This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

A Vector Angle based Evolutionary Algorithm for Unconstrained Many-Objective Optimization

Yi Xiang, Yuren Zhou, Miqing Li and Zefeng Chen

***Abstract*—Taking both convergence and diversity into consideration, this paper suggests a vector angle based evolutionary algorithm for unconstrained (with box constraints only) many-objective optimization problems. In the proposed algorithm, the *maximum-vector-angle-first* principle is used in the environmental selection to guarantee the wideness and uniformity of the solution set. With the help of the *worse-elimination* principle, worse solutions in terms of the convergence (measured by the sum of normalized objectives) are allowed to be conditionally replaced by other individuals. Therefore, the selection pressure toward the Pareto-optimal front is strengthened. The proposed method is compared with other four state-of-the-art many-objective evolutionary algorithms on a number of unconstrained test problems with up to fifteen objectives. The experimental results have shown the competitiveness and effectiveness of our proposed algorithm in keeping a good balance between convergence and diversity. Furthermore, it was shown by the results on two problems from practice (with irregular Pareto fronts) that our method significantly outperforms its competitors in terms of both the convergence and diversity of the obtained solution sets. Notably, the new algorithm has the following good properties: (1) it is free from a set of supplied reference points or weight vectors; (2) it has less algorithmic parameters and (3) the time complexity of the algorithm is low. Given both good performance and nice properties, the suggested algorithm could be an alternative tool when handling optimization problems with more than three objectives.**

***Index Terms*—Many-objective optimization, decomposition, evolutionary algorithm, convergence and diversity.**

I. Introduction

Multi-Objective Problems (MOPs) with at least four conflicting objectives are informally known as Many-Objective Optimization Problems (MaOPs) [1], [2]. MaOPs exist widely in real-world applications [1], such as engineering design [3], water distribution system design [4], airfoil designing problem [5] and automotive engine calibration problem [6]. Recently, MaOPs have caught wide attention from the Evolutionary Multi-objective Optimization (EMO) community. A number of Many-Objective Evolutionary Algorithms (MaOEAs) have been proposed to deal with optimization problems having more than three objectives [7]–[14].

Generally, an MOP with only box constraints can be stated as follows.

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

where m is the number of objectives. For MaOPs, $m \geq 4$. In (1), $\Omega = \prod_{i=1}^{n}[l_i, u_i] \subseteq \mathbb{R}^n$ is called the decision space. Here n is the number of decision variables; u_i and l_i are the upper and lower bounds of the ith decision variable, respectively.

Unlike in single-objective optimization problems, there is no single solution available for an MOP, but a set of non-dominated solutions known as a Pareto-optimal Set (*PS*). For $\forall$ $\mathbf{x} \in PS$, its objective vector $\mathbf{F}(\mathbf{x})$ constitutes the Pareto Front (*PF*) [15]. Since Multi-Objective Evolutionary Algorithms (MOEAs) are typically capable of finding multiple trade-off solutions in a single simulation run, they have been widely and successfully used to solve optimization problems with mostly two or three objectives [15]–[17].

It has been shown that Pareto-based MOEAs, such as Non-dominated Sorting Genetic Algorithm II (NSGA-II) [16], the improved Strength Pareto Evolutionary Algorithm (SPEA2) [18], the Pareto Archived Evolution Strategy (PAES) [19], and Pareto envelope-based selection algorithm II (PESAII) [20] encountered difficulties when handling MaOPs, mainly due to the obstacles caused by the Dominance Resistance (DR) and the Active Diversity Promotion (ADP) phenomena [21], [22]. The DR phenomenon refers to the incomparability of solutions in terms of the Pareto dominance relation, and the main reason is that the proportion of non-dominated solutions in a population tends to rise rapidly as the number of objectives increases [1]. Actually, the probability of any two solutions being comparable is $\eta = 1/2^{m-1}$ in an m-dimensional objective space [10]. For two or three objectives, this probability is 1/2 and 1/4, respectively. However, η is already as low as 0.002 when m takes the value of 10. As a result of DR phenomenon, the dominance-based primary selection criterion fails to distinguish between solutions, and then the diversity-based secondary selection criterion is activated to determine the survival of solutions in the environmental selection. This is the so-called ADP problem which may be harmful to the convergence of the approximated Pareto fronts. According to the experiment carried out in [10], the NSGA-II without the diversity maintenance mechanism obtains even better convergence metric (CM) [23] results than the original NSGA-

Manuscript received November 02, 2015; revised February 19, 2016, and May xx, 2016; accepted July xx, 2016. Date of publication June x, 2016; date of current version March xx, 2016. This paper is supported by the Natural Science Foundation of China (Grant nos. 61472143 and 61170081), and the Scientific Research Special Plan of Guangzhou Science and Technology Programme (Grant no. 201607010045).

Yi Xiang, Yuren Zhou and Zefeng Chen are with the School of Data and Computer Science & Collaborative Innovation Center of High Performance Computing, Sun Yat-sen University, Guangzhou 510006, P. R. China. (E-mail: gzhuxiang_yi@163.com (Y. Xiang), yrzhou@scut.edu.cn (Y. Zhou)).

Miqing Li is with the Centre of Excellence for Research in Computational Intelligence and Applications, School of Computer Science, University of Birmingham, Birmingham B15 2TT, U.K.

* Corresponding Author: Y. Zhou

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

Digital Object Identifier xxxxxxxxxxxx

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

II on the 5-objective DTLZ2 problem. Furthermore, for the 10-objective problem, the diversity maintenance mechanism even makes the population gradually move away from the true Pareto front.

Many efforts have been made to improve the scalability of Pareto-based MOEAs to make them suitable for MaOPs. To address the DR phenomenon, a number of modified or relaxed Pareto dominance relations have been proposed to enhance the selection pressure toward the true Pareto front. The ϵ-dominance [24], CDAS-dominance [25], α-dominance [26], fuzzy Pareto dominance [27], L-dominance [28], θ-dominance [29], and so on [30]–[32] have been experimentally verified to be more effective than the original Pareto dominance relation when searching toward the optimum.

Regarding the ADP phenomenon, several customized diversity-based approaches have been developed to weaken or avoid the adverse impact of diversity maintenance. Wagner *et al.* [33] showed that assigning the crowding distance of boundary solutions a zero value, instead of infinity, could significantly improve the convergence of NSGA-II on MaOPs. Adra *et al.* [34] introduced a diversity management method named DM1 that adaptively determines whether or not to activate diversity promotion according to the distribution of the population. It has been shown by the experimental results on a set of test problems with up to 20 objectives, that NSGA-II with DM1 operator performed consistently better than NSGA-II in terms of both the convergence and diversity [1]. Focused on the secondary selection criterion for Pareto-based algorithms, Li *et al.* [22] introduced a shift-based density estimation (SDE) strategy and applied it to three popular Pareto-based algorithms, i.e., NSGA-II, SPEA2 and PESAII, forming three new algorithms (i.e., NSGA-II+SDE, SPEA2+SDE and PESAII+SDE). In SDE, poorly converged individuals are pulled into crowded regions and assigned a high density value, thus they are easily eliminated during the evolutionary process. It has been demonstrated by experimental results on DTLZ and TSP test problems with up to 10 objectives that SPEA2+SDE was very effective in balancing convergence and diversity. Recently, Deb *et al.* [8] adopted a new diversity maintenance mechanism in the framework of NSGA-II, and proposed a reference-point-based evolutionary algorithm (NSGA-III) for many-objective optimization. In each generation of NSGA-III, the population is divided into different layers by using the non-dominated sorting procedure. Then, objective vectors are adaptively normalized. Next, each population member is associated with a particular reference line based on a proximity measure. Finally, a niche preservation operation is applied to ensure a set of diversified solutions. In NSGA-III, the reference points can be either generated systematically or supplied by the users.

In MaOEAs, balancing convergence and diversity is of high importance, and is also the main challenge in the field. Yang *et al.* [7] proposed a grid based evolutionary algorithm (GrEA) for MaOPs. In GrEA, the grid dominance and grid difference defined in the grid environment are used to increase the selection pressure toward the true Pareto-optimal front. Besides, three grid-based criteria, i.e., grid ranking, grid crowding distance and grid coordinate point distance, are introduced to guarantee the good distribution among solutions. It was revealed by the experimental results on 52 test instances that GrEA was effective and competitive in balancing convergence and diversity. Praditwong and Yao [35], Wang *et al*. [36] proposed the two-archive algorithm (Two_Arch) and its improved version Two_Arch2. In these algorithms, the non-dominated solution set is divided into two archives, namely Convergence Archive (CA) and Diversity Archive (DA). In Two_Arch2, the two archives are assigned different selection principles, i.e., indicator-based and Pareto-based for CA and DA, respectively. Additionally, a new L_p-norm-based ($p < 1$) diversity maintenance scheme is designed in Two_Arch2 for MaOPs. The experimental results showed that Two_Arch2 performed well on a set of MaOPs with satisfactory convergence, diversity, and complexity. Li *et al*. [11] developed a steady-state algorithm named MOEA/DD for MaOPs by combining dominance- and decomposition-based approaches to achieve a balance between convergence and diversity. In MOEA/DD, a two-layer weight vector generation method is designed to produce a set of well-distributed weight vectors. Cheng *et al*. [37] proposed a many-objective evolutionary algorithm (named MaOEA-DDFC) which is based on directional diversity (DD) and favorable convergence (FC). In this algorithm, a mating selection based on FC is used to enhance the selection pressure, and the environmental selection based on DD and FC is applied to balance diversity and convergence. Experimental results showed that the proposed algorithm performed competitively to seven state-of-the-art algorithms on a set of test instances.

Decomposition-based algorithms decompose an MOP into a set of subproblems, and optimize them in a collaborative way. The multi-objective evolutionary algorithm based on decomposition (MOEA/D) [38] is a representative of this kind of algorithms. In these algorithms, a predefined set of weight vectors are used to specify multiple search directions towards different parts of the PF. Since the weight vectors of subproblems are widely distributed, the obtained solutions are expected to have a wide spread over the PF. Decomposition-based methods seem to be exempt from the insufficient selection pressure problem, because they do not rely on Pareto-dominance relation to distinguish between individuals. MOEA/D and MOEA/DD were demonstrated to be highly competitive when handling MaOPs [8], [11].

However, decomposition-based algorithms face their own challenges. First, decomposition-based methods need to specify a set of weight vectors which significantly affect the algorithms' performance in terms of diversity [1]. But the configuration of weight vectors is not always an easy task. Although the two-layer weight vector generation method is adopted in MOEA/DD to produce weight vectors, it still suffers from the curse of dimensionality. For instance, in an $m = 50$ objective space, this method will generate $\binom{50+2-1}{2}$ + $\binom{50+1-1}{1}$ = 1325 weight vectors even for $h_1 = 2$ and $h_2 = 1$. Here, h_1 and h_2 are the number of divisions in the boundary layer and inside layer, respectively. Second, how to maintain the uniformity of intersection points of the specified search directions and the problems' PF is a key

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

issue in decomposition-based EMO algorithms. When facing problems with a highly irregular PF (e.g., a discontinuous or degenerate front), the uniformly-distributed weight vectors cannot guarantee the uniformity of the intersection points [39].

Given the above, we here consider a new optimizer based on the search directions of the evolutionary population itself. In the evolutionary process, the current population is an approximation of the true PF, and this approximation becomes more and more accurate as the evolution proceeds [40]. Therefore, the search can be guided by the current population. In this way, the new method does not require reference points or weight vectors. Specifically, this paper proposes a Vector angle based Evolutionary Algorithm (VaEA) to solve MaOPs. VaEA takes both convergence and diversity into consideration, and is expected to achieve a good balance between them. In each generation of VaEA, the objective vectors are normalized. Then, the critical layer F_l is identified by using the non-dominated sorting method. In the environmental selection, the *maximum-vector-angle-first* principle is applied to selecting solutions from F_l one by one so as to ensure the diversity of the population set. By using the *worse-elimination* principle, VaEA allows worse solutions in terms of the convergence (measured by the sum of each normalized objective) to be conditionally replaced by other individuals so as to keep a balance between convergence and diversity.

It should be noted here that approximating the PF in multiple search directions is one of the major features in decomposition-based algorithms. From this point of view, VaEA shares the same idea as in decomposition-based methods. This is because individuals in the current population define multiple search directions towards the PF. However, there exists a significant difference between VaEA and decomposition-based methods: In decomposition-based algorithms, an MOP is usually converted into a number of scalar optimization subproblems [38] or a set of simple MOPs [41], but this is not the case in VaEA. The main properties of VaEA can be summarized as follows.

- No weight vectors or reference points need to be specified beforehand. Actually, the evolutionary process in VaEA is dynamically guided by the current population because (1) the population in different generations is usually distinct from each other and (2) the search directions in each generation are dynamically selected according to the *maximum-vector-angle-first* principle. From this point of view, VaEA shares a similar idea as in some decomposition algorithms [42]–[44], where the weight vectors are adaptively adjusted according to the distribution of the current population. Thus, in these algorithms the search directions are also dynamically changed.
- Parameter-Less property of VaEA. Apart from some common GA parameters, such as the population size, termination condition and parameters in genetic operators, VaEA does not need to configure any new algorithmic parameter. In MOEA/DD, three more parameters are involved, i.e., the penalty parameter in PBI, the neighborhood size and the probability used to select in the neighborhood [11]. These parameters have to be carefully configured in the practical application of the algorithm.
- Lower time complexity. As will be shown in Section II-C, the worst time complexity of VaEA is $max\{O(Nlog^{m-2}N), O(mN^2)\}$ which is equal to that of NSGA-III, MOEA/DD and Two_Arch2, and lower than MaOEA-DDFC, GrEA and SPEA2+SDE.

In [25], the concept of vector angle was also adopted in the design of MOEAs. However, our method is totally different from the method in [25]. First and foremost, the vector angle in [25] is used to constrict or expand the dominance area of solutions. The aim is to enhance the selection pressure toward the optimal fronts. However, in this paper, the vector angle is adopted for the purpose of diversity promotion. Secondly, the method in [25] considers the vector angle between a solution and each coordinate axis. In VaEA, we calculate angle between each pair of solutions as a measurement of closeness. Finally, the vector angle in [25] can be set to any value between 0 and π. While the angle in VaEA is restricted to the interval $[0,\pi/2]$. The recently proposed MOEA/D-M2M [41] also used the concept of vector angle. In MOEA/D-M2M, with the help of vector angles, an MOP is divided into a number of simple multiobjective subproblems which are solved in a collaborative way. The major difference between VaEA and MOEA/D-M2M is that the vector angle in the latter is used to divide the whole objective space into a number of subregions so as to specify different subpopulations for approximating a small part of the true Pareto front. However, as stated before, the angle in VaEA is adopted to measure the closeness between a pair of solutions. Furthermore, a set of direction vectors should be chosen in MOEA/D-M2M. But, in VaEA, we don't need to specify any direction vector.

In the remainder of this paper, we first outline our proposed VaEA in detail in Section II. Then, we present the simulation results on a set of test instances in Section III. Thereafter, VaEA is applied to two problems from practice, and is compared with two competitive algorithms NSGA-III and MOEA/DD. Finally, Section V concludes this paper and gives some remarks for future studies.

II. Proposed algorithm: VaEA

A. General Framework

The pseudo code of the proposed method is shown in Algorithm 1. The VaEA shares a common framework that is employed by many evolutionary algorithms. First, a population with N solutions is randomly initialized in the whole decision space Ω. Then more potential solutions are selected into the mating pool according to the fitness value of each individual. In what followed, a set of offspring solutions Q is generated by applying crossover and mutation operations. Finally, N solutions are selected from the union set of P and Q by adopting an environmental selection procedure. The above steps continue until the number of generations G reaches to its maximum value, i.e., G_{max}.

As we will describe in the following section, the VaEA performs a careful elitist preservation strategy and maintains diversity among solutions by putting more emphasis on solutions that have the maximum vector angle to individuals which have already survived. Therefore, we do not use any special

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

Algorithm 1 Framework of the proposed VaEA

1: $Initialization(P)$
2: $G = 1$
3: **While** $G \leq G_{max}$ **do**
4: $\quad P' = Mating\_selection(P)$
5: $\quad Q = Variation(P')$
6: $\quad S = P \cup Q$
7: $\quad P = Environmental\_selection(S)$
8: $\quad G{+}{+}$
9: **End While**
10: **Return** P

reproduction operations in VaEA. The offspring generation Q is constructed by applying the usual crossover and mutation operations to parents that are randomly picked from P. As in NSGA-III, the well-known SBX crossover [45] (with a relatively larger value of distribution index) and the polynomial mutation operators are used in the $Variation$ step (line 5) in Algorithm 1.

B. Environmental Selection

The framework of the environmental selection in VaEA is similar to that of NSGA-II [16] or NSGA-III [8], but the niche-preservation operation is quite different from other existing MaOEAs. Algorithm 2 shows the general framework of the environmental selection procedure. First, the set S is normalized by the function $Normalization$ (line 2 in Algorithm 2). Then the norm of each solution in S is calculated in the normalized objective space (line 3). The non-dominated sorting procedure is used to divide solutions into different layers, and then the last layer F_l is determined (lines 4-8). If the population is full, then return P. Otherwise, $K = N - |P|$ solutions from F_l are added into P one by one by using functions $Associate$ and $Niching$ (lines 13 and 14) which are designed on the basis of vector angles. In what follows, we will describe them in more details.

