

# A Decomposition-Based Many-Objective Evolutionary Algorithm With Two Types of Adjustments for Direction Vectors

Xinye Cai, *Member, IEEE*, Zhiwei Mei, and Zhun Fan, *Senior Member, IEEE*

**Abstract**—Decomposition-based multiobjective evolutionary algorithm has shown its advantage in addressing many-objective optimization problem (MaOP). To further improve its convergence on MaOPs and its diversity for MaOPs with irregular Pareto fronts (PFs, e.g., degenerate and disconnected ones), we proposed a decomposition-based many-objective evolutionary algorithm with two types of adjustments for the direction vectors (MaOEA/D-2ADV). At the very beginning, search is only conducted along the boundary direction vectors to achieve fast convergence, followed by the increase of the number of the direction vectors for approximating a more complete PF. After that, a Pareto-dominance-based mechanism is used to detect the effectiveness of each direction vector and the positions of ineffective direction vectors are adjusted to better fit the shape of irregular PFs. The extensive experimental studies have been conducted to validate the efficiency of MaOEA/D-2ADV on many-objective optimization benchmark problems. The effects of each component in MaOEA/D-2ADV are also investigated in detail.

**Index Terms**—Adjustment of direction vectors, convergence, decomposition, diversity, many-objective optimization.

## I. INTRODUCTION

**M**OST real-world optimization problems involves the simultaneous optimization of multiple objectives. Different from a single-objective optimization problem that has an optimal solution, a set of Pareto-optimal solutions that

Manuscript received April 26, 2017; revised July 17, 2017; accepted July 28, 2017. This work was supported in part by the National Program on Key Basic Research Project (973 Program) under Grant 744901, Grant 744903, and Grant 744904, in part by the National Natural Science Foundation of China under Grant 61300159 and Grant 61473241, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20130808, in part by the China Post-Doctoral Science Foundation under Grant 2015M571751, and in part by the Fundamental Research Funds for the Central Universities of China under Grant NZ2013306 and Grant NS2017070. This paper was recommended by Associate Editor Q. Zhang. (*Corresponding author: Xinye Cai.*)

X. Cai and Z. Mei are with the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China, and also with the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China (e-mail: xinye@nuaa.edu.cn; zwmei@nuaa.edu.cn).

Z. Fan is with the Department of Electronic Engineering, School of Engineering, Shantou University, Shantou 515063, China (e-mail: zfan@stu.edu.cn).

This paper has supplementary downloadable multimedia material available at <http://ieeexplore.ieee.org> provided by the authors.

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

Digital Object Identifier 10.1109/TCYB.2017.2737554

represent the tradeoff relationship between different objectives, exist in an multiobjective optimization problem (MOP). The set of such solutions is called Pareto set (PS) and the image of (PS) on the objective vector space is called Pareto front (PF) [54].

In the past decades, multiobjective evolutionary algorithms (MOEAs) have been recognized as a major methodology for approximating the PF [9], [11], [13], [14], [16], [19], [25], [33], [63], [71]. However, the existing MOEAs mainly focus on addressing MOPs with two or three objectives. The performance of MOEAs, especially Pareto-dominance-based MOEAs, deteriorate when the number of objectives increases to four or above [56], [66]. How to design effective MOEAs to address MOPs with more than three objectives, commonly referred to as many-objective optimization problems (MaOPs) [38], has drawn a great deal of attention. MaOPs are very challenging due to the following reasons [38], [49].

- 1) With the increase of the number of objectives, the selection pressure of Pareto dominance-based MOEAs deteriorate drastically, due to the facts that most solutions become nondominated to each other [26], [45], [57]. Some classical MOEAs (e.g., NSGA-II [19]) work well on MOPs with two or three objectives but cannot work well in MaOPs [37].
- 2) It is well-known that, the PF of an  $m$ -objective nondegenerate MOP (or MaOP) is an  $(m - 1)$ -dimensional manifold [34], [43] [PFs of degenerate MOPs (or MaOPs) are less than  $(m - 1)$ -dimensional], which indicates that the number of solutions required to approximate the entire PF increases exponentially with the increase of the number of objectives. Nevertheless, it is impractical to use such a huge population for an optimizer in MaOPs due to the unaffordable computational and space complexity [49].

