2) the elimination of nonextreme solutions. Before the procedure starts, the acute angle (in the objective space) of every two individuals in S is calculated and stored in a matrix $\mathbf{M}_{2N \times 2N}$, and this operation needs $O(mN^2)$ multiplications. For every member in S, it has a unique identity. With the help of the angle matrix $\mathbf{M}_{2N \times 2N}$ and the identities of individuals, the angle between any two individuals can be obtained in the time complexity of $O(1)$.

For each objective k, we define a unit vector $\mathbf{e}_k = (0, \ldots, 1, \ldots, 0)$, where the kth element is 1 and all the other ones are 0's. For this vector, in the objective space, find the solution $\mathbf{x}_k$ to which $\mathbf{e}_k$ has the minimum angle. The $\mathbf{x}_k$ is called an *extreme solution*. The inclusion of extreme solutions may be good for the diversity promotion. However, some extreme solutions may be far away from the true PF. This will

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ZHOU *et al.*: SCALAR PROJECTION AND ANGLE-BASED EVOLUTIONARY ALGORITHM FOR MaOPs

7

have a side effect on the convergence of the whole population if these poorly converged solutions are directly included.

To handle the above problem, the following strategy is used in the proposed algorithm (refer to line 2 in Algorithm 4). First, find out the individuals $\mathbf{x}_k$ and $\mathbf{x}_h$ that have the minimum and the second minimum angle to $\mathbf{e}_k$. Second, a choice between $\mathbf{x}_k$ and $\mathbf{x}_h$ is made according to the length of their objective vectors. The $\|\mathbf{F}(\mathbf{x}_k)\|$ is the distance from $\mathbf{F}(\mathbf{x}_k)$ to $\mathbf{z}^{\min}$, which can reflect the convergence of the individual $\mathbf{x}_k$. The selection logic is as follows: if $\|\mathbf{F}(\mathbf{x}_k)\| - \|\mathbf{F}(\mathbf{x}_h)\| \leq \alpha \|\mathbf{F}(\mathbf{x}_h)\|$, then $\mathbf{x}_k$ is added into P', and is also removed from S. Otherwise, $\mathbf{x}_h$ is selected. Note that a parameter $\alpha > 0$ is used in the selection condition. According to this selection strategy, if $\mathbf{F}(\mathbf{x}_k)$ is extremely far away from the true PF, then it would not be included in the new population. Hence, the diversity and convergence can be balanced when adding extreme solutions.

For the remaining individuals in S, we first find out a pair of solutions, denoted by $\mathbf{x}_r$ and $\mathbf{x}_t$, which have the minimum angle among all the pairs of individuals (line 4 in Algorithm 4). Then, we identify the worse one (denoted by $\mathbf{x}$) in terms of the length of the objective vectors (line 5 in Algorithm 4). In case that a "tie" occurs, it will be broken randomly. Next, $\mathbf{x}$ is removed from S (line 6 in Algorithm 4). The above procedure is repeated if $|P'| + |S| > N$, where $|\cdot|$ is the cardinal number of a set. Finally, Algorithm 4 returns the union of P' and S.

It should be noted that He and Yen [53] have recently proposed a similar procedure to eliminate solutions one by one, which introduces a parameter t whose value should be set carefully according to characteristics of the problems at hand [53]. Our procedure, however, ignores the use of this parameter. In addition, we here present a fast implementation of the above elimination procedures. Naively, the time complexity of lines 3–7 in Algorithm 4 is $O(N^3)$. In each of the N loops, we need to find a pair of solutions with the minimum angle to each other from all the $O(N^2)$ pairs of solutions. Actually, lines 3–7 in Algorithm 4 can be speeded up by the following routine. First, we have the following preprocessing: sorting angle values of all pairs of solutions in an ascending order by using Quicksort [54]. Since there are $\binom{2N-m}{2}^1 = ([(2N-m)(2N-m-1)]/2) = O(N^2)$ pairs of solutions, this requires $O(N^2 \log N^2) = O(N^2 \log N)$ comparisons by using Quicksort. After sorting, the minimum angle can be found in the first place of the angle array. We delete one of the two solutions associated with this angle according to lines 5 and 6 in Algorithm 4. Subsequently, the angles related to the deleted solution should be removed from the array. From a programmatic perspective, this can be done by setting these angles to a large number (e.g., M). Note that, for each solution, the array indexes of angles related to this solution can be recorded during the quick sort process. Therefore, the marking of these angles needs only $O(N)$ operations. Then, the procedure continues scanning the array and the first angle that is not equal to M is the second minimum angle. Similarly, an associated solution is removed, so do angles related to this solution. The above procedures are repeated until N solutions

