A Reference Vector Guided Evolutionary Algorithm for Many-objective Optimization

Ran Cheng, Yaochu Jin, *Fellow, IEEE*, Markus Olhofer and Bernhard Sendhoff, *Senior Member, IEEE*

***Abstract*—In evolutionary multi-objective optimization, maintaining a good balance between convergence and diversity is particularly crucial to the performance of the evolutionary algorithms. In addition, it becomes increasingly important to incorporate user preferences because it will be less likely to achieve a representative subset of the Pareto optimal solutions using a limited population size as the number of objectives increases. This paper proposes a reference vector guided evolutionary algorithm for many-objective optimization. The reference vectors can not only be used to decompose the original multi-objective optimization problem into a number of single-objective sub-problems, but also to elucidate user preferences to target a preferred subset of the whole Pareto front. In the proposed algorithm, a scalarization approach, termed angle penalized distance, is adopted to balance convergence and diversity of the solutions in the high-dimensional objective space. An adaptation strategy is proposed to dynamically adjust the distribution of the reference vectors according to the scales of the objective functions. Our experimental results on a variety of benchmark test problems show that the proposed algorithm is highly competitive in comparison with five state-of-the-art evolutionary algorithms for many-objective optimization. In addition, we show that reference vectors are effective and cost-efficient for preference articulation, which is particularly desirable for many-objective optimization. Furthermore, a reference vector regeneration strategy is proposed for handling irregular Pareto fronts. Finally, the proposed algorithm is extended for solving constrained many-objective optimization problems.**

***Index Terms*—Evolutionary multiobjective optimization, reference vector, many-objective optimization, convergence, diversity, preference articulation, angle penalized distance**

I. Introduction

MULTIOBJECTIVE optimization problems (MOPs), which involve more than one conflicting objective to be optimized simultaneously, can be briefly formulated as follows:

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_M(\mathbf{x})) \\ \text{s.t.} \quad & \mathbf{x} \in X \end{aligned} \tag{1}$$

where $X \subseteq \mathbb{R}^n$ is the decision space with $\mathbf{x} = (x_1, x_2, ..., x_n) \in X$ being the decision vector. Due to the conflicting nature of the objectives, usually, there does not exist one single solution that is able to optimize all the objectives simultaneously. Instead, a set of optimal solutions representing the trade-offs between different objectives, termed Pareto optimal solutions, can be achieved. The Pareto optimal solutions are known as the Pareto front (PF) in the objective space and the Pareto set (PS) in the decision space, respectively.

Evolutionary algorithms (EAs), as a class of population based search heuristics, are able to obtain a set of solutions in a single run. Thanks to this attractive property, multiobjective evolutionary algorithms (MOEAs) have witnessed a boom of development over the past two decades [1]. MOEAs have been shown to perform well on a wide range of MOPs with two or three objectives; however, MOEAs have experienced substantial difficulties when they are adopted to tackle MOPs with more than three objectives [2]–[5], often referred to as the many-objective problems (MaOPs) nowadays. As a result, MaOPs have attracted increasing attention in evolutionary optimization [6].

One major reason behind the failure of most conventional MOEAs in solving MaOPs can be attributed to the loss of selection pressure, i.e., the pressure for the population to converge towards the PF, when dominance is adopted as a criterion for selecting individuals with a limited population size [7]. For example, the elitist non-dominated sorting genetic algorithm (NSGA-II) [8] and the strength Pareto evolutionary algorithm 2 (SPEA2) [9], both of which use dominance-based selection, will fail to work properly for MaOPs, since most candidate solutions generated in a population of a limited size are non-dominated, making the dominance based selection criterion hardly possible to distinguish the candidate solutions, even in a very early stage of the search.

Another important reason for the degraded performance of MOEAs on MaOPs is the difficulty in maintaining a good population diversity in a high-dimensional objective space. Generally speaking, since the PF of most continuous MOPs is piecewise continuous [10], [11], it is practically unlikely to approximate all Pareto optimal solutions. Instead, most MOEAs aim to find a set of evenly distributed representative solutions to approximate the PF. When the number of objectives is two or three, where the PF is typically a one-dimensional curve or two-dimensional surface, maintaining a good diversity of the solutions is relatively straightforward. As the dimension of the objective space increases, it becomes increasingly challenging to maintain a good population diversity, as the candidate solutions distribute very sparsely in the high-dimensional objective space, causing immense difficulties to the diversity management strategies widely used in MOEAs, e.g., the crowding distance base diversity method in NSGA-

Ran Cheng and Yaochu Jin are with the Department of Computer Science, University of Surrey, Guildford, Surrey, GU2 7XH, United Kingdom (e-mail: {r.cheng;yaochu.jin}@surrey.ac.uk). Y. Jin is also with the College of Information Sciences and Technology, Donghua University, Shanghai 201620, P. R. China.

Markus Olhofer and Bernhard Sendhoff are with the Honda Research Institute Europe, 63073 Offenbach, Germany (e-mail: {markus.olhofer;bernhard.sendhoff}@honda-ri.de).

II [12]–[14].

To enhance the performance of most traditional MOEAs in solving MaOPs, a number of approaches have been proposed [15], [16], which can be roughly divided into three categories.

The first category covers various approaches to convergence enhancement. Since the loss of convergence pressure in most traditional MOEAs is directly caused by the inability of the canonical dominance to distinguish solutions, the most intuitive idea for convergence enhancement is to modify the dominance relationship to increase the selection pressure towards the PF. Examples of modified dominance definitions include ϵ-dominance [17], [18], L-optimality [19], preference order ranking [20] and fuzzy dominance [21]. In [22], a grid dominance based metric is defined for solving MaOPs, termed grid-based evolutionary algorithm (GrEA), which eventually modifies the dominance criterion to accelerate the convergence in many-objective optimization. Another typical idea in this category is to combine the Pareto dominance based criterion with additional convergence-related metrics. For example, Köppen and Yoshida [23] propose to use some substitute distances to describe the degree to which a solution almost dominates another to improve the performance of NSGA-II. In [24], a binary ϵ-indicator based preference is combined with dominance to speed up convergence of NSGA-II for solving MaOPs. In [25], a shift-based density estimation strategy is proposed to penalize poorly converged solutions by assigning them a high density value in addition to dominance comparison. In [26], a mating selection based on favorable convergence is applied to strengthen convergence pressure while an environmental selection based on directional diversity and favorable convergence is designed to balance diversity and convergence. In the recently proposed knee point driven evolutionary algorithm (KnEA) [27], a knee point based secondary selection is designed on tope of non-dominated sorting to enhance convergence pressure.

The second category is often known as the decomposition based approaches, which divide a complex MOP into a number of sub-problems and solve them in a collaborative manner [28], [29]. There are mainly two types of decomposition based approaches [30]. In the first type of decomposition based approaches, an MOP is decomposed into a group of single-objective problems (SOPs), including the weighted aggregation based approaches in early days [31], [32], and the more recent multiobjective evolutionary algorithm based on decomposition (MOEA/D), where more explicit collaboration strategies between the solutions of the sub-problems were introduced. Several variants of MOEA/D have been proposed for enhancing the selection strategy for each sub-problem to strike a better balance between convergence and diversity [30], [33]–[37].

In the second type of decomposition based approaches, an MOP is decomposed into a group of sub-MOPs. For instance, MOEA/D-M2M [29], [38] divides the whole PF into a group of segments, and each segment can be regarded as a sub-problem. Another MOEA that essentially falls under this category is NSGA-III [39], where a set of predefined, evenly distributed reference points to manage the diversity of the candidate solutions, eventually contribute to enhanced convergence of the algorithm. Although the second type of decomposition strategy has been reported very efficient in some recent works [40], [41], compared to the first type, its development is still in the infancy.

The third category is known as the performance indicator based approaches, e.g., the indicator based evolutionary algorithm (IBEA) [42], the S-metric selection based evolutionary multi-objective algorithm (SMS-EMOA) [43], a dynamic neighborhood multi-objective evolutionary algorithm based on hypervolume indicator (DNMOEA/HI) [44], and the fast hypervolume based evolutionary algorithm (HypE) [45]. These approaches are not subject to the issues that dominance based MOEAs have for solving MaOPs. Unfortunately, the computational cost for the calculation of the performance becomes prohibitively expensive when the number of objectives is large [46].

There are also a few approaches that do not fall into any of the above three main categories. For example, some researchers propose to use interactive user preferences [47] or reference points [48] during the search while others suggest to solve MaOPs by using a reduced set of objectives [49]–[51]. Another example is a recently proposed evolutionary many-objective optimization algorithm based on both dominance and decomposition (MOEA/DD) [41], the motivation of which is to exploit the merits offered by both dominance and decomposition based approaches. More recently, a two-archive algorithm for many-objective optimization (Two_Arch2) has been proposed based on indicator and dominance [52].

While most existing MOEAs focusing on convergence enhancement and diversity maintenance, it is noted that use of preferences will become particularly important for many-objective optimization, not only because the user may be interested in only part of Pareto optimal solutions, but also because it is less practical to achieve a representative subset of the whole PF using an evolutionary algorithm of a limited population size.

As already shown in [39], reference points can also be used to generate a subset of preferred Pareto optimal solutions, although NSGA-III can be seen as a decomposition based approach if the reference points are evenly distributed in the whole objective space. Motivated by ideas in decomposition based approaches and the aim to achieve the preferred part of the PF when the number of objectives is large, we propose a reference vector guided evolutionary algorithm (RVEA) for solving MaOPs. Compared with existing decomposition based approaches, the main new contributions of this work can be summarized as follows:

(1) A scalarization approach, termed as the angle penalized distance (APD), is designed to dynamically balance convergence and diversity in many-objective optimization according to the number of objectives and the number of generations. In the proposed APD, the convergence criterion is measured by the distance between the candidate solutions and the ideal point[1], and the diversity criterion

[1]For a minimization problem, the ideal point is a vector that consists of the minimum value of each objective function.

is measured by the acute angle between the candidate solutions and the reference vectors. Compared to the penalty-based boundary intersection (PBI) approach [28] that relies on the Euclidean distance, the angle-based distance metric makes it easier for normalization and is more scalable to the number of objectives, which is essential for many-objective optimization. Note that if the reference vectors are used to represent user preferences, this angle also indicates the degree of satisfaction of the user preferences.

(2) An adaptive strategy is proposed to adjust the reference vectors to deal with objective functions that are not well normalized. The adaptive strategy adjusts the distribution of the reference vectors according to the ranges of different objective functions in order to ensure a uniform distribution of the candidate solutions in the objective space, even if the objective functions are not well normalized or the geometrical structure of a PF is highly asymmetric. This strategy is mainly for achieving an evenly distributed Pareto optimal subset.

(3) It is shown that reference vectors can also provide an effective and computationally efficient approach to preference articulation. Such preference articulation is particularly valuable in many-objective optimization, where it is very unlikely to obtain a representative approximation of the whole PF [53]. By specifying a central vector and a radius, we propose a reference vector based preference articulation approach that is able to generate evenly distributed Pareto optimal solutions in a preferred region in the objective space.

(4) To enhance the performance of the proposed RVEA on problems with irregular[2] Pareto fronts, a reference vector regeneration strategy is proposed. The basic idea is to use an additional reference vector set to perform exploration in the objective space so that the density of the solutions obtained by RVEA on problems with irregular Pareto fronts can be improved.

(5) The proposed RVEA is further extended for solving constrained MaOPs.

The rest of this paper is organized as follows. Section II introduces some background knowledge. The details of the proposed RVEA are described in Section III. Section IV presents empirical results that compare the performance of RVEA with five state-of-the-art MOEAs for solving MaOPs. In addition, preference articulation using reference vectors is exemplified in Section V, a reference vector regeneration strategy for irregular Pareto fronts handling is presented in Section VI, and the extension of RVEA to handling constrained MaOPs is presented in Section VII. Finally, conclusions and future work are given in Section VIII.

II. Background

In this section, we first present a brief review of decomposition based MOEAs. Then, an introduction to the reference vectors used in this work is given, including how to sample uniformly distributed reference vectors and how to measure the spacial relationship between two vectors.

A. Decomposition based MOEAs

In weight aggregation based decomposition approaches, a set of *weight vectors* are used to convert an MOP into a number of SOPs using a scalarization method [28], [32], [54]. Among others, the weighted sum approach, the weighted Tchebycheff approach [55] and the PBI approach [28] are most widely used.

More recently, a set of weight vectors are used to divide an MOP into a number of sub-problems by partitioning the entire objective space into some subspaces, where each sub-problem remains an MOP. This type of decomposition strategy was first proposed by Liu *et al.* in [29], where a set of *direction vectors* are used to divide the whole PF into a number of segments, each segment being a multi-objective sub-problem. Such a decomposition strategy has attracted certain interests. For example, in NSGA-III [39], a set of *reference points* or *reference lines* are used for niche preservation to manage diversity in each subspace for many-objective optimization, which effectively enhances convergence by giving priority to solutions closer to the reference points. Most recently, an inverse model based MOEA (IM-MOEA) [40] has been suggested, where a set of *reference vectors* are used to partition the objective space into a number of subspaces and then inverse models that map objective vectors onto decision vectors are built inside each subspace for sampling new candidate solutions.

Since *weight vectors* are typically used to denote the importance of objectives in weighted aggregation, different terminologies have been coined in the second type of decomposition based approaches to refer to vectors that decompose the original objective space, including *direction vectors* [29], *reference lines* [39], and *reference vectors* [40]. In essence, these vectors play a similar role of partitioning the objective space into a number of subspaces. In this work, we adopt the term *reference vectors*.

When a set of evenly distributed reference vectors are generated for achieving representative solutions of the whole PF, the proposed RVEA can be considered as one of the second type of decomposition based approaches. However, if user preferences are available and a set of specific reference vectors are generated for achieving only a preferred section of the PF, RVEA can also be seen as a preference based approach. For example in [56], [57], a set of reference vectors have been used to achieve preferred subset of the Pareto-optimal solutions. In this sense, RVEA differs from most existing reference point based MOEAs [58]–[60] in that these algorithms use dominance and preference to search for preferred subset of the PF only. It is worth noting that there are other preference articulation methods tailored for the decomposition based MOEAs. For example in [61], Gong *et al.* have proposed an interactive MOEA/D for multi-objective decision making. The idea is to dynamically adjust the distribution of the weight vectors according to the preferred region specified by a hypersphere. In [62], Ma *et al.* have proposed to apply the light beam

[2]In this work, irregular Pareto fronts refer to disconnected and degenerate Pareto fronts.

search (LBS) [63] in MOEA/D to incorporate user preferences, where the preference information is specified by an aspiration point and a reservation point, together with a preference neighborhood parameter. Most recently, Mohammadi *et al.* have also proposed to integrate user preferences for many-objective optimization [64], where the preferred region is specified by a hypercube. These methods try to define some preferred regions and generate weight vectors inside them to guide the search of the MOEAs. The main difference between RVEA proposed in this work and the above methods lies in the fact that RVEA defines preferred regions using a central vector and a radius, and a new angle-based selection criterion is proposed.