Over the recent years, numerous efforts have been made to address MaOPs and a number of many-objective evolutionary algorithms (MaOEA) have emerged. These MaOEA can be roughly divided into two categories.

- 1) Objective space reduction [20], [39], [40], [61].
- 2) Design of new selection methods [1], [5], [32], [33], [37], [53], [60].

The former category can be further divided into two sub-categories.

- 1) *The objective reduction* [8], [31], [40] takes advantage of the correlation between objectives or conducts machine

learning techniques such as feature selection, to reduce the number of objectives either online or offline.

- 2) *Preference-based methods* [10], [27], [44], [48] integrate users' preference information to reduce the search region in the objective vector space and make the search focus on one or several parts of the PFs.

The latter category focuses on designing new selection methods of solutions to achieve better balance between *convergence* and *diversity* for obtaining good approximation to the set of Pareto optimal solutions [7], [22]. Convergence can be measured as the distance of solutions toward the PF, which should be as small as possible. Diversity can be measured as the spread of solutions along the PF, which should be as uniform as possible. Based on different selection methods, the latter category can be further classified into modified-Pareto-dominance-based [24], [37], [59], [60], [68]–[70], [72], diversity-based [1], [53], indicator-based [5], [6], [73], co-evolutionary-based [65]–[67], and decomposition-based [4], [32], [33], [71] MaOEAs, as follows.

- 1) Modified-Pareto-dominance-based approaches directly relax the Pareto-dominance relation to have solutions further distinguished to each other, to further increase the selection pressure toward PFs for MaOPs. Such modified-Pareto-dominance-based approaches include  $\epsilon$ -dominance [18], [46], grid-based-dominance [69], volume-dominance [47], and subspace-dominance [2], [3].
- 2) Diversity-based approaches further enhance the selection pressure of MOEAs by better diversity maintenance. Adra and Fleming [1] introduced a diversity management mechanism based on the spread of the population. Li *et al.* [53] proposed a shift-based density estimation as a diversity maintenance scheme to further enhance the selection pressure for MaOPs.
- 3) Indicator-based approaches use indicator metric directly as the selection criteria. For example, hypervolume [74] is a widely used indicator, where the higher the hypervolume, the better the approximation. Emmerich *et al.* [23] proposed a  $S$ -metric selection by maximizing the hypervolume of the solution sets for MaOPs. To further reduce the computational complexity of calculating hypervolume, Bader and Zitzler [5] proposed a hypervolume estimation algorithm, where, instead of calculating the exact values of hypervolume, Monte Carlo sampling is adopted to approximate it.
- 4) Co-evolutionary-based approaches use the co-evolution of decision-maker preferences together with a population of candidate solutions [65]. It has been further hybridized with the decomposition approach in [66] and with brushing technique in [67].
- 5) Decomposition-based approaches decomposes an MOP or MaOP into a number of subproblems by linear or nonlinear aggregation functions and solve them simultaneously. MOEA based on decomposition (MOEA/D) [71] is a representative of such algorithms. In MOEA/D, each solution is associated with a subproblem, and two subproblems are called neighbors if their direction vectors are close to each other.

Recent research indicates that decomposition-based approaches [34], [35] (e.g., MOEA/D [71]) has very good performance on MaOPs. However, the diversity of MOEA/D is maintained by a set of preset direction vectors, which is usually very sensitive to the shape of unknown PFs.

Some recent works focus on the hybridization of decomposition and dominance approaches [17], [28], [31], [52], [62]. For instance, Deb and Jain [17], [41] proposed a reference-point-based MaOEA (NSGA-III) as an extension of NSGA-II [19]. A MaOEA based on both dominance and decomposition is also proposed to address MaOPs [52]. In addition, some other recent studies focus on the association methods between solutions and subproblems (direction vectors) (e.g., [4] and [10]), for the decomposition-based MaOEA.

Since the decomposition-based approaches have shown their great potential to address MaOPs, we continue to conduct research along this direction. In this paper, we propose a MaOEA based on decomposition with two types of adjustments for the direction vectors (MaOEA/D-2ADV)—one type aims to adjust the number of direction vectors for approximating a more complete PF after fast convergence along the boundary direction vectors and the other type aims to adjust the positions of the ineffective direction vectors for MaOPs with irregular PFs (e.g., disconnected and/or degenerate).