1) Normalization: The normalization procedure is shown in Algorithm 3. The ideal point $\mathbf{Z}^{min} = (z_1^{min}, z_2^{min}, \ldots, z_m^{min})^T$ is determined by finding the minimum value of each objective for all solutions in S, i.e., $z_i^{min} = \min_{j=1}^{2N} f_i(\mathbf{x}_j), \mathbf{x}_j \in S, i = 1, 2, \ldots, m$. Similarly, the nadir point $\mathbf{Z}^{max} = (z_1^{max}, z_2^{max}, \ldots, z_m^{max})^T$ can be worked out (line 2 in Algorithm 3). For each solution $\mathbf{x}_j \in S$, its objective vector $\mathbf{F}(\mathbf{x}_j)$ is normalized to $\mathbf{F}'(\mathbf{x}_j) = (f_1'(\mathbf{x}_j), f_2'(\mathbf{x}_j), \ldots, f_m'(\mathbf{x}_j))^T$ through the following equation.

$$f_i'(\mathbf{x}_j) = \frac{f_i(\mathbf{x}_j) - z_i^{min}}{z_i^{max} - z_i^{min}}, i = 1, 2, \ldots, m \tag{2}$$

During the normalization process, the fitness value of $\mathbf{x}_j$ (denoted by $fit(\mathbf{x}_j)$) is also calculated. This fitness assignment method considers the sum of each normalized objective value. Mathematically,

Algorithm 2 $Environmental\_selection(S)$

Input: S, the union of P and Q
Output: P, the new population
1: $P = \emptyset, i = 1$
2: *Normalization(S)*
3: *Compute_norm(S)*
4: $(F_1, F_2, \ldots) =$ *Non-dominated-sorting(S)*
5: **While** $|P| + |F_i| \leq N$
6: $\quad P = P \bigcup F_i$ and $i = i + 1$
7: **End While**
8: The last front to be included $F_l = F_i$
9: **If** $|P| = N$, **then**
10: $\quad$ return P
11: **Else**
12: $\quad$ Solutions to be chosen from F_l: $K = N - |P|$
13: $\quad$ Associate each member of F_l with a solution in P: $Associate(P, F_l)$
14: $\quad$ Choose K solutions one by one from F_l to construct final P: $P = Niching(P, F_l, K)$
15: **End If**
16: **Return** P

Algorithm 3 $Normalization(S)$

Input: S, the union of P and Q
1: Find $\mathbf{Z}^{min}$, where $z_i^{min} = \min_{j=1}^{2N} f_i(\mathbf{x}_j), i = 1, 2, \ldots, m$
2: Find $\mathbf{Z}^{max}$, where $z_i^{max} = \max_{j=1}^{2N} f_i(\mathbf{x}_j), i = 1, 2, \ldots, m$
3: **For** $j = 1$ to $2N$
4: $\quad fit(\mathbf{x}_j) = 0$
5: $\quad$ **For** $i = 1$ to m
6: $\quad\quad f_i'(\mathbf{x}_j) = (f_i(\mathbf{x}_j) - z_i^{min})/(z_i^{max} - z_i^{min})$
7: $\quad\quad fit(\mathbf{x}_j) = fit(\mathbf{x}_j) + f_i'(\mathbf{x}_j)$
8: $\quad$ **End For**
9: **End For**

$$fit(\mathbf{x}_j) = \sum_{i=1}^{m} f_i'(\mathbf{x}_j). \tag{3}$$

This fitness value roughly reflects the convergence information of an individual. Two factors — the number of objectives and the performance in each objective — determine this estimation function. A solution with good performance in the majority of objectives is likely to obtain a better (i.e., lower) *fit* value. It is notable that the accuracy of the estimation can be influenced by the shape of an MaOP's Pareto front [10]. However, as will be shown in Algorithm 5, the fitness value in VaEA is used to determine which one to survive between two closest solutions in terms of the vector angle (in the objective space). Hence, the influence of the shape of the Pareto front may be alleviated.

2) Computation of norm and vector angles: The function $Compute\_norm(S)$ (line 3 in Algorithm 2) calculates the norm (in the normalized objective space) of each solution in

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

S. For $\mathbf{x}_j$, its norm (denoted as $norm(\mathbf{x}_j)$) is defined as

$$norm(\mathbf{x}_j) \triangleq \sqrt{\sum_{i=1}^{m} f_i'(\mathbf{x}_j)^2}. \quad (4)$$

The norm is stored in an attribute named *norm* and is used to calculate vector angles between two solutions in the normalized objective space. The vector (acute) angle between solutions $\mathbf{x}_j$ and $\mathbf{y}_k$ is defined as below:

$$angle(\mathbf{x}_j, \mathbf{y}_k) \triangleq \arccos \left| \frac{\mathbf{F}'(\mathbf{x}_j) \bullet \mathbf{F}'(\mathbf{y}_k)}{norm(\mathbf{x}_j) \cdot norm(\mathbf{y}_k)} \right| \quad (5)$$

where $\mathbf{F}'(\mathbf{x}_j) \bullet \mathbf{F}'(\mathbf{y}_k)$ returns the inner product between $\mathbf{F}'(\mathbf{x}_j)$ and $\mathbf{F}'(\mathbf{y}_k)$, and its definition is given below.

$$\mathbf{F}'(\mathbf{x}_j) \bullet \mathbf{F}'(\mathbf{y}_k) = \sum_{i=1}^{m} f_i'(\mathbf{x}_j) \cdot f_i'(\mathbf{y}_k) \quad (6)$$

Obviously, $angle(\mathbf{x}_j, \mathbf{y}_k) \in [0, \pi/2]$.

3) Association: After determining the last layer F_l by using the non-dominated sorting procedure, the proposed algorithm selects $K = N - |P|$ solutions from F_l on the basis of vector angles if the population P is not full. Before the description of the association operation, we first give some basic definitions.

Definition 1: The target solution of $\mathbf{x}_j \in F_l$ is defined as the solution in P to which $\mathbf{x}_j$ has the minimum vector angle. We denote this minimum vector angle as $\theta(\mathbf{x}_j)$, and the corresponding index of the target solution is written as $\gamma(\mathbf{x}_j)$. If $\mathbf{x}_j$ has multiple target solutions, we can simply select a random one.

Definition 2: The vector angle from a solution $\mathbf{x}_j \in F_l$ to the population P is defined as the angle between $\mathbf{x}_j$ and its target solution, i.e., $angle(\mathbf{x}_j, P) = \theta(\mathbf{x}_j)$.

Definition 3: Extreme solutions in a solution set are defined as solutions that have minimum angle to m vectors $(1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, 0, \ldots, 1)$. We denote by $\mathbf{e}_i, i = 1, 2, \ldots, m$ these extreme solutions.

The association procedure is given in Algorithm 4. If P is empty (this is common when a higher dimensional objective space is considered), we first sort solutions in F_l in ascending order according to fitness values defined by (3). Then, m extreme solutions in F_l are first added into P (lines 3-4 in Algorithm 4). The inclusion of extreme solutions aims at the promotion of the diversity, especially the extensiveness of the population. Thereafter, the first m best converged solutions in terms of the fitness value are considered and added into P (lines 5-6 in Algorithm 4). Note here that well converged solutions may be also extreme solutions. In this case, we only add these solutions once. Once a solution has been added, it will be removed from F_l. For each of the remaining members in F_l (i.e., $\mathbf{x}_j$), its $\gamma(\mathbf{x}_j)$ and $\theta(\mathbf{x}_j)$ are initialized to -1 and ∞, respectively (lines 9-10 in Algorithm 4). Then for each $\mathbf{y}_k \in P$, the angle between $\mathbf{x}_j$ and $\mathbf{y}_k$ is calculated according to (5). Finally, $\gamma(\mathbf{x}_j)$ and $\theta(\mathbf{x}_j)$ are updated accordingly if a smaller angle is found (lines 13-16 in Algorithm 4). Fig. 1 illustrates the association operation in a two-dimensional objective space. According to vector angles, both $\mathbf{x}_1$ and $\mathbf{x}_2$ are associated with $\mathbf{y}_1$; $\mathbf{x}_3$ is associated with $\mathbf{y}_2$, and $\mathbf{x}_4$ is associated with $\mathbf{y}_4$. For $\mathbf{x}_3$, the $\theta(\mathbf{x}_3)$ and $\gamma(\mathbf{x}_3)$ are σ and 2, respectively.

Algorithm 4 $Association(P, F_l)$

Input: P and F_l
1: **If** $P = \emptyset$
2: $F_l = Sort(F_l)$ // Sort F_l according to fitness values
3: // Firstly, add m extreme solutions $\mathbf{e}_i$ into P
4: $P = P \cup \{\mathbf{e}_i\}, i = 1, \ldots, m$
5: // Secondly, add the first m best converged solutions
6: $P = P \cup \{\mathbf{x}_j\}, j = 1, \ldots, m$
7: **End If**
8: **For** each $\mathbf{x}_j \in F_l$
9: $\gamma(\mathbf{x}_j) = -1$
10: $\theta(\mathbf{x}_j) = \infty$
11: **For** each $\mathbf{y}_k \in P$
12: Calculate *angle* between $\mathbf{x}_j$ and $\mathbf{y}_k$ using (5)
13: **If** $angle < \theta(\mathbf{x}_j)$
14: $\gamma(\mathbf{x}_j) = k$
15: $\theta(\mathbf{x}_j) = angle$
16: **End If**
17: **End For**
18: **End For**

4) Niche-Preservation: In VaEA, the niche preservation procedure shown in Algorithm 5 considers both the convergence and diversity of the population that are guaranteed by the *worse-elimination* and *maximum-vector-angle-first* principles, respectively. Each member in the last layer F_l has a flag value that indicates whether this solution has been added into the population P or not. And it is initialized as $false$ at the beginning of the procedure (lines 1 and 2 in Algorithm 5). Next, for each $k \leq K$, the indexes of the members in F_l that have the maximum and minimum vector angles to P (the definition of vector angles from a solution to a set is given by *Definition 2*) are worked out and denoted by ρ and μ, respectively (lines 4 and 5).

Line 6 in Algorithm 5 is the *maximum-vector-angle-first*

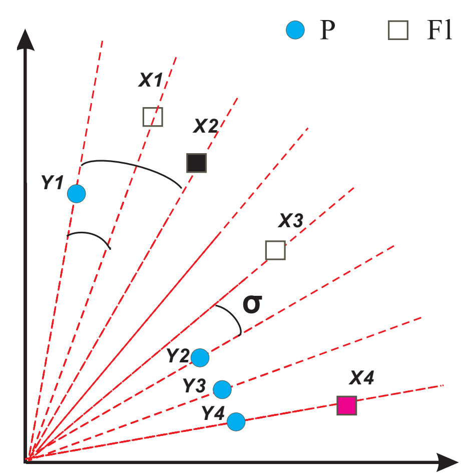


Fig. 1. Illustration of the association operation.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

principle that aims to select the best solution in terms of the maximum vector angle, and add it into the population to construct the new P. Details of this principle is shown in Algorithm 6. If ρ is *null* (meaning that all members in F_l have been added), then the procedure returns P. Otherwise, $\mathbf{x}_\rho$ is added into P and its *flag* is modified to *true* (lines 2-3). When $\mathbf{x}_\rho$ has been added, the vector angles from the remaining individuals in F_l to the new population P should be updated accordingly. To achieve this, we just need to calculate the vector angle between $\mathbf{x}_\rho$ and each member in F_l whose flag value is *false* (line 6). If a smaller angle is found, then the update is executed according to lines 8 and 9.

For example, in Fig. 1, $\mathbf{x}_2$ has the maximum vector angle (2σ) to $P = \{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4\}$. According to the *maximum-vector-angle-first* principle, $\mathbf{x}_2$ will be added. Clearly, the improvement of the distribution (mainly the uniformity) of P by adding $\mathbf{x}_2$ is more obvious than by adding other solutions, e.g., $\mathbf{x}_3$ and $\mathbf{x}_4$. After $\mathbf{x}_2$ is added, the region around it can be exploited by the algorithm and better solutions along the direction of $\mathbf{x}_2$ may be attainable. Now suppose two more solutions need to be added, and then the choice will be $\mathbf{x}_1$ and $\mathbf{x}_3$. On the contrary, $\mathbf{x}_4$ can be never considered because it has a zero vector angle to P. This is also understandable because a better solution $\mathbf{Y}_4$ in the same direction as $\mathbf{x}_4$ has already been included in the population P.

According to the above procedure, our proposed VaEA selects solutions dynamically and it is expected to keep a well distributed population. However, as discussed elsewhere [10], most individuals in a population are incomparable in terms of convergence since Pareto dominance often loses its effectiveness to differentiate individuals when dealing with MaOPs. In fact, for many-objective problems, most individuals fall into the first layer in the non-dominated sorting procedure, thereby providing insufficient selection pressure towards the true Pareto front. As a result, the diversity-based selection criterion will play a crucial role in determining the survival of individuals, which may make the individuals in the final population have a good distribution but may be also far away from the desired Pareto front.

Algorithm 5 $P = Niching(P, F_l, K)$

Input: P, F_l and K
Output: The new P
1: $T = |F_l|$ // $|F_l|$ *returns the cardinality of* F_l
2: $flag(\mathbf{x}_j) = false, \mathbf{x}_j \in F_l, j = 1, 2, \ldots, T$
3: **For** $k = 1$ to K
   // *Find the index with the maximum vector angle*
4: $\rho = \arg\max_{j=1}^{T}\{\theta(\mathbf{x}_j) | \mathbf{x}_j \in F_l \wedge (flag(\mathbf{x}_j) == false)\}$
   // *Find the index with the minimum vector angle*
5: $\mu = \arg\min_{j=1}^{T}\{\theta(\mathbf{x}_j) | \mathbf{x}_j \in F_l \wedge (flag(\mathbf{x}_j) == false)\}$
6: P = *Maximum-vector-angle-first* (P, ρ, T)
7: P = *Worse-elimination*(P, μ, T)
8: **End For**
9: **Return** P

Algorithm 6 P = *Maximum-vector-angle-first* (P, ρ, T)

Input: P, ρ and T
Output: The new P
1: **If** $\rho == null$, **return** P
2: $P = P \cup \{\mathbf{x}_\rho\}$ // $\mathbf{x}_\rho$ *is orderly added*
3: $flag(\mathbf{x}_\rho) = true$
4: **For** $j = 1$ to T
5: **If** $flag(\mathbf{x}_j) == false$ // $\mathbf{x}_j$ *is a member of* F_l
6: Calculate *angle* between $\mathbf{x}_j$ and $\mathbf{x}_\rho$ using (5)
7: **If** $angle < \theta(\mathbf{x}_j)$ // $\mathbf{x}_j$ *is associated with* $\mathbf{x}_\rho$
   // *Update the vector angle from* $\mathbf{x}_j$ *to* P
8: $\theta(\mathbf{x}_j) = angle$
   // *Update the corresponding index.*
9: $\gamma(\mathbf{x}_j) = |P|$
10: **End If**
11: **End If**
12: **End For**
13: **return** P

Algorithm 7 P = *Worse-elimination*(P, μ, T)

Input: P, μ and T
Output: The new P
1: **If** $(\mu \neq null) \wedge (\theta(\mathbf{x}_\mu) < \frac{\pi/2}{N+1})$
2: $r = \gamma(\mathbf{x}_\mu)$ // $\mathbf{y}_r$ *is the rth solution in* P
3: **If** $(fit(\mathbf{y}_r) > fit(\mathbf{x}_\mu)) \wedge (flag(\mathbf{x}_\mu) == false)$
4: Replace $\mathbf{y}_r$ with $\mathbf{x}_\mu$
5: $flag(\mathbf{x}_\mu) = true$
6: **For** $j = 1$ to T
7: **If** $flag(\mathbf{x}_j) == false$
8: Calculate *angle* between $\mathbf{x}_j$ and $\mathbf{x}_\mu$ by (5)
9: **If** $\gamma(\mathbf{x}_j) \neq \gamma(\mathbf{x}_\mu)$
10: **If** $angle < \theta(\mathbf{x}_j)$
11: $\theta(\mathbf{x}_j) = angle$
12: $\gamma(\mathbf{x}_j) = r$
13: **End If**
14: **Else**
15: $\theta(\mathbf{x}_j) = angle$
16: **End If**
17: **End If**
18: **End For**
19: **End If**
20: **End If**
21: **Return** P

To alleviate the above phenomenon, VaEA allows worse solutions in terms of the convergence to be conditionally replaced by other individuals so as to keep a balance between convergence and diversity. This is called the *worse-elimination* principle (line 7 in Algorithm 5) and it is described in Algorithm 7 in details.

Specifically, if the angle between $\mathbf{x}_\mu$ (not null) and its target solution is smaller than $\sigma = \frac{\pi/2}{N+1}$, where N is the population size, then the target solution $\mathbf{y}_r \in P$ (where $r = \gamma(\mathbf{x}_\mu)$, the index of the target solution) is replaced by $\mathbf{x}_\mu$ when $\mathbf{x}_\mu$ is not added and is better than $\mathbf{y}_r$ in terms of the fitness value

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

calculated according to (3) (see lines 1-5 in Algorithm 7). The basic idea behind the *worse-elimination* principle is that it is often unnecessary to keep multiple solutions with identical search directions. In VaEA, the similarity of search directions is measured by the vector angle. If the angle between two solutions is smaller than the threshold $\sigma = \frac{\pi/2}{N+1}$, then they are deemed to search in the similar direction. In this case, only the better one can survive. The value of σ is the angle between two solutions in the ideal direction division with N solutions. Take Fig. 1 as an example, $\sigma = \frac{\pi/2}{8+1} = 0.1745 = 10°$. Usually, the population size N is a larger integer (e.g., more than 100), then σ takes a very small value which is small enough to determine the same search directions.

After replacement, the angles from each member $\mathbf{x}_j \in F_l$ to the new population P should be updated accordingly. We consider the following two scenarios:

1) $\gamma(\mathbf{x}_j) \neq \gamma(\mathbf{x}_\mu)$ means that $\mathbf{x}_j$ and $\mathbf{x}_\mu$ are associated with different individuals in P. In this case, we just need to check whether the *angle* between $\mathbf{x}_j$ and $\mathbf{x}_\mu$ is smaller than $\theta(\mathbf{x}_j)$ or not. If so, $\theta(\mathbf{x}_j)$ and $\gamma(\mathbf{x}_j)$ are updated to *angle* and r, respectively (lines 6-12 in Algorithm 7). Otherwise, they are kept unchanged.
2) $\gamma(\mathbf{x}_j) = \gamma(\mathbf{x}_\mu)$ indicates that $\mathbf{x}_j$ and $\mathbf{x}_\mu$ are associated with the same individual in P (i.e., $\mathbf{y}_r$). In this situation, we just need to update the value of $\theta(\mathbf{x}_j)$ (line 15 in Algorithm 7). It should be noted here that a smaller vector angle may be found if the angle between $\mathbf{x}_j$ and $\mathbf{x}_\mu$ is larger than $\theta(\mathbf{x}_j)$, but this happens in a very extreme situation. This is because $\mathbf{x}_j$ is closest to $\mathbf{y}_r$ and the angle between $\mathbf{x}_\mu$ and $\mathbf{y}_r$ is very small, hence, with great probability, $\mathbf{x}_j$ has closest vector angle to $\mathbf{x}_\mu$ as well.