are removed. Therefore, the routine needs $O(N^2)$ deletions of angles (by marking them with M), and $O(N^2)$ scans of the angle array. As $O(N^2 \log N)$ is larger than $O(N^2)$, the worst time complexity of the above routine is $O(N^2 \log N)$, which is lower than $O(N^3)$.

The overall worst-case time complexity of PAEA at one generation is $\max\{O(N^2 \log N), O(mN^2)\}$. For detailed analyses and comparisons, we direct readers to Section V in the supplementary materials. Moreover, we give some discussions on the differences/relations between PAEA and related algorithms in Section I in the supplementary materials.

IV. EXPERIMENTAL STUDY

In this section, the proposed PAEA is compared with MOEA/DD [9], NSGA-III [8], MOEA/D [21], 1by1EA [55], GWASF-GA [42], MOEA/D-AWA [39], and MOEA/D-TPN [43] on a large number of test problems. These state-of-the-art algorithms were demonstrated to be effective when handling MaOPs. All the algorithms, except for 1by1EA, are reference point/weight vector based approaches. Similar to PAEA, the 1by1EA does not require any predefined weight vector. Since PAEA uses both ideal and nadir points, we include 1by1EA, GWASF-GA, and MOEA/D-TPN as peer algorithms as they also use the two reference points in different ways.

A. Test Problems

In the empirical study, we consider 32 test problems that are selected from two test suites. One is the DTLZ test suite consisting of 14 problems: DTLZ1−DTLZ7 [56], ConvexDTLZ2 [8], ScaledDTLZ1-2 [8], and DTLZ1^{-1}−DTLZ4^{-1} [24]. The other is the WFG test suite containing 18 test problems: WFG1−WFG9 [57] and WFG1^{-1}−WFG9^{-1} [24]. All the test problems can be scaled to any number of objectives. In this paper, $m = 5, 8, 10$, and 15 are considered.

In all the experiments, the number of decision variables for DTLZ test problems is set to $n = m + k - 1$, where $k = 5$ for DTLZ1 and ScaledDTLZ1, $k = 20$ for DTLZ7 and $k = 10$ for other DTLZ test problems [56]. According to suggestions in [57], the number of decision variables in WFG test problems is set to $n = 2 \times (m-1) + l$, where l is the distance-related variable that is set to 20. Recently, Ishibuchi *et al.* [24] have proposed the DTLZ^{-1} and WFG^{-1} test problems which are minus versions of DTLZ and WFG, respectively. The minus problems are created by multiplying all objectives in the original DTLZ and WFG by (-1). In this paper, the settings of decision variables and objectives are the same as in the original DTLZ and WFG test problems.

B. Performance Metrics

In this paper, the inverted generational distance (IGD) [58], [59], the generational distance (GD) [60] and the pure diversity (PD) [30] are chosen as performance metrics. The IGD can provide combined information on convergence and diversity of the obtained solutions, therefore it is widely used in the evaluation of approximated solution sets for both

---

[1]Since m extreme solutions are removed from S according to line 2 of Algorithm 4, there are exactly $2N - m$ remaining solutions in S.

TABLE I
POPULATION SIZE N FOR DIFFERENT NUMBERS OF OBJECTIVES

m	MOEA/DD, MOEA/D MOEA/D-AWA, MOEA/D-TPN	PAEA, NSGA-III 1by1EA	GWASF-GA
5	210	212	210
8	156	156	156
10	275	276	274
15	135	136	134

TABLE II
SETTINGS OF THE *maxFEs* FOR DIFFERENT NUMBERS OF
OBJECTIVES ON EACH TEST PROBLEM

m	DTLZ1	DTLZ2	DTLZ3	DTLZ4	Others
5	212×600	212×350	212×1000	212×1000	212×1250
8	156×750	156×500	156×1000	156×1250	156×1500
10	276×1000	276×750	276×1500	276×2000	276×2000
15	136×1500	136×1000	136×2000	136×3000	136×3000

MOPs [58], [61], [62] and MaOPs [8]–[10], [23], [49], [63]. The GD assesses only the convergence while PD gives a pure measurement on the diversity.