In the remainder of this paper, some preliminaries are given in Section II. The motivations are presented in Section III. The proposed algorithm, MaOEA/D-2ADV, is detailed in Section IV. In Section V, the experimental results are analyzed and discussed in detail. The effects of components in MaOEA/D-2ADV are also investigated in this section. Finally, the conclusion is drawn in Section VI.

## II. PRELIMINARIES

### A. Basic Definitions

Without loss of generality, an MOP can be defined as

$$\begin{aligned} & \text{minimize } F(x) = (f_1(x), \dots, f_m(x))^T \\ & \text{subject to } x \in \Omega \end{aligned} \quad (1)$$

where  $\Omega$  is the *decision space*,  $F : \Omega \rightarrow R^m$  consists of  $m$  real-valued objective functions, and  $\{F(x)|x \in \Omega\}$  is the *attainable objective set*.

Let  $u, v \in R^m$ ,  $u$  is said to *dominate*  $v$ , denoted by  $u < v$ , if and only if  $u_j \leq v_j$  for every  $j \in \{1, \dots, m\}$  and  $u_k < v_k$  for at least one index  $k \in \{1, \dots, m\}$ .<sup>1</sup> Given a set  $S$  in  $R^m$ , a solution  $x \in S$  is called *nondominated* if no other solution in  $S$  can dominate it. A solution  $x^* \in \Omega$  is *Pareto-optimal* if  $F(x^*)$  is nondominated in the attainable objective set.  $F(x^*)$  is then called a *Pareto-optimal (objective) vector*. In other words, any improvement in one objective of a Pareto optimal solution is bound to deteriorate at least another objective [54].

<sup>1</sup>In the case of maximization, the inequality signs should be reversed.

The *ideal objective vector* and *nadir objective vector* are two important concepts to define the ranges of a PF. The *ideal objective vector*  $z^*$  is a vector  $z^* = (z_1^*, \dots, z_m^*)^T$ , where  $z_j^* = \min_{x \in \Omega} f_j(x)$ ,  $j \in \{1, \dots, m\}$ . The *nadir objective vector*  $z^{\text{nad}}$  is a vector  $z^{\text{nad}} = (z_1^{\text{nad}}, \dots, z_m^{\text{nad}})^T$ , where  $z_i^{\text{nad}} = \max_{x \in \text{PS}} f_i(x)$ ,  $i \in \{1, \dots, m\}$ .

### B. Decomposition Approaches

In principle, many methods can be used to decompose an MOP into a number of scalar optimization subproblems [54]. Penalty boundary intersection (PBI) approach [71], which can be seen as a variant of normal-boundary intersection approach [15], is one of the popular decomposition approaches.

Let  $\lambda^i = (\lambda_1, \dots, \lambda_m)^T$  be a direction vector for the  $i$ th subproblem, where  $\lambda_j \geq 0$ ,  $j \in 1, \dots, m$  and  $\sum_{j=1}^m \lambda_j = 1$ . The  $i$ th subproblem is defined as

$$\begin{aligned} \text{minimize } & g^{\text{pbi}}(x|\lambda^i, z^*) = d_1^i + \theta d_2^i \\ & d_1^i = (F(x) - z^*)^T \lambda^i / \|\lambda^i\| \\ & d_2^i = \|F(x) - z^* - (d_1^i / \|\lambda^i\|) \lambda^i\| \\ \text{subject to } & x \in \Omega \end{aligned} \quad (2)$$

where  $\|\cdot\|$  denotes  $L_2$ -norm and  $\theta$  is the penalty parameter.

### C. Approaches With Adaptive Direction Vectors

To address MOPs with irregular PFs, there have already been several related works on adjusting the direction vectors (weight vectors) for MOEA/D. For example, the direction vectors were adjusted using a linear interpolation of the nondominated solutions to approximate the PF in [29] and [30]. However, as it is difficult to give a method of estimating the PF and uniformly generate points on the high-dimensional PF [29], [30].