To illustrate this update procedure, we give an example in Fig. 2 through a two-dimensional objective space. Suppose all 8 solutions are in the first layer and $\mathbf{Y}_1 = (1, 0)$ has been added into P. According to the *maximum-vector-angle-first* principle, $\mathbf{Y}_2 = (0, 1)$ and $\mathbf{Y}_r = (0.5, 0.5)$ are then included orderly. Once $\mathbf{Y}_r$ has been added, the procedure identifies a solution $\mathbf{x}_\mu = (0.45, 0.51)$ that searches in a similar direction. Since the advantage of $\mathbf{x}_\mu$ over $\mathbf{Y}_r$ in the first objective is more than the disadvantage in the second objective, $\mathbf{x}_\mu$ is better than $\mathbf{Y}_r$ in terms of the fitness value ($fit(\mathbf{x}_\mu) = 0.96 < fit(\mathbf{Y}_r) = 1$). Therefore, $\mathbf{Y}_r$ is replaced by $\mathbf{x}_\mu$. Next, the angles from $\mathbf{X}_i (i = 1, 2, 3, 4)$ to P should be modified (or equivalently, the target solutions should be updated). For $\mathbf{X}_1$, its target solution is different from that of $\mathbf{X}_\mu$ (the former is associated with $\mathbf{Y}_2$, while the latter is associated with $\mathbf{Y}_r$), but the angle between $\mathbf{X}_1$ and $\mathbf{X}_\mu$ is smaller than the original angle (i.e., angle between $\mathbf{X}_1$ and $\mathbf{Y}_2$). Therefore, according to the first scenario, the target solution for $\mathbf{X}_1$ is modified to $\mathbf{X}_\mu$. For $\mathbf{X}_4$, its target solution is also different from that of $\mathbf{X}_\mu$, but the angle between $\mathbf{X}_4$ and $\mathbf{X}_\mu$ is larger than the original one. Hence, it is unnecessary to change its target solution.

The $\mathbf{X}_2$, $\mathbf{X}_3$ and $\mathbf{X}_\mu$ have the same target solution $\mathbf{Y}_r$. This falls into the second scenario. Therefore, both $\mathbf{X}_2$ and $\mathbf{X}_3$ are still associated with the *r*th solution in P ($\mathbf{X}_\mu$ is now in the *r*th position). However, $\theta(\mathbf{X}_2)$ and $\theta(\mathbf{X}_3)$ should be modified accordingly after replacing $\mathbf{Y}_r$ by $\mathbf{X}_\mu$.

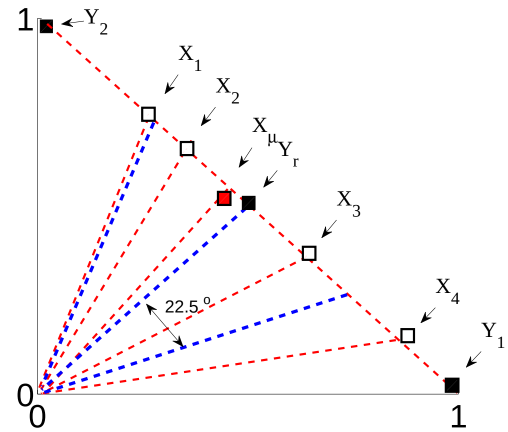


Fig. 2. Illustration of updating angles in the *Worse-elimination* principle.

TABLE I
TIME COMPLEXITY OF PEER ALGORITHMS

Algorithm	Worst Time Complexity
VaEA	$max\{O(Nlog^{m-2}N), O(mN^2)\}$
NSGA-III	$max\{O(Nlog^{m-2}N), O(mN^2)\}$ [8]
MOEA/DD	$O(mN^2)$ [11], [47]
Two_Arch2	$max\{O(Nlog^{m-2}N), O(mN^2)\}$ [36]
MaOEA-DDFC	$O(mN^2 + N^2 \cdot log_2 N)$ [37]
GrEA	$O(N^3)$ [7]
SPEA2+SDE	$O(N^3)$ [18]

C. Computational Complexity Analysis

The normalization (line 2 in Algorithm 2) of a population with $2N$ solutions having m objectives requires $O(mN)$ divisions. The calculation of norm (line 3 in Algorithm 2) for a population of size $2N$ also needs $O(mN)$ additions. The non-dominated sorting (line 4 in Algorithm 2) has a time complexity of $O(Nlog^{m-2}N)$ [46] in terms of the number of comparisons. Both the association and niching operations (lines 13 and 14 in Algorithm 2) require $O(mN^2)$ additions in the worst case. Therefore, the overall worst-case complexity of one generation in VaEA is $max\{O(Nlog^{m-2}N), O(mN^2)\}$[1]. The worst time complexity of some MaOEAs is summarized in Table I.

D. Discussion

After describing details of VaEA, this section presents similarities and differences between VaEA and NSGA-III as well as MOEA/DD. Particularly, the relationship between VaEA and MOEA/D is well discussed. First and foremost, as stated previous, the major difference is that our proposed

[1]If we only consider comparisons, the time complexity of VaEA would be $max\{O(Nlog^{m-2}N), O(N^2)\}$. The non-dominated sorting requires $O(Nlog^{m-2}N)$ comparisons, and both *Association* and *Niching* operations need $O(N^2)$ comparisons

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

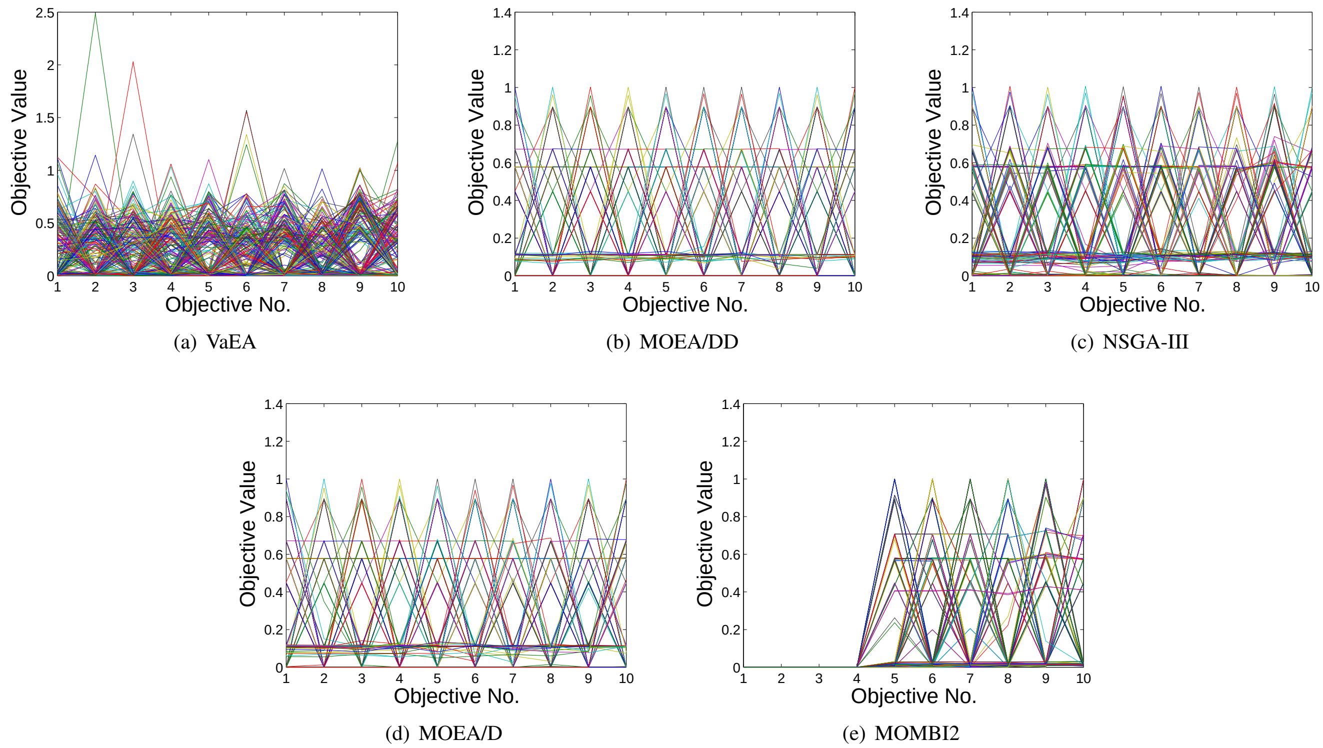


Fig. 3. The final solution set of the five algorithms on the ten-objective DTLZ3, shown by parallel coordinates.

VaEA does not need to provide a set of reference points or weight vectors beforehand. Just as described in Section I, it is not easy to specify a number of well-distributed reference points or weight vectors. Other differences can be summarized as follows.

1) Similarities and differences between VaEA and NSGA-III:

- Both of them use non-dominated sorting procedure to divide the union population into different layers. And the last layer is important and should be identified first.
- Both of them involve the normalization of the union population. However, the normalization procedure of NSGA-III is much more complicated than that of VaEA: In VaEA, the population is normalized by using only ideal points and nadir points of the population, while the normalization in NSGA-III needs to calculate another component, i.e., the intercepts along each objective axis, which is based on the solution of a system of linear equations [29]. Therefore, the normalization of VaEA is time cheaper than that of NSGA-III.
- Both VaEA and NSGA-III consider a niching technique for the balance between the diversity and convergence. In NSGA-III, each population member is associated with a reference line based on the perpendicular distance that could be measured by angles to some extent. The niche-preserving operation in both NSGA-III and VaEA adds solutions in F_l one by one. However, the differences are also clear. First, the association targets are different: In NSGA-III, the members are associated with reference lines that are determined by a set of predefined reference points, while members in VaEA are associated with already included solutions. Second, in NSGA-III, all members in $S_t = \bigcup_{i=1}^{l} F_i$ need an association operation, while in VaEA, only members in F_l need this operation. Finally, the association in VaEA is dynamic, while that in NSGA-III is static. In VaEA, the member in F_l may be associated with another solution when a new solution is added into the population. However, the member in NSGA-III is associated with a fixed reference line.

2) Similarities and differences between VaEA and MOEA/DD:

- Both of them divide the population into several non-domination levels based on the concept of Pareto dominance. However, the selection procedure in MOEA/DD does not fully obey the decision made by Pareto dominance. Particularly, a solution that is associated with an isolated subregion, will be preserved without reservation even if it belongs to the last non-domination level. Although, this mechanism benefits the diversity maintenance, it will slow the convergence speed of MOEA/DD [11].
- Unlike VaEA, MOEA/DD is a steady-state algorithm. That is to say, once a new solution has been generated, it is used to update the non-domination level structure, as well as the population.
- Both of them use a scalar function to measure the convergence of a solution. In VaEA, the convergence is evaluated by the fitness value defined in Eq. (3), while in MOEA/DD the PBI objective function is used.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

Although PBI function provides a joint measurement of both convergence and diversity, it introduces a parameter θ that should be determined beforehand. Note that we don't consider the diversity when evaluating a solution in the *worse-elimination* principle, this is because the diversity among solutions is guaranteed by the *maximum-vector-angle-first* principle.

3) The relationship between VaEA and MOEA/D: In the *Association* procedure of VaEA (Algorithm 4), by computing the angles, a solution in the last front is actually associated with a search direction pointing towards the ideal point in the normalized objective space. One of the major features in MOEA/D is that it approximates the Pareto front along multiple search directions. That is, the search in MOEA/D is guided by a number of search directions (or subproblems). From this point of view, VaEA uses the same idea as in MOEA/D with dynamical decomposition methods [42] and [44], where the search directions are adaptively or dynamically changed during the search process so as to obtain a better distribution. However, VaEA does not convert an MOP into a number of scalar optimization subproblems, which is one of the major differences between the two algorithms.

III. Simulation Results

This section is devoted to the experimental study for the verification of the performance of the VaEA algorithm[2]. We compare the proposed algorithm with MOEA/DD[3] [11], NSGA-III[4] [8], MOEA/D[5] [38], and MOMBI2[6] [48] on test problems from both DTLZ [49] and WFG test suites [50]. These peer algorithms have been demonstrated to be effective when handling MaOPs, and they can be divided into two classes: the reference-points/weight-vectors based algorithms (MOEA/DD, NSGA-III and MOEA/D) and indicator based algorithm (MOMBI2). Brief descriptions of MOEA/DD, MOEA/D and NSGA-III can be found in Section I. MOMBI2, one of the recently proposed indicator-based MaOEAs which uses $R2$ indicator as the selection criterion. Compared with the hypervolume [51], $R2$ is attractive due to its low computational cost and weak-Pareto compatibility [48]. In MOMBI2, individuals are ranked by using the achievement scalarizing function (ASF). Like MOEA/D and MOEA/DD, MOMBI2 also needs a set of supplied weight vectors. As shown by the experimental results in [48], MOMBI2 presented superiorities over other MOEAs, and could be a suitable alternative for solving MaOPs.

A. Test Problems

For the purpose of performance comparison, four test problems from the DTLZ test suite (DTLZ1 to DTLZ4), and nine ones from the WFG test suite (WFG1 to WFG9) are chosen for our empirical studies. All these problems can be scaled to any number of objectives and decision variables. For each problem, the number of objectives is set to 3, 5, 8, 10 and 15, respectively. According to [49], the number of decision variables is set to $n = m + r - 1$ for DTLZ test instances, where $r = 5$ for DTLZ1 and $r = 10$ for DTLZ2 to DTLZ4. Following the suggestions in [50], the number of decision variables is set to $n = k+l$ for WFG test instances, where the position-related variable $k = 2 \times (m-1)$, and the distance-related variable $l = 20$.

According to [50], Pareto fronts of the above test problems have various characteristics (e.g., linear, convex, concave, mixed, multi-modal etc.) and they pose a significant challenge for an EMO algorithm to find a well-converged and well-distributed solution set.

B. Performance Metrics

In our experimental study, three widely used metrics are chosen to evaluate the performance of each algorithm. They are the Inverted Generational Distance (IGD) [52], generalized SPREAD (or Δ) [53] and Hypervolume (HV) [51]. IGD and HV can provide a joint measurement of both the convergence and diversity of obtained solutions, while the generalized SPREAD concentres on evaluating the distribution of an approximated solution set.

1) IGD: [52], [38]: Let P be an approximation set, and P^* be a set of non-dominated points uniformly distributed along the true Pareto front, and then the IGD metric is defined as follows:

$$IGD(P) = \frac{1}{|P^*|} \sum_{z^* \in P^*} dist(z^*, P), \tag{7}$$

where $dist(z^*, P)$ is the Euclidean distance between z^* and its nearest neighbor in P, and $|P^*|$ is the cardinality of P^*. The advantages of IGD measure are twofold [54]: One is its computational efficiency. The other is its generality: If $|P^*|$ is large enough to cover the true Pareto front very well, then both convergence and diversity of approximate set P can be measured by $IGD(P)$. Due to these two nice properties, IGD has been frequently used to evaluate the performance of MOEAs for MaOPs [7], [8], [22], [55], etc. For an EMO algorithm, a smaller IGD value is desirable[7].

2) Generalized SPREAD [53] : To measure the extent of spread by the set of computed solutions, the metric SPREAD [56] was first proposed for only two-objective problems. However, in the context of many-objective optimization, this metric is not applicable. In this paper, we use the generalized SPREAD (denoted by Δ) instead, which was first proposed in [53].

$$\Delta(P) = \frac{\sum_{i=1}^{m} dist(e_i, S) + \sum_{z \in S} |dist(z, S) - \bar{d}|}{\sum_{i=1}^{m} dist(e_i, S) + |S| * \bar{d}}, \tag{8}$$

[2]The code of VaEA is available at https://www.researchgate.net/profile/Xiang_Yi9/publications.

[3]The code of MOEA/DD is downloaded from http://www.cs.bham.ac.uk/ likw/code/MOEADD.zip.

[4]The code of NSGA-III is downloaded from http://learn.tsinghua.edu.cn:8080/2012310563/ManyEAs.rar which is maintained by X. Yao's research team.