B. Reference Vector

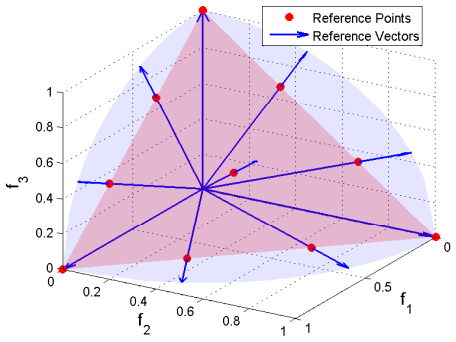


Fig. 1. An illustration of how to generate the uniformly distributed reference vectors in a three-objective space. In this case, 10 uniformly distributed reference points are first generated on a hyperplane, and then they are mapped to a hypersphere to generate the 10 reference vectors.

Without loss of generality, all the reference vectors used in this work are unit vectors inside the first quadrant with the origin being the initial point. Theoretically, such a unit vector can be easily generated via dividing an arbitrary vector by its norm. However, in practice, uniformly distributed unit reference vectors are required for a uniformly distributed coverage of the objective space. In order to generate uniformly distributed reference vectors, we adopt the approach introduced in [40]. Firstly, a set of uniformly distributed points on a unit hyperplane are generated using the canonical simplex-lattice design method [65]:

$$\begin{cases} \mathbf{u_i} = (u_i^1, u_i^2, ..., u_i^M), \\ u_i^j \in \{\frac{0}{H}, \frac{1}{H}, ..., \frac{H}{H}\}, \sum_{j=1}^{M} u_i^j = 1, \end{cases} \tag{2}$$

where $i = 1, ..., N$ with N being the number of uniformly distributed points, M is the objective number, and H is a positive integer for the simplex-lattice design. Then, the corresponding unit reference vectors $\mathbf{v_i}$ can be obtained by the following transformation:

$$\mathbf{v_i} = \frac{\mathbf{u_i}}{\|\mathbf{u_i}\|}, \tag{3}$$

which maps the reference points from a hyperplane to a hypersphere, an example of which is shown in Fig. 1. According to the property of the simplex-lattice design, given H and M, a total number of $N = \binom{H+M-1}{M-1}$ uniformly distributed reference vectors can be generated.

Given two vectors $\mathbf{v_1}$ and $\mathbf{v_2}$, the cosine value of the acute angel θ between the two vectors can be used to measure the spatial relationship between them, which is calculated as follows:

$$\cos\theta = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\|\|\mathbf{v_2}\|}, \tag{4}$$

where $\|\cdot\|$ calculates the norm, i.e., the length of the vector. As will be introduced in the following section, in the proposed RVEA, (4) can be used to measure the spacial relationship between an objective vector and a reference vector in the objective space. Since they have already been normalized as in (2) when generated, the reference vectors no longer need to be normalized again in calculating the cosine values.

III. The Proposed RVEA

A. Elitism Strategy

Algorithm 1 The main framework of the proposed RVEA.

1: **Input:** the maximal number of generations t_{max}, a set of unit reference vectors $V_0 = \{\mathbf{v}_{0,1}, \mathbf{v}_{0,2}..., \mathbf{v}_{0,N}\}$;
2: **Output:** final population $P_{t_{max}}$;
3: /*Initialization*/
4: **Initialization**: create the initial population P_0 with N randomized individuals;
5: /*Main Loop*/
6: **while** $t < t_{max}$ **do**
7: Q_t = offspring-creation(P_t);
8: $P_t = P_t \cup Q_t$;
9: P_{t+1} = reference-vector-guided-selection(t, P_t, V_t);
10: V_{t+1} = reference-vector-adaptation(t, P_{t+1}, V_t, V_0);
11: $t = t + 1$;
12: **end while**

The main framework of the proposed RVEA is listed in Algorithm 1, from which we can see that RVEA adopts an elitism strategy similar to that of NSGA-II [8],where the offspring population is generated using traditional genetic operations such as crossover and mutation, and then the offspring population is combined with the parent population to undergo an elitism selection. The main new contributions in the RVEA lie in the two other components, i.e., the reference vector guided selection and the reference vector adaptation. In addition, RVEA requires a set of predefined reference vectors as the input, which can either be uniformly generated using (2) and (3), or specified according to the user preferences, which will be introduced in Section V. In the following subsections, we will introduce the three main components in Algorithm 1, i.e., offspring creation, reference vector guided selection and reference vector adaptation.

B. Offspring Creation

In the proposed RVEA, the widely used genetic operators, i.e., the simulated binary crossover (SBX) [66] and the polynomial mutation [67] are employed to create the offspring

population, as in many other MOEAs [22], [27], [28], [39], [45]. Here, we do not apply any explicit mating selection strategy to create the parents; instead, given N individuals in the current population P_t, a number of $\lfloor N/2 \rfloor$ pair of parents are randomly generated, i.e., each of the N individuals has an equal probability to participate in the reproduction procedure. This is made possible partly thanks to the reference vector guided selection strategy, which is able to effectively manage the convergence and diversity inside small subspaces of the objective space such that the individual inside each subspace can make an equal contribution to the population. Nevertheless, specific mating selection strategies can be helpful in solving problems having a multimodal landscape or a complex Pareto set [68].

C. Reference Vector Guided Selection

Similar to MOEA/D-M2M [29], [38], RVEA partitions the objective space into a number of subspaces using the reference vectors, and selection is performed separately inside each subspace. The objective space partition is equivalent to adding a constraint to the subproblem specified each reference vector, which is shown to be able to help balance the convergence and diversity in decomposition based approaches [69]. To be specific, the proposed reference vector guided selection strategy consists of four steps: objective value translation, population partition, angle penalized distance calculation and the elitism selection.

1) Objective Value Translation: According to the definition in Section II-B, the initial point of the reference vectors used in this work is always the coordinate origin. To be consistent with this definition, the objective values of the individuals in population P_t, denoted as $F_t = \{\mathbf{f}_{t,1}, \mathbf{f}_{t,2}, ..., \mathbf{f}_{t,|P_t|}\}$, where t is the generation index, are translated[3] into F_t' via the following transformation:

$$\mathbf{f'}_{t,i} = \mathbf{f}_{t,i} - \mathbf{z}_t^{min}, \tag{5}$$

where $i = 1, ..., |P_t|$, $\mathbf{f}_{t,i}$, $\mathbf{f'}_{t,i}$ are the objective vectors of individual i before and after the translation, respectively, and $\mathbf{z}_t^{min} = (z_{t,1}^{min}, z_{t,2}^{min}, ..., z_{t,m}^{min})$ represents the minimal objective values calculated from F_t. The role of translation operation is twofold: (1) to guarantee that all the translated objective values are inside the first quadrant, where the extreme point of each objective function is on the corresponding coordinate axis, thus maximizing the coverage of the reference vectors; (2) to set the ideal point to be the origin of the coordinate system, which will simplify the formulations to be presented later on. Some empirical results showing the significance of the objective value translation can be found in Supplementary Materials I.

2) Population Partition: After the translation of the objective values, population P_t is partitioned into N subpopulations $\overline{P}_{t,1}, \overline{P}_{t,2}, ..., \overline{P}_{t,N}$ by associating each individual with its closest reference vector, referring to Fig. 2, where N is the number of reference vectors. As introduced in Section II-B, the spacial relationship of two vectors is measured by the acute angle between them, i.e., the cosine value between an objective vector and a reference vector can be calculated as:

$$\cos\theta_{t,i,j} = \frac{\mathbf{f'}_{t,i} \cdot \mathbf{v}_{t,j}}{\|\mathbf{f'}_{t,i}\|}, \tag{6}$$

where $\theta_{t,i,j}$ represents the angle between objective vector $\mathbf{f'}_{t,i}$ and reference vector $\mathbf{v}_{t,j}$.

In this way, an individual $I_{t,i}$ is allocated to a subpopulation $\overline{P}_{t,k}$ if and only if the angle between $\mathbf{f'}_{t,i}$ and $\mathbf{v}_{t,k}$ is minimal (i.e., the cosine value is maximal) among all the reference vectors:

$$\overline{P}_{t,k} = \{I_{t,i} | k = \underset{j \in \{1,...,N\}}{\operatorname{argmax}} \cos\theta_{t,i,j}\}, \tag{7}$$

where $I_{t,i}$ denotes the I-th individual in P_t, with $i = 1, ..., |P_t|$.

3) Angle-Penalized Distance (APD) Calculation: Once the population P_t is partitioned into N subpopulations $\overline{P}_{t,1}, \overline{P}_{t,2}, ..., \overline{P}_{t,N}$, one elitist can be selected from each subpopulation to create P_{t+1} for the next generation.

Since our motivation is to find the solution on each reference vector that is closest to the ideal point, the selection criterion consists of two sub-criteria, i.e., the convergence criterion and the diversity criterion, with respect to the reference vector that the candidate solutions are associated with.

Specifically, given a translated objective vector $\mathbf{f'}_{t,i}$ in subpopulation j, the convergence criterion can be naturally represented by the distance from $\mathbf{f'}_{t,i}$ to the ideal point[4], i.e., $\|\mathbf{f'}_{t,i}\|$; and the diversity criterion is represented by the acute angle between $\mathbf{f'}_{t,i}$ and $\mathbf{v}_{t,j}$, i.e., $\theta_{t,i,j}$, as the inverse function value of $\cos\theta_{t,i,j}$ calculated in (6). In order to balance between the convergence criterion $\|\mathbf{f'}_{t,i}\|$ and the diversity criterion $\theta_{t,i,j}$, a scalarization approach, i.e., the *angle-penalized distance (APD)* is proposed as follows:

$$d_{t,i,j} = (1 + P(\theta_{t,i,j})) \cdot \|\mathbf{f'}_{t,i}\|, \tag{8}$$

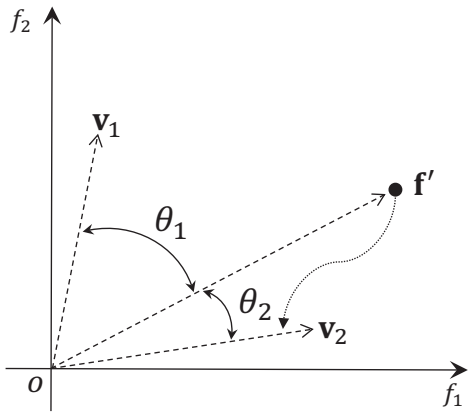


Fig. 2. An example showing how to associate an individual with a reference vector. In this example, $\mathbf{f'}$ is a translated objective vector, $\mathbf{v}_1$ and $\mathbf{v}_2$ are two unit reference vectors. θ_1 and θ_2 are the angles between $\mathbf{f'}$ and $\mathbf{v}_1$, $\mathbf{v}_2$, respectively. Since $\theta_2 < \theta_1$, the individual denoted by $\mathbf{f'}$ is associated with reference vector $\mathbf{v}_2$.

[3]In Euclidean geometry, a translation is a rigid motion that moves every point a constant distance in a specified direction.

[4]As the objective values have been translated by subtracting the minimal value of each objective function in (5), the ideal point is always the coordinate origin. Therefore, the distance from a translated objective vector to the ideal point equals the norm (length) of the translated objective vector.

Algorithm 2 The reference vector guided selection strategy in the proposed RVEA.

Input: generation index t, population P_t, unit reference vector set $V_t = \{\mathbf{v}_{t,1}, \mathbf{v}_{t,2}, ..., \mathbf{v}_{t,N}\}$;
Output: population P_{t+1} for next generation;
/*Objective Value Translation*/
Calculate the minimal objective values $\mathbf{z}_t^{min}$;
for $i = 1$ to $|P_t|$ **do**
  $\mathbf{f'}_{t,i} = \mathbf{f}_{t,i} - \mathbf{z}_t^{min}$; //refer to (5)
end for
/*Population Partition*/
for $i = 1$ to $|P_t|$ **do**
  for $j = 1$ to N **do**
    $\cos\theta_{t,i,j} = \frac{\mathbf{f'}_{t,i} \cdot \mathbf{v}_{t,j}}{\|\mathbf{f'}_{t,i}\|}$; //refer to (6)
  end for
end for
for $i = 1$ to $|P_t|$ **do**
  $k = \underset{j \in \{1,...,N\}}{\operatorname{argmax}} \cos\theta_{t,i,j}$;
  $\overline{P}_{t,k} = \overline{P}_{t,k} \cup \{I_{t,i}\}$; //refer to (7)
end for
/*Angle-Penalized Distance (APD) Calculation*/
for $j = 1$ to N **do**
  for $i = 1$ to $|\overline{P}_{t,j}|$ **do**
    $d_{t,i,j} = (1 + P(\theta_{t,i,j})) \cdot \|\mathbf{f'}_{t,i}\|$; //refer to (8) (9) (10)
  end for
end for
/*Elitism Selection*/
for $j = 1$ to N **do**
  $k = \underset{i \in \{1,...,|\overline{P}_{t,j}|\}}{\operatorname{argmin}} d_{t,i,j}$;
  $P_{t+1} = P_{t+1} \cup \{I_{t,k}\}$;
end for

with $P(\theta_{t,i,j})$ being a penalty function related to $\theta_{t,i,j}$:

$$P(\theta_{t,i,j}) = M \cdot (\frac{t}{t_{max}})^{\alpha} \cdot \frac{\theta_{t,i,j}}{\gamma_{\mathbf{v}_{t,j}}}, \tag{9}$$

and

$$\gamma_{\mathbf{v}_{t,j}} = \min_{i \in \{1,...,N\}, i \neq j} \langle \mathbf{v}_{t,i}, \mathbf{v}_{t,j} \rangle, \tag{10}$$

where M is the number of objectives, N is the number of reference vectors, t_{max} is the predefined maximal number of generations, $\gamma_{\mathbf{v}_{t,j}}$ is the smallest angle value between reference vector $\mathbf{v}_{t,j}$ and the other reference vectors in the current generation, and α is a user defined parameter controlling the rate of change of $P(\theta_{t,i,j})$. The detailed design of the penalty function $P(\theta_{t,i,j})$ in the APD calculation is based on the following empirical observations.

Firstly, in many-objective optimization, since the candidate solutions are sparsely distributed in the high-dimensional objective space, it is not the best to apply a constant pressure on convergence and diversity in the entire search process. Ideally, at the early stage of the search process, a high selection pressure on convergence is exerted to push the population towards the PF, while at the late search stage, once the population is close to the PF, population diversity can be emphasized in selection to generate well distributed candidate solutions. The penalty function , $P(\theta_{t,i,j})$, is exactly designed to meet these requirements. Specifically, at the early stage of the search process (i.e, $t \ll t_{max}$), $P(\theta_{t,i,j}) \approx 0$, and thus $d_{t,i,j} \approx \|\mathbf{f'}_{t,i}\|$ can be satisfied, which means that the value of $d_{t,i,j}$ is mainly determined by the convergence criterion $\|\mathbf{f'}_{t,i}\|$; while at the late stage of the search process, with the value of t approaching t_{max}, the influence of $P(\theta_{t,i,j})$ will be gradually accumulated to emphasize the importance of the diversity criterion $\theta_{t,i,j}$.