Jiang *et al.* [42] proposed Pareto-adaptive weight vectors ( $pa\lambda$ ) based on hypervolume metric for MOEA/D. However, this approach needs to assume that the PF is symmetric of the form  $f_1^p + f_2^p + \dots + f_m^p = 1$  ( $m$  is the number of objectives and  $p$  is a parameter to estimate the shape of PF), which limits its use on MaOPs with PFs that do not follow this assumption.

In EMOSA [50], two sets of weight vectors (direction vectors) are maintained:  $Q$  evenly distributed weight vectors and  $C_{H+m-1}^{m-1}$  predefined candidate weight vector set  $\Lambda$ , where  $H$  is a positive integer number and  $C_{H+m-1}^{m-1} \gg Q$ . The adjustment of weight vectors are implemented as follows. For the current solution  $x^s$  (corresponding to  $\lambda^s$ ), its closest nondominated neighbor  $x^t$  (corresponding to  $\lambda^t$ ), is tracked. The new weight vector  $\lambda$  is picked from the candidate weight vector set  $\Lambda$  to replace the current weight vector  $\lambda^s$ , based on two conditions: 1)  $\text{dist}(\lambda^s, \lambda^t) < \text{dist}(\lambda, \lambda^t)$  and 2)  $\text{dist}(\lambda, \lambda^s) \leq \text{dist}(\lambda, \Lambda)$ . EMOSA is applied to bi- and tri-objective knapsack and traveling salesman problems. However, it may not be able to be extended to MaOPs, due to the following reason. The number of the predefined candidate weight vectors,  $|\Lambda|$ , would increase dramatically, as  $C_{H+m-1}^{m-1}$  becomes extremely large when  $m$  is larger than 5 and  $H > m$ , where ( $H > m$ ) is to guarantee

not all the generated weight vectors are on the boundaries of PFs.

In NSGA-III [41], the reference points (direction vectors) are adjusted by simply deleting nonuseful ones (the ones with no solution associated with) and adding a simplex of  $m$  points centering around an existing reference point. These inserted  $m$  points have the same distance as the distance between two consecutive reference points on the original simplex. This method can be applied to many-objective problems with irregular PFs. However, it generates  $m$  new reference points by adding  $m$  fixed vectors to each of the original reference points, which may lead to low efficient adjustment, especially on MaOPs with the degenerate PFs, as most of the newly generated reference points would be ineffective.

In MOEA/D-AWA [58], a decomposition working population and an external archive are adopted. The direction vectors are adaptively adjusted by estimating the sparsity of both working population and nondominated solutions in the external archive. More specifically, the direction vectors are deleted in the overcrowded area of the working population and the new direction vectors are inserted by using the objective vectors of the nondominated solutions (of the external archive), with the best sparsity level in the working population. MOEA/D-AWA is demonstrated to perform well on MaOPs with degenerate PFs. However, a parameter  $nus$  is used to control the number of direction vectors to be deleted, which is very difficult to set. The improper setting of it would lower the efficiency of the adjustment. In addition, MOEA/D-AWA needs to maintain a large number of nondominated solutions, stored in an external archive. The extra computational time is needed for density estimation and nondominated sorting on this external archive, both of which are known to be computational expensive, especially on MaOPs.

Cheng *et al.* [10] also proposed a reference (direction) vector regeneration strategy in RVEA to improve the performance on problems with irregular PFs. In RVEA [10], the new direction vectors are randomly generated inside the range specified by the minimum and maximum objective values calculated from the candidate solutions. Since the reference vectors are generated globally and randomly, the local solution density is not guaranteed [10]. This is very likely to lead to a low efficient adjustments on MaOPs whose effective PFs only account for a very small proportion of the whole ( $m - 1$ )-dimensional manifold ( $m$  is the number of objectives).

Wang *et al.* [66] proposed a co-evolutionary algorithm with weights (PICEA-w), in which weights (direction vectors) are co-evolving with the candidate solutions during the optimization process. The new weights are generated randomly in each generation and the weight vectors with the highest contributions to nondominated solutions are maintained based on two criteria. First, for each candidate solution, the selected weight must be the one that ranks the candidate solution as the best. Second, if more than one weight is found by the first criterion, the one that is the furthest from the candidate solution is chosen. PICEA-w shows its effectiveness on MaOPs with irregular PFs. However, like the method in [10], the globally random generation of weight vectors may not be very efficient on MaOPs with the degenerate PFs.