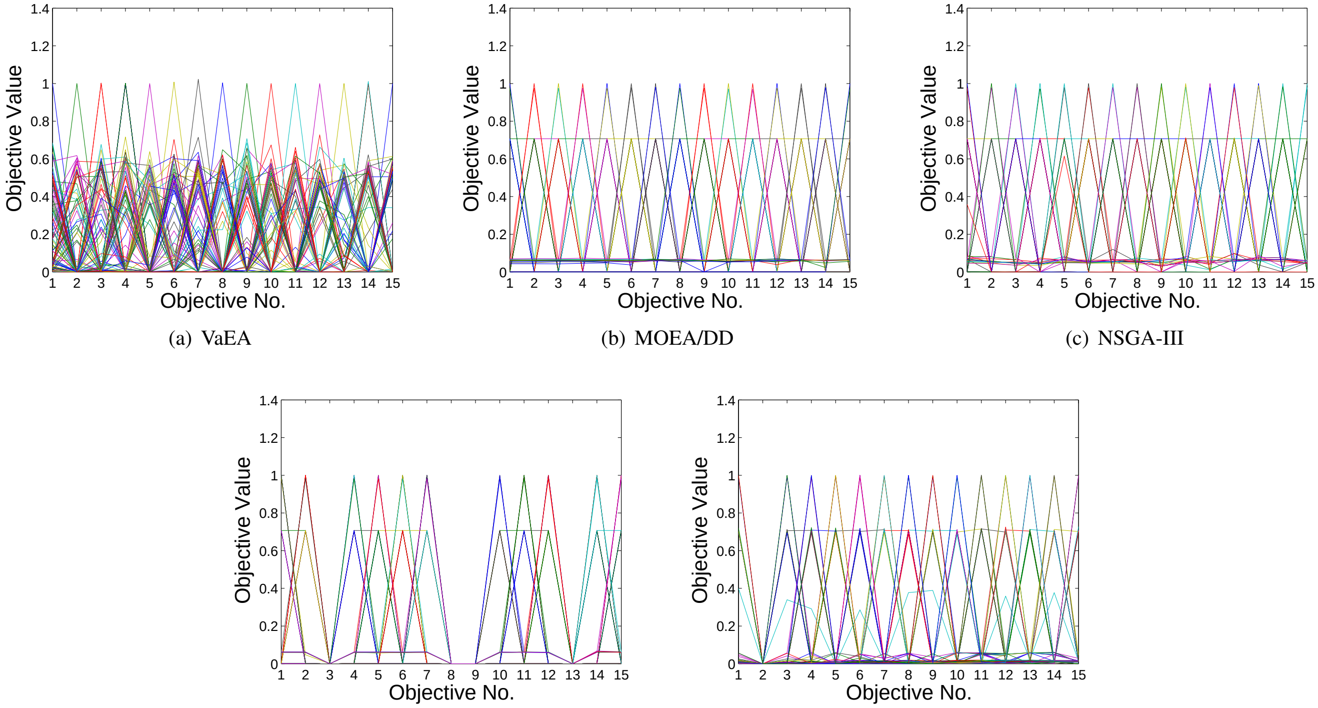
[5]The code of MOEA/D is downloaded from http://dces.essex.ac.uk/staff/zhang/webofmoead.htm

[6]The code of MOMBI2 is downloaded from http://jmetal.github.io/jMetal/

[7]In our empirical studies, IGD metric is implemented by the codes from http://dces.essex.ac.uk/staff/qzhang/moeacompetition09.htm.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

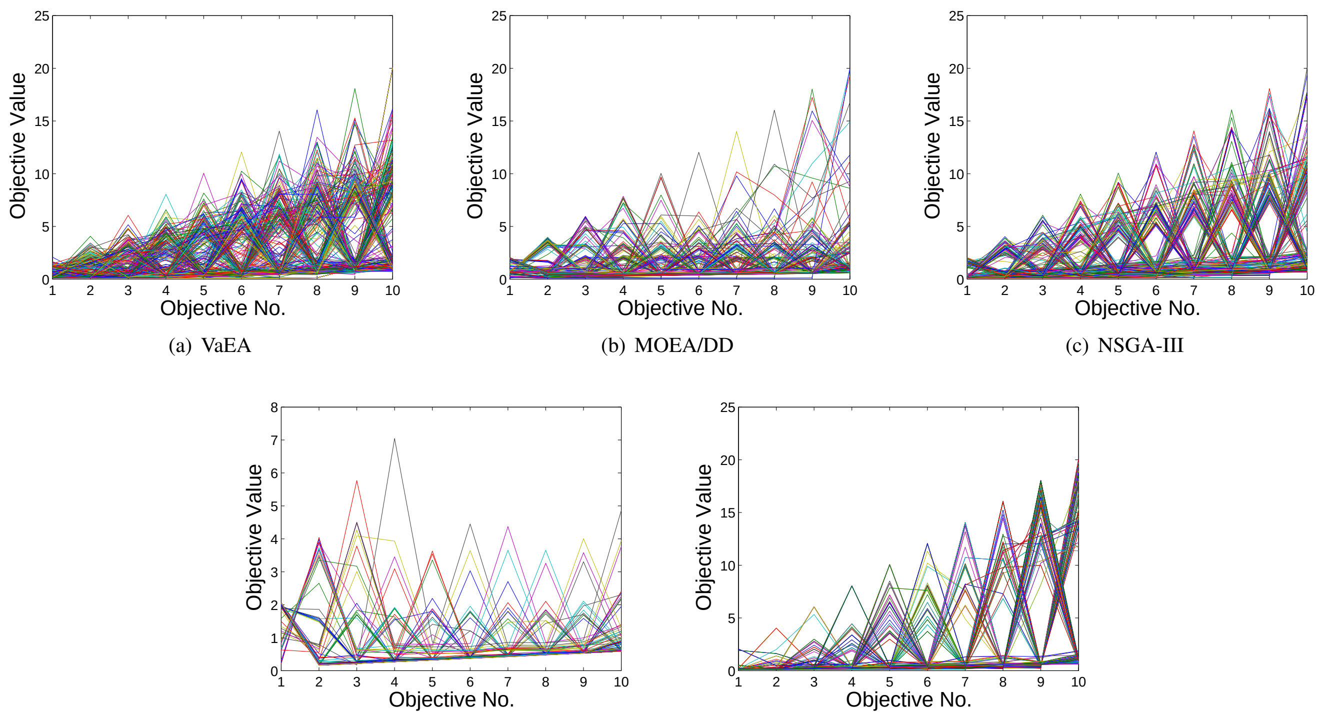
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation



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

(d) MOEA/D (e) MOMBI2

Fig. 4. The final solution set of the five algorithms on the fifteen-objective DTLZ4, shown by parallel coordinates.



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

(d) MOEA/D (e) MOMBI2

Fig. 5. The final solution set of the five algorithms on the ten-objective WFG9, shown by parallel coordinates.

where S is a set of solutions, S^* is the set of Pareto optimal solutions, $e_1, e_2, \ldots, e_m$ are m extreme solutions in S^* and

$$dist(z, S) = \min_{y \in S, y \neq z} ||\mathbf{F}(z) - \mathbf{F}(y)||^2, \quad (9)$$

$$\bar{d} = \frac{1}{|S^*|} \sum_{x \in S^*} dist(x, S). \quad (10)$$

In this study, the Δ metric is implemented by the open

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

source software jMetal[8] [57].

In the calculation of both $IGD(P)$ and $\Delta(P)$, a set of reference Pareto-optimal solutions is needed. For DTLZ1 to DTLZ4, the same method as in [11] is used to generate reference Pareto-optimal points. Specifically, a set of well distributed weight vectors is first produced. For $m \leq 5$, the Das and Dennis's [58] systematic approach is utilized to produce weight vectors where a parameter p should be specified. For $m > 5$, weight vectors are generated by the two-layer generation method [8], [11] which involves two parameters h_1 and h_2, denoting the number of divisions for the boundary and inside layers, respectively. Thereafter, the intersecting points of weight vectors and the Pareto-optimal surface of DTLZ test problems can be exactly located (because analytical forms of the exact Pareto-optimal surfaces of these test problems are known). Finally, these intersecting points can be regarded as the reference points. The Pareto-optimal surface of DTLZ1 is a hyperplane, while that of DTLZ2 to DTLZ4 is a hypersphere. As discussed in [54], the number of reference points for calculating IGD should be large enough so as to cover the true front as well as possible. Therefore, we use relatively larger values of divisions in the weight vectors generation method. The number of divisions for different numbers of objectives is listed in Table II, and the last column gives the corresponding number of reference points for DTLZ test problems.

TABLE II
THE NUMBER OF DIVISIONS WHEN GENERATING WEIGHT VECTORS.

m	$h_1(p)$	h_2	No. of Reference Points (DTLZ)
3	25	-	351
5	13	-	2,380
8	7	6	5,148
10	6	5	7,007
15	5	4	14,688

The two-layer reference point/weight vector generation method is applied only for the number of objectives greater than 5.

For WFG test problems, the range of the ith ($i = 1, 2, \ldots, m$) objective is $[0, 2i]$. Since the Pareto-optimal front of WFG4-9 is also a hypersphere, we can get the reference points of WFG4-9 by multiplying the ith objective of the reference points of DTLZ2-4 by $2i$. Hence, the number of reference points is the same as DTLZ test problems, which is shown in the last column of Table III. For WFG1 and WFG2, their Pareto fronts are irregular. Because their shape functions are known, we sample a large number of points in the underlying space $[0, 1]^{m-1}$, and then calculate objective values according to analytic expressions of the shape functions. Finally, we remove all dominated points, and the remaining points constitute the final reference set. Based on the above method, we obtain the number of reference points for WFG1 and WFG2, which is given in columns 2 and 3 in Table III. For WFG3, it was originally designed to have a degenerate Pareto front, i.e., a straight line in the objective space. However, according to the latest study [59], WFG3 with three or more objectives also has non-degenerate Pareto Front whose derivation, even for the case of three objectives, is extremely difficult. Given this, as suggested in [59], we construct the reference point set by selecting all the non-dominated solutions from all the returned solutions by 20 runs of each algorithm. The number of reference points for each objective is presented in column 4 of Table III.

[8]http://jmetal.github.io/jMetal/

TABLE III
THE NUMBER OF REFERENCE POINTS USED FOR THE CALCULATION OF IGD AND Δ ON WFG TEST PROBLEMS.

Objectives (m)	WFG1	WFG2	WFG3	WFG4-9
3	421	148	5,138	351
5	2,801	1,601	19,570	2,380
8	5,467	4,690	15,353	5,148
10	20,705	13,634	29,750	7,007
15	49,151	32,768	15,232	14,688

3) HV: This metric measures the volume of the objective space between the obtained solutions set and a specified reference point in the objective space. For HV, a larger value is preferable. For the calculation of HV, two issues are crucial: One is the scaling of the search space, and the other is the choice of the reference point. We normalize the objective value of obtained solutions according to the range of the problems' Pareto fronts, and set reference point to 1.1 times of the upper bounds of the true Pareto fronts. In our study, the Monte Carlo sampling [60] is applied to approximate HV, where the number of sampling points is set to 1,000,000. Note that solutions dominated by the reference point are discarded for the calculation of HV. In this paper, the HV is implemented by the codes in [10].

C. General Experimental Settings

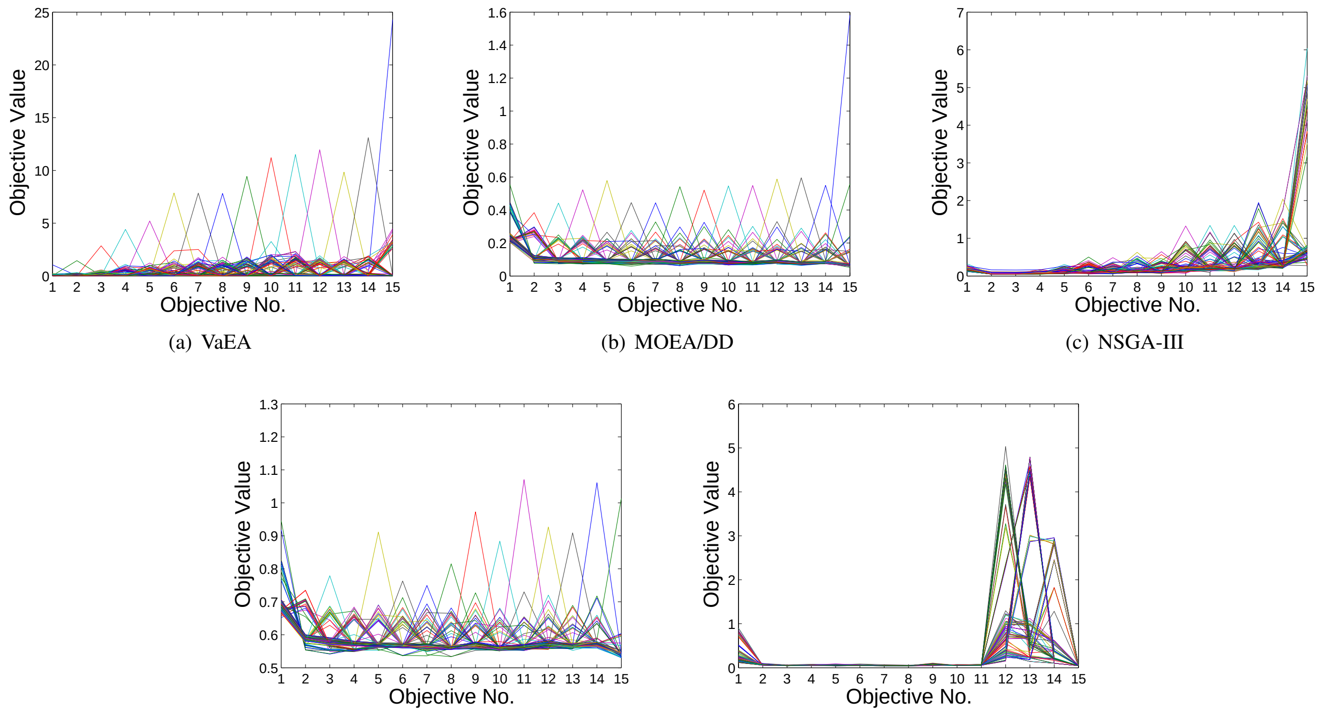
The parameter settings for this experiment are listed as below unless otherwise mentioned.

1) Population size: According to [8], the population size in NSGA-III is set as the smallest multiple of four larger than the number of reference points (H) produced by the so-called two-layer reference point (or weight vector) generation method. The population size in VaEA keeps the same as in NSGA-III. Since MOMBI2 involves a binary tournament selection, it requires the population size to be even numbers. Therefore, in MOMBI2, we use the same population size as in NSGA-III or VaEA. For MOEA/DD and MOEA/D, the population size is set to $N = H$, as recommend by their developers [11], [38]. The value of N for different number of objectives is summarized in Table IV.

2) Number of independent runs and termination condition: All algorithms are independently run 20 times on each test instance and terminated when a predefined maximum function evaluations (MFE) reaches. The settings of MFE for different numbers of objectives are listed in Table V. For VaEA, the termination condition can be easily determined by $G_{max} = MFE/N$.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation



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

(d) MOEA/D (e) MOMBI2

Fig. 6. The final solution set of the five algorithms on the fifteen-objective WFG1, shown by parallel coordinates.

TABLE IV
THE POPULATION SIZE (N) FOR DIFFERENT NUMBERS OF OBJECTIVES

m	H	NSGA-III VaEA MOMBI2	MOEA/DD MOEA/D
3	91	92	91
5	210	212	210
8	156 ($h_1 = 3, h_2 = 2$)	156	156
10	275 ($h_1 = 3, h_2 = 2$)	276	275
15	135 ($h_1 = 2, h_2 = 1$)	136	135

TABLE V
THE SETTINGS OF MFE FOR DIFFERENT NUMBERS OF OBJECTIVES

Test instance	DTLZ1	DTLZ2	DTLZ3	DTLZ4	WFG1-9
$m = 3$	36,800	23,000	92,000	55,200	$92,000$
$m = 5$	127,200	74,200	212,000	212,000	$265,000$
$m = 8$	117,000	78,000	156,000	195,000	$234,000$
$m = 10$	276,000	207,000	414,000	552,000	$552,000$
$m = 15$	204,000	136,000	272,000	405,000	$405,000$

3) Parameter settings for operators: In all algorithms, the simulated binary crossover (SBX) and polynomial mutation are used to generate offspring solutions. The crossover probability p_c and mutation probability p_m are set to 1.0 and $1/n$, respectively. For SBX operator, its distribution index is $\eta_c = 30$, and the distribution index of mutation operator is $\eta_m = 20$ [8], [11].

4) Parameter settings for algorithms: Following the practice in [38] and [11], the PBI approach is used in MOEA/D and MOEA/DD with a penalty parameter $\theta = 5$. In both algorithms, the neighborhood size T is set to 20. For MOEA/DD, a probability δ is set for choosing a neighboring partner of a parent in the mating selection procedure. As recommend by its developers, this value is set to $\delta = 0.9$. Following the practice in [48], two parameters in MOMBI2 are set as $\epsilon = 1e-3$ and $\alpha = 0.5$, respectively.

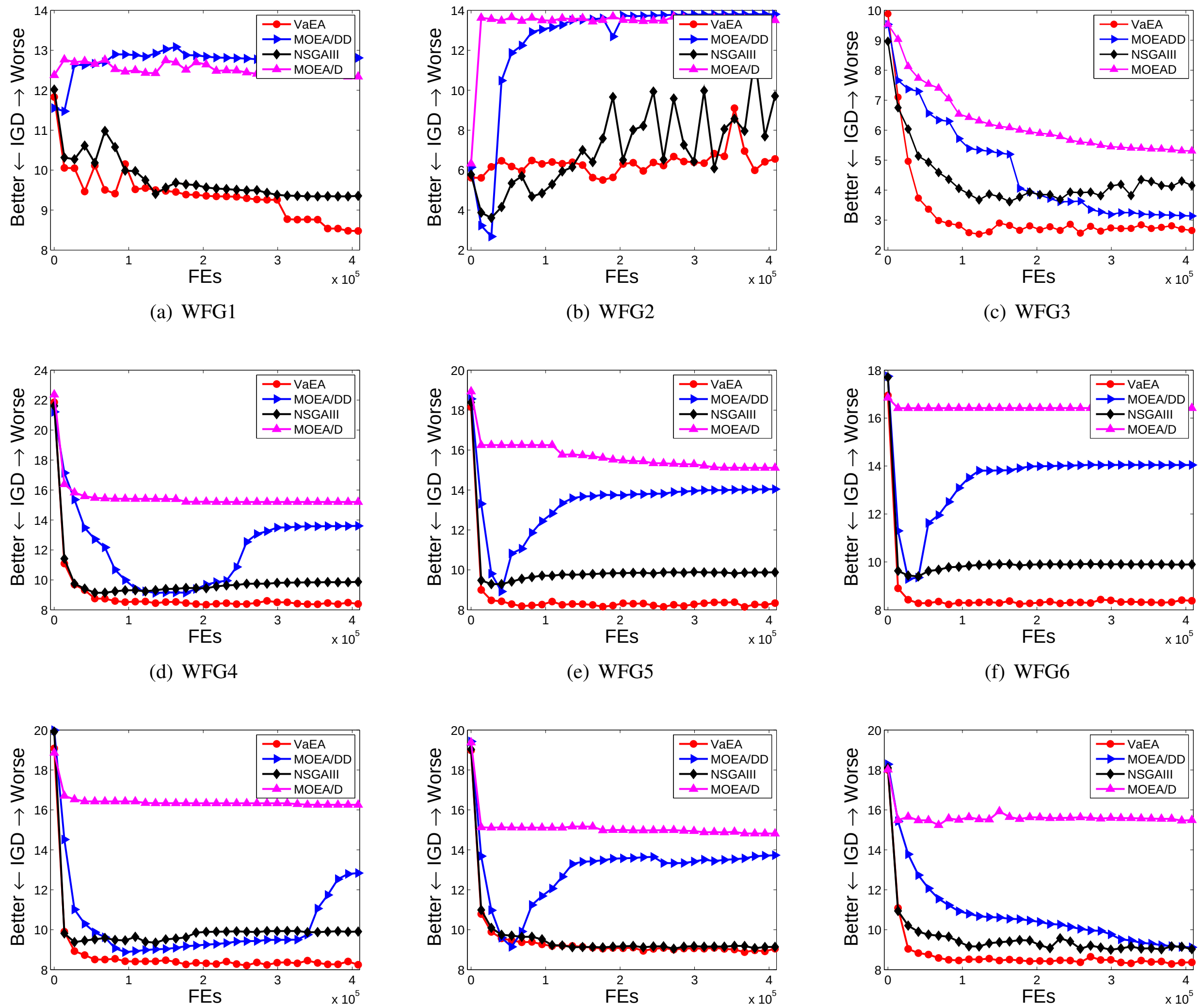
D. Results and Analysis

Table VI shows the median and Inter Quartile Range (IQR) results on all the twenty DTLZ test instances in terms of the IGD metric. The significance of difference between VaEA and the peer algorithms is determined by using the well known Wilcoxon's rank sum test [61]. As shown, MOEA/DD is the most effective algorithm in terms of the number of the best or the second best results it obtains. NSGA-III performs very competitively to MOEA/DD. For MOEA/D, it obtains the best performance on DTLZ1, achieving the best results for 8-, 10- and 15-objective test instances. The performance of MOMBI2 is poor than that of its competitors, and it performs best only on DTLZ2 and DTLZ4 with five objectives. Our proposed VaEA obtains significantly better performance than other algorithms on 10- and 15-objective DTLZ2 and DTLZ4, and shows clear improvements over MOMBI2 and MOEA/D on 13 and 7 out of 20 test instances, respectively.