Angle $\gamma_{\mathbf{v}_{t,j}}$ is used to normalize the angles in the subspace specified by $\mathbf{v}_{t,j}$. This angle normalization process is particularly meaningful when the distribution of some reference vectors is either too dense (or too sparse), resulting in extremely small (or large) angles between the candidate solutions and the reference vectors. Compared to most existing objective normalization approaches, e.g., the one adopted in NSGA-III [39], the proposed angle normalization approach has two major differences: (1) normalizing the angles (instead of the objectives) will not change the actual positions of the candidate solutions, which is important convergence information for the proposed RVEA; (2) angle normalization, which is independently carried out inside each subspace, does not influence the distribution of the candidate solutions in other subspaces.

In addition, since the sparsity of the distribution of the candidate solutions is directly related to the dimension of the objective space, i.e., the value of M, the penalty function $P(\theta_{t,i,j})$ is also related to M to adaptively adjust the range of the penalty function values.

According to the angle penalized distances calculated using (8) and (9), the individual in each subpopulation having the minimal distance is selected as the elitist to be passed to the population for the next generation. The pseudo code of the reference vector guided selection procedure is summarized in Algorithm 2.

It is worth noting that the formulation of the proposed APD shares some similarity to PBI [28], which is widely adopted in the decomposition based MOEAs. However, there are two major differences between APD and PBI:

(1) In APD of the proposed RVEA, the angle between the reference vector and the solution vector is calculated for measuring diversity or the degree of satisfaction of the user preference, while in PBI, the Euclidean distance of the solution to the weight vector is calculated, which is a sort of diversity measure. Calculation of the difference in angle has certain advantages over calculation of the distance for the following two reasons. First, no matter what the exact distance a candidate solution is from the ideal point, the angle between the candidate solution and a reference vector is constant. Second, angles can be more easily normalized into the same range, e.g., [0, 1].
(2) The penalty item $P(\theta_{t,i,j})$ in APD is tailored for many-objective optimization, which is adaptive to the search process as well as the number of objectives, while the penalty item θ in PBI is a fixed parameter, which was originally designed for multiobjective optimization. As pointed out in [70], there is no unique setting for the parameter θ in PBI that works well on different types of problems with different numbers of objectives. By

contrast, our empirical results in Section IV demonstrate that APD works robustly well on a variety of problems with different numbers of objectives without changing the setting for parameter α. This is mainly due to the fact that the penalty function $P(\theta_{t,i,j})$ in APD is able to be normalized to a certain range, given any angles between the candidate solutions and the reference vector. Such a normalized penalty function provides a stable balancing between convergence and diversity, no matter whether the distribution of the reference vectors is sparse or dense.

Empirical results on comparing the proposed APD approach and the PBI approach can be found in Section IV-F.

D. Reference Vector Adaptation

Given a set of uniformly distributed unit reference vectors, the proposed RVEA is expected to obtain a set of uniformly distributed Pareto optimal solutions that are the intersection points between each reference vector and the PF, as shown in Fig. 3(a). However, this happens only if the function values of all objectives can be easily normalized into the same range, e.g., $[0, 1]$. Unfortunately, in practice, there may exist MaOPs where different objectives are scaled to different ranges, e.g., the WFG test problems [71] and the scaled DTLZ problems [39]. In this case, uniformly distributed reference vectors will not produce uniformly distributed solutions, as shown in Fig. 3(b).

One intuitive way to address the above issue is to carry out objective normalization dynamically as the search proceeds. Unfortunately, it turns out that objective normalization is not suited for the proposed RVEA, mainly due to the following reasons:

(1) Objective normalization, as a transformation that maps the objective values from a scaled objective space onto the normalized objective space, changes the actual objective values, which will further influence the selection criterion, i.e., the angle penalized distance.
(2) Objective normalization has to be repeatedly performed as the scales of the objective values change in each generation.

As a consequence, performing objective normalization will cause instability in the convergence of the proposed RVEA due to the frequently changed selection criterion.

Nevertheless, it should be noted that although objective normalization is not suited for the proposed RVEA, it can work well for dominance based approaches such as NSGA-III, because the transformation does not change the partial orders, i.e., the dominance relations, between the candidate solutions, which is vital in dominance based approaches.

To illustrate the discussions above, we show below an empirical example. Given two translated objective vectors, $\mathbf{f'}_1 = (0.1, 2)$ and $\mathbf{f'}_2 = (1, 10)$, where $\mathbf{f'}_1$ dominates $\mathbf{f'}_2$; after objective normalization, the two vectors become $\overline{\mathbf{f}}'_1 = (0.1, 0.2)$ and $\overline{\mathbf{f}}'_2 = (1, 1)$. It can be seen that the dominance relation is not changed, where $\overline{\mathbf{f}}'_1$ still dominates $\overline{\mathbf{f}}'_2$. However, the difference between the two vectors has been substantially changed, from $\|\mathbf{f'}_2 - \mathbf{f'}_1\| = 8.0$ to $\|\overline{\mathbf{f}}'_2 - \overline{\mathbf{f}}'_1\| = 1.2$. It is very likely such a substantial change will influence the results generated by the angle penalized distance in (8), thus causing instability in the selection process.

Therefore, instead of normalizing the objectives, we propose to adapt the reference vectors according to the ranges of the objective values in the following manner:

$$\mathbf{v}_{t+1,i} = \frac{\mathbf{v}_{0,i} \circ (\mathbf{z}^{max}_{t+1} - \mathbf{z}^{min}_{t+1})}{\|\mathbf{v}_{0,i} \circ (\mathbf{z}^{max}_{t+1} - \mathbf{z}^{min}_{t+1})\|}, \tag{11}$$

where $i = 1, ..., N$, $\mathbf{v}_{t+1,i}$ denotes the i-th adapted reference vector for the next generation $t + 1$, $\mathbf{v}_{0,i}$ denotes the i-th uniformly distributed reference vector, which is generated in the initialization stage (on Line 1 in Algorithm 1), $\mathbf{z}^{max}_{t+1}$ and $\mathbf{z}^{min}_{t+1}$ denote the maximum and minimum values of each objective function in the $t + 1$ generation, respectively. The $\circ$ operator denotes the Hadamard product that element-wisely multiplies two vectors (or matrices) of the same size.

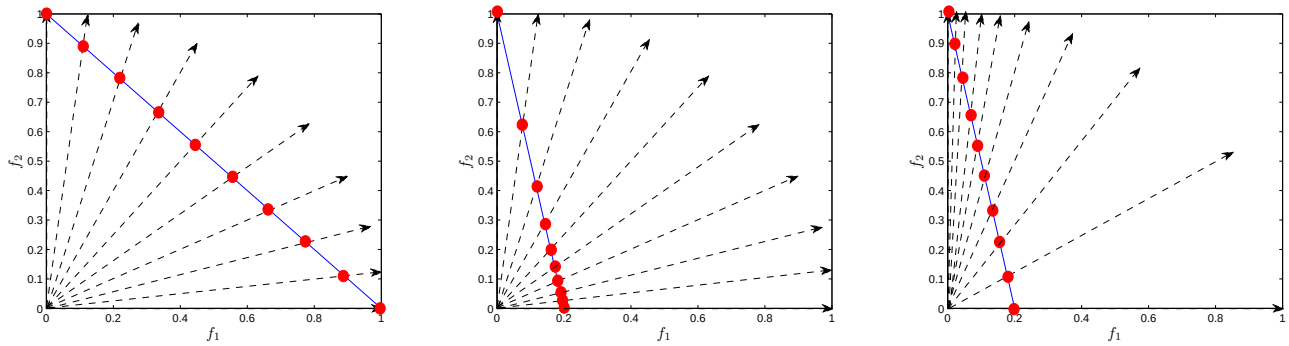
With the reference vector adaptation strategy described above, the proposed RVEA will be able to obtain uniformly distributed solutions, even if the objective functions are not normalized to the same range, as illustrated in Fig. 3(c). Furthermore, some empirical results on the influence of the reference vector adaption strategy can be found in Appendix II.

Algorithm 3 The reference vector adaptation strategy in the proposed RVEA.

1: **Input:** generation index t, population P_{t+1}, current unit reference vector set $V_t = \{\mathbf{v}_{t,1}, \mathbf{v}_{t,2}, ..., \mathbf{v}_{t,N}\}$, initial unit reference vector set $V_0 = \{\mathbf{v}_{0,1}, \mathbf{v}_{0,2}, ..., \mathbf{v}_{0,N}\}$;
2: **Output:** reference vector set V_{t+1} for next generation;
3: **if** $(\frac{t}{t_{max}} \mod f_r) == 0$ **then**
4: Calculate the minimal and maximal objective values $\mathbf{z}^{min}_{t+1}$ and $\mathbf{z}^{max}_{t+1}$, respectively;
5: **for** i = 1 to N **do**
6: $\mathbf{v}_{t+1,i} = \frac{\mathbf{v}_{0,i} \circ (\mathbf{z}^{max}_{t+1} - \mathbf{z}^{min}_{t+1})}{\|\mathbf{v}_{0,i} \circ (\mathbf{z}^{max}_{t+1} - \mathbf{z}^{min}_{t+1})\|}$; //refer to (11);
7: **end for**
8: **else**
9: $\mathbf{v}_{t+1,i} = \mathbf{v}_{t,i}$;
10: **end if**

However, as pointed out by Giagkiozis *et al.* in [72], the reference vector adaptation strategy should not be employed very frequently during the search process to ensure a stable convergence. Fortunately, unlike objective normalization, the reference vector adaptation does not have to be performed in each generation. Accordingly, a parameter f_r (Line 3 in Algorithm 3) is introduced to control the frequency of employing the adaptation strategy. For instance, if f_r is set to 0.2, the reference vector will only be adapted at generation $t = 0$, $t = 0.2 \times t_{max}$, $t = 0.4 \times t_{max}$, $t = 0.6 \times t_{max}$ and $t = 0.8 \times t_{max}$, respectively. The detailed sensitivity analysis of f_r can be found in Supplementary Materials III.

Note that since the proposed reference vector adaptation strategy is only motivated to deal with problems with scaled objectives, it does not guarantee a uniform distribution of the reference vectors on any type of PFs, especially on those with irregular geometrical features, e.g., disconnection or degeneration. In order to handle such irregular PFs, we



(a) Pareto optimal solutions specified by 10 uniformly distributed reference vectors on a PF with objectives normalized to the same range.

(b) Pareto optimal solutions specified by 10 uniformly distributed reference vectors on a PF with objectives scaled different ranges.

(c) Pareto optimal solutions specified by 10 adapted reference vectors on a PF with objectives scaled to different ranges.

Fig. 3. The Pareto optimal solutions (solid dots) specified by different reference vectors (dashed arrows) on different PF (solid lines).

have proposed another reference vector regeneration strategy in Section VI. Nevertheless, it is conceded that the proposed reference vector adaptation (as well as regeneration) strategy is not able to comfortably handle all specific situations, e.g., when a PF has low tails or sharp peaks [73].

E. Computational Complexity of the Proposed RVEA

To analyze the computational complexity of the proposed RVEA, we consider the main steps in one generation in the main loop of Algorithm 1. Apart from genetic operations such as crossover and mutation, the main computational cost is resulted from the reference vector guided selection procedure and the reference vector adaptation mechanism.

As shown in Algorithm 2, the reference vector guided selection procedure consists of the following components: objective value translation, population partition, angle-penalized distance (APD) calculation and elitism selection. We will see that the computational complexity of each component is very low, as will be analyzed in the following. The time complexity for the objective value translation is $O(MN)$, where M is the objective number and N is the population size. The time complexity for population partition is $O(MN^2)$. In addition, calculation of APD and elitism selection hold a computational complexity of $O(MN^2)$ and $O(N^2)$ in the worst case, respectively. Finally, the computational complexity for the reference vector adaptation procedure is $O(MN/(f_r \cdot t_{max}))$, where f_r and t_{max} denote the frequency to employ the reference vector adaptation strategy and maximal number of generations, respectively.

To summarize, apart from the genetic variations, the worst-case overall computational complexity of RVEA within one generation is $O(MN^2)$, which indicates that RVEA is computationally efficient.

IV. Comparative Studies

In this section, empirical experiments are conducted on 15 benchmark test problems taken from two widely used test suites, i.e., the DTLZ [74] test suite (including the scaled version [39]) and the WFG test suite [71], to compare RVEA with five state-of-the-art MOEAs for many-objective optimization, namely, MOEA/DD [41], NSGA-III [39], MOEA/D-PBI [28], GrEA [22], and KnEA [27]. For each test problem, objective numbers varying from 3 to 10, i.e., $M \in \{3, 6, 8, 10\}$ are considered.

In the following subsections, we first present a brief introduction to the benchmark test problems and the performance indicator used in our comparative studies. Then, the parameter settings used in the comparisons are given. Then, each algorithm is run for 20 times on each test problem independently, and the Wilcoxon rank sum test is adopted to compare the results obtained by RVEA and those by five compared algorithms at a significance level of 0.05. Symbol '$+$' indicates that the compared algorithm is significantly outperformed by RVEA according to a Wilcoxon rank sum test, while '$-$' means that RVEA is significantly outperformed by the compared algorithm. Finally, '$\approx$' means that there is no statistically significant difference between the results obtained by RVEA and the compared algorithm.

A. Benchmark Test Problems

The first four test problems are DTZL1 to DTLZ4 taken from the DTLZ test suite [74]. As recommended in [74], the number of decision variables is set to $n = M + K - 1$, where M is the objective number, $K = 5$ is used for DTLZ1, $K = 10$ is used for DTLZ2, DTLZ3 and DTLZ4.

We have also used the scaled version of the DTLZ1 and DTLZ3 (denoted as SDTLZ1 and SDTLZ3) for comparisons to see if the proposed RVEA is capable of handling strongly scaled problems. The scaling approach is recommended in [39], where each objective is multiplied by a coefficient p^{i-1}, where p is a parameter that controls the scaling size and $i = 1, .., M$ is the objective index. For example, given $p = 10$, the objectives of a 3-objective problem will be scaled to be $10^0 \times f_1$, $10^1 \times f_2$ and $10^2 \times f_3$. In our experiments, the values of p are set to $10, 5, 3, 2$ for problems with an objective number $M = 3, 6, 8, 10$, respectively .

The other nine test problems are WFG1 to WFG9 taken from the WFG test suite [71], [75], which are designed by introducing difficulties in both the decision space (e.g. non-separability, deception and bias) and the objective space (e.g.

TABLE I
SETTINGS OF POPULATION SIZES IN RVEA, MOEA/DD, NSGA-III AND MOEA/D-PBI.

M	(H_1, H_2)	Population size
3	(13, 0)	105
6	(4, 1)	132
8	(3, 2)	156
10	(3, 2)	275

H_1 and H_2 are the the simplex-lattice design factors for generating uniformly distributed reference (or weight) vectors on the outer boundaries and the inside layers, respectively.

mixed geometrical structures of the PFs). As suggested in [71], the number of decision variables is set as $n = K + L - 1$, where M is the objective number, the distance-related variable $L = 10$ is used in all test problems, and the position-related variable $K = 4, 10, 7, 9$ are used for test problems with $M = 3, 6, 8, 10$, respectively.