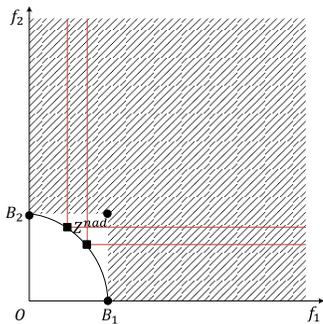


Fig. 1. Illustration of the domination regions of different Pareto optimal solutions in a bi-objective optimization problem. The boundary solutions ( $B_1$  and  $B_2$ ) dominate the largest regions (denoted by shaded regions) compared with other Pareto optimal solutions (denoted by  $\blacksquare$ ). The nadir point  $z^{\text{nad}}$ , approximated by the boundary solutions, can be used to define the domination regions by boundary solutions.

### III. MOTIVATIONS

Although the current MOEAs/D are very promising to address MaOPs, the following issues need to be further considered [36].

- 1) In MOPs, the neighborhood subproblems play an important role in helping each other speeding up the convergence. However, in many-objective optimization where the number of direction vectors is very limited compared to the huge objective space, the direction vectors of subproblems may be too far away to have much effects on helping each other during the evolutionary process. Arguably, the boundary solutions of a PF provide more domination information as they dominate more regions in objective space than other Pareto optimal solutions [64], as shown in Fig. 1. The subproblems along the boundary direction vectors are most important as the Pareto optimal solutions of these subproblems may help most for the convergence of other parts of a PF. After fast convergence along the boundary direction vectors, proper adjustments by inserting new direction vectors can be further adopted for achieving better diversity.
- 2) As described in Section I, the diversity of decomposition-based multiobjective optimization approaches are implicitly achieved by a set of predefined direction vectors. An underlying assumption is that each subproblem has a unique Pareto optimal solution. This is true when the PF of a MaOP is regular, as shown in Fig. 2(a). However, for MaOPs with irregular PFs (e.g., degenerate ones), as shown in Fig. 2(b), the optimal solutions of many subproblems are not Pareto-optimal.<sup>2</sup> In other words, many of the predefined direction vectors are ineffective for MaOPs with irregular PFs. To maintain better diversity for such MaOPs, the positions of the ineffective direction vectors need to be adjusted once the direction vectors are verified as ineffective.

<sup>2</sup>Assume PBI is used with a large  $\theta$  value for decomposition.

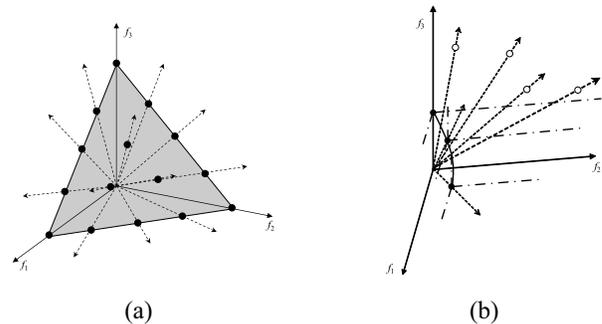


Fig. 2. Illustrations of tri-objective optimization problems with regular and irregular (degenerate) PFs. A arrowed dashed line denotes a direction vector of a subproblem. (a) Illustration of a tri-objective optimization problem with a regular PF, where the optimal solutions of all subproblems are also the Pareto optimal solutions, denoted by “ $\bullet$ .” The direction vectors of all the subproblems are effective. (b) Illustration of a tri-objective optimization problem with an irregular (degenerate) PF, where the optimal solutions of most subproblems are not the Pareto optimal solutions, denoted by “ $\circ$ .” The direction vectors of such subproblems are ineffective.

Actually, the above two issues are not independent to each other. On one hand, the adjustment of direction vectors can be effective only when the population is converged to some extent [58]. On the other hand, it has been demonstrated in [66] that adapting direction vectors during the search may affect the convergence performance of MaOEA/D. It is even more desirable to achieve better balance between convergence and diversity when adjusting the direction vectors for MaOPs with irregular PFs [49].