When calculating both IGD and GD, a reference point set is needed. In this paper, the same method as in [64] is used to sample points on the true PF. For more details, please refer to Section II-A in the supplementary material. For both IGD and GD, small values are preferable. The PD is a recently proposed diversity assessment in many-objective optimization. Details of the calculation of PD can be found in [30]. It is worth mentioning that an approximate set that is far away from the true PF may also present a satisfactory PD result if solutions are distributed properly. However, this PD value may be meaningless and may lead to misleading results. To avoid this situation, PD values in this paper are calculated only among those solutions whose objective values are within the regions of the true PF. Thus, worse converged solutions in any objective contribute zero to the PD value. In this way, the PD metric can also measure the convergence to some extent. For PD metric, a large value is desirable. In the study, this metric is implemented by the code[2] provided in [30].

C. General Experimental Settings

The experimental settings in this paper are listed below unless otherwise mentioned.
1) *Population Size N:* According to [8], the population size in NSGA-III is set to the smallest multiple of four larger than the number of weight vectors (denoted by H), which are created by the Das and Dennis's systematic approach [50] when $m = 5$ and by the two-layer weight/reference vectors generation method [8], [9] when $m > 5$. In PAEA and 1by1EA, the population size keeps the same as in NSGA-III. While in MOEA/DD, MOEA/D, MOEA/D-AWA, and MOEA/D-TPN, the population size is set to H directly. For GWASF-GA, N is set to H when $m = 5$ and 8, and $H - 1$ when $m = 10$ and 15. Since a binary tournament selection is used in GWASF-GA, it requires the population size to be even. In case that the number of generated weight vectors is odd, a random one will be removed as done in [42]. The values of N for different numbers of objectives are summarized in Table I.
2) *Number of Independent Runs and the Termination Condition:* All algorithms are independently run 30 times in each test instance, and are terminated when the

---

[2]The code of PD can be found at http://www.surrey.ac.uk/cs/people/handing_wang/.

objective function evaluations reach *maxFEs*. The settings of *maxFEs* for different numbers of objectives on each test problem are summarized in Table II.
3) *Algorithmic Parameter Settings:* In our proposed PAEA, there are two additional parameters θ and α, which are set to 10 and 0.5, respectively. The parameter study on θ and α is available in Section IV in the supplementary material. For other peer algorithms, the parameter settings are kept the same as in their original studies. More details can be found in Section II-B in the supplementary material.

D. Comparison of Computational Results

For performance comparisons, we first record the average IGD and PD for the DTLZ test suite as shown in Tables II and III in the supplementary material. As we can see from these tables, PAEA obtains the best or the second best results in most of the test instances, showing a great superiority over other state-of-the-art algorithms. Specifically, among all the 56 DTLZ and DTLZ^{-1} test instances, the proportion of the best or the second best results PAEA obtains is $34/56 \approx 61\%$ and $52/56 \approx 93\%$ for IGD and PD, respectively. To make statistical comparisons, the Wilcoxon's rank sum test [65] is applied to determining whether the differences between PAEA and each peer algorithm in each test instance are significant or not. The symbol ● denotes that PAEA shows a significant improvement over its competitors with a level of significance $\alpha = 0.05$, while ○ indicates the opposite. If no significant difference is found, then the symbol ‡ will be used. The test results on DTLZ test suite are summarized in Table III, where we can find the proportion of test instances where PAEA is better than (●), worse than (○) and equal to (‡) each of the peer algorithms.