B. Performance Indicators

To make empirical comparisons between the results obtained by each algorithm, the hypervolume (HV) [46] is used as the performance indicator in the comparisons.

Let $y^* = (y_1^*, ..., y_M^*)$ be a reference point in the objective space that is dominated by all Pareto optimal solutions, and P be the approximation to PF obtained by an MOEA. The HV value of P (with respect to y^*) is the volume of the region which dominates y^* and is dominated by P.

In this work, $y^* = (1.5, 1.5, ..., 1.5)$ is used for DTLZ1, SDTLZ1; $y^* = (2, 2, ..., 2)$ is used for DTLZ2, DTLZ3, SDTLZ3, DTLZ4; and $y^* = (3, 5, ..., 2M + 1)$ is used for WFG1 to WFG9. For problems with fewer than 8 objectives, the recently proposed fast hypervolume calculation method is adopted to calculate the exact hypervolume [76], while for 8-objective and 10-objective problems, the Monte Carlo method [43] with 1,000,000 sampling points is adopted to obtain the approximate hypervolume values. All hypervolume values presented in this work are all normalized to $[0, 1]$ by dividing $\prod_{i=1}^{m} y_i^*$.

C. Parameter Settings

In this subsection, we first present the general parameter settings for the experiments, and afterwards, the specific parameter settings for each algorithm in comparison are given.

1) Settings for Crossover and Mutation Operators: For the simulated binary crossover [66], the distribution index is set to $\eta_c = 30$ in RVEA, MOEA/DD and NSGA-III, $\eta_c = 20$ in the other four algorithms, and the crossover probability $p_c = 1.0$ is used in all algorithms; for the polynomial mutation [67], the distribution index and the mutation probability are set to $\eta_m = 20$ and $p_m = 1/n$, respectively, as recommended in [77].

2) Population Size: For RVEA, MOEA/DD, NSGA-III and MOEA/D-PBI, the population size is determined by the simplex-lattice design factor H together with the objective number M, referring to (2). As recommended in [39], [41], for problems with $M \geq 8$, a two-layer vector generation strategy can be applied to generate reference (or weight) vectors not only on the outer boundaries but also on the inside layers of the Pareto fronts. The detailed settings of the population sizes in RVEA, MOEA/DD, NSGA-III and MOEA/D-PBI are summarized in Table I. For the other two algorithms, GrEA and KnEA, the population sizes are also set according to Table I, with respect to different objective numbers M.

TABLE II
PARAMETER SETTING OF div IN GREA ON EACH TEST INSTANCE

Problem	M = 3	M = 4	M = 6	M = 8	M = 10
DTLZ1	16	10	10	10	11
DTLZ2	15	10	10	8	12
DTLZ3	17	11	11	10	11
DTLZ4	15	10	8	8	12
SDTLZ1	16	10	10	10	11
SDTLZ3	17	11	11	10	11
WFG1	10	8	9	7	10
WFG2	12	11	11	11	11
WFG3	22	18	18	16	22
WFG4–9	15	10	9	8	12

TABLE III
PARAMETER SETTING OF T IN KNEA ON EACH TEST INSTANCE

Problem	M = 3	M = 4	M = 6	M = 8	M = 10
DTLZ1	0.6	0.6	0.2	0.1	0.1
DTLZ3	0.6	0.4	0.2	0.1	0.1
SDTLZ1	0.6	0.6	0.2	0.1	0.1
SDTLZ3	0.6	0.4	0.2	0.1	0.1
others	0.5	0.5	0.5	0.5	0.5

3) Termination Condition: The termination condition of each run is the maximal number of generations. For DTLZ1, SDTLZ1, DTLZ3, SDTLZ3 and WFG1 to WFG9, the maximal number of generations is set to 1000. For DTLZ2 and DTLZ4, the maximal number of generations is set to 500.

4) Specific Parameter Settings in Each Algorithm: For MOEA/D-PBI, the neighborhood size T is set to 20, and the penalty parameter θ in PBI is set to 5, as recommended in [28], [39]. For MOEA/DD, T and θ are set to the same values as in MOEA/D-PBI, and the neighborhood selection probability is set to $\delta = 0.9$, as recommended in [41]. For GrEA and KnEA, the detailed parameter settings are listed in Table II and Table III, respectively, which are all recommended settings by the authors [22], [27].

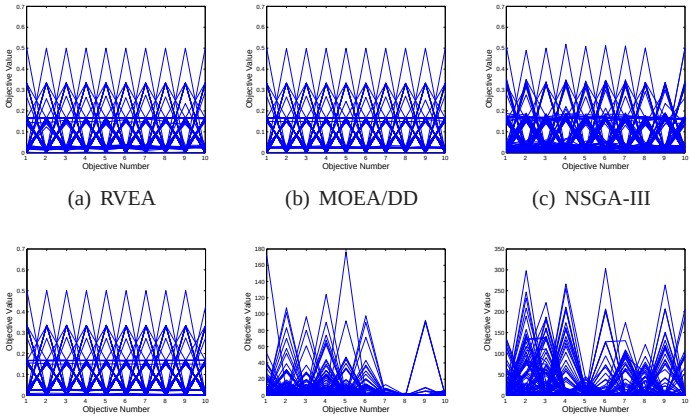
Two parameters in RVEA require to be predefined, i.e., the index α used to control the rate of change of the penalty function in (9), and the frequency f_r to employ the reference vector adaptation in Algorithm 3. In the experimental comparisons, $\alpha = 2$ and $f_r = 0.1$ is used for all test instances. A sensitivity analysis of α and f_r is provided in Supplementary Materials III.

In order to reduce the time cost of non-dominated sorting, the efficient non-dominated sorting approach ENS-SS reported in [78] has been adopted in NSGA-III, GrEA and KnEA, and a steady-state non-dominated sorting approach as reported in [79] has been adopted in MOEA/DD. By contrast, neither RVEA nor MOEA/D-PBI uses dominance comparisons. All the algorithms are realized in Matlab R2012a[5]

[5]The Matlab source code of RVEA can be downloaded from: http://www.soft-computing.de/jin-pub_year.html

except MOEA/DD, which is implemented in the jMetal framework [80].

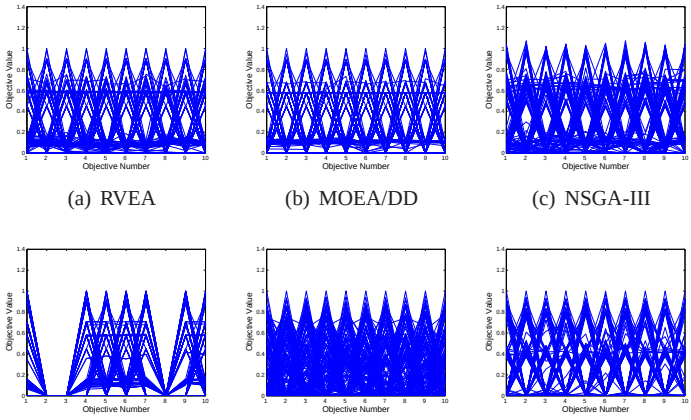
D. Performance on DTLZ1 to DTLZ4, SDTLZ1 and SDTLZ3



(a) RVEA (b) MOEA/DD (c) NSGA-III

(d) MOEA/D-PBI (e) GrEA (f) KnEA

Fig. 4. The parallel coordinates of non-dominated front obtained by each algorithm on 10-objective DTLZ1 in the run associated with the median HV value.



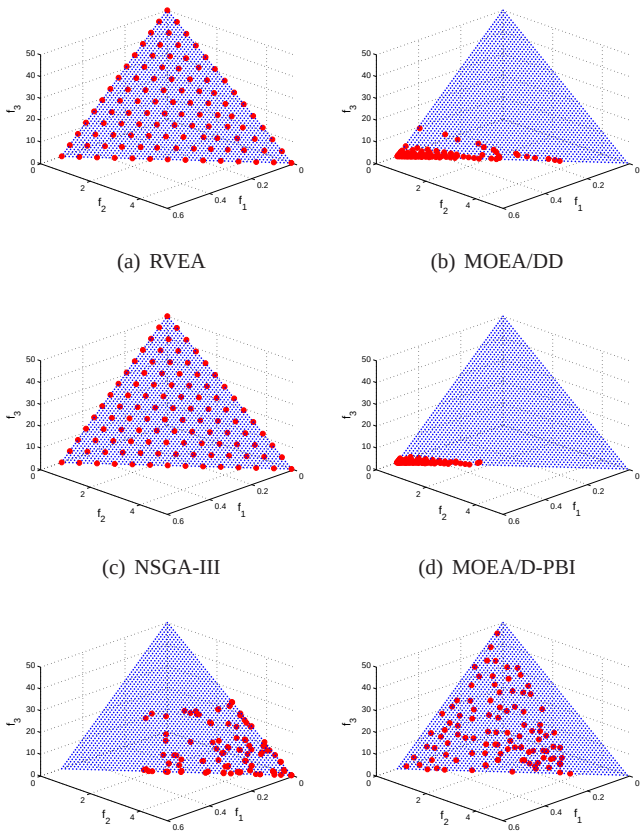
(a) RVEA (b) MOEA/DD (c) NSGA-III

(d) MOEA/D-PBI (e) GrEA (f) KnEA

Fig. 5. The parallel coordinates of non-dominated front obtained by each algorithm on 10-objective DTLZ4 in the run associated with the median HV value.

The statistical results of the HV values obtained by the six algorithms over 20 independent runs are summarized in Table IV, where the best results are highlighted. It can be seen that RVEA, together with MOEA/DD, shows best overall performance among the six compared algorithms on the four original DTLZ test instances, while NSGA-III shows the best overall performance on the scaled DTLZ test instances.

As can be observed from Fig. 4, the approximate PFs obtained by RVEA, MOEA/DD and MOEA/D-PBI show promising convergence performance as well as a good distribution on 10-objective DTLZ1. The statistical results in Table IV also indicate that RVEA, MOEA/DD and MOEA/D-PBI have achieved the best performance among the six algorithms on all DTLZ1 instances, where MOEA/DD shows best performance on 3-objective instance and RVEA shows best performance



(a) RVEA (b) MOEA/DD

(c) NSGA-III (d) MOEA/D-PBI

(e) GrEA (f) KnEA

Fig. 6. The non-dominated solutions obtained by each algorithm on 3-objective SDTLZ1 in the run associated with the median HV value.

on 6-objective and 8-objective instances. Meanwhile, the PFs approximated by NSGA-III is also of high quality. By contrast, neither GrEA nor KnEA is able to converge to the true PF of 10-objective DTLZ1.

Similar observations can be made about the results on DTLZ2, a relatively simple test problem, to those on DTLZ1. RVEA shows the best performance on the 8-objective instance, while MOEA/DD outperforms RVEA on 3-objective and 6-objective instances. NSGA-III and MOEA/D-PBI are slightly outperformed by RVEA. Compared to the performance on 10-objective DTLZ1, the performance of GrEA and KnEA on this instance is much better.

For DTLZ3, which is a highly multimodal problem, RVEA and MOEA/DD have also obtained an approximate PF of high quality. It seems that the performance of NSGA-III and MOEA/D-PBI is not very stable on high dimensional (8-objective and 10-objective) instances of this problem, as evidenced by the statistical results in Table IV, while GrEA and KnEA completely fail to reach the true PF of this problem.

DTLZ4 is test problem where the density of the points on the true PF is strongly biased. This test problem is designed to verify whether an MOEA is able to maintain a proper distribution of the candidate solutions. From the results, we can see that RVEA and MOEA/DD remain to show the best overall performance. By contrast, MOEA/D-PBI is generally outperformed by the other five algorithms. As can be observed

TABLE IV
THE STATISTICAL RESULTS (MEAN AND STANDARD DEVIATION) OF THE HV VALUES OBTAINED BY RVEA, MOEA/DD, NSGA-III, MOEA/D-PBI, GREA AND KNEA ON DTLZ1 TO DTLZ4, SDTLZ1 AND SDTLZ3. THE BEST RESULTS ARE HIGHLIGHTED.