Based on the above considerations, in this paper, we propose an MaOEA/D-2ADV. At the beginning, search is only conducted along the boundary direction vectors for fast convergence to approximate boundary Pareto optimal solutions. After that, the number of direction vectors is expanded for approximating a more complete PF. For MaOPs with irregular PFs (e.g., disconnected and degenerate), a simple Pareto-dominance-based approach is used to detect the effectiveness of each direction vector and the positions of the ineffective direction vectors are adjusted by iteratively deleting ineffective ones and inserting new ones between effective direction vectors. It is worth noting that the approximated boundary Pareto optimal solutions remain helpful for maintaining the convergence of the population during the adaptations of the direction vectors and the fast convergence and the adaptation of direction vectors are complementarily key steps for MaOEA/D-2ADV for achieving better balance between convergence and diversity.

In addition, different from schemes of direction vectors adjustments in [29], [30], and [50], our approach can be easily extended to MaOPs. Different from method in [58], our approach does not have any parameter for adjusting direction vectors and does not need to use a large number of nondominated solutions (usually maintained in an external archive). Alternatively, it takes advantages of the effective neighboring direction vector pairs. This is based on two of our perspectives.

- 1) A direction vector can be considered as a representative of a subregion in the objective space. When the

**Algorithm 1: Main Framework of MaOEA/D-2ADV**


---

**Input** : a stopping criterion;  
 $N$ : maximum population size;  
 $T$ : neighborhood size;  
 $m$ : the number of objectives.

**Output**: a solution set  $P$

```

/*  $K$  is the population size          */
1  $K = m$ ;
/* Initialization                      */
2  $[P, DV, B, z^*, z^{nad}] = \text{INITIALIZATION}(K, m)$ ;
3  $t = 1$ ;
/* Main loop                          */
4 while the stopping criteria is not satisfied do
5    $[Q, z^*] = \text{VARIATION}(P, B, z^*, N, m)$ ;
6    $P = \text{ASSOCIATION\_BASED\_SELECTION}(P \cup Q, DV, z^*, z^{nad}, N)$ ;
7   if  $\text{mod}(t, \phi_1) == 0$  and  $K == m$  then // The
      adjustment for the number of
      direction vectors
8     if  $\Delta_t < 10^{-4}$  then
9        $[P, DV, B, z^{nad}] =$ 
10         $\text{DV\_ADJUSTMENT1}(P, DV, N, T, m)$ ;
11         $K = N$ ;
12     end
13   if  $\text{mod}(t, \phi_2) == 0$  and  $K == N$  then // The
      adjustment for the positions of the
      ineffective direction vectors
14      $[DV, B] = \text{DV\_ADJUSTMENT2}(P, DV, T)$ ;
15   end
16    $t = t + 1$ ;
17 end

```

---

accurate density estimation of a large number of the non-dominated solutions becomes much more expensive in MaOPs [58], the effective direction vectors provides a cheaper alternative.

- 2) Without knowing the shape of PF *a priori*, inserting new direction vectors at the midpoint of two distant and effective neighboring direction vectors would be more efficient than globally random generation of direction vectors [10], [66] or generating around each of the original reference points [41], especially on MaOP whose effective PF only accounts for a small proportion of the whole  $(m - 1)$ -dimensional manifold (e.g., degenerate PF).

## IV. MAOEA/D-2ADV

## A. Main Framework

Algorithm 1 presents the framework of MaOEA/D-2ADV. The algorithm maintains the following.

- 1) A population  $P$ .
- 2) A set of direction vectors  $DV = \{\lambda^1, \dots, \lambda^K\}$ , where  $\lambda^i$  is the direction vector of the  $i$ th subproblem and  $K$  is the number of direction vectors and/or the population size.