It can be observed from Table III that the proportion of the DTLZ and DTLZ^{-1} test instances where PAEA obtains significantly better IGD than its competitors is 44/56, 37/56, 48/56, 33/56, 42/56, 39/56, and 48/56 for MOEA/DD, NSGA-III, MOEA/D, 1by1EA, GWASF-GA, MOEA/D-AWA, and MOEAD-TPN, respectively. Conversely, the proportion of test instances where PAEA is inferior to the peer algorithms is 8/56, 15/56, 8/56, 16/56, 13/56, 16/56, and 7/56, respectively. For the PD metric, as shown in Table III, PAEA outperforms GWASF-GA in all the 56 test instances, and outperforms both MOEA/DD and MOEA/D in 55 test instances. Compared with NSGA-III, 1by1EA, MOEA/D-AWA, and MOEAD-TPN, PAEA obtains better PD in 51, 44, 49, and 53 out of 56 test instances, respectively.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ZHOU *et al.*: SCALAR PROJECTION AND ANGLE-BASED EVOLUTIONARY ALGORITHM FOR MaOPs

9

TABLE III
PROPORTION OF DTLZ AND DTLZ^{-1} TEST INSTANCES WHERE PAEA IS BETTER THAN ($\bullet$), WORSE THAN ($\circ$) AND
EQUAL TO ($\ddagger$) EACH OF THE PEER ALGORITHMS ACCORDING TO THE WILCOXON'S RANK SUM TEST

PAEA v.s.		MOEA/DD	NSGA-III	MOEA/D	1by1EA	GWASF-GA	MOEA/D-AWA	MOEA/D-TPN
IGD	$\bullet$	44/56	37/56	48/56	33/56	42/56	39/56	48/56
	$\circ$	8/56	15/56	8/56	16/56	13/56	16/56	7/56
	$\ddagger$	4/56	4/56	0/56	7/56	1/56	1/56	1/56
PD	$\bullet$	55/56	51/56	55/56	44/56	56/56	49/56	53/56
	$\circ$	0/56	2/56	0/56	7/56	0/56	2/56	3/56
	$\ddagger$	1/56	3/56	1/56	5/56	0/56	5/56	0/56

TABLE IV
PROPORTION OF WFG AND WFG^{-1} TEST INSTANCES WHERE PAEA IS BETTER THAN ($\bullet$), WORSE THAN ($\circ$) AND EQUAL
TO ($\ddagger$) EACH OF THE PEER ALGORITHMS ACCORDING TO THE WILCOXON'S RANK SUM TEST

PAEA v.s.		MOEA/DD	NSGA-III	MOEA/D	1by1EA	GWASF-GA	MOEA/D-AWA	MOEA/D-TPN
IGD	$\bullet$	70/72	52/72	68/72	57/72	44/72	49/72	64/72
	$\circ$	1/72	16/72	1/72	3/72	23/72	23/72	6/72
	$\ddagger$	1/72	4/72	3/72	12/72	5/72	0/72	2/72
PD	$\bullet$	71/72	67/72	71/72	68/72	72/72	62/72	66/72
	$\circ$	1/72	4/72	1/72	3/72	0/72	8/72	3/72
	$\ddagger$	0/72	1/72	0/72	1/72	0/72	2/72	3/72

The raw experimental results on the WFG test suite are provided in Tables IV and V in the supplementary material. To have a preliminary understanding of the comparisons, main statistics are summarized as follows: among all the 72 WFG and WFG^{-1} test instances, the proportion of the best or the second best results PAEA obtains is $50/72 \approx 69\%$ and $67/72 \approx 93\%$ for IGD and PD, respectively. To make the comparisons easier, the Wilcoxon's rank sum test results are summarized in Table IV where we can find that the largest and smallest proportion of WFG and WFG^{-1} test instances where PAEA performs better in terms of IGD is $70/72 \approx 97\%$ (PAEA versus MOEA/DD) and $44/72 \approx 61\%$ (PAEA versus GWASF-GA), respectively. Similarly, the largest proportion of test instances where PAEA obtains better PD values is $72/72 = 100\%$ (PAEA versus GWASF-GA), and the smallest proportion is $62/72 \approx 86\%$ (PAEA versus MOEA/D-AWA). It is clear from the table that PAEA shows significantly better performance than other state-of-the-art algorithms in the majority of the test instances. Intuitive comparisons of the distribution of the final solutions are provided in Section III in the supplementary material. As a summary, it can be found from these materials that the solutions of PAEA are distributed more widely than those of its competitors.