Problem	M	RVEA	MOEA/DD	NSGA-III	MOEA/D-PBI	GrEA	KnEA
DTLZ1	3	0.992299 (0.000027)	**0.992321 (0.000005)**$^-$	0.992275 (0.000017)$^+$	0.992217 (0.000068)$^+$	0.951715 (0.053104)$^+$	0.961327 (0.022927)$^+$
	6	0.999966 (0.000021)	0.999965 (0.000017)$^\approx$	0.999951 (0.000060)$^\approx$	**0.999969 (0.000007)**$^\approx$	0.806853 (0.101147)$^+$	0.902769 (0.064009)$^+$
	8	**0.999999 (0.000000)**	0.999984 (0.000016)$^+$	0.999993 (0.000022)$^+$	0.999998 (0.000001)$^+$	0.949778 (0.081134)$^+$	0.774819 (0.054991)$^+$
	10	**0.999999 (0.000000)**	0.999995 (0.000005)$^+$	0.999992 (0.000027)$^\approx$	0.999999 (0.000000)$^\approx$	0.950476 (0.088517)$^+$	0.906688 (0.070628)$^+$
DTLZ2	3	0.926994 (0.000041)	**0.927292 (0.000002)**$^-$	0.927012 (0.000032)$^\approx$	0.926808 (0.000042)$^+$	0.926675 (0.000122)$^+$	0.925387 (0.000225)$^+$
	6	0.995935 (0.000175)	**0.996096 (0.000146)**$^-$	0.995689 (0.000062)$^+$	0.995872 (0.000078)$^\approx$	0.996049 (0.000141)$^\approx$	0.995249 (0.000250)$^+$
	8	**0.999338 (0.000096)**	0.999330 (0.000079)$^\approx$	0.998223 (0.002551)$^+$	0.999333 (0.000024)$^+$	0.999059 (0.000065)$^+$	0.999104 (0.000119)$^+$
	10	0.999912 (0.000040)	**0.999920 (0.000022)**$^\approx$	0.999804 (0.000222)$^+$	0.999917 (0.000011)$^+$	0.999921 (0.000022)$^\approx$	0.999841 (0.000209)$^+$
DTLZ3	3	0.924421 (0.001273)	**0.926921 (0.000316)**$^-$	0.925650 (0.000664)$^+$	0.924153 (0.001546)$^+$	0.869279 (0.120728)$^+$	0.894023 (0.041540)$^+$
	6	0.995596 (0.000228)	**0.995848 (0.000203)**$^-$	0.768098 (0.402591)$^+$	0.970814 (0.077681)$^+$	0.592477 (0.120285)$^+$	0.523015 (0.073019)$^+$
	8	**0.999350 (0.000090)**	0.999338 (0.000069)$^\approx$	0.683011 (0.452841)$^+$	0.997552 (0.004021)$^+$	0.249152 (0.262630)$^+$	0.596495 (0.129071)$^+$
	10	**0.999919 (0.000027)**	0.999914 (0.000033)$^\approx$	0.588230 (0.507488)$^+$	0.883623 (0.195053)$^+$	0.173715 (0.287635)$^+$	0.665098 (0.163428)$^+$
DTLZ4	3	0.926922 (0.000049)	**0.927295 (0.000001)**$^-$	0.926947 (0.000035)$^\approx$	0.761109 (0.188313)$^+$	0.913877 (0.040960)$^+$	0.925410 (0.000274)$^+$
	6	0.995886 (0.000218)	**0.995972 (0.000150)**$^\approx$	0.992253 (0.008107)$^+$	0.958608 (0.028688)$^+$	0.995811 (0.000204)$^\approx$	0.995341 (0.000240)$^+$
	8	0.999359 (0.000043)	**0.999374 (0.000057)**$^\approx$	0.999016 (0.000055)$^+$	0.987237 (0.013487)$^+$	0.999164 (0.000079)$^+$	0.999181 (0.000041)$^+$
	10	0.999915 (0.000036)	0.999916 (0.000019)$^\approx$	0.999844 (0.000023)$^+$	0.998947 (0.000615)$^+$	**0.999921 (0.000031)**$^\approx$	0.999904 (0.000014)$^\approx$
SDTLZ1	3	0.942902 (0.000374)	0.842188 (0.000255)$^+$	**0.943320 (0.000258)**$^-$	0.752965 (0.026594)$^+$	0.811754 (0.135200)$^+$	0.890970 (0.020920)$^+$
	6	0.987277 (0.005948)	0.825625 (0.005232)$^+$	**0.994307 (0.004067)**$^-$	0.784053 (0.009784)$^+$	0.787636 (0.148348)$^+$	0.822863 (0.119886)$^+$
	8	0.977781 (0.013207)	0.918495 (0.004078)$^+$	**0.994885 (0.005614)**$^-$	0.879281 (0.008595)$^+$	0.603896 (0.285257)$^+$	0.638502 (0.124225)$^+$
	10	0.997845 (0.003597)	0.968728 (0.005022)$^+$	**0.999669 (0.000593)**$^-$	0.961194 (0.002466)$^+$	0.699631 (0.172086)$^+$	0.784741 (0.125824)$^+$
SDTLZ3	3	0.728992 (0.004089)	0.611762 (0.005177)$^+$	**0.731667 (0.002343)**$^\approx$	0.510602 (0.018017)$^+$	0.704035 (0.063554)$^+$	0.687029 (0.038566)$^+$
	6	0.645483 (0.099350)	0.628237 (0.011612)$^+$	**0.694999 (0.360772)**$^-$	0.545299 (0.002836)$^+$	0.008744 (0.027535)$^+$	0.003786 (0.009766)$^+$
	8	**0.673466 (0.071693)**	0.667774 (0.005031)$^+$	0.337208 (0.378458)$^+$	0.556130 (0.000727)$^+$	0.057487 (0.119544)$^+$	0.370506 (0.213103)$^+$
	10	**0.869416 (0.044609)**	0.573527 (0.005793)$^+$	0.753836 (0.397503)$^+$	0.727759 (0.004431)$^+$	0.018452 (0.019051)$^+$	0.470806 (0.187499)$^+$

$+$: RVEA shows significantly better performance in the comparison.
$-$: RVEA shows significantly worse performance in the comparison.
$\approx$: There is no significant difference between the compared results.

from Fig. 5, it appears that MOEA/D-PBI is only able to find some parts of the true PF. The performance of NSGA-III is similar to that of RVEA. An interesting observation is that, although the distribution of the approximate PFs obtained by GrEA and KnEA look slightly noisy, the solutions are still relatively evenly distributed, and thus the HV values obtained by these two algorithms are very encouraging, especially on the 8-objective and 10-objective instances.

Compared with the original DTLZ problems, the SDTLZ1 and SDTLZ3 are challenging due to the strongly scaled objective function values. It turns out that NSGA-III shows the best overall performance on SDTLZ1, 3-objective and 6-objective SDTLZ3, while RVEA shows the best overall performance on 8-objective and 10-objective SDTLZ3. As shown in Fig. 6, RVEA and NSGA-III are the only two algorithms that are able to generate evenly distributed solutions on the 3-objective SDTLZ1, while the other four algorithms have all failed. Such observations are consistent with those reported in [39], where it has been shown that even the normalized version of MOEA/D still does not work on such scaled DTLZ problems.

E. Performance on WFG1 to WFG9

As evidenced by statistical results of the HV values summarized in Table V, RVEA has shown the most competitive performance on WFG4, WFG5, WFG6, WFG7 and WFG9, while MOEA/DD, NSGA-III, GrEA and GrEA have achieved the best performance on WFG8, WFG2, WFG3 and WFG1, respectively. By contrast, the performance of MOEA/D-PBI on the WFG test functions is not as good as that on the DTLZ test functions. In the following, some discussions on the experimental results will be presented.

WFG1 is designed with flat bias and a mixed structure of the PF. Although RVEA is slightly outperformed by GrEA and KnEA, its performance is still significantly better than NSGA-III and MOEA/D-PBI. WFG2 is a test problem which has a disconnected PF. It can be observed that although the HV values obtained by each algorithm vary on this test problem, the overall performance is generally very good. More specifically, the overall performance of RVEA is better than MOEA/D-PBI and GrEA, while NSGA-III has achieved the best overall performance. WFG3 is a difficult problem where the PF is degenerate and the decision variables are non-separable. On this problem, RVEA has achieved comparable higher HV values than MOEA/D-PBI, but has been outperformed by the other four algorithms, where GrEA has achieved the largest HV values on all the instances of this problem.

WFG4 to WFG9 are designed with different difficulties in the decision space, e.g., multimodality for WFG4, landscape deception for WFG5 and non-separability for WFG6, WFG8 and WFG9, though the true PFs are of the same convex structure. As can be observed in Table V, RVEA shows the most competitive overall performance on these six problems by achieving the best results on 14 out of 24 instances. By contrast, GrEA shows high effectiveness on most 3-objective instances, KnEA shows promising performance on some 8-objective and 10-objective instances; MOEA/DD and NSGA-

TABLE V
THE STATISTICAL RESULTS (MEAN AND STANDARD DEVIATION) OF THE HV VALUES OBTAINED BY RVEA, MOEA/DD, NSGA-III, MOEA/D-PBI, GREA AND KNEA ON WFG1 TO WFG9. THE BEST RESULTS ARE HIGHLIGHTED.

Problem	M	RVEA	MOEA/DD	NSGA-III	MOEA/D-PBI	GrEA	KnEA
WFG1	3	0.865072 (0.044144)	0.853637 (0.014545)$^{\approx}$	0.880642 (0.037913)$^{\approx}$	0.808559 (0.039850)$^{+}$	0.948634 (0.004706)$^{-}$	**0.958812 (0.003336)**$^{-}$
	6	0.838645 (0.052536)	0.847766 (0.036001)$^{\approx}$	0.760115 (0.030498)$^{+}$	0.937861 (0.034231)$^{+}$	0.978189 (0.002217)$^{-}$	**0.987990 (0.021758)**$^{-}$
	8	0.828420 (0.071859)	0.874047 (0.029795)$^{\approx}$	0.636388 (0.041800)$^{+}$	0.873566 (0.084635)$^{\approx}$	0.984504 (0.003077)$^{-}$	**0.993004 (0.003330)**$^{-}$
	10	0.912619 (0.059865)	0.862892 (0.004600)$^{+}$	0.570728 (0.048639)$^{+}$	0.802624 (0.118040)$^{+}$	0.989678 (0.002609)$^{-}$	**0.995399 (0.003227)**$^{-}$
WFG2	3	0.911168 (0.070014)	0.911115 (0.070477)$^{\approx}$	0.913209 (0.069967)$^{\approx}$	0.850708 (0.072264)$^{+}$	0.907856 (0.069149)$^{\approx}$	**0.941823 (0.046468)**$^{-}$
	6	0.929838 (0.084468)	0.969481 (0.004744)$^{\approx}$	0.974124 (0.058171)$^{-}$	0.824505 (0.068663)$^{+}$	0.959854 (0.055499)$^{\approx}$	**0.992704 (0.000971)**$^{-}$
	8	0.985362 (0.002434)	0.964892 (0.005227)$^{+}$	**0.995848 (0.001354)**$^{-}$	0.952490 (0.002702)$^{+}$	0.974283 (0.002684)$^{+}$	0.994149 (0.000685)$^{-}$
	10	0.990577 (0.002493)	0.962456 (0.004293)$^{+}$	**0.997515 (0.000882)**$^{-}$	0.953995 (0.001703)$^{+}$	0.984013 (0.002625)$^{+}$	0.995553 (0.000637)$^{-}$
WFG3	3	0.548860 (0.008423)	0.547597 (0.003333)$^{\approx}$	0.571840 (0.003352)$^{-}$	0.536581 (0.023869)$^{\approx}$	**0.581615 (0.001209)**$^{-}$	0.572039 (0.005376)$^{-}$
	6	0.146476 (0.033719)	0.193116 (0.016724)$^{-}$	0.324412 (0.026666)$^{-}$	0.120196 (0.030687)$^{\approx}$	**0.375592 (0.001591)**$^{-}$	0.218213 (0.038518)$^{-}$
	8	0.022206 (0.028688)	0.092131 (0.011635)$^{-}$	0.165832 (0.045706)$^{-}$	0.001958 (0.003823)$^{+}$	**0.299918 (0.002411)**$^{-}$	0.172217 (0.027005)$^{-}$
	10	0.010805 (0.015590)	0.049190 (0.009707)$^{-}$	0.103724 (0.050683)$^{-}$	0.000000 (0.000000)$^{+}$	**0.277491 (0.001015)**$^{-}$	0.124849 (0.023234)$^{-}$
WFG4	3	0.722421 (0.001194)	0.727762 (0.000320)$^{-}$	0.716893 (0.002451)$^{+}$	0.693973 (0.006281)$^{+}$	**0.727825 (0.000843)**$^{-}$	0.718904 (0.001444)$^{+}$
	6	0.862869 (0.008112)	**0.884741 (0.001449)**$^{-}$	0.822742 (0.035516)$^{+}$	0.685602 (0.031404)$^{+}$	0.865653 (0.004690)$^{\approx}$	0.854176 (0.003003)$^{+}$
	8	**0.931277 (0.001998)**	0.905293 (0.005383)$^{+}$	0.880990 (0.012305)$^{+}$	0.595068 (0.028301)$^{+}$	0.873270 (0.003215)$^{+}$	0.920423 (0.004544)$^{+}$
	10	**0.957305 (0.003455)**	0.903194 (0.006120)$^{+}$	0.903346 (0.008478)$^{+}$	0.666928 (0.045705)$^{+}$	0.936195 (0.004850)$^{+}$	0.941286 (0.005063)$^{+}$
WFG5	3	**0.691464 (0.002019)**	0.687568 (0.003520)$^{+}$	0.685643 (0.001488)$^{+}$	0.677525 (0.001467)$^{+}$	0.691238 (0.000467)$^{\approx}$	0.685238 (0.001966)$^{+}$
	6	**0.849224 (0.004178)**	0.826789 (0.001754)$^{+}$	0.822939 (0.004400)$^{+}$	0.713608 (0.022588)$^{+}$	0.831251 (0.002928)$^{+}$	0.826200 (0.003230)$^{+}$
	8	**0.893446 (0.001405)**	0.852921 (0.005773)$^{+}$	0.876518 (0.002323)$^{+}$	0.661992 (0.024267)$^{+}$	0.843985 (0.005326)$^{+}$	0.881075 (0.003052)$^{+}$
	10	**0.915470 (0.000684)**	0.846836 (0.002304)$^{+}$	0.897372 (0.001389)$^{+}$	0.698976 (0.012554)$^{+}$	0.903894 (0.002522)$^{+}$	0.907063 (0.001088)$^{+}$
WFG6	3	**0.688069 (0.008413)**	0.683466 (0.009075)$^{\approx}$	0.675559 (0.006628)$^{+}$	0.664349 (0.013807)$^{+}$	0.687321 (0.008883)$^{\approx}$	0.677556 (0.009792)$^{+}$
	6	**0.839476 (0.018356)**	0.829075 (0.018391)$^{+}$	0.827357 (0.021420)$^{\approx}$	0.571130 (0.077250)$^{+}$	0.831224 (0.009766)$^{+}$	0.821218 (0.017134)$^{+}$
	8	**0.875538 (0.027472)**	0.829947 (0.019581)$^{+}$	0.858480 (0.013466)$^{+}$	0.513394 (0.026226)$^{+}$	0.840416 (0.017071)$^{+}$	0.842453 (0.013495)$^{+}$
	10	**0.895348 (0.017421)**	0.816800 (0.020135)$^{+}$	0.881965 (0.012155)$^{+}$	0.517089 (0.013090)$^{+}$	0.894585 (0.015375)$^{+}$	0.893298 (0.016330)$^{+}$
WFG7	3	0.729925 (0.000389)	0.728672 (0.000178)$^{+}$	0.727215 (0.001203)$^{+}$	0.713092 (0.003740)$^{+}$	**0.731639 (0.000274)**$^{-}$	0.726503 (0.000914)$^{+}$
	6	**0.898577 (0.004851)**	0.889818 (0.000888)$^{+}$	0.875921 (0.041999)$^{+}$	0.703274 (0.045351)$^{+}$	0.890273 (0.002230)$^{\approx}$	0.892166 (0.003037)$^{+}$
	8	**0.937539 (0.002340)**	0.915130 (0.005266)$^{+}$	0.932110 (0.002068)$^{+}$	0.587896 (0.018747)$^{+}$	0.910003 (0.003575)$^{+}$	0.932208 (0.005226)$^{+}$
	10	0.967352 (0.001810)	0.922808 (0.003271)$^{+}$	0.951173 (0.010827)$^{+}$	0.583912 (0.004645)$^{+}$	0.970362 (0.001082)$^{-}$	**0.971008 (0.000842)**$^{-}$
WFG8	3	0.612003 (0.004350)	0.627505 (0.000737)$^{-}$	0.612757 (0.004250)$^{\approx}$	0.601874 (0.009641)$^{+}$	**0.632657 (0.002170)**$^{-}$	0.615693 (0.001789)$^{-}$
	6	0.675580 (0.043175)	**0.846348 (0.020575)**$^{-}$	0.657068 (0.027923)$^{+}$	0.423092 (0.075422)$^{+}$	0.793957 (0.024650)$^{-}$	0.728246 (0.037332)$^{-}$
	8	0.701451 (0.102467)	**0.855585 (0.017945)**$^{-}$	0.761618 (0.020636)$^{\approx}$	0.241909 (0.032016)$^{+}$	0.750641 (0.057737)$^{\approx}$	0.782234 (0.021242)$^{\approx}$
	10	0.781393 (0.101569)	**0.878498 (0.011198)**$^{-}$	0.804495 (0.009241)$^{\approx}$	0.269943 (0.096638)$^{+}$	0.844060 (0.007662)$^{-}$	0.864081 (0.035497)$^{-}$
WFG9	3	0.672551 (0.042637)	0.678869 (0.032140)$^{-}$	0.669388 (0.041506)$^{+}$	0.648216 (0.024442)$^{+}$	**0.697834 (0.001936)**$^{-}$	0.687694 (0.001493)$^{\approx}$
	6	**0.803601 (0.009998)**	0.772578 (0.007200)$^{+}$	0.733317 (0.058866)$^{+}$	0.619386 (0.043888)$^{+}$	0.793840 (0.004384)$^{+}$	0.792788 (0.039524)$^{\approx}$
	8	**0.870772 (0.043400)**	0.819450 (0.011535)$^{+}$	0.858150 (0.010268)$^{+}$	0.519427 (0.092773)$^{+}$	0.841801 (0.007196)$^{+}$	0.861791 (0.004099)$^{+}$
	10	0.911081 (0.008760)	0.788889 (0.021221)$^{+}$	0.867112 (0.040977)$^{+}$	0.493999 (0.141804)$^{+}$	0.909293 (0.005168)$^{+}$	**0.911170 (0.003321)**$^{\approx}$

$+$: RVEA shows significantly better performance in the comparison.
$-$: RVEA shows significantly worse performance in the comparison.
$\approx$: There is no significant difference between the compared results.