**Algorithm 2: Initialization (INITIALIZATION)**


---

**Input** :  $K$ : the population size or number of direction vectors;  
 $m$ : the number of objectives;

**Output**: an initial population  $P = \{x^1, \dots, x^K\}$ ;  
initial direction vectors  $DV = \{\lambda^1, \dots, \lambda^K\}$ ;  
neighborhood index sets  $B = \{B^1, \dots, B^K\}$ ;  
the initial ideal point  $z^*$ ;  
the initial nadir point  $z^{nad}$ .

```

1 Randomly generate an initial population,
   $P = \{x^1, \dots, x^K\}$ ;
2 for  $j = 1$  to  $m$  do
3    $\lambda_j^j = 1, \lambda_i^j = 0, i = 1, \dots, m, i \neq j$ ;
4    $B^j = \emptyset$ ;
5    $z_j^* = \min_{x \in P} f_j(x)$ ;
6    $z_j^{nad} = +\infty$ ;
7 end

```

---

**Algorithm 3: Offspring Generation (VARIATION)**


---

**Input** :  $P = \{x^1, \dots, x^K\}$ : parent solutions;  
 $B = \{B^1, \dots, B^K\}$ : neighbourhood index set;  
 $z^*$ : the ideal point;  
 $N$ : maximum population size;  
 $m$ : the number of objectives;

**Output**: an offspring population  $Q = \{y^1, \dots, y^K\}, z^*$ .

```

1  $Q = \emptyset$ ;
2 foreach  $x^i \in P, i = 1, 2, \dots, K$  do
  /* Selection of neighborhood index
  set                                     */
3    $D = \begin{cases} \emptyset & K = m \\ B^i & K = N \text{ and } \text{rand}() < \delta \\ \{1, \dots, K\} & \text{otherwise} \end{cases}$  (3)

  /* Reproduction                       */
4   if  $D = \emptyset$  then
5     Perform a mutation operator on  $x^i$  with probability  $p_m$ 
      to produce a new solution  $y^i$ ;
6   else
7     Randomly select two indexes  $r_1$  and  $r_2$  from  $D$ , and
      then generate a solution  $y^i$  from  $x^i, x^{r_1}$  and  $x^{r_2}$  by a
      DE operator;
8   end
  /* Update  $z^*$                          */
9   for  $j = 1$  to  $m$  do
10    if  $z_j^* > f_j(y^i)$  then
11       $z_j^* = f_j(y^i)$ ;
12    end
13  end
14   $Q = Q \cup \{y^i\}$ ;
15 end

```

---

- 3) The neighborhood index sets  $B = \{B^1, \dots, B^K\}$ , where  $B^i$  is the neighborhood index set for the  $i$ th subproblem. It is mainly used in restricted mating (variation). It is worth noting that the number of direction vectors  $K$  may change during the evolutionary process.

**Algorithm 4:** Association (ASSOCIATION)

---

**Input** :  $P$ : current population;  
 $z^*$ : ideal point;  
 $DV = \{\lambda^1, \dots, \lambda^K\}$ : direction vectors.  
**Output**: Solutions associated with direction vectors  
 $S^1, \dots, S^K$ .

```

1  $S^k = \emptyset$  for all  $k = 1, \dots, K$ ;
2 foreach  $x^i$  in  $P$  do
3   for  $j = 1$  to  $m$  do
4      $x_j^i = x_j^i - z_j^*$ ;
5   end
6   /* Refer to (4) */
7    $k = \operatorname{argmin}_{\lambda^k \in DV} \arccos(\frac{x^i \cdot \lambda^k}{\|x^i\| \|\lambda^k\|})$ ;
8    $S^k = S^k \cup \{x^i\}$ ;
9 end

```

---

The main procedures of MaOEA/D-2ADV include the following steps.

- 1) The initialization of  $P$ ,  $DV$ , and  $B$ .
- 2) Variation.
- 3) Association-based selection.
- 4) The adjustment for the number of direction vectors.
- 5) The adjustment for the position of ineffective direction vectors.

In the following sections, each step of MaOEA/D-2ADV is introduced in detail.

**B. Initialization**

In the initialize procedure (Algorithm 2), the population size and/or the number of the direction vectors is set to  $m$ . A population  $P$  is randomly generated and  $m$  initial direction vectors  $DV$  along the boundary objective directions [i.e., all the permutations of  $(1, 0, \dots, 0)$ ] are initialized. The neighborhood index set  $B^i$  of the  $i$ th subproblem is set to  $\emptyset$ . Each objective of  $z^*$  is initialized as the minimum value of the objective in  $P$  and each objective of  $z^{\text{nadir}}$  is set to  $+\infty$  as the initial value.