It is found from the results that better performance on DTLZ test problems is obtained by decomposition and reference point based algorithms, i.e., MOEA/DD, MOEA/D and NSGA-III. On one hand, this finding is consistent with that in [8] and [11]. On the other hand, the phenomenon can be well interpreted:

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation



(a) WFG1 (b) WFG2 (c) WFG3

(d) WFG4 (e) WFG5 (f) WFG6

(g) WFG7 (h) WFG8 (i) WFG9

Fig. 7. Evolutionary trajectories of IGD on all the fifteen-objective WFG test problems.

Since Pareto fronts of DTLZ test problems are regular, being either a hyperplane or a hypersphere, decomposition or reference points based algorithms predefine a set of well distributed weight vectors or reference points which can promote the diversity of the obtained solutions. Furthermore, all algorithms adopt specific mechanisms to keep a good balance between diversity and convergence. For example, the PBI approach used in MOEA/DD and MOEA/D simultaneously measures diversity and convergence of a solution. In NSGA-III, the non-dominated sorting method and a well designed niching procedure collaboratively make the algorithm useful.

To visually understand the solutions' distribution, Figs. 3 and 4 plot, by parallel coordinates, the final solutions of one run with respect to the ten-objective DTLZ3 and fifteen-objective DTLZ4, respectively. This run is associated with the particular run that obtains the closest result to the median value of IGD. As shown in Fig. 3, the solutions obtained by MOEA/D, MOEA/DD and NSGA-III distribute similarly, while those found by MOMBI2 fail to cover the first three objectives well. Though solutions obtained by VaEA have a good

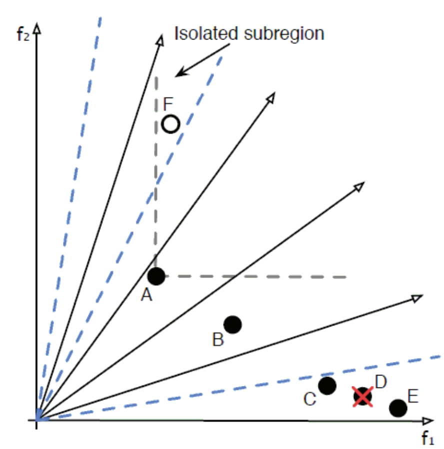


Fig. 8. F belongs to the worst non-domination level, but it is associated with an isolated subregion. Therefore, it survives in MOEA/DD. According to [11], this mechanism may slow down the convergence speed. Note that this figure is borrowed from the original study of MOEA/DD [11].

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

TABLE VI
MEDIAN AND IQR (IN BRACKETS) OF IGD METRIC ON DTLZ TEST INSTANCES. THE BEST AND THE SECOND BEST RESULTS FOR EACH TEST INSTANCE ARE SHOWN WITH DARK AND LIGHT GRAY BACKGROUND, RESPECTIVELY.

	m	VaEA	MOEA/DD	NSGA-III	MOEA/D	MOMBI2
DTLZ1	3	2.379E−02 (9.7E−04)	2.007E−02 (7.7E−05)◦	2.061E−02 (8.6E−04)◦	2.028E−02 (3.6E−04)◦	2.238E−02 (4.3E−03)‡
	5	5.615E−02 (8.8E−04)	5.292E−02 (3.6E−05)◦	5.288E−02 (1.7E−04)◦	5.295E−02 (4.9E−05)◦	5.528E−02 (1.4E−02)‡
	8	1.042E−01 (2.9E−03)	9.332E−02 (1.3E−04)◦	9.418E−02 (6.1E−04)◦	9.317E−02 (5.0E−04)◦	1.666E−01 (4.9E−02)•
	10	1.148E−01 (3.4E−03)	1.088E−01 (1.5E−04)◦	1.094E−01 (3.8E−04)◦	1.086E−01 (3.0E−04)◦	2.252E−01 (1.3E−01)•
	15	2.126E−01 (1.1E−02)	1.703E−01 (1.6E−03)◦	1.773E−01 (1.7E−03)◦	1.656E−01 (1.5E−03)◦	3.201E−01 (5.3E−02)•
DTLZ2	3	5.686E−02 (7.2E−04)	5.240E−02 (9.8E−06)◦	5.241E−02 (1.0E−04)◦	5.241E−02 (9.7E−06)◦	5.289E−02 (8.9E−04)◦
	5	1.681E−01 (2.0E−03)	1.659E−01 (1.3E−05)◦	1.658E−01 (2.3E−04)◦	1.659E−01 (2.1E−05)◦	1.627E−01 (8.3E−04)◦
	8	3.772E−01 (7.4E−03)	3.400E−01 (1.9E−04)◦	3.422E−01 (9.6E−04)◦	3.400E−01 (4.4E−05)◦	4.085E−01 (8.4E−04)•
	10	4.186E−01 (4.9E−03)	4.211E−01 (3.9E−04)•	4.215E−01 (9.3E−04)•	4.214E−01 (4.3E−04)•	4.464E−01 (8.1E−04)•
	15	6.061E−01 (5.9E−03)	6.192E−01 (4.0E−04)•	6.199E−01 (1.3E−03)•	6.186E−01 (3.5E−04)•	8.676E−01 (4.0E−02)•
DTLZ3	3	5.888E−02 (1.3E−03)	5.261E−02 (1.4E−04)◦	5.473E−02 (7.7E−03)◦	5.348E−02 (1.1E−03)◦	6.737E−02 (1.6E−02)•
	5	1.703E−01 (2.7E−03)	1.661E−01 (1.3E−04)◦	1.658E−01 (3.5E−04)◦	1.662E−01 (2.8E−04)◦	2.148E−01 (2.1E−02)•
	8	3.893E−01 (3.2E−02)	3.411E−01 (1.3E−03)◦	3.464E−01 (5.7E−03)◦	3.419E−01 (4.9E−03)◦	5.165E−01 (2.0E−01)‡
	10	4.256E−01 (1.2E−02)	4.213E−01 (5.1E−04)◦	4.216E−01 (1.1E−03)◦	4.206E−01 (1.6E−03)◦	7.075E−01 (2.1E−01)•
	15	6.938E−01 (1.5E−01)	6.201E−01 (6.9E−04)◦	6.249E−01 (4.9E−03)◦	1.168E+00 (1.8E−01)•	1.052E+00 (9.1E−02)•
DTLZ4	3	1.013E−01 (1.9E−01)	5.240E−02 (1.6E−06)◦	5.239E−02 (2.7E−04)◦	3.771E−01 (3.5E−01)‡	1.889E−01 (2.8E−01)‡
	5	1.694E−01 (1.1E−03)	1.660E−01 (1.4E−06)◦	1.657E−01 (1.1E−04)◦	4.051E−01 (4.7E−01)•	1.618E−01 (2.9E−04)◦
	8	3.712E−01 (3.0E−03)	3.398E−01 (4.6E−05)◦	3.398E−01 (6.3E−04)◦	5.508E−01 (1.1E−01)•	4.106E−01 (3.7E−02)•
	10	4.154E−01 (4.8E−03)	4.218E−01 (3.7E−04)•	4.209E−01 (6.8E−04)•	5.463E−01 (6.8E−02)•	4.480E−01 (9.9E−04)•
	15	5.992E−01 (2.2E−03)	6.176E−01 (2.2E−04)•	6.184E−01 (7.4E−04)•	7.198E−01 (4.0E−02)•	6.461E−01 (1.8E−02)•

• indicates that VaEA significantly improves the peer algorithm at a 0.05 level by the Wilcoxon's rank sum test, whereas ◦ indicates the opposite, i.e., the peer algorithm shows significant improvements over VaEA. If no significant difference is detected, it will be marked by the symbol "‡". They have the same meanings in other tables.

distribution, some extreme values for the second, the third and the sixth objectives are detected. As a result, VaEA has a larger IGD value than MOEA/D, MOEA/DD and NSGA-III do (because of its relatively poor convergence), but performs better than MOMBI2 whose solutions have an extremely worse distribution. For solutions of the fifteen-objective DTLZ4 (see 4), they perform similarly in terms of the convergence, i.e., values of each objective are within [0,1]. However, there indeed exit differences concerning the distribution. Solutions obtained by MOEA/D are unable to cover the third, eighth, ninth and thirteenth objectives well, and those obtained by MOMBI2 fail to cover the second objective. It seems that all values of the second objective of obtained solutions tend to zero. Compared with the solutions of MOEA/DD and NSGA-III, those found by VaEA are able to cover the whole Pareto front. It seems that solutions obtained by MOEA/DD and NSGA-III mainly concentre on the boundary, or on the middle parts of the front.

Table VII gives the median and IQR results on all the WFG test instances in terms of the IGD metric. As shown, VaEA and NSGA-III perform best, presenting a clear advantage over the other three algorithms on the majority of the test instances. More specifically, VaEA obtains the best and second best IGD results on 24 and 11 out of the 45 test instances respectively. And NSGA-III obtains 11 best results and 24 second best results. VaEA performs best on WFG3, WFG4, WFG5, WFG6, WFG7 and WFG9, while NSGA-III outperforms other algorithms on WFG2 and WFG8. For WFG3, MOEA/DD is a competitive algorithm to VaEA. On the three-objective WFG4 to WFG7, MOMBI2 performs best but struggles on higher numbers of objectives. Statistically, VaEA shows significant improvement over some other algorithms on most of the test instances. Specifically, the proportion of the test instances where VaEA performs better than MOEA/DD, NSGA-III MOEA/D and MOMBI2 is 36/45, 25/45, 42/45 and 33/45, respectively. Conversely, the proportion that VaEA is defeated by the peer algorithms is 7/45, 14/45, 2/45 and 10/45 for MOEA/DD, NSGA-III, MOEA/D and MOMBI2, respectively. It is demonstrated by the results that MOEA/DD and MOEA/D obtain relatively poor performance on this test suite compared with their counterparts. This may be due to the fact that the Pareto fronts of WFG test problems are irregular, being discontinued or mixed, and are scaled with different ranges in each objective. A set of well distributed weight vectors can not guarantee a good distribution of obtained solutions. For discontinued Pareto fronts, some weight vectors are not associated with any Pareto-optimal solution. Another reason accounting for the failure of MOEA/DD may be its lack of a normalization procedure before the evaluation of a solution. The reason for the better performance of VaEA on some WFG test problems may be that the *maximum-vector-angle-first* principle can not only ensure the width of the search region, but also dynamically adjust the search direction of the whole population. Therefore, unlike MOEA/DD or MOEA/D where the search directions are fixed by weight vectors, VaEA is not easy to be trapped into local optima, or to bias the search to certain regions of the true Pareto front, on biased problems such as WFG1 and WFG9, or on multimodal problems such as WFG4. What's more, the diversity and convergence in VaEA are kept balanced by introducing the *worse-elimination*

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

TABLE VII
MEDIAN AND IQR (IN BRACKETS) OF IGD METRIC ON ALL THE WFG TEST INSTANCES.

	m	VaEA	MOEA/DD	NSGA-III	MOEA/D	MOMBI2
WFG1	3	1.318E+00 (2.0E−01)	1.395E+00 (2.1E−01)●	1.282E+00 (1.5E−01)‡	1.191E+00 (1.3E−01)○	1.936E+00 (1.4E−01)●
	5	3.116E+00 (3.4E−01)	3.299E+00 (2.1E−01)●	2.943E+00 (4.3E−01)‡	3.687E+00 (1.5E−01)●	3.212E+00 (3.0E−01)‡
	8	5.139E+00 (1.3E−01)	6.355E+00 (3.4E−01)●	5.230E+00 (1.8E−01)‡	6.623E+00 (1.5E−01)●	3.904E+00 (1.4E+00)○
	10	7.226E+00 (4.4E−02)	9.201E+00 (4.0E−01)●	7.064E+00 (8.0E−02)○	9.637E+00 (2.2E−01)●	5.522E+00 (2.4E+00)○
	15	9.060E+00 (3.3E−01)	1.215E+01 (6.4E−01)●	9.082E+00 (1.7E−01)‡	1.241E+01 (2.6E−01)●	1.193E+01 (1.3E+00)●
WFG2	3	3.156E−01 (8.0E−02)	8.684E−01 (1.3E−02)●	1.846E−01 (1.4E−01)○	1.471E+00 (8.0E−02)●	2.743E−01 (1.3E−01)○
	5	8.134E−01 (2.8E−01)	3.628E+00 (3.8E−01)●	5.936E−01 (1.5E−01)○	3.971E+00 (2.2E−01)●	9.510E−01 (5.1E−01)○
	8	1.989E+00 (2.6E−01)	8.619E+00 (3.8E−01)●	2.865E+00 (1.2E+00)●	8.806E+00 (5.8E−01)●	3.406E+00 (1.7E+00)●
	10	3.670E+00 (4.8E−01)	1.264E+01 (4.6E−02)●	2.923E+00 (5.1E−01)○	1.131E+01 (1.2E+00)●	7.671E+00 (3.2E+00)●
	15	5.250E+00 (6.4E−01)	1.384E+01 (6.2E−02)●	6.223E+00 (2.0E+00)●	1.346E+01 (3.4E−01)●	1.365E+01 (1.2E+00)●
WFG3	3	1.300E−01 (7.9E−03)	1.708E−01 (6.0E−03)●	3.867E−01 (7.6E−02)●	1.817E−01 (2.9E−02)●	1.868E−01 (2.6E−02)●
	5	4.871E−01 (7.5E−03)	5.319E−01 (3.0E−02)●	5.215E−01 (8.2E−03)●	5.096E−01 (3.8E−02)●	1.178E+00 (1.3E−01)●
	8	1.315E+00 (2.0E−02)	1.216E+00 (3.2E−02)○	1.709E+00 (2.0E−01)●	1.789E+00 (3.1E−02)●	1.191E+01 (3.1E+00)●
	10	1.674E+00 (1.8E−02)	1.651E+00 (2.3E−02)‡	2.223E+00 (2.3E−01)●	2.560E+00 (4.9E−02)●	6.186E+00 (5.8E+00)●
	15	2.758E+00 (3.3E−01)	3.024E+00 (9.7E−02)●	4.155E+00 (1.8E−01)●	5.167E+00 (9.0E−02)●	5.825E+00 (8.4E−01)●
WFG4	3	2.296E−01 (3.7E−03)	2.280E−01 (4.3E−04)○	2.119E−01 (4.3E−04)○	2.375E−01 (2.9E−03)●	2.109E−01 (1.5E−03)○
	5	9.447E−01 (7.5E−03)	1.009E+00 (1.1E−03)●	9.883E−01 (4.5E−03)●	1.423E+00 (9.4E−02)●	9.869E−01 (3.0E−03)●
	8	3.023E+00 (2.6E−02)	3.945E+00 (1.2E−01)●	3.175E+00 (8.7E−03)●	6.703E+00 (2.6E−01)●	4.191E+00 (5.3E−01)●
	10	3.982E+00 (2.5E−02)	6.397E+00 (1.6E−01)●	4.512E+00 (1.1E−02)●	9.146E+00 (2.6E−01)●	5.559E+00 (1.0E+00)●
	15	8.387E+00 (1.0E−01)	1.349E+01 (4.4E+00)●	9.871E+00 (2.2E−02)●	1.519E+01 (8.4E−03)●	1.541E+01 (3.0E+00)●
WFG5	3	2.419E−01 (3.8E−03)	2.362E−01 (8.5E−04)○	2.229E−01 (6.3E−03)○	2.423E−01 (1.7E−03)‡	2.226E−01 (2.4E−03)○
	5	9.501E−01 (1.2E−02)	9.904E−01 (1.3E−03)●	9.780E−01 (5.4E−03)●	1.184E+00 (6.3E−02)●	9.632E−01 (3.1E−03)●
	8	3.112E+00 (2.6E−02)	4.136E+00 (1.6E−01)●	3.151E+00 (6.5E−03)●	5.886E+00 (1.9E−01)●	3.794E+00 (5.5E−02)●
	10	3.973E+00 (2.3E−02)	6.444E+00 (1.2E−01)●	4.515E+00 (1.1E−02)●	8.747E+00 (2.1E−01)●	4.818E+00 (6.7E−02)●
	15	8.292E+00 (6.2E−02)	1.401E+01 (4.8E−02)●	9.872E+00 (1.5E−02)●	1.511E+01 (4.9E−03)●	9.973E+00 (1.3E−01)●
WFG6	3	2.467E−01 (6.5E−03)	2.369E−01 (3.4E−03)○	2.207E−01 (4.1E−03)○	2.516E−01 (9.8E−03)●	2.164E−01 (3.3E−03)○
	5	9.739E−01 (1.6E−02)	1.002E+00 (9.5E−04)●	9.843E−01 (2.5E−03)●	1.626E+00 (1.4E−01)●	9.659E−01 (1.9E−03)○
	8	3.264E+00 (5.9E−02)	3.923E+00 (1.7E−01)●	3.165E+00 (6.3E−03)○	7.425E+00 (3.1E−01)●	3.728E+00 (1.2E−02)●
	10	4.056E+00 (3.6E−02)	6.301E+00 (1.7E−01)●	4.528E+00 (6.7E−03)●	9.517E+00 (2.7E−01)●	4.750E+00 (2.8E−02)●
	15	8.316E+00 (9.9E−02)	1.400E+01 (5.6E−02)●	9.887E+00 (1.9E−02)●	1.640E+01 (5.2E−02)●	1.042E+01 (6.0E−01)●
WFG7	3	2.274E−01 (5.3E−03)	2.284E−01 (1.1E−04)‡	2.119E−01 (2.6E−04)○	2.676E−01 (2.8E−02)●	2.091E−01 (9.1E−04)○
	5	9.642E−01 (1.3E−02)	1.011E+00 (6.1E−04)●	9.983E−01 (1.5E−03)●	1.610E+00 (8.9E−02)●	9.791E−01 (2.4E−03)●
	8	3.190E+00 (3.9E−02)	3.699E+00 (1.6E−01)●	3.194E+00 (1.2E−02)‡	7.122E+00 (2.6E−01)●	3.796E+00 (4.5E−02)●
	10	4.007E+00 (2.9E−02)	5.817E+00 (9.5E−02)●	4.537E+00 (6.7E−03)●	9.419E+00 (1.4E−01)●	4.818E+00 (5.6E−02)●
	15	8.330E+00 (9.0E−02)	1.336E+01 (1.4E+00)●	9.910E+00 (3.2E−02)●	1.633E+01 (1.1E−01)●	1.133E+01 (9.6E−01)●
WFG8	3	3.208E−01 (8.8E−03)	2.823E−01 (4.8E−03)○	2.790E−01 (4.0E−03)○	2.959E−01 (7.5E−03)○	3.193E−01 (6.2E−03)‡
	5	1.101E+00 (1.8E−02)	1.030E+00 (1.4E−03)○	1.005E+00 (7.4E−03)○	1.171E+00 (2.9E−01)●	1.327E+00 (4.0E−02)●
	8	3.289E+00 (1.9E−02)	3.895E+00 (3.8E−01)●	3.284E+00 (2.7E−02)‡	6.145E+00 (2.7E−01)●	4.473E+00 (2.2E−01)●
	10	4.331E+00 (2.2E−02)	6.009E+00 (3.7E−01)●	4.307E+00 (9.4E−02)○	8.365E+00 (4.6E−01)●	6.445E+00 (8.4E−01)●
	15	8.964E+00 (1.3E−01)	1.335E+01 (1.2E+00)●	9.594E+00 (5.5E−01)●	1.431E+01 (1.1E+00)●	1.271E+01 (1.4E+00)●
WFG9	3	2.772E−01 (2.4E−02)	2.320E−01 (4.1E−02)○	2.149E−01 (4.9E−02)○	2.825E−01 (3.0E−02)●	2.621E−01 (4.5E−02)○
	5	9.877E−01 (1.2E−02)	1.005E+00 (6.4E−03)●	9.318E−01 (8.9E−03)○	1.221E+00 (9.9E−02)●	1.048E+00 (2.1E−02)●
	8	3.008E+00 (3.0E−02)	3.552E+00 (1.6E−01)●	3.141E+00 (2.0E−02)●	6.187E+00 (1.3E−01)●	4.412E+00 (1.1E−01)●
	10	3.970E+00 (3.9E−02)	5.613E+00 (3.3E−01)●	4.315E+00 (6.6E−02)●	8.445E+00 (6.7E−01)●	5.837E+00 (1.4E−01)●
	15	8.239E+00 (9.3E−02)	1.064E+01 (9.0E−01)●	9.147E+00 (8.0E−02)●	1.548E+01 (9.1E−01)●	1.118E+01 (6.3E−01)●