III also show generally competitive performance.

F. Comparisons between APD, TCH and PBI

In principle, most scalarization approaches used in the decomposition based MOEAs are also applicable to the proposed RVEA. In this section, we will carry out some comparative studies on the original RVEA, RVEA with the weighted Tchebycheff (TCH) approach [55], and RVEA with the penalty-based boundary intersection (PBI) approach [28]. The detailed definitions of the scalarization functions of TCH approach and PBI approach can be found in the Supplementary Materials IV.

In order to apply the TCH and PBI approaches in RVEA, we simply replace the weight vectors with the reference vectors, and then replace the APD part (between Line 19 to Line 23 in Algorithm 2) with the two approaches. For simplicity, RVEA with the TCH and PBI approaches are denoted as RVEA-TCH and RVEA-PBI, respectively.

The performance of RVEA-TCH and RVEA-PBI has been verified on six benchmark test problems selected from different test suites, including DTLZ1, DTLZ3, SDTLZ1, SDTLZ3, WFG4 and WFG5. The results are compared with those obtained by the original RVEA. As shown by the statistical results summarized in Table VI, RVEA shows the best performance on SDTLZ1, SDTLZ3, WFG4, WFG5, and significantly outperforms RVEA-TCH on DTLZ1 and DTLZ3.

It is also observed that RVEA-PBI shows very close performance to RVEA on the original DTLZ problems (DTLZ1 and DTLZ3), but is completely outperformed on the scaled DTLZ problems (SDTLZ1 and SDTLZ3). It implies that the proposed APD has better capability of handling strongly scaled problems than PBI. The superiority of APD over PBI can be attributed to two main reasons. One is that the penalty function in APD is relatively insensitive to the ranges (either scaled or not) of the objective functions, as the penalty function is angle (instead of distance) based, where the angle is normalized inside each subspace according to the local density of the

TABLE VI
THE STATISTICAL RESULTS (MEAN AND STANDARD DEVIATION) OF THE HV VALUES OBTAINED BY RVEA, RVEA-TCH AND RVEA-PBI. THE BEST RESULTS ARE HIGHLIGHTED.

Problem	M	DTLZ1	DTLZ3	SDTLZ1	SDTLZ3	WFG4	WFG5
RVEA	3	**0.992299 (0.000027)**	0.924421 (0.001273)	**0.942750 (0.002020)**	**0.728992 (0.004089)**	**0.722421 (0.001194)**	**0.691464 (0.002019)**
	6	0.999966 (0.000022)	**0.995568 (0.000221)**	**0.931802 (0.026874)**	**0.645483 (0.099350)**	**0.862869 (0.008112)**	**0.849224 (0.004178)**
	8	**0.999999 (0.000000)**	0.999265 (0.000094)	**0.929137 (0.041125)**	**0.673466 (0.071693)**	**0.931277 (0.001998)**	**0.893446 (0.001405)**
	10	**0.999999 (0.000000)**	0.999916 (0.000040)	**0.994282 (0.002017)**	**0.869416 (0.044609)**	**0.957305 (0.003455)**	**0.915470 (0.000684)**
RVEA-TCH	3	0.953517 (0.038431)$^+$	0.819342 (0.120837)$^+$	0.843940 (0.064003)$^+$	0.630476 (0.073516)$^+$	0.717267 (0.004447)$^+$	0.659718 (0.006905)$^+$
	6	0.959546 (0.028536)$^+$	0.860139 (0.161875)$^+$	0.742058 (0.038357)$^+$	0.503601 (0.003011)$^+$	0.548469 (0.041835)$^+$	0.625649 (0.020886)$^+$
	8	0.971264 (0.018376)$^+$	0.973105 (0.051183)$^+$	0.769960 (0.033828)$^+$	0.510808 (0.010686)$^+$	0.547145 (0.041040)$^+$	0.666948 (0.012961)$^+$
	10	0.979620 (0.008317)$^+$	0.974545 (0.026168)$^+$	0.829314 (0.050620)$^+$	0.516442 (0.013395)$^+$	0.614155 (0.042705)$^+$	0.675949 (0.008709)$^+$
RVEA-PBI	3	0.992282 (0.000023)$^\approx$	**0.925089 (0.001512)**$^\approx$	0.855031 (0.059168)$^+$	0.710234 (0.004145)$^+$	0.720274 (0.002474)$^+$	0.690480 (0.002499)$^+$
	6	**0.999970 (0.000017)**$^\approx$	0.995549 (0.000273)$^\approx$	0.743207 (0.041016)$^+$	0.600545 (0.088439)$^+$	0.862551 (0.007594)$^+$	0.838670 (0.003321)$^+$
	8	0.999999 (0.000000)$^\approx$	**0.999278 (0.000095)**$^\approx$	0.905666 (0.022996)$^+$	0.552537 (0.091864)$^+$	0.909863 (0.007500)$^+$	0.875984 (0.003191)$^+$
	10	0.999999 (0.000000)$^\approx$	**0.999925 (0.000023)**$^\approx$	0.971815 (0.007036)$^+$	0.823095 (0.035745)$^+$	0.922820 (0.003179)$^+$	0.879818 (0.001235)$^+$

$+$: RVEA shows significantly better performance in the comparison.
$\approx$: There is no significant difference between the compared results.

reference vectors. The other is that the adaptive strategy in APD emphasizes the convergence at the initial search stage, which is particularly helpful when the objective vectors are scaled to be distant from the ideal point.

V. PREFERENCE ARTICULATION

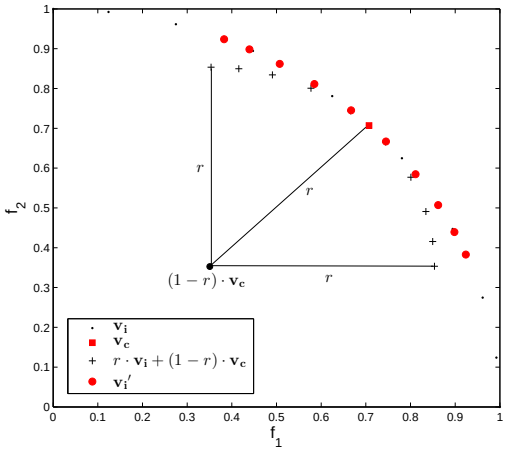
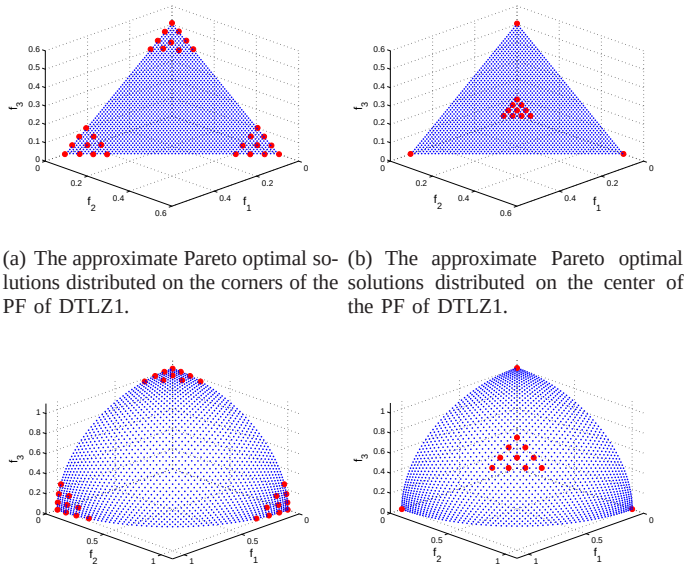


Fig. 7. A visualized illustration of the transformation procedure to generate reference vectors inside a region specified by a central vector $\mathbf{v_c}$ and a radius r. In this example, 10 uniformly distributed reference vectors are generated inside a region in a bi-objective space specified by $\mathbf{v_c} = (\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2})$ and $r = 0.5$.

The proposed RVEA, a method based on reference vectors, is inherently capable of articulating user preferences. Preference articulation is particularly meaningful in many-objective optimization as we can no longer obtain a good (representative) approximation of a high-dimensional PF using a limited population size.

In comparison with the reference point based preference articulation methods [39], [58]–[60], reference vector based preference articulation is more intuitive, as each reference vector specifies a line instead of a point, which means that the Pareto optimal solutions can always be found by RVEA as long as there exists one along the direction specified by the reference vector, regardless where the solution is exactly located. In this section, we will demonstrate the ability of RVEA in preference articulation by providing a few illustrative examples on the 3-objective DTLZ1 and DTLZ2.



(a) The approximate Pareto optimal solutions distributed on the corners of the PF of DTLZ1.

(b) The approximate Pareto optimal solutions distributed on the center of the PF of DTLZ1.

(c) The approximate Pareto optimal solutions distributed on the corners of the PF of DTLZ2.

(d) The approximate Pareto optimal solutions distributed on the center of the PF of DTLZ2.

Fig. 8. The preferred Pareto optimal solutions approximated by RVEA on DTLZ1 and DTLZ2 with reference vectors articulated with different preferences.

To begin with, a preference based reference vector generation method is proposed to uniformly generate reference vectors in a user specified subspace of the objective space. To specify a preferred subspace, a user may first identify a central vector $\mathbf{v_c}$ and a radius r, where $\mathbf{v_c}$ is a unit vector and $r \in (0, 1)$. Then, the reference vectors inside the subspace can

be generated using the following transformation:

$$\mathbf{v_i}' = \frac{r \cdot \mathbf{v_i} + (1-r) \cdot \mathbf{v_c}}{\|r \cdot \mathbf{v_i} + (1-r) \cdot \mathbf{v_c}\|}, \tag{12}$$

where $i = 1, ..., N$ is the index of each reference vector, $\mathbf{v_i}$ denotes a uniformly distributed vector generated with (2) and (3), and $\mathbf{v_i}'$ denotes a reference vector transformed from $\mathbf{v_i}$, which is inside the subspace specified by $\mathbf{v_c}$ and r. A visualized illustration of the above transformation procedure can be found in Fig. 7. With such a preference based reference vector generation method, we are now able to generate uniformly distributed reference vectors inside the preferred subspaces in the objective space.

Firstly, we show an example where the reference vectors are distributed in the corners of the objective space of DTLZ1 and DTLZ2, as shown in Fig. 8(a) and Fig. 8(c). In this example, 10 preference vectors are uniformly generated in each corner of the objective space by setting $\mathbf{v_c} = \{(0,0,1),(0,1,0),(1,0,0)\}$ and $r = 0.2$. As a consequence, RVEA has successfully obtained all the solutions specified by the reference vectors on both test problems.

The second example is to examine if RVEA is capable of dealing with preferences in the center of the objective space. As shown in Fig. 8(b) and Fig. 8(d), the 10 solutions obtained by RVEA show good convergence as well as distribution. This example also implies that RVEA is still able to work effectively with a very small population size, even on a difficult multimodal problem like DTLZ1. In this example, reference vectors are generated with $\mathbf{v_c} = (\frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3}, \frac{\sqrt{3}}{3})$ and $r = 0.2$.

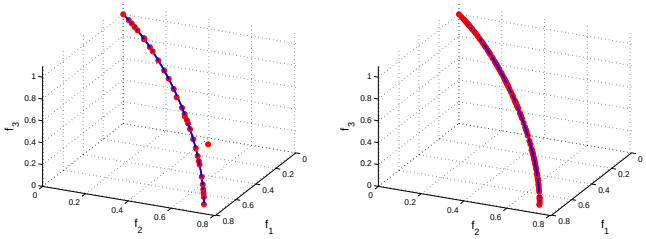
It is worth noting that we have not changed any settings in RVEA to obtain the preferred solutions shown above, except that the reference vector adaptation procedure (on Line 10 in Algorithm 1) has been switched off, such that the distribution of the reference vectors specified by the user preferences will not be changed during the search process. Another point to note is that the extreme vectors, i.e., $(0,0,1)$, $(0,1,0)$ and $(1,0,0)$ in the 3-objective case are always included, because the translation operation, as introduced in Section III-C1, requires the extreme values of each objective function.

VI. Handling Irregular Pareto Fronts

As presented in Section IV, the proposed RVEA is able to perform robustly on a variety of test problems with a set of uniformly distributed reference vectors. In fact, using uniform distributed reference vectors is based on a general assumption that the PF has a regular geometrical structure, i.e., it is smooth, continuous and well spread. However, in practice, the geometrical structure of a PF can be quite irregular due to various reasons. For example, some parts of the objective space are infeasible due to some constraints, or the Pareto optimal solutions only exist in some specific subspaces in the objective space while the other subspaces only contain dominated solutions. In these cases, the PFs can become disconnected or even degenerate. On such problems, if we still use uniformly distributed reference vectors, some of the reference vectors may be associated with no individuals (termed invalid reference vectors hereafter), thus considerably

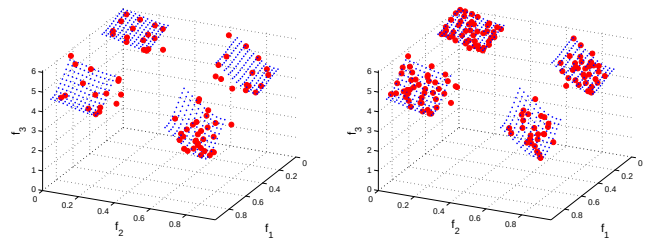
Algorithm 4 The reference vector regeneration strategy for handling irregular Pareto fronts.