V. DISCUSSION

In this section, we explain in depth why the state-of-the-art algorithms perform as they do. For MOEA/DD and MOEA/D, they perform competitively on normalized problems (e.g., DTLZ1-4), but degenerate on problems with differently scaled objective functions (e.g., ScaledDTLZ1-2, WFG1-9, and WFG1-9^{-1}). In MOEA/D, a new solution is generated for a weight vector by selecting parents from the neighbors. The new solution is compared with all of its neighbors. If the new solution is better, then the current solution is replaced. Hence, there is a risk that many neighboring solutions are replaced by a same good solution. This will do harm to the

diversity of the solutions [24], which is reflected by the parallel coordinates as shown in Figs. 1(d), 2(d), and 3(d) in the supplementary material. The MOEA/DD performs not well on the WFG test problems, and this may be due to the lack of an efficient objective normalization mechanism in MOEA/DD [24]. For GWASF-GA, it performs poorly in terms of the diversity of obtained solutions on the original DTLZ and WFG test problems. As stated in [42], the final population of GWASF-GA depends highly on the distribution of the weight vectors used. In GWASF-GA, the weight vectors are predefined and are not recalculated as the generations are increased. As a result, the algorithm cannot dynamically adapt the evolution of the population to the shapes of the PF.

The NSGA-III is significantly worse than PAEA in terms of the PD metric. This may be explained as follows. Since the majority of solutions in a high-dimensional objective space are nondominated with each other, they fall into the same layer according to the nondominated sorting procedure adopted by NSGA-III. When this primary selection criterion fails to distinguish individuals, the reference line-based diversity preservation strategy is activated. However, it is possible that some reference lines have multiple solutions, while other reference lines have no solutions [24]. In addition, the reference points generated by the two-layer approach in NSGA-III are mainly distributed on two layers of the hyper-plane [8]. Therefore, solutions found by NSGA-III concentrate mainly on the boundary or middle parts of the true PF. Hence, worse PD values are obtained by NSGA-III compared with PAEA.

The 1by1EA is highly competitive with the proposed PAEA on the DTLZ test suite. The algorithm selects individuals one-by-one-based on convergence indicators in the environmental selection. Once a solution is selected, its neighbors are de-emphasized using a niche technique to guarantee the diversity of the population. Hence, a good balance between convergence and diversity may be obtained. However, the performance of 1by1EA is related to the convergence indicators used. The 1by1EA prefers the convergence indicator

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

10   IEEE TRANSACTIONS ON CYBERNETICS

whose contour lines have a similar shape to that of the PF of a given problem. Unfortunately, the shape of the PF of a practical optimization problem is usually unknown in advance. Therefore, the enhancements of 1by1EA may be possible by adaptively choosing an appropriate convergence indicator or using an ensemble of multiple convergence indicators during the evolution of the algorithm [55].

The MOEA/D-AWA performs competitively with PAEA on both DTLZ and WFG test suites with respect to both IGD and PD metrics. The success of MOEA/D-AWA may be attributed to the adaptive weight vector adjustment, which deletes overcrowded weight vectors and adds new ones into the sparse regions. Hence, the diversity among solutions is promoted. Thanks to this effective diversity maintenance strategy, MOEA/D-AWA obtains competitive IGD and PD results compared with PAEA. However, since MOEA/D-AWA maintains an external population and needs to calculate the crowdedness of solutions, it requires more computational costs. As shown in Section V in the supplementary material, MOEA/D-AWA runs slowest among all the peer algorithms.

The MOEA/D-TPN was demonstrated to be promising when handling 2- and 3-objective MOPs with complex PFs [43], which is also verified by our experimental results on test functions F1–F9 [36] as shown in Section III in the supplementary material. However, the performance of MOEA/D-TPN degenerates on many-objective problems. One possible reason for the performance deterioration is that MOEA/D-TPN is sensitive to some key control parameters. Different settings of these parameters may be required for MaOPs. As another likely explanation, MOEA/D-TPN uses fixed weight vectors, being unable to adapt to the shapes of the PFs.