principle.

The HV results of WFG test instances are given in Table VIII. It can be seen from the table that VaEA performs best on WFG2, WFG3 and WFG9, while NSGA-III outperforms other algorithms on WFG5, WFG7 and WFG8. For MOMBI2, it is competitive on some test instances, such as 3- and 5-objective WFG3 and WFG4, etc. For MOEA/DD and MOEA/D, they obtain worse HV values on almost all the test instances, except for WFG1 with three and five objectives. Note that for problems with complicated fronts (such as WFG1 and WFG2), and with higher dimensional objective space (such as 15-objective WFG3-4 and WFG6-9), VaEA obtains the best performance in terms of the HV metric.

Take ten-objective WFG9 and fifteen-objective WFG1 as examples, the final solutions obtained by all the algorithms are shown in Figs. 5 and 6. It is clear from Fig. 5 that solutions of VaEA are with the highest quality in terms of both the convergence and distribution. For NSGA-III and MOMBI2,

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

TABLE VIII
MEDIAN AND IQR (IN BRACKETS) OF HV METRIC ON ALL THE WFG TEST INSTANCES.

	m	VaEA	MOEA/DD	NSGA-III	MOEA/D	MOMBI2
WFG1	3	8.927E-01 (4.9E-02)	9.577E-01 (4.6E-02)○	8.551E-01 (5.3E-02)●	8.689E-01 (1.1E-01)‡	7.930E-01 (5.3E-02)●
	5	8.494E-01 (7.9E-02)	1.191E+00 (4.0E-02)○	8.207E-01 (8.2E-02)‡	1.349E+00 (4.1E-02)○	1.280E+00 (1.6E-02)○
	8	1.604E+00 (5.6E-02)	1.519E+00 (8.6E-02)●	1.455E+00 (6.9E-02)●	1.444E+00 (1.4E-01)●	1.666E+00 (1.2E-01)○
	10	2.206E+00 (3.1E-02)	2.161E+00 (5.7E-02)●	2.157E+00 (3.0E-02)●	1.563E+00 (2.0E-01)●	2.293E+00 (9.0E-02)○
	15	3.640E+00 (3.1E-02)	3.008E+00 (2.6E-01)●	3.629E+00 (7.3E-02)‡	1.404E+00 (1.3E-01)●	3.336E+00 (4.4E-01)●
WFG2	3	1.217E+00 (1.8E-01)	1.227E+00 (5.7E-03)‡	1.231E+00 (1.8E-01)‡	1.001E+00 (2.7E-02)●	1.047E+00 (1.8E-01)●
	5	1.550E+00 (8.4E-02)	1.491E+00 (1.2E-01)●	1.565E+00 (8.5E-02)○	1.326E+00 (1.0E-01)●	1.478E+00 (1.4E-01)‡
	8	2.069E+00 (1.1E-01)	1.973E+00 (1.0E-01)●	1.969E+00 (1.9E-01)●	1.645E+00 (1.3E-01)●	1.833E+00 (1.6E-01)●
	10	2.561E+00 (4.2E-03)	2.459E+00 (2.1E-02)●	2.483E+00 (1.8E-01)‡	2.078E+00 (1.4E-01)●	2.479E+00 (1.9E-01)‡
	15	4.016E+00 (2.6E-01)	3.834E+00 (1.9E-01)●	3.744E+00 (3.7E-01)●	3.341E+00 (2.8E-01)●	3.723E+00 (3.4E-01)●
WFG3	3	8.150E-01 (7.4E-03)	7.861E-01 (1.3E-02)●	8.288E-01 (6.3E-03)○	7.638E-01 (2.5E-02)●	8.353E-01 (1.2E-02)○
	5	9.903E-01 (1.3E-02)	9.571E-01 (1.4E-02)●	1.008E+00 (1.0E-02)○	9.053E-01 (5.5E-02)●	1.029E+00 (1.5E-02)○
	8	1.254E+00 (5.3E-02)	1.094E+00 (3.3E-02)●	1.191E+00 (1.3E-01)●	7.103E-01 (4.8E-02)●	3.975E-01 (2.8E-01)●
	10	1.521E+00 (7.3E-02)	1.284E+00 (2.3E-02)●	1.509E+00 (1.4E-01)‡	5.299E-01 (6.3E-02)●	1.214E+00 (4.5E-01)●
	15	2.635E+00 (4.7E-02)	1.503E+00 (5.9E-02)●	2.295E+00 (1.1E-01)●	5.582E-01 (1.3E-02)●	1.454E+00 (8.9E-02)●
WFG4	3	7.151E-01 (6.7E-03)	7.257E-01 (1.6E-03)○	7.331E-01 (2.2E-03)○	7.103E-01 (5.8E-03)●	7.349E-01 (1.7E-03)○
	5	1.183E+00 (7.0E-03)	1.255E+00 (2.2E-03)○	1.263E+00 (5.4E-03)○	1.155E+00 (2.9E-02)●	1.279E+00 (2.5E-02)○
	8	1.883E+00 (1.2E-02)○	1.764E+00 (2.1E-02)●	1.768E+00 (2.0E-02)●	9.645E-01 (1.2E-01)●	1.741E+00 (1.3E-01)●
	10	2.225E+00 (2.2E-02)	2.117E+00 (3.3E-02)●	2.414E+00 (1.1E-02)○	1.026E+00 (1.1E-01)●	2.254E+00 (1.6E-01)○
	15	4.057E+00 (1.8E-02)	3.087E+00 (5.5E-01)●	3.635E+00 (5.5E-02)●	1.319E+00 (1.6E-01)●	2.910E+00 (2.7E-01)●
WFG5	3	6.866E-01 (6.2E-03)	6.707E-01 (5.7E-03)●	6.811E-01 (6.5E-03)●	6.628E-01 (5.7E-03)●	6.759E-01 (4.3E-03)●
	5	1.161E+00 (5.6E-03)	1.168E+00 (3.6E-03)○	1.210E+00 (3.4E-03)○	1.132E+00 (1.4E-02)●	1.203E+00 (2.6E-03)○
	8	1.741E+00 (1.6E-02)	1.572E+00 (2.9E-02)●	1.816E+00 (5.2E-03)○	1.126E+00 (5.9E-02)●	1.800E+00 (1.7E-02)○
	10	2.193E+00 (1.4E-02)	1.891E+00 (2.6E-02)●	2.314E+00 (5.7E-03)○	1.161E+00 (9.8E-02)●	2.282E+00 (1.4E-02)○
	15	3.540E+00 (1.8E-02)	2.135E+00 (1.4E-01)●	3.821E+00 (2.0E-03)○	1.159E+00 (1.2E-01)●	3.585E+00 (1.1E-01)○
WFG6	3	6.746E-01 (1.3E-02)	6.814E-01 (1.0E-02)○	6.886E-01 (9.4E-03)○	6.662E-01 (1.7E-02)●	6.898E-01 (9.6E-03)○
	5	1.150E+00 (1.3E-02)	1.193E+00 (1.1E-02)○	1.215E+00 (8.9E-03)○	9.470E-01 (1.5E-01)●	1.216E+00 (9.7E-03)○
	8	1.792E+00 (2.3E-02)	1.693E+00 (2.7E-02)●	1.849E+00 (1.6E-02)○	6.076E-01 (7.5E-02)●	1.859E+00 (2.2E-02)○
	10	2.345E+00 (1.9E-02)	2.018E+00 (4.1E-02)●	2.326E+00 (2.9E-02)●	7.292E-01 (1.1E-01)●	2.247E+00 (2.6E-02)●
	15	3.876E+00 (5.1E-02)	2.312E+00 (2.0E-01)●	3.665E+00 (4.8E-02)●	5.378E-01 (1.8E-01)●	3.557E+00 (1.3E-01)●
WFG7	3	7.258E-01 (4.1E-03)	7.308E-01 (1.2E-03)○	7.400E-01 (1.3E-03)○	6.994E-01 (1.8E-02)●	7.345E-01 (1.4E-03)○
	5	1.235E+00 (5.7E-03)	1.265E+00 (2.4E-03)○	1.293E+00 (1.5E-03)○	1.091E+00 (2.9E-02)●	1.285E+00 (1.4E-03)○
	8	1.895E+00 (7.3E-03)	1.822E+00 (2.3E-02)●	1.943E+00 (3.1E-03)○	7.624E-01 (5.7E-02)●	1.936E+00 (5.1E-02)○
	10	2.388E+00 (1.2E-02)	2.258E+00 (9.2E-03)●	2.481E+00 (3.0E-03)○	9.015E-01 (4.5E-02)●	2.420E+00 (2.8E-02)○
	15	4.118E+00 (5.3E-03)	3.274E+00 (3.4E-01)●	3.866E+00 (2.5E-02)●	6.885E-01 (1.4E-01)●	3.621E+00 (1.8E-01)●
WFG8	3	5.998E-01 (6.3E-03)	6.259E-01 (2.9E-03)○	6.326E-01 (2.8E-03)○	6.123E-01 (6.8E-03)○	6.120E-01 (5.7E-03)○
	5	1.000E+00 (1.1E-02)	1.097E+00 (4.9E-03)○	1.103E+00 (4.0E-03)○	9.239E-01 (1.7E-01)‡	1.072E+00 (3.5E-03)○
	8	1.449E+00 (2.4E-02)	1.482E+00 (1.0E-01)‡	1.624E+00 (1.3E-02)○	1.552E-01 (9.8E-02)●	1.548E+00 (4.8E-02)○
	10	1.911E+00 (2.9E-02)	1.901E+00 (1.4E-01)‡	2.140E+00 (2.1E-02)○	1.857E-01 (1.7E-01)●	2.004E+00 (1.3E-01)○
	15	3.689E+00 (9.0E-02)	2.205E+00 (3.3E-01)●	3.407E+00 (5.1E-02)●	8.324E-01 (9.8E-01)●	3.258E+00 (2.9E-01)●
WFG9	3	6.119E-01 (3.4E-02)	6.722E-01 (7.2E-02)○	6.743E-01 (7.3E-02)○	5.949E-01 (4.3E-02)●	6.163E-01 (8.4E-02)○
	5	1.031E+00 (7.3E-03)	1.090E+00 (3.8E-02)○	1.092E+00 (4.1E-02)○	1.027E+00 (3.3E-02)‡	1.056E+00 (4.9E-03)○
	8	1.548E+00 (3.9E-02)	1.379E+00 (5.7E-02)●	1.492E+00 (2.6E-02)●	6.850E-01 (1.7E-01)●	1.364E+00 (5.2E-02)●
	10	1.985E+00 (7.5E-02)	1.643E+00 (5.2E-02)●	1.871E+00 (4.5E-02)●	7.992E-01 (1.3E-01)●	1.790E+00 (6.6E-02)●
	15	3.243E+00 (1.3E-01)	1.904E+00 (2.1E-01)●	2.867E+00 (1.2E-01)●	5.521E-01 (4.5E-01)●	2.508E+00 (1.7E-01)●

their solutions are distributed similarly: the middle parts of each objective are not well covered. The final solutions found by MOEA/DD fail to sufficiently cover the top half of the sixth to tenth objectives. The solutions obtained by MOEA/D concentre only on a small parts of the true Pareto fronts, resulting in the largest (or smallest) IGD (or HV) values. WFG1 is a hard problem due to its mixed and biased Pareto optimal front. For the fifteen-objective WFG1, all algorithms encounter some difficulties when solving this problem (see Fig. 6). The solutions obtained by MOEA/DD and MOEA/D converge into a small middle part of the Pareto front, while those obtained by NSGA-III and MOMBI2 struggle to cover a wider region in some objectives, e.g, the 2nd to the 11th objectives. For VaEA, much better boundary values are found for each objective, which may be attributed to the survival of extreme solutions in the *Association* operation in Algorithm 4.

Table IX shows the results concerning the Δ metric on 5-

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

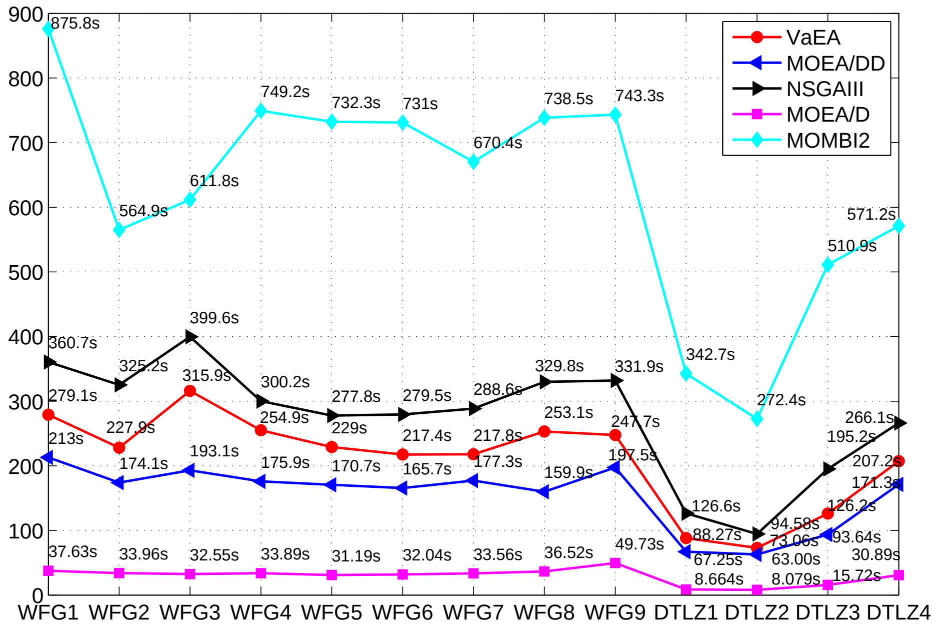


Fig. 9. The running time of all the algorithms on the ten-objective test problems.