1: **Input:** population P_t, the additional reference vector set $V_t^* = \{\mathbf{v}_{t,1}^*, \mathbf{v}_{t,2}^*, ..., \mathbf{v}_{t,N}^*\}$;
2: **Output:** the additional reference vector set V_{t+1}^* for next generation;
3: Remove the dominated solutions in P_t;
4: /*Objective Value Translation*/
5: Perform objective value translation using the operations from Line 4 to Line 7 in Algorithm 2;
6: /*Population Partition*/
7: Perform population partition using the operations from Line 9 to Line 17 in Algorithm 2;
8: /*Reference Vector Regeneration*/
9: Calculate maximal objective values $\mathbf{z}_t^{max}$;
10: **for** i = 1 to $|V_t^*|$ **do**
11: **if** $\overline{P}_{t,i} == \emptyset$ **then**
12: **for** j = 1 to N **do**
13: $\mathbf{u}_{r,j} \leftarrow$ a value randomized within $[0, \mathbf{z}_{t,j}^{max}]$;
14: **end for**
15: $\mathbf{v}_{t+1,i}^* = \frac{\mathbf{u}_r}{\|\mathbf{u}_r\|}$;
16: **else**
17: $\mathbf{v}_{t+1,i}^* = \mathbf{v}_{t,i}^*$;
18: **end if**
19: **end for**



(a) The Pareto optimal solutions approximated by RVEA on 3-objective DTLZ5.

(b) The Pareto optimal solutions approximated by RVEA* on 3-objective DTLZ5.



(c) The Pareto optimal solutions approximated by RVEA on 3-objective DTLZ7.

(d) The Pareto optimal solutions approximated by RVEA* on 3-objective DTLZ7.

Fig. 9. The Pareto optimal solutions approximated by RVEA* and RVEA in the run associated with the median HV values.

reducing the density of the Pareto optimal solutions obtained by RVEA.

To tackle the issue caused by irregular PFs as discussed above, one intuitive idea is to regenerate the invalid reference vectors, as adopted in some existing adaptive approaches for generating reference points [81], [82] as well as reference vectors [56] and weight vectors [83], [84]. However, one difficulty is that we have no idea when the regeneration procedure should be carried out, because even for a regular PF, it is still likely that some reference vectors can be occasionally invalid during the search process. To guarantee a wide spread in the distribution of the candidate solutions, as suggested in the reference point adaptation strategy for NSGA-III [81], the original uniformly distributed reference point set should always be maintained, and some new reference points can be added in the neighborhood of each existing reference point. A newly added reference point will be included in the reference point set if and only if it has an niche count of one. A merit of this adaptive approach is that the new reference points are added in the neighborhood of existing reference points, while a potential demerit is that although a large number of reference points are added in each generation, only a small portion of them can survive, since the sum of niche count is always N. This reduces the efficiency of the algorithm.

In this work, we also propose to maintain the original uniform reference vector set V, while an additional reference vector set (denoted as V^*) is introduced to perform exploration for handling the irregular Pareto fronts. Compared to the reference point adaptive strategy in NSGA-III, the proposed reference vector regeneration strategy is based on "replacement", instead of "addition-and-deletion", which results in higher efficiency. However, since the reference vectors are generated globally, the local solution density is not guaranteed, which can be a potential disadvantage.

The detailed procedure of the proposed reference vector regeneration strategy is summarized in Algorithm 4. At the beginning, all the dominated solutions in the current population are removed. Afterwards, the objective value translation and population partition operations are performed. Based on the result of the population partition, for each empty subpopulation, the associated reference vector will be replaced with a unit vector, which is randomly generated inside the range specified by the minimum[6] and maximum objective values calculated from the candidate solutions in the current population; while for the nonempty subpopulations, the associated reference vectors will remain unchanged. To use the reference vector regeneration strategy in RVEA, the only minor modification needed is to insert Algorithm 4 into the end of the main loop (between Line 10 and Line 11) in Algorithm 1, and in each generation, $V_t \cup V_t^*$ (instead V_t) can be used to guide the elitism selection performed in Algorithm 2.

To assess the performance of the proposed reference vector regeneration strategy, empirical experiments are conducted on three typical test problems having irregular PFs, i.e., DTLZ5, DTLZ6 and DTLZ7. Specifically, DTLZ5 and DTLZ6 have a degenerate PF, which is always a 2-dimensional curve in the hyperspace, regardless of the number of objectives; DTLZ7 has a disconnected PF, where the number of segments can be as large as 2^{M-1}, and M is the number of objectives.

TABLE VII
THE STATISTICAL RESULTS (MEAN AND STANDARD DEVIATION) OF THE HV VALUES OBTAINED BY RVEA AND RVEA* ON DTLZ5, DTLZ6 AND DTLZ7. THE BEST RESULTS ARE HIGHLIGHTED.

Problem	M	RVEA*	RVEA
DTLZ5	3	**0.730371 (0.000086)**	0.725655 (0.000524)$^+$
	6	**0.655164 (0.006010)**	0.607517 (0.119848)$^+$
	8	**0.640410 (0.009400)**	0.581582 (0.122095)$^+$
	10	**0.643795 (0.002908)**	0.609602 (0.002982)$^+$
DTLZ6	3	**0.729752 (0.000153)**	0.725599 (0.000785)$^+$
	6	**0.639908 (0.045042)**	0.537950 (0.101526)$^+$
	8	**0.640109 (0.007304)**	0.553392 (0.072805)$^+$
	10	**0.633461 (0.006110)**	0.540260 (0.069931)$^+$
DTLZ7	3	**0.464756 (0.004527)**	0.461050 (0.003502)$^+$
	6	**0.451512 (0.008209)**	0.292165 (0.008900)$^+$
	8	**0.284548 (0.037903)**	0.187609 (0.028100)$^+$
	10	**0.242016 (0.061984)**	0.113275 (0.016875)$^+$

$+$: RVEA* shows significantly better performance in the comparison.

In the experiments, the RVEA embedded with the reference vector regeneration strategy (denoted as RVEA* hereafter) is compared with the original RVEA. To measure the quality of the solutions obtained by RVEA* and RVEA, the HV indicator is used, and the reference points used in the HV calculation are set as $(z_1^{max}+1, z_2^{max}+1, ..., z_M^{max}+1)$, where z_1^{max} to z_M^{max} are the maximum objective values calculated from the true PF[7] of each test problem. All the HV values are normalized to $[0, 1]$. For fair comparisons, an additional reference vector set is also added to RVEA, where all the reference vectors are randomly initialized, though they do not undergo the reference vector regeneration strategy in each generation.

As shown in Table VII and Fig. 9, RVEA*, in which the reference vector regeneration strategy is embedded, significantly outperforms the original RVEA on all test instances.

VII. HANDLING CONSTRAINTS

The proposed RVEA has shown competitive performance on a variety of unconstrained optimization problems. However, constrained optimization problems are widely seen in solving practical problems. Therefore, in this section, we extend RVEA to handling constraints (denoted as C-RVEA) by making some minor modifications to the selection strategy in Algorithm 2. The performance of C-RVEA is assessed on three representative constrained problems in comparison with C-MOEA/DD, which is the constrained version of MOEA/DD [41].

To begin with, the constraint violation function suggested in [81] is adopted to evaluate the degree of violation of a candidate solution:

$$CV(\mathbf{x}) = \sum_{j=1}^{J} \langle g_j(\mathbf{x}) \rangle + \sum_{k=1}^{K} |h_k(\mathbf{x})|, \tag{13}$$

[6]Since the objective values have already been translated, the minimum value of each objective is always 0.

[7]Since the PFs of DTLZ5, DTLZ6 and DTLZ7 have an irregular geometrical structure, the true PFs are approximated by sampling 10000 Pareto optimal solutions in the decision space.

where $g_j \geq 0$ and $h_j = 0$ are constraints, the operator $\langle\alpha\rangle$ returns a negative value of α if $\alpha < 0$, and returns 0 otherwise.

Algorithm 5 The elitism selection strategy for handling constraints.

for j = 1 to N **do**
  $S = \emptyset$; // index set of infeasible solutions
  for i = 1 to $|\overline{P}_{t,j}|$ **do**
    if $CV(\mathbf{x}_{t,i}) > 0$ **then**
      $S = S \cup \{i\}$
    end if
  end for
  if $|S| == |\overline{P}_{t,j}|$ **then**
    $k = \underset{i\in\{1,...,|\overline{P}_{t,j}|\}}{\operatorname{argmin}} CV(\mathbf{x}_{t,i})$
  else
    $k = \underset{i\in\{1,...,|\overline{P}_{t,j}|\},i\notin S}{\operatorname{argmin}} d_{t,i,j}$
  end if
  $P_{t+1} = P_{t+1} \cup \{I_{t,k}\}$;
end for

As summarized in Algorithm 5, the basic idea of the elitism selection strategy for handling constraints is to take the constraint violations into consideration in the selection process. Firstly, we pick out the infeasible solutions and record the indices in S. Afterwards, we check the size of S. If the size of S equals that of the current subpopulation $\overline{P}_{t,j}$, which means that every solution is infeasible, we will select the one having the minimum degree of constraint violation to be passed to the next generation; otherwise, we select the one with the minimal APD value among the feasible solutions.

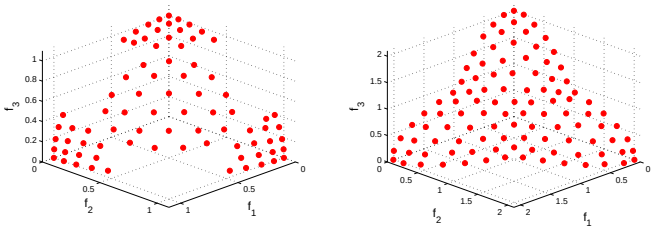
In order to assess the performance of the proposed elitism selection strategy for handling constraints, empirical studies have been conducted on three representative constrained problems, namely, C1-DTLZ1, C2-DTLZ2 and C3-DTLZ3, where the detailed definitions of the constraints can be found in Supplementary Materials V.

TABLE VIII
THE STATISTICAL RESULTS (MEAN AND STANDARD DEVIATION) OF THE HV VALUES OBTAINED BY C-RVEA AND C-MOEA/DD ON C1-DTLZ1, C2-DTLZ2 AND C3-DTLZ3. THE BEST RESULTS ARE HIGHLIGHTED.

Problem	M	C-RVEA	C-MOEA/DD
C1-DTLZ1	3	0.990636 (0.001579)	**0.991204 (0.001073)**$^{\approx}$
	6	**0.998719 (0.001306)**	0.996792 (0.001743)$^{+}$
	8	**0.998628 (0.000931)**	0.997427 (0.002109)$^{+}$
	10	**0.998613 (0.002182)**	0.997635 (0.001826)$^{+}$
C2-DTLZ2	3	0.918574 (0.000035)	**0.918809 (0.000023)**$^{-}$
	6	0.993848 (0.000417)	**0.993871 (0.000282)**$^{\approx}$
	8	0.998513 (0.000106)	**0.998585 (0.000124)**$^{\approx}$
	10	**0.999746 (0.000067)**	0.999738 (0.000050)$^{\approx}$
C3-DTLZ4	3	0.918766 (0.000201)	**0.919660 (0.000274)**$^{-}$
	6	0.996745 (0.000288)	**0.997242 (0.000225)**$^{-}$
	8	0.999456 (0.000081)	**0.999685 (0.000041)**$^{-}$
	10	**0.999971 (0.000029)**	0.999969 (0.000013)$^{\approx}$

+: RVEA shows significantly better performance in the comparison.
−: RVEA shows significantly worse performance in the comparison.
≈: There is no significant difference between the compared results.

Both C-RVEA and C-MOEA/DD are run 20 times independently on each test instance. In each singe run, 1000 generations are run for C1-DTLZ1, and 500 generations for C2-DTLZ2, C3-DTLZ4. We again use the HV as the performance indicator, and the reference points for C1-DTLZ1, C2-DTLZ2 and C3-DTLZ4 are $(1.5, 1.5, ..., 1.5)$, $(2, 2, ..., 2)$ and $(3, 3, ..., 3)$, respectively.



(a) C2-DTLZ2 (b) C3-DTLZ4

Fig. 10. The approximate Pareto optimal solutions obtained by C-RVEA on the 3-objective C2-DTLZ2 and C3-DTLZ4 in the run associated with the median HV values.

Generally speaking, C-RVEA and C-MOEA/DD show comparable performance, as shown by the results in Table VIII. Specifically, C-RVEA outperforms C-MOEA/DD on C1-DTLZ1, but is outperformed on C3-DTLZ4, and the two algorithms show comparable performance on C2-DTLZ2. In addition to the statistical results, the approximate Pareto optimal solutions obtained by C-RVEA on the 3-objective C2-DTLZ2 and C3-DTLZ4 are plotted in Fig. 10.

VIII. CONCLUSION

In this paper, we have proposed a reference vector based MOEA, termed RVEA, for solving MaOPs. In the proposed RVEA, the search process is guided by a set of predefined reference vectors inspired from the direction vectors proposed in MOEA/D-M2M, a decomposition based approach. The basic idea is to partition the objective space into a number small subspaces using a set of reference vectors. Inside each subspace, an elitism selection strategy is employed. As the selection criterion, a scalarization approach, known as the angle-penalized distance (APD), is proposed to measure the distance of the solutions to the ideal point and the closeness of the solutions to the reference vectors, which can be seen as a diversity measure or a degree of satisfaction to the preferences. Thus, the proposed RVEA can be largely categorized into the decomposition based approaches, when the reference vectors are uniformly generated to cover the whole PF. On the other hand, the proposed RVEA is also able to find a set of preferred Pareto optimal set, which can be used for elucidation of user preferences. We demonstrate that reference vectors are well suited for precise preference articulation by defining a central vector and a radius. The effectiveness of the reference vector based preference articulation is demonstrated by some illustrative empirical examples.

To obtain a set of uniformly distributed solutions in the objective space, a strategy for adapting the reference vectors has been suggested to tune the reference vectors according to the distribution of the candidate solutions. This reference vector adaptation strategy has shown high efficiency in dealing

with problems where the objective functions are not well normalized.

To assess the performance of RVEA, empirical comparisons have been conducted by comparing RVEA with five state-of-the-art MOEAs, namely, MOEA/DD, NSGA-III, MOEA/D-PBI, GrEA and KnEA on widely used MaOP test functions. Our experimental results indicate that RVEA shows robust performance on various benchmark problems with the objective number varying from 3 to 10, including normalized DTLZ problems, scaled DTLZ problems, and WFG problems. We have also conducted some experiments to compare the performance of APD with two popular scalarization approaches widely used in the decomposition based MOEAs, i.e., the Weighted Tchebycheff approach and the PBI approach.

Furthermore, a reference vector regeneration strategy has been proposed to improve the performance of RVEA on problems with irregular Pareto fronts. The strategy can be inserted into the framework of RVEA and shows high effectiveness. Finally, RVEA has been extended for solving constrained MaOPs.

The proposed RVEA has shown competitive performance on the studied MaOP test problems. One interesting feature of the proposed RVEA is that it is able to approximate the Pareto optimal solutions specified by the predefined reference vectors when the PF distributes in the whole objective space. However, the question remains open as to how the reference vectors can be adapted to the distribution of the candidate solutions according to the estimated geometrical features of the PF. It is still unclear which type of reference vectors is most practical in many-objective optimization. In addition, the use of reference vectors for interactive preference articulation is to be investigated [85].

ACKNOWLEDGEMENTS

This work was supported in part by the Honda Research Institute Europe, and in part by the Joint Research Fund for Overseas Chinese, Hong Kong and Macao Scholars of the National Natural Science Foundation of China (Grant No. 61428302).