As can be found from the experimental results, the performance of MOEA/D, MOEA/DD, and NSGA-III degrades on DTLZ^{-1} and WFG^{-1} test problems. As explained in [24], this performance deterioration is attributed to the inconsistencies between the shapes of the PFs and those of the weight vectors used in these algorithms. For our proposed PAEA, however, it performs well on both DTLZ/DTLZ^{-1} and WFG/ WFG^{-1} test problems. First, PAEA uses adaptive search directions such that it can handle problems with irregular PFs, such as DTLZ5-6 and WFG3. Second, each solution in PAEA is simultaneously evaluated on two fitness functions taking into account different reference points. Therefore, PAEA can take advantages of the complementary effects of both ideal and nadir points for either concave or convex PFs. Besides, the diversity among solutions in the environmental selection is maintained by an angle-based one-by-one elimination procedure. Finally, some other techniques, such as the adaptive normalization of the population and a smart procedure for including extreme solutions are also factors for the success of our proposed PAEA. For further discussions on PAEA, we direct readers to Section VI in the supplementary material.

VI. Conclusion

This paper proposes a new decomposition-based many-objective optimizer, i.e., PAEA, by using two primary concepts: 1) scalar projection and 2) angle. In PAEA, search directions are defined based on the current solutions and two reference points. For each solution, binary search directions are considered. One is the direction from the current individual to the ideal point, while the other is from the nadir point to the current solution. The scalar projection motivates us to develop two fitness functions, which are deduced to be the differences between two PBI functions and two IPBI functions, respectively. Each individual is then evaluated on the two fitness functions simultaneously, and solutions with the best values on each function are emphasized. Finally, the angle information is adopted to select diversified solutions for the next generation. The proposed PAEA is compared with seven state-of-the-art algorithms on a large number of test problems. As shown in the experimental study, PAEA obtains promising results on these test problems with up to 15 objectives, regarding both the quality of the final solutions and the efficiency of the algorithm.

Major advantages of PAEA are that: 1) it uses adaptive search directions, therefore problems with irregular PFs could be effectively handled and 2) PAEA uses two fitness functions with different reference points to distinguish individuals, hence it takes advantages of the complementary effects of both ideal and nadir points. Consequently, PAEA is able to properly deal with problems with both concave and convex PFs. A potential shortcoming of PAEA would be that it currently considers only the distance to the ideal point as the convergence metric when eliminating solutions one by one as shown in Algorithm 4. Actually, other convergence metrics [55] could be used, which will be one of our future studies. As another meaningful research direction, besides the angle-based clustering method developed in this paper (see Section VI-B in the supplementary material), it is possible to use ideas from other related works [66]–[69] to match between parents and offspring so as to further improve the performance of the proposed algorithm. In addition, applying PAEA to imbalanced problems [70], [71] or problems from practice would be also one of our subsequent research subjects.

References

[1] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in *Proc. Annu. Meeting North Amer. Fuzzy Inf. Process. Soc.*, New Orleans, LA, USA, 2002, pp. 233–238.

[2] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surveys*, vol. 48, no. 1, pp. 1–35, Sep. 2015.

[3] R. J. Lygoe, M. Cary, and P. J. Fleming, "A real-world application of a many-objective optimisation complexity reduction process," in *Evolutionary Multi-Criterion Optimization* (LNCS 7811). Heidelberg, Germany: Springer, 2013, pp. 641–655.

[4] E. S. Matrosov, I. Huskova, and J. J. Harou, "Using many-objective optimization and robust decision making to identify robust regional water resource system plans," in *Proc. AGU Fall Meeting Abstracts*, Dec. 2015, Art. no. PA11B-2153. [Online]. Available: http://adsabs.harvard.edu/abs/2015AGUFMPA11B2153M

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

ZHOU *et al.*: SCALAR PROJECTION AND ANGLE-BASED EVOLUTIONARY ALGORITHM FOR MaOPs

11

[5] K. Narukawa and T. Rodemann, "Examining the performance of evolutionary many-objective optimization algorithms on a real-world application," in *Proc. 6th Int. Conf. Genet. Evol. Comput. (ICGEC)*, 2012, pp. 316–319.

[6] Y. Xiang, Y. Zhou, Z. Zheng, and M. Li, "Configuring software product lines by combining many-objective optimization and SAT solvers," *ACM Trans. Softw. Eng. Methodol.*, vol. 26, no. 4, pp. 1–46, Feb. 2018.

[7] M. Li, S. Yang, and X. Liu, "Bi-goal evolution for many-objective optimization problems," *Artif. Intell.*, vol. 228, pp. 45–65, Nov. 2015.