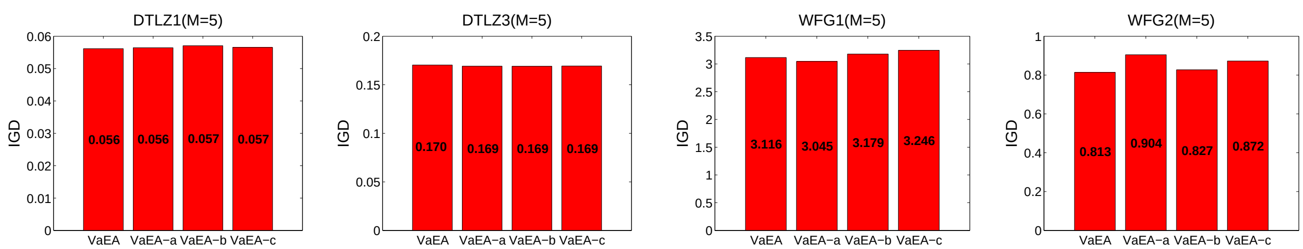


Fig. 10. The comparison between VaEA and its three variants on five-objective DTLZ1, DTLZ3, WFG1 and WFG2. VaEA-a: the algorithm adding only extreme solutions; VaEA-b: the algorithm adding only better converged solutions; VaEA-c: the algorithm adding random solutions.

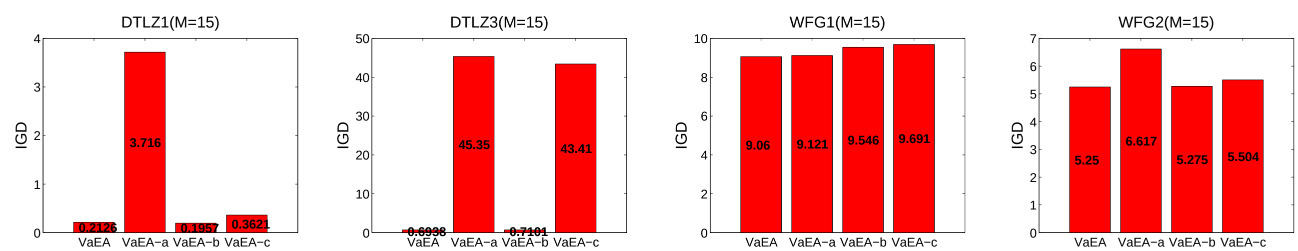


Fig. 11. The comparison between VaEA and its three variants on fifteen-objective DTLZ1, DTLZ3, WFG1 and WFG2. VaEA-a: the algorithm adding only extreme solutions; VaEA-b: the algorithm adding only better converged solutions; VaEA-c: the algorithm adding random solutions.

, 10- and 15-objective DTLZ and WFG test problems. As seen, VaEA obtains significantly better performance than all the peer algorithms on almost all the tDest instances, except for DTLZ1, DTLZ3, 5-objective WFG1 and 5-objective WFG2. On DTLZ1 and DTLZ3, MOEA/DD performs best. For WFG1 and WFG2 with five objectives, MOEA/D and NSGA-III are two most effective algorithms. Good distributions (in terms of both uniformness and extensiveness) of solutions obtained by VaEA are reflected by Figs. 3, 4, 5 and 6, and the better performance of VaEA in terms of the Δ metric is attributed to the *maximum-vector-angle-first* principle which dynamically adds solutions, and emphasizes solutions locating at the most sparsest region in terms of the vector angles.

E. Comparison of Evolutionary Trend and Running Time

In this section, evolutionary trends of selected algorithms are plotted in Fig. 7, where all the fifteen-objective WFG test problems are considered. In these figures, the *x*-coordinate is

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

TABLE IX
MEDIAN AND IQR (IN BRACKETS) OF Δ METRIC ON ALL THE 5-, 10- AND 15-OBJECTIVE TEST INSTANCES.

	m	VaEA	MOEA/DD	NSGA-III	MOEA/D	MOMBI2
DTLZ1	5	2.300E−01 (6.8E−02)	1.458E−03 (3.9E−04)∘	3.852E−02 (5.0E−03)∘	2.717E−03 (4.6E−03)∘	2.999E−01 (5.1E−01)‡
	10	3.380E−01 (4.5E−01)	5.664E−02 (3.5E−03)∘	6.359E−02 (6.5E−03)∘	6.154E−02 (6.1E−03)∘	1.036E+00 (1.5E−01)•
	15	8.765E−01 (4.8E−01)	1.636E−02 (4.4E−03)∘	3.819E−02 (6.3E−03)∘	1.579E−01 (3.7E−02)∘	1.004E+00 (1.0E−02)•
DTLZ2	5	1.131E−01 (7.8E−03)	1.474E−01 (2.0E−04)•	1.644E−01 (7.0E−03)•	1.474E−01 (4.4E−04)•	1.559E−01 (1.2E−02)•
	10	1.074E−01 (7.3E−03)	1.456E−01 (2.3E−03)•	1.517E−01 (2.5E−03)•	1.466E−01 (1.0E−03)•	3.504E−01 (2.6E−02)•
	15	2.425E−01 (1.4E−02)	2.890E−01 (4.3E−03)•	3.071E−01 (2.3E−02)•	2.849E−01 (3.4E−03)•	1.011E+00 (2.4E−02)•
DTLZ3	5	1.586E−01 (3.2E−02)	1.476E−01 (2.5E−04)∘	1.668E−01 (4.9E−03)‡	1.476E−01 (2.7E−04)∘	8.107E−01 (1.9E−01)•
	10	3.294E−01 (2.3E−01)	1.463E−01 (2.5E−03)∘	1.516E−01 (6.4E−03)∘	1.487E−01 (8.7E−03)∘	1.038E+00 (1.2E−01)•
	15	1.093E+00 (7.3E−01)	2.901E−01 (4.6E−03)∘	3.085E−01 (2.0E−02)∘	1.061E+00 (2.3E−02)‡	1.005E+00 (9.8E−03)‡
DTLZ4	5	1.168E−01 (1.0E−02)	1.475E−01 (3.2E−05)•	1.631E−01 (5.3E−03)•	8.377E−01 (1.1E+00)•	1.476E−01 (2.1E−03)•
	10	1.079E−01 (6.0E−03)	1.470E−01 (1.8E−03)•	1.505E−01 (3.7E−03)•	7.172E−01 (4.7E−01)•	3.464E−01 (9.3E−03)•
	15	2.497E−01 (3.7E−02)	2.816E−01 (1.3E−03)•	2.973E−01 (7.5E−03)•	1.033E+00 (2.3E−01)•	6.003E−01 (2.7E−01)•
WFG1	5	5.078E−01 (4.0E−02)	4.552E−01 (3.1E−02)∘	5.714E−01 (5.4E−02)•	4.414E−01 (3.0E−02)∘	8.812E−01 (5.6E−02)•
	10	6.928E−01 (1.3E−02)	8.088E−01 (1.5E−02)•	7.210E−01 (2.6E−02)•	7.850E−01 (4.3E−02)•	9.936E−01 (1.5E−02)•
	15	9.058E−01 (9.0E−03)	9.593E−01 (3.9E−03)•	9.062E−01 (1.5E−02)‡	9.802E−01 (8.6E−03)•	9.949E−01 (6.4E−03)•
WFG2	5	4.070E−01 (3.6E−02)	3.478E−01 (4.4E−02)∘	2.199E−01 (1.5E−02)∘	4.785E−01 (2.5E−02)•	3.729E−01 (6.4E−02)∘
	10	4.766E−01 (2.1E−02)	6.358E−01 (7.5E−03)•	6.256E−01 (7.3E−02)•	8.080E−01 (2.1E−02)•	8.920E−01 (8.0E−02)•
	15	7.465E−01 (3.2E−02)	9.584E−01 (4.7E−03)•	8.973E−01 (7.1E−02)•	9.882E−01 (5.2E−02)•	9.886E−01 (9.1E−03)•
WFG3	5	2.197E−01 (8.6E−03)	3.251E−01 (3.0E−02)•	5.579E−01 (4.2E−02)•	3.885E−01 (3.5E−02)•	1.108E+00 (5.7E−02)•
	10	1.571E−01 (6.0E−03)	4.702E−01 (4.3E−02)•	7.415E−01 (4.0E−02)•	6.189E−01 (7.5E−03)•	1.025E+00 (4.9E−02)•
	15	3.271E−01 (1.5E−01)	7.143E−01 (2.4E−02)•	9.903E−01 (9.7E−02)•	1.125E+00 (8.0E−03)•	9.995E−01 (3.6E−02)•
WFG4	5	1.091E−01 (8.1E−03)	3.174E−01 (2.0E−03)•	1.640E−01 (7.4E−03)•	3.618E−01 (3.4E−02)•	1.584E−01 (7.4E−03)•
	10	9.757E−02 (4.9E−03)	4.829E−01 (1.5E−02)•	1.521E−01 (4.2E−03)•	7.238E−01 (4.9E−02)•	1.226E+00 (9.8E−02)•
	15	2.488E−01 (1.4E−02)	7.411E−01 (1.5E−01)•	3.017E−01 (1.7E−02)•	7.780E−01 (6.0E−03)•	1.024E+00 (3.9E−02)•
WFG5	5	1.112E−01 (1.2E−02)	3.205E−01 (2.0E−03)•	1.677E−01 (1.0E−02)•	3.563E−01 (1.4E−02)•	1.573E−01 (7.9E−03)•
	10	9.863E−02 (3.3E−03)	4.890E−01 (1.6E−02)•	1.550E−01 (2.9E−03)•	7.411E−01 (2.4E−02)•	5.011E−01 (1.2E−01)•
	15	2.438E−01 (7.5E−03)	7.526E−01 (1.7E−02)•	3.026E−01 (1.9E−02)•	7.705E−01 (1.2E−02)•	7.362E−01 (4.1E−01)•
WFG6	5	1.107E−01 (6.5E−03)	3.187E−01 (1.5E−03)•	1.667E−01 (3.5E−03)•	4.211E−01 (8.7E−02)•	1.506E−01 (5.8E−03)•
	10	1.010E−01 (8.1E−03)	4.789E−01 (1.2E−02)•	1.560E−01 (4.3E−03)•	6.893E−01 (3.8E−02)•	3.521E−01 (1.3E−01)•
	15	2.394E−01 (1.3E−03)	7.529E−01 (2.3E−02)•	3.088E−01 (2.1E−02)•	7.706E−01 (6.9E−03)•	1.046E+00 (2.8E−01)•
WFG7	5	1.083E−01 (9.3E−03)	3.161E−01 (1.1E−03)•	1.632E−01 (4.8E−03)•	3.926E−01 (4.6E−02)•	1.536E−01 (7.1E−03)•
	10	1.029E−01 (6.0E−03)	4.681E−01 (1.3E−02)•	1.583E−01 (4.6E−03)•	6.849E−01 (2.5E−02)•	1.018E+00 (3.7E−01)•
	15	2.830E−01 (4.2E−02)	7.565E−01 (8.5E−02)•	3.102E−01 (2.0E−02)•	7.439E−01 (1.4E−02)•	1.137E+00 (1.1E−01)•
WFG8	5	1.253E−01 (9.2E−03)	3.595E−01 (8.5E−03)•	2.453E−01 (1.9E−02)•	3.913E−01 (1.7E−02)•	8.222E−01 (6.3E−02)•
	10	1.483E−01 (9.0E−03)	4.746E−01 (2.1E−02)•	3.052E−01 (6.1E−02)•	1.009E+00 (4.9E−01)•	1.165E+00 (3.1E−01)•
	15	2.805E−01 (1.4E−02)	7.796E−01 (1.9E−02)•	4.121E−01 (3.2E−01)•	1.053E+00 (2.8E−01)•	1.028E+00 (5.1E−02)•
WFG9	5	1.111E−01 (9.9E−03)	3.665E−01 (9.2E−03)•	1.884E−01 (1.1E−02)•	3.940E−01 (3.7E−02)•	6.088E−01 (6.8E−02)•
	10	1.025E−01 (4.1E−03)	4.966E−01 (2.2E−02)•	2.087E−01 (6.0E−02)•	1.158E+00 (9.1E−02)•	1.100E+00 (1.0E−01)•
	15	2.543E−01 (2.8E−03)	8.667E−01 (3.4E−01)•	3.040E−01 (3.7E−02)•	1.151E+00 (2.5E−02)•	1.045E+00 (1.2E−01)•

the number of function evaluations, while the *y*-coordinate is the value of IGD metric. It is observed from Fig. 7 that VaEA performs better than other algorithms on almost all the test problems during the whole evolutionary process. For WFG8, VaEA and NSGA-III obtain very close performance. Intuitively, VaEA and NSGA-III are better than MOEA/DD and MOEA/D. For WFG2, larger fluctuations of the values of IGD are found by all algorithms, which may be due to the discontinuity of the Pareto-optimal front. It is interesting to find that the trajectory line of MOEA/DD rises after a certain number of function evaluations on almost all the test problems, except for WFG9. This phenomenon may be attributed to the mechanism adopted in MOEA/DD that solutions belonging to the worst non-domination level but being associated with an isolated subregion, are deemed to be important for population diversity and are preserved without reservation [11] (see Fig. 8). These solutions may be far away from the Pareto-optimal front. Once they have been found, as discussed in [11], they would definitely do harm to the convergence of the population, resulting in a rise of IGD value.

To evaluate the speed of all the considered algorithms, we record the actual running time (in seconds: s) of each algorithm on each test problem with 10 objectives. The hardware configurations are as follows: Intel (R) Core (TM) i5-5200U CPU 2.20 GHz (processor), 8.00 GB (RAM). All algorithms are implemented by JAVA codes, and the experimental results

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

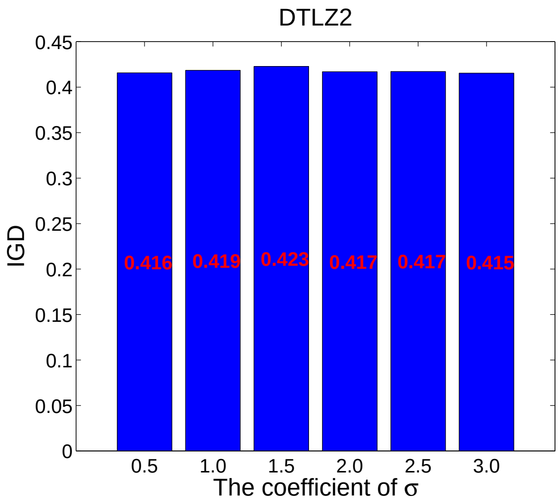


Fig. 12. The comparison of IGD values for different coefficients of σ on the ten-objective DTLZ2 test problem.

are shown in Fig. 9. Clearly, MOEA/D and MOMBI2 are the fastest and slowest algorithms, respectively. For VaEA, it is faster than NSGA-III, but slower than MOEA/DD. Although VaEA, NSGA-III and MOEA/DD have the same worst time complexity $O(mN^2)$, they behave differently in terms of the actual running speed. For the pairwise comparison between VaEA and MOEA/DD, both of them divide the union population into different layers, however, MOEA/DD uses an efficient non-domination level update method [47] to renew the non-domination level structure which saves the running time to some extent. Compare with NSGA-III, VaEA utilizes a simpler normalization procedure. In fact, the normalization in NSGA-III needs to determine intercepts of a hyperplane which involves solving a system of linear equations [8]. Hence, VaEA runs relatively faster than NSGA-III. Similar observations are found for other numbers of objectives.

F. Further Discussions

Recall that m extreme solutions and the first m best solutions in terms of the fitness value (or better converged solutions) are preferentially added if the population P is empty (see Algorithm 4). Does the inclusion of these types of solutions really contribute to the performance improvement of VaEA? To answer this question, we carry out the following experiment to compare the performance of VaEA with its three variants: the algorithm adding only extreme solutions (denoted by VaEA-a), the algorithm adding only better converged solutions (denoted by VaEA-b), and the algorithm adding m random solutions (denoted by VaEA-c). The IGD values on five-objective DTLZ1, DTLZ3, WFG1 and WFG2 are shown in Fig. 10. The main reason for choosing the above four test problems are that they are representatives of DTLZ and WFG test suites, and are deemed to be hard problems. As shown in Fig. 10, the differences among all the algorithms are really tiny on DTLZ1 and DTLZ3. For WFG1, the best algorithm is VaEA-a, while VaEA performs best on WFG2. Generally speaking, the four algorithms perform competitively on the five-objective test problems.

Now, we consider fifteen-objective test problems, and the results are shown in Fig. 11. It can be found that the performance of VaEA-a seriously degenerates on DTLZ1 and DTLZ3, indicating that adding only extreme solutions is not enough for the performance improvements. Similarly, the inclusion of random solutions can also lead to the deterioration of the performance (see VaEA-c on DTLZ3). For VaEA-b, it obtains slightly better IGD than VaEA on DTLZ1, but performs worse on DTLZ3, WFG1 and WFG2. From the analysis above, the algorithm adding both extreme and better converged solutions (i.e., VaEA) is able to provide better overall performance, indicating the usefulness of the joint addition of two types of solutions.

Finally, we investigate the condition in the *worse-elimination* principle. As described in Section II-B4, if the angle between a solution and its target solution is smaller than $\sigma = \frac{\pi/2}{N+1}$, then a worse solution may be replaced according to the *worse-elimination* principle. In the previous experimental studies, the threshold is fixed to $\sigma = \frac{\pi/2}{N+1}$. In this section, we multiply σ by a coefficient, such as 0.5, 1.0, 1.5, 2.0, 2.5 and 3.0, and run the VaEA algorithm under each configuration. Take ten-objective DTLZ2 as an example, the IGD values for different coefficients are given in Fig. 12, where we find no significant differences among all the considered coefficients, which implies that VaEA is not sensitive to the settings of the condition in the *worse-elimination* principle.

IV. Practical Applications in Two Problems from Practice

In this part, VaEA is compared with the two most competitive algorithms MOEA/DD and NSGA-III on two engineering problems. The first problem has three objectives and the second one has nine objectives.