REFERENCES

[1] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P. N. Suganthan, and Q. Zhang, "Multiobjective evolutionary algorithms: A survey of the state of the art," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 32–49, 2011.

[2] K. Ikeda, H. Kita, and S. Kobayashi, "Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?" in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2001, pp. 957–962.

[3] D. Brockhoff, T. Friedrich, N. Hebbinghaus, C. Klein, F. Neumann, and E. Zitzler, "On the effects of adding objectives to plateau functions," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 591–603, 2009.

[4] O. Schütze, A. Lara, and C. A. C. Coello, "On the influence of the number of objectives on the hardness of a multiobjective optimization problem," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 444–455, 2011.

[5] L. S. Batista, F. Campelo, F. G. Guimaraes, and J. A. Ramírez, "A comparison of dominance criteria in many-objective optimization problems," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2011, pp. 2359–2366.

[6] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *Proceedings of the International Conference on Evolutionary multi-criterion optimization*. Springer, 2005, pp. 14–32.

[7] D. W. Corne and J. D. Knowles, "Techniques for highly multiobjective optimisation: some nondominated points are better than others," in *Proceedings of the International Conference on Genetic and Evolutionary Computation*, 2007, pp. 773–780.

[8] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

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

[10] Y. Jin and B. Sendhoff, "Connectedness, regularity and the success of local search in evolutionary multi-objective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 3. IEEE, 2003, pp. 1910–1917.

[11] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[12] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 770–784, 2007.

[13] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 2, pp. 183–195, 2011.

[14] M. Li, S. Yang, and X. Liu, "Diversity comparison of pareto front approximations in many-objective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2568–2584, 2014.

[15] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2008, pp. 2419–2426.

[16] Z. He and G. G. Yen, "Ranking many-objective evolutionary algorithms using performance metrics ensemble," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2013, pp. 2480–2487.

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

[18] D. Hadka and P. Reed, "Borg: An auto-adaptive many-objective evolutionary computing framework," *Evolutionary Computation*, vol. 21, no. 2, pp. 231–259, 2013.

[19] X. Zou, Y. Chen, M. Liu, and L. Kang, "A new evolutionary algorithm for solving many-objective optimization problems," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 38, no. 5, pp. 1402–1412, 2008.

[20] F. di Pierro, S.-T. Khu, and D. A. Savic, "An investigation on preference order ranking scheme for multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 17–45, 2007.

[21] G. Wang and H. Jiang, "Fuzzy-dominance and its application in evolutionary many objective optimization," in *2007 International Conference on Computational Intelligence and Security Workshops*, 2007, pp. 195–198.

[22] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721–736, 2013.

[23] M. Köppen and K. Yoshida, "Substitute distance assignments in NSGA-II for handling many-objective optimization problems," in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, 2007, pp. 727–741.

[24] A. López, C. A. Coello Coello, A. Oyama, and K. Fujii, "An alternative preference relation to deal with many-objective optimization problems," in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, 2013, pp. 291–306.

[25] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348–365, 2014.

[26] J. Cheng, G. Yen, and G. Zhang, "A many-objective evolutionary algorithm with enhanced mating and environmental selections," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 592–605, 2015.

[27] X. Zhang, Y. Tian, and Y. Jin, "A knee point driven evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 761–776, 2015.

[28] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.

[29] H.-L. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 2450–455, 2014.

[30] Y. Yuan, H. Xu, B. Wang, B. Zhang, and X. Yao, "Balancing convergence and diversity in decomposition-based many-objective optimizers," *IEEE Transactions on Evolutionary Computation*, accepted in 2015.

[31] Y. Jin, M. Olhofer, and B. Sendhoff, "Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?" in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2001, pp. 1042–1049.

[32] T. Murata, H. Ishibuchi, and M. Gen, "Specification of genetic search directions in cellular multiobjective genetic algorithms," in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, 2001, pp. 82–95.

[33] Z. Wang, Q. Zhang, M. Gong, and A. Zhou, "A replacement strategy for balancing convergence and diversity in MOEA/D," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2014, pp. 2132–2139.

[34] K. Li, Q. Zhang, S. Kwong, M. Li, and R. Wang, "Stable matching-based selection in evolutionary multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 6, pp. 909–923, 2014.

[35] K. Li, S. Kwong, Q. Zhang, and K. Deb, "Interrelationship-based selection for decomposition multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 10, pp. 2076–2088, 2014.

[36] M. Asafuddoula, T. Ray, and R. Sarker, "A decomposition based evolutionary algorithm for many objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 445–460, 2015.

[37] S. B. Gee, K. C. Tan, V. A. Shim, and N. Pal, "Online diversity assessment in evolutionary multiobjective optimization: A geometrical perspective," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 542–559, 2015.

[38] L. Chen, H.-L. Liu, C. Lu, Y.-m. Cheung, and J. Zhang, "A novel evolutionary multi-objective algorithm based on s metric selection and m2m population decomposition," in *Proceedings of the Asia Pacific Symposium on Intelligent and Evolutionary Systems*. Springer, 2015, pp. 441–452.

[39] 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 Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[40] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multiobjective evoltuionary algorithm using Gaussian process based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.

[41] K. Li, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 5, pp. 694–716, 2015.

[42] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *8th International Conference on Parallel Problem Solving from Nature*, 2004, pp. 832–842.

[43] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, 2007.

[44] K. Li, S. Kwong, J. Cao, M. Li, J. Zheng, and R. Shen, "Achieving balance between proximity and diversity in multi-objective evolutionary algorithm," *Information Sciences*, vol. 182, no. 1, pp. 220–242, 2012.

[45] J. Bader and E. Zitzler, "HypE: an algorithm for fast hypervolume-based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, 2011.

[46] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 29–38, 2006.

[47] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius, "An interactive evolutionary multi-objective optimization method based on progressively approximated value functions," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 5, pp. 723–739, 2010.

[48] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 474 – 494, 2013.

[49] D. K. Saxena and K. Deb, "Dimensionality reduction of objectives and constraints in multi-objective optimization problems: a system design perspective," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2008, pp. 3204–3211.

[50] H. K. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 4, pp. 539–556, 2011.

[51] S. Bandyopadhyay and A. Mukherjee, "An algorithm for many-objective optimization with reduced objective computations: A study in differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 3, pp. 400–413, 2015.

[52] H. Wang, L. Jiao, and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 4, pp. 524–541, Aug 2015.

[53] R. C. Purshouse, C. Jalbă, and P. J. Fleming, "Preference-driven co-evolutionary algorithms show promise for many-objective optimisation," in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2011, pp. 136–150.

[54] E. J. Hughes, "MSOPS-II: A general-purpose many-objective optimiser," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2007, pp. 3944–3951.

[55] K. Miettinen, *Nonlinear multiobjective optimization*. Springer Science & Business Media, 1999, vol. 12.

[56] R. Cheng, Y. Jin, and K. Narukawa, "Adaptive reference vector generation for inverse model based evolutionary multiobjective optimization with degenerate and disconnected pareto fronts," in *Proceedings of the Evolutionary Multi-Criterion Optimization*. Springer, 2015, pp. 127–140.

[57] R. Cheng, M. Olhofer, and Y. Jin, "Reference vector based a posteriori preference articulation for evolutionary multiobjective optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2015.

[58] K. Deb, J. Sundar, U. Bhaskara, and S. Chaudhuri, "Reference point based multi-objective optimization using evolutionary algorithms," *International Journal of Computational Intelligence Research*, vol. 2, no. 3, pp. 635–642, 2006.

[59] J. Molina, L. V. Santana, A. G. Hernández-Díaz, C. A. C. Coello, and R. Caballero, "g-dominance: reference point based dominance for multiobjective metaheuristics," *European Journal of Operational Research*, vol. 197, no. 2, pp. 685–692, 2009.

[60] L. Thiele, K. Miettinen, P. J. Korhonen, and J. Molina, "A preference-based evolutionary algorithm for multi-objective optimization," *Evolutionary Computation*, vol. 17, no. 3, pp. 411–436, 2009.

[61] M. Gong, F. Liu, W. Zhang, L. Jiao, and Q. Zhang, "Interactive MOEA/D for multi-objective decision making," in *Proceedings of the Genetic and Evolutionary Computation Conference*. ACM, 2011, pp. 721–728.

[62] X. Ma, F. Liu, Y. Qi, L. Li, L. Jiao, X. Deng, X. Wang, B. Dong, Z. Hou, Y. Zhang *et al.*, "MOEA/D with biased weight adjustment inspired by user preference and its application on multi-objective reservoir flood control problem," *Soft Computing*, pp. 1–25, 2015.

[63] A. Jaszkiewicz and R. Słowiński, "The light beam search approach–an overview of methodology applications," *European Journal of Operational Research*, vol. 113, no. 2, pp. 300–314, 1999.

[64] A. Mohammadi, M. N. Omidvar, X. Li, and K. Deb, "Integrating user preferences and decomposition methods for many-objective optimization," in *IEEE Congress on Evolutionary Computation*. IEEE, 2014, pp. 421–428.

[65] J. A. Cornell, *Experiments with mixtures: designs, models, and the analysis of mixture data*. John Wiley & Sons, 2011.

[66] K. Deb and R. B. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, no. 4, pp. 115–148, 1995.

[67] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, no. 4, pp. 30–45, 1996.

[68] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[69] L. Wang and Q. Zhang, "Constrained subproblems in decomposition based multiobjective evolutionary algorithm," *IEEE Transaction on Evolutionary Computation*, 2015 (accepted).

[70] A. Mohammadi, M. N. Omidvar, X. Li, and K. Deb, "Sensitivity analysis of penalty-based boundary intersection on aggregation-based EMO algorithms," in *Proceedings of the IEEE Congress on Evolutionary Computation*. IEEE, 2015, pp. 2891–2898.

[71] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE*

Transactions on Evolutionary Computation, vol. 10, no. 5, pp. 477–506, 2006.
[72] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Towards understanding the cost of adaptation in decomposition-based optimization algorithms." in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2013, pp. 615–620.
[73] Y. Qi, X. Ma, F. Liu, L. Jiao, J. Sun, and J. Wu, "MOEA/D with adaptive weight adjustment," *Evolutionary computation*, vol. 22, no. 2, pp. 231–264, 2014.
[74] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," in *2001 TIK-Technical Report*, no. 112, 2001.
[75] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multiobjective test problem toolkit," in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, 2005, pp. 280–294.
[76] L. While, L. Bradstreet, and L. Barone, "A fast way of calculating exact hypervolumes," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 86–95, 2012.
[77] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York: Wiley, 2001.
[78] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An efficient approach to non-dominated sorting for evolutionary multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, 2014, DOI. 10.1109/TEVC.2014.2308305.
[79] K. Li, K. Deb, Q. Zhang, and S. Kwong, "Efficient non-domination level update approach for steady-state evolutionary multiobjective optimization," *Department of Electtrical and Computer Engineering, Michigan State University, East Lansing, USA, Tech. Rep. COIN Report*, no. 2014014, 2014.
[80] J. J. Durillo and A. J. Nebro, "jmetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, 2011.
[81] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2013.
[82] ——, "An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization," in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 2013, pp. 307–321.
[83] Y. Jin, T. Okabe, and B. Sendhoff, "Adapting weighted aggregation for multiobjective evolution strategies," in *Proceedings of the International Conference on Evolutionary multi-criterion optimization*. Springer, 2001, pp. 96–110.
[84] I. Giagkiozis, R. C. Purshouse, and P. J. Fleming, "Generalized decomposition and cross entropy methods for many-objective optimization," *Information Sciences*, vol. 282, pp. 363–387, 2014.
[85] P. Eskelinen, K. Miettinen, K. Klamroth, and J. Hakanen, "Pareto navigator for interactive nonlinear multiobjective optimization," *OR Spectrum*, vol. 32, pp. 211–227, 2010.

Ran Cheng received the B. Eng. degree from Northeastern University, Shenyang, China, in 2010. He is currently working towards the Ph.D. degree at the Department of Computer Science, University of Surrey, United Kingdom.

His current research interests include multi-/many-objective optimization, estimation of distribution algorithms, and swarm intelligence.

Yaochu Jin (M'98-SM'02-F'16) received the B.Sc., M.Sc., and Ph.D. degrees from Zhejiang University, Hangzhou, China, in 1988, 1991, and 1996 respectively, and the Dr.-Ing. degree from Ruhr University Bochum, Germany, in 2001.

He is a Professor in Computational Intelligence, Department of Computer Science, University of Surrey, Guildford, U.K., where he heads the Nature Inspired Computing and Engineering Group. He is also a Finland Distinguished Professor funded by the Finnish Agency for Innovation (Tekes) and a Changjiang Distinguished Visiting Professor appointed by the Ministry of Education, China. His science-driven research interests lie in the interdisciplinary areas that bridge the gap between computational intelligence, computational neuroscience, and computational systems biology. He is also particularly interested in nature-inspired, real-world driven problem-solving. He has (co)authored over 200 peer-reviewed journal and conference papers and been granted eight patents on evolutionary optimization. His current research is funded by EC FP7, UK EPSRC and industry. He has delivered 20 invited keynote speeches at international conferences.

He is Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENTAL SYSTEMS and Complex & Intelligent Systems. He is also an Associate Editor or Editorial Board Member of the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NANOBIOSCIENCE, Evolutionary Computation, BioSystems, Soft Computing, and Natural Computing. Dr Jin was an IEEE Distinguished Lecturer (2013-2015) and Vice President for Technical Activities of the IEEE Computational Intelligence Society (2014-2015). He was the recipient of the Best Paper Award of the 2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology and the 2014 IEEE Computational Intelligence Magazine Outstanding Paper Award. He is a Fellow of IEEE.

Markus Olhofer obtained his Dipl.-Ing. in Electrical Engineering in 1997 and his PhD degree in 2000, both from Ruhr-University Bochum, Germany. He joined the Future Technology Research Division at Honda R&D Europe (Deutschland) in 1998 and has been Chief Scientist and head of the Complex Systems Optimisation and Analysis Group at the Honda Research Institute Europe since 2010. His research interests include the extension of soft computing methods to meet requirements in complex engineering problems, ranging from evolutionary design optimisation to engineering data mining.

Dr Olhofer is a Visiting Professor with the Department of Computer Science, University of Surrey, UK.

Bernhard Sendhoff (M'99-SM'05) obtained a PhD in Applied Physics in May 1998, from the Ruhr-Universit'at Bochum, Germany. From 1999 to 2002 he worked for Honda R&D Europe GmbH, and since 2003, he has been with the Honda Research Institute Europe GmbH. Since 2007 he is Honorary Professor of the School of Computer Science of the University of Birmingham, UK. Since 2008, he is Honorary Professor at the Technical University of Darmstadt, Germany. Since 2011 he is President of the Honda Research Institute Europe GmbH.

Bernhard Sendhoff is a Senior Member of the IEEE and of the ACM. His research focuses on methods from computational intelligence and their applications in development, production and services. He has authored and co-authored over 150 scientific papers and over 30 patents.