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

[9] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.

[10] 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.

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

[12] D. Gong, J. Sun, and Z. Miao, "A set-based genetic algorithm for interval many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 22, no. 1, pp. 47–60, Feb. 2018.

[13] Y. Liu, D. Gong, X. Sun, and Y. Zhang, "Many-objective evolutionary optimization based on reference points," *Appl. Soft Comput.*, vol. 50, pp. 344–355, Jan. 2017.

[14] Y. Xiang, Y. Zhou, L. Tang, and Z. Chen, "A decomposition-based many-objective artificial bee colony algorithm," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2017.2772250.

[15] Z. Chen, Y. Zhou, and Y. Xiang, "A many-objective evolutionary algorithm based on a projection-assisted intra-family election," *Appl. Soft Comput.*, vol. 61, pp. 394–411, Dec. 2017.

[16] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*, vol. 16. Chichester, U.K.: Wiley, 2001.

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

[18] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," Comput. Eng. Netw. Lab., Dept. Elect. Eng., Swiss Federal Inst. Technol. Zurich, Zürich, Switzerland, TIK-Rep. 103, 2001.

[19] 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.

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

[21] 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.

[22] Y. Zhou, Z. Chen, and J. Zhang, "Ranking vectors by means of the dominance degree matrix," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 34–51, Feb. 2017.

[23] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 645–665, Oct. 2016.

[24] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, Apr. 2017.

[25] H. Ishibuchi, Y. Hitotsuyanagi, H. Ohyanagi, and Y. Nojima, "Effects of the existence of highly correlated objectives on the behavior of MOEA/D," in *Evolutionary Multi-Criterion Optimization*. Heidelberg, Germany: Springer, 2011, pp. 166–181.

[26] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Inf. Sci.*, vol. 282, pp. 363–387, Oct. 2014.

[27] 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.

[28] 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.

[29] 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.

[30] H. Wang, Y. Jin, and X. Yao, "Diversity assessment in many-objective optimization," *IEEE Trans. Cybern.*, vol. 47, no. 6, pp. 1510–1522, Jun. 2017.

[31] M. Li, S. Yang, and X. Liu, "A performance comparison indicator for Pareto front approximations in many-objective optimization," in *Proc. Genet. Evol. Comput. Conf.*, Madrid, Spain, 2015, pp. 703–710.

[32] 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. (CEC)*, Cancún, Mexico, 2013, pp. 2488–2495.

[33] R. H. Gómez and C. A. C. Coello, "Improved metaheuristic based on the R2 indicator for many-objective optimization," in *Proc. Genet. Evol. Comput. Conf.*, Madrid, Spain, 2015, pp. 679–686.

[34] K. Bringmann, "Bringing order to special cases of Klee's measure problem," in *Mathematical Foundations of Computer Science 2013*. Heidelberg, Germany: Springer, 2013, pp. 207–218.

[35] M. Wagner and F. Neumann, "A fast approximation-guided evolutionary multi-objective algorithm," in *Proc. 15th Annu. Conf. Genet. Evol. Comput.*, Amsterdam, The Netherlands, 2013, pp. 687–694.

[36] Z. Wang, Q. Zhang, H. Li, H. Ishibuchi, and L. Jiao, "On the use of two reference points in decomposition based multiobjective evolutionary algorithms," *Swarm Evol. Comput.*, vol. 34, pp. 89–102, Jun. 2017.

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

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

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

[40] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.

[41] H. Sato, "Inverted PBI in MOEA/D and its impact on the search performance on multi and many-objective optimization," in *Proc. Annu. Conf. Genet. Evol. Comput.*, Vancouver, BC, Canada, 2014, pp. 645–652.

[42] R. Saborido, A. B. Ruiz, and M. Luque, "Global WASF-GA: An evolutionary algorithm in multiobjective optimization to approximate the whole Pareto optimal front," *Evol. Comput.*, vol. 25, no. 2, pp. 309–349, Jun. 2017.

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

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

[45] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, Feb. 2016.

[46] 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.

[47] 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.

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

[49] H. Wang, L. Jiao, and X. Yao, "Two_Arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 4, pp. 524–541, Aug. 2015.

[50] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.