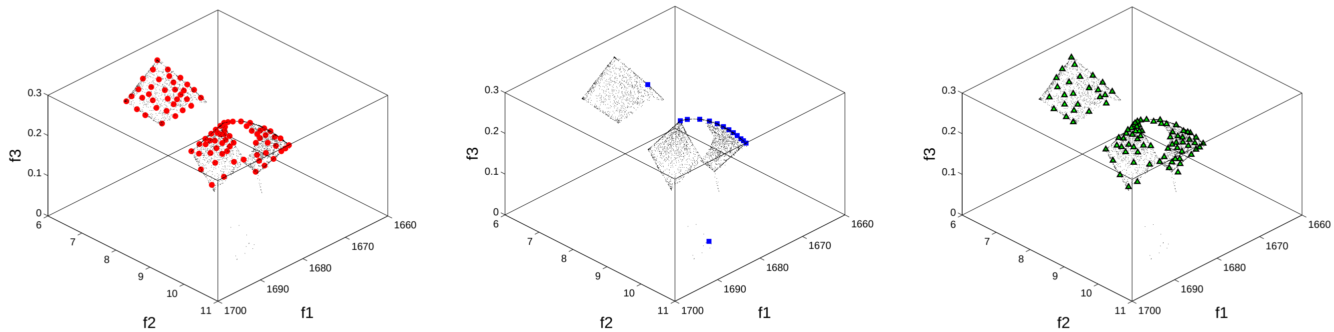
A. Problem Description

1) Crashworthiness Design of Vehicles: This problem aims at the structural optimization for crashworthiness criteria in automotive industry [8]. The thickness of five reinforced members around the frontal structure is chosen as the design variables (denoted as $t_i, i = 1, 2, \ldots, 5$) which could significantly affect the crash safety. The mass of vehicle (*Mass*), the integration of collision acceleration (A_{in}) and the toe board intrusion in the "offset-frontal crash" (*Intrusion*) are taken as objectives. For each variable, the upper and lower boundaries are 1*mm* and 3*mm*, respectively. Mathematical formulation for the three objectives is available in the original study [62].

2) Car Side-Impact Problem: The original car side-impact problem described in [63] and [64] is a single objective optimization problem with 10 constraints (g_1 to g_{10}). There are seven decision variables (x_1 to x_7) and four stochastic parameters x_8 and x_{11}. In this paper, each constraint is converted to an objective by minimizing the constraint violation value. Thus, there are 11 objectives in total (10 converted objectives as well as the original objective function f). As suggested in [63], g_5, g_6 and g_7 can be combined by taking their average value. Therefore, the number of objectives reduces to 9 in the new version of the car side-impact problem. A mathematical

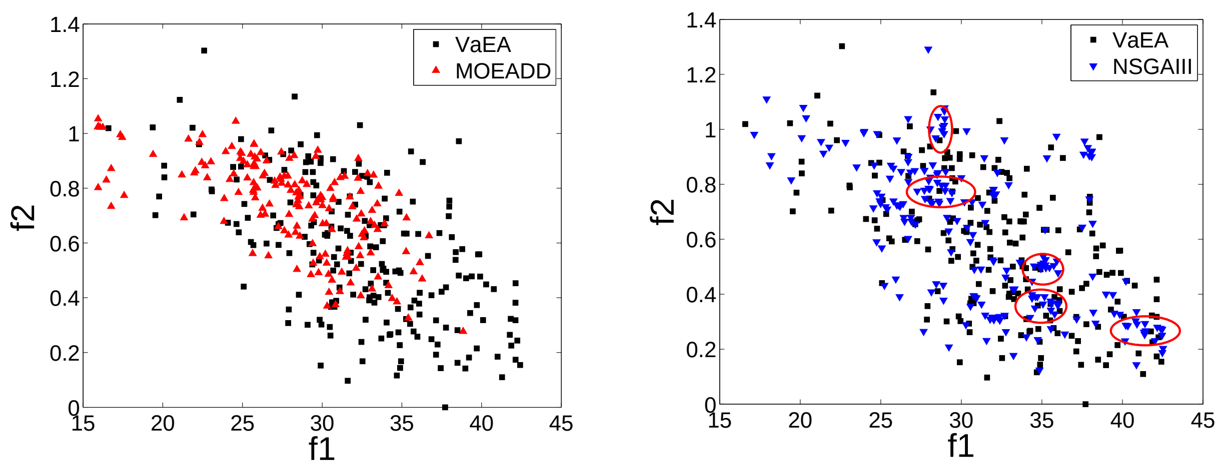
1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation



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

Fig. 13. Final solutions obtained by VaEA, MOEA/DD and NSGA-III on the *Crashworthiness Design* problem.



(a) VaEA vs MOEA/DD (b) VaEA vs NSGA-III

Fig. 14. Pairwise comparison between VaEA and MOEA/DD as well as NSGA-III on the *Car Side-Impact* problem. The final solutions of the algorithms are shown regarding the two-dimensional objective space f_1 and f_2

TABLE X
RESULTS (MEDIAN AND IQR OF Δ AND IGD) OF VAEA, MOEA/DD AND NSGA-III ON TWO ENGINEERING PROBLEMS.

		VaEA	MOEA/DD	NSGA-III
Crashworthiness Design	Δ	$7.101E-01$ $(4.1E-02)$	$1.170E+00$ $(1.7E-01)\bullet$	$8.036E-01$ $(2.1E-01)\bullet$
	IGD	$2.669E-01$ $(9.9E-03)$	$2.289E+00$ $(4.6E-01)\bullet$	$2.937E-01$ $(6.6E-02)\bullet$
Car Side-Impact Problem	Δ	$3.142E-01$ $(2.0E-02)$	$3.352E-01$ $(3.2E-02)\bullet$	$4.200E-01$ $(3.2E-02)\bullet$
	IGD	$1.965E+00$ $(1.5E-01)$	$8.762E+00$ $(3.7E+00)\bullet$	$2.177E+00$ $(3.1E-01)\bullet$

formulation of the original objective as well as all constraints can be found elsewhere [64].

B. Results on Two Problems from Practice

Since the true Pareto fronts of the above two practical problems are not known, we independently run VaEA, MOEA/DD and NSGA-III on each problem 40 times, and combine all returned solutions for each run of each algorithm to construct the reference Pareto fronts (and dominated points are removed) which are used to calculate the Δ and IGD metrics. For the nine-objective *Car Side-Impact Problem*, the population size N is 210 by letting $h_1 = 3$ and $h_2 = 2$, and MFE is set to $210 * 2000$. All other experimental configurations are kept the same as in Section III-C. Table X gives the median and IQR of each metric on the two engineering problems mentioned before. From statistical results, VaEA shows a significant improvement over other two algorithms in terms of both metrics on both practical problems, which strongly demonstrates the usefulness of our proposed VaEA for practical applications.

Fig. 13 plots the Pareto fronts obtained by VaEA, MOEA/DD and NSGA-III on the *Crashworthiness Design* problem. Obviously, VaEA and NSGA-III outperform MOEA/DD concerning the distribution of the final solutions.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

For MOEA/DD, it seems to find solutions only on the boundaries of the front. A large portion of the front is not covered by any non-dominated points. As a result, much larger values of Δ and IGD are obtained by the algorithm. Solutions found by VaEA and NSGA-III are distributed similarly. However, the number of non-dominated points found by VaEA in the top left part is larger than that found by NSGA-III (34 vs 25). Furthermore, some parts in the approximate set of NSGA-III is more crowded than that of VaEA. Therefore, our proposed VaEA presents better metric values.

In the modified *Car Side-Impact* problem, there are nine objectives which makes the visualization much more difficult. As suggested in [10], the solutions are shown in Fig. 14 regarding a two-dimensional objective space f_1 and f_2. In this problem, the regions of f_1 and f_2 in the reference Pareto front are [15.6154,42.5883] and $[0, 1.3964]$, respectively. Apparently, as shown in Fig. 14 (a), the solutions of VaEA are distributed more widely than those of MOEA/DD. Compared with MOEA/DD, the solutions found by VaEA have lower f_2 values, meaning that a much wider area has been explored by our approach. It seems that the solutions returned by MOEA/DD are mainly centralized at the middle part of the two-dimensional space. For the pairwise comparison between VaEA and NSGA-III, both algorithms have found a set of widely distributed solutions. However, there indeed exist differences in terms of the uniformity of the solutions. It is observed in Fig. 14 (b) that obvious clusters (marked with red ellipses) are identified among the solutions of NSGA-III. Therefore, VaEA achieves better performance in terms of the Δ metric.

V. Conclusion and Future Work

To balance convergence and diversity, this paper proposes a new MaOEA named VaEA by using the concept of vector angle. The proposed algorithm is tested on DTLZ and WFG test problems with up to 15 objectives. It is demonstrated by the experimental results that VaEA is quite competitive compared with NSGA-III, and is better than MOEA/DD, MOEA/D and MOMBI2 on the majority of all the test instances. Especially, VaEA shows significant improvement over other peer algorithms in terms of the distribution of the obtained solutions. Our proposed VaEA is able to find a set of properly distributed solutions, and keeps a good balance between the convergence and diversity.

Furthermore, the VaEA has been applied to two practical problems with irregular Pareto fronts. As shown, our algorithm outperforms its competitors in terms of both IGD and generalized SPREAD metrics. VaEA could be particularly suitable for practical applications because of its inherent properties of being free from reference points or weight vectors, introducing no additional algorithmic parameters and having a lower time complexity. Applying VaEA to constrained many-objective optimization problems is one of the subsequent studies.

Acknowledgment

The authors would like to thank the anonymous reviewers for providing valuable comments to improve this paper, and add special thanks to Dr. K. Li (University of Birmingham) for his insightful suggestions.

References

[1] B. Li, J. Li, K. Tang, and X. Yao, “Many-objective evolutionary algorithms: A survey,” *ACM Computing Surveys*, vol. 48, no. 1, pp. 13:1–13:35, Sep 2015.

[2] M. Farina and P. Amato, “On the optimal solution definition for many-criteria optimization problems,” in *Annual Meeting of the North American Fuzzy Information Processing Society*. IEEE, 2002, pp. 233–238.

[3] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, “Many-objective optimization: An engineering design perspective,” in *Evolutionary Multi-criterion Optimization*. Springer, 2005, pp. 14–32.

[4] G. Fu, Z. Kapelan, J. Kasprzyk, and P. Reed, “Optimal design of water distribution systems using many-objective visual analytics,” *Journal of Water Resources Planning and Management*, vol. 139, pp. 624–33, Nov.-Dec. 2013.

[5] U. Wickramasinghe, R. Carrese, and X. Li, “Designing airfoils using a reference point based evolutionary many-objective particle swarm optimization algorithm,” in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2010, pp. 1–8.

[6] R. J. Lygoe, M. Cary, and P. J. Fleming, “A Real-World Application of a Many-Objective Optimisation Complexity Reduction Process,” in *Evolutionary Multi-criterion Optimization*, ser. Lecture Notes in Computer Science, vol. 7811. Springer, 2013, pp. 641–655.

[7] 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.

[8] K. Deb and H. Jain, “An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part I: Solving problems with box constraints,” *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[9] Miqing Li, Shengxiang Yang, and Xiaohui Liu, “Diversity Comparison of Pareto Front Approximations in Many-Objective Optimization,” *IEEE Transactions on Cybernetics*, vol. 44, pp. 2568–84, Dec. 2014.

[10] M. Li, S. Yang, and X. Liu, “Bi-goal evolution for many-objective optimization problems,” *Artificial Intelligence*, vol. 228, pp. 45–65, 2015.

[11] K. Li, K. Deb, 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, Oct 2015.

[12] 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, June 2015.

[13] 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, June 2015.

[14] P. C. Roy, M. M. Islam, K. Murase, and X. Yao, “Evolutionary Path Control Strategy for Solving Many-Objective Optimization Problem,” *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 702–715, APR 2015.

[15] K. Deb, “Multi-objective optimization,” *Multi-objective optimization using evolutionary algorithms*, pp. 13–46, 2001.

[16] 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.

[17] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, 2004.

[18] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm,” Computer Engineering and Networks Laboratory,Department of Electrical Engineering, Swiss Federal Institute of Technology (ETH) Zurich,Switzerland, Tech. Rep., 2001.

[19] J. Knowles and D. Corne, “The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation,” in *Proceeding of Congress on Evolutionary Computation, CEC’99*, vol. 1. IEEE, 1999, pp. 98–105.

[20] D. W. Corne, N. R. Jerram, J. D. Knowles, M. J. Oates, and M. J, “PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001)*. Morgan Kaufmann Publishers, 2001, pp. 283–290.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

[21] 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, Dec. 2007.

[22] 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, pp. 348–65, June 2014.

[23] K. Deb and S. Jain, "Running performance metrics for evolutionary multi-objective optimization." Indian Institute of Technology, KanGAL, Tech. Rep., 2002.

[24] 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, Sep. 2002.

[25] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling Dominance Area of Solutions and Its Impact on the Performance of MOEAs," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007, vol. 4403, pp. 5–20.

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

[27] Z. He, G. G. Yen, and J. Zhang, "Fuzzy-Based Pareto Optimality for Many-Objective Evolutionary Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 2, pp. 269–285, APR 2014.

[28] 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, p. 1402C1412, 2008.

[29] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, p. 1, 2015.

[30] F. di Pierro, S.-T. Khu, and D. A. Savić, "An investigation on preference order ranking scheme for multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 17–45, Feb. 2007.

[31] M. Farina and P. Amato, "A fuzzy definition of "optimality" for many-criteria optimization problems," *IEEE Transactions on Systems Man and Cybernetics Part A-Systems and Humans*, vol. 34, no. 3, pp. 315–326, May 2004.

[32] A. L. Jaimes, C. A. Coello Coello, H. Aguirre, and K. Tanaka, "Adaptive objective space partitioning using conflict information for many-objective optimization," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2011, vol. 6576, pp. 151–165.

[33] T. Wagner, N. Beume, and B. Naujoks, "Pareto-, aggregation-, and indicator-based methods in many-objective optimization," in *Evolutionary Multi-Criterion Optimization*, ser. Lecture Notes in Computer Science, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Springer Berlin Heidelberg, 2007, vol. 4403, pp. 742–756.

[34] S. F. Adra and P. J. Fleming, "Diversity Management in Evolutionary Many-Objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 15, pp. 183–195, 2011.

[35] K. Praditwong and X. Yao, "A new multi-objective evolutionary optimization algorithm: The two-archive algorithm," in *Proceedings of the Computational Intelligence and Security*, vol. 1, Guangzhou, China, 2006, pp. 286–291.

[36] 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, pp. 524–541, August 2015.

[37] J. Cheng, G. 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.

[38] 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.

[39] M. Li, S. Yang, and X. Liu, "Pareto or non-pareto: Bi-criterion evolution in multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. PP, no. 99, pp. 1–1, 2015.

[40] H.-L. Liu, L. Chen, Q. Zhang, and K. Deb, "An evolutionary many objective optimisation algorithm with adaptive region decomposition," Guangdong University of Technology, Tech. Rep. COIN Report Number 2016014, 2016.

[41] 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. 450 – 455, 2014.

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

[43] J. Siwei, C. Zhihua, Z. Jie, and O. Yew-Soon, "Multiobjective optimization by decomposition with pareto-adaptive weight vectors," in *The Seventh International Conference on Natural Computation (ICNC)*, vol. 3, July 2011, pp. 1260–1264.

[44] 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, June 2014.

[45] K. Deb and R. Agrawal, "Simulated binary crossover for continuous search space," *Complex Systems*, vol. 9, pp. 115–48, April 1995.

[46] M. T. Jensen, "Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 5, pp. 503–515, October 2003.

[47] 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, COIN Report Number 2014014, 2014.

[48] R. Hernández Gómez and C. A. Coello Coello, "Improved Metaheuristic Based on the R2 Indicator for Many-Objective Optimization," in *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*. ACM, 2015, pp. 679–686.

[49] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable test problems for evolutionary multiobjective optimization*. Springer, 2005.

[50] 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.

[51] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[52] C. A. Coello Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-objective Problems*, 2nd ed. Springer Science + Business Media, LLC, New York, NY, 2007.

[53] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion," in *IEEE Congress on Evolutionary Computation (CEC 2006)*. IEEE, 2006, pp. 892–899.

[54] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima, "Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems," in *Computational Intelligence in Multi-Criteria Decision-Making (MCDM), 2014 IEEE Symposium on*. IEEE, 2014, pp. 170–177.

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

[56] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[57] J. J. Durillo and A. J. Nebro, "jMetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, pp. 760–771, 2011.

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

[59] H. Ishibuchi, H. Masuda, and Y. Nojima, "Pareto fronts of many-objective degenerate test problems," *IEEE Transactions on Evolutionary Computation*, vol. pp, no. 99, pp. 1–1, 2015.

[60] J. Bader and E. Zitzler, "HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, Jul. 2011.

[61] J. Derrac, S. Garcia, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 7, pp. 3–18, March 2011.

[62] X. Liao, Q. Li, X. Yang, W. Zhang, and W. Li, "Multiobjective optimization for crash safety design of vehicles using stepwise regression model," *Structural and Multidisciplinary Optimization*, vol. 35, pp. 561–9, June 2008.

[63] L. Gu, R. Yang, C. Tho, M. Makowski, O. Faruque, and Y. Li, "Optimization and robustness for crashworthiness of side impact," *International Journal of Vehicle Design*, vol. 26, no. 4, pp. 348–360, 2001.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

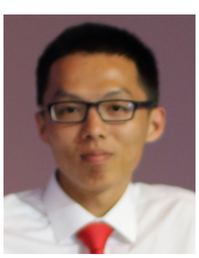
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TEVC.2016.2587808, IEEE Transactions on Evolutionary Computation

[64] K. Deb, S. Gupta, D. Daum, J. Branke, A. K. Mall, and D. Padmanabhan, "Reliability-based optimization using evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 1054–1073, Oct 2009.

Yi Xiang born in Hubei, China, on November 10, 1986. He received the M.S. degree in applied mathematics from the Guangzhou Universiy in 2013. He is currently a Ph.D. candidate in Sun Yat-sen University, Guangzhou, P. R. China. His current research interests include many-objective optimization, mathematical modeling and data mining.

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

Miqing Li received the B.Sc. degree in computer science from the School of Computer and Communication, Hunan University, and the M.Sc. degree in computer science from the College of Information Engineering, Xiangtan University. He received the Ph.D degree in the Department of Computer Science, Brunel University London. His current research interests include evolutionary computation, many-objective optimization and dynamic optimization.



Zefeng Chen received the B.Sc. degree from Sun Yat-sen University, and the M.Sc. degree from the South China University of Technology. He is currently a Ph.D candidate in School of data and computer science, Sun Yat-sen University. His current research interests include evolutionary computation, multi-objective optimization, and machine learning.

1089-778X (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.