[51] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evol. Comput.*, vol. 19, no. 4, pp. 561–595, Dec. 2011.

[52] F. Gu, H. L. Liu, and K. C. Tan, "A multiobjective evolutionary algorithm using dynamic weight design method," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 5B, pp. 3677–3688, 2012.

[53] Z. He and G. G. Yen, "Many-objective evolutionary algorithms based on coordinated selection strategy," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 220–233, Apr. 2017.

[54] C. A. R. Hoare, "Quicksort," *Comput. J.*, vol. 5, no. 1, pp. 10–16, 1962.

[55] Y. Liu, D. Gong, J. Sun, and Y. Jin, "A many-objective evolutionary algorithm using a one-by-one selection strategy," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2689–2702, Sep. 2017.

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

[57] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Evolutionary Multi-Criterion Optimization* (LNCS 3410). Heidelberg, Germany: Springer, 2005, pp. 280–295.

[58] Q. Zhang *et al.*, "Multi-objective optimization test instances for the CEC 2009 special session and competition," Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, School Comput. Sci. Electron. Eng., Univ. Essex, Colchester, U.K., School Elect. Electron. Eng., Nanyang Technol. Univ., Singapore, and Dept. Mech. Eng., Clemson Univ., Clemson, SC, USA, Rep. CES-487, 2009.

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

[60] D. A. Van Veldhuizen, *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, Air Force Inst. Technol., Wright-Patterson AFB, OH, USA, 1999.

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

[62] Y. Xiang, Y. Zhou, and H. Liu, "An elitism based multi-objective artificial bee colony algorithm," *Eur. J. Oper. Res.*, vol. 245, no. 1, pp. 168–193, 2015.

[63] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-based Pareto optimality for many-objective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 18, no. 2, pp. 269–285, Apr. 2014.

[64] Y. Xiang, Y. Zhou, M. Li, and Z. Chen, "A vector angle-based evolutionary algorithm for unconstrained many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 1, pp. 131–152, Feb. 2017.

[65] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

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

[67] M. Wu, K. Li, S. Kwong, Y. Zhou, and Q. Zhang, "Matching-based selection with incomplete lists for decomposition multiobjective optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 554–568, Aug. 2017, doi: 10.1109/TEVC.2017.2656922.

[68] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Trans. Evol. Comput.*, vol. 20, no. 2, pp. 180–198, Apr. 2016.

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

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

[71] H.-L. Liu, L. Chen, K. Deb, and E. D. Goodman, "Investigating the effect of imbalance between convergence and diversity in evolutionary multiobjective algorithms," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 408–425, Jun. 2017.

Yuren Zhou received the B.Sc. degree in mathematics from Peking University, Beijing, China, in 1988, the M.Sc. degree in mathematics and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 1991 and 2003, respectively. He is currently a Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His current research interests include design and analysis of algorithms, evolutionary computation, and social networks.
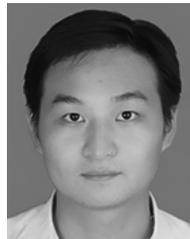
Yi Xiang received the M.Sc. degree in applied mathematics from Guangzhou University, Guangzhou, China, in 2013. He is currently pursuing the Ph.D. degree with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China.

His current research interests include evolutionary algorithms and many-objective optimization and search-based software engineering.

Zefeng Chen received the B.Sc. degree from Sun Yat-sen University, Guangzhou, China, and the M.Sc. degree from the South China University of Technology, Guangzhou. He is currently pursuing the Ph.D. degree with the School of Data and Computer Science, Sun Yat-sen University.

His current research interests include evolutionary computation, multiobjective optimization, and machine learning.

Jun He (M'06–SM'18) received the B.S. and M.Sc. degrees in mathematics and the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 1989, 1992, and 1995, respectively.

He is currently an Associate Professor of computer science with the School of Science and Technology, Nottingham Trent University, Nottingham, U.K. His current research interests include evolutionary computation, global optimization, data analysis, and network security.

Jiahai Wang (M'07) received the Ph.D. degree from Toyama University, Toyama, Japan, in 2005.

In 2005, he joined Sun Yat-sen University, Guangzhou, China, where he is currently a Professor with the School of Data and Computer Science. His current research interest includes computational intelligence and its applications.