**C. Variation**

In the variation step, an offspring population  $Q$  is generated from  $P$  by variation (Algorithm 3).

In Algorithm 3, for each solution  $x^i$  in  $P$ , a mating pool  $D$  is used to store its mating solutions for generating a new offspring. When  $K = m$ ,  $D$  is set to  $\emptyset$  and a simple mutation is applied to  $x^i$ ; otherwise,  $D$  is set to either  $B^i$  or  $\{1, \dots, K\}$  based on a small probability  $\delta$ . The differential evolution (DE) [55] with polynomial mutation operator [16] is applied on  $x^i$  and  $D$  to generate a new offspring  $y^i$ . After that,  $y^i$  is used to update  $z^*$ . The whole process is iterated until  $K$  offspring solutions are generated for  $Q$ .

**D. Association-Based Selection**

After combining the parent population  $P$  with the offspring population  $Q$ , association-based selection (Algorithm 5)

**Algorithm 5:** Association-Based Selection (ASSOCIATION\_BASED\_SELECTION)

---

**Input** :  $P$ : current population;  
 $DV = \{\lambda^1, \dots, \lambda^K\}$ : direction vectors;  
 $z^*$ : the initial ideal point;  
 $z^{\text{nadir}}$ : the initial nadir point;  
 $N$ : maximum population size.  
**Output**: the selected population  $Q$ .

```

1  $S = P, Q = \emptyset$ ;
2 if  $K == N$  then
3   /* Space reduction */
4    $S = \emptyset$ ;
5   foreach  $x$  in  $P$  do
6     if  $f_j(x) \leq z_j^{\text{nadir}}$  for all  $j = 1, \dots, m$  then
7        $S = S \cup \{x\}$ ;
8     end
9   end
10  $[S^1, \dots, S^K] = \text{ASSOCIATION}(S, DV, z^*)$ ;
11 /* Selection */
12 for  $k = 1$  to  $K$  do
13   if  $S^k = \emptyset$  then
14     randomly select a solution from  $S$  and add to  $Q$ ;
15   end
16   if  $|S^k| > 1$  then
17      $x = \operatorname{argmin}_{x \in S} g^{\text{pbi}}(x | \lambda^k, z^*)$ ;
18      $Q = Q \cup \{x\}$ ;
19   end
20 end

```

---

**Algorithm 6:** Adjustment for the Number of Direction Vectors (DV\_ADJUSTMENT1)

---

**Input** :  $P$ : current population;  
 $DV$ : the old set of direction vectors;  
 $N$ : maximum population size;  
 $T$ : neighborhood size;  
 $m$ : the number of objectives.  
**Output**: new population  $Q$ ;  
a new set of direction vectors  $DV$ ;  
updated neighborhood index set  $B$ ;  
the nadir point  $z^{\text{nadir}}$ .

```

1 for  $k = 1$  to  $m$  do
2    $z_k^{\text{nadir}} = \max_{x \in P} f_k(x)$ ;
3 end
4 Generate  $N$  direction vectors  $DV = \{\lambda^1, \dots, \lambda^N\}$ ;
5  $B = \text{TRACK\_NEIGHBORS}(DV, T)$ ;
6  $Q = \text{ASSOCIATION\_BASED\_SELECTION}(P, DV)$ ;

```

---

is called, where  $K$  solutions are selected from the merged population as follows.

As explained in Fig. 1, the nadir points contain the information of domination regions by the boundary solutions, which can be used to reduce the possible objective space that PF

**Algorithm 7:** Track Neighborhood Index Set  $B$  (TRACK\_NEIGHBORS)

---

**Input** :  $DV = \{\lambda^1, \dots, \lambda^K\}$ : direction vectors;  
 $T$ : neighborhood size;  
**Output**: neighborhood index set  $B$ .

```

1 for  $i = 1$  to  $K$  do
2    $d_{i,i} = +\infty$ ;
3   for  $j = i + 1$  to  $K$  do
4      $d_{i,j} = d_{j,i} = |\lambda^i - \lambda^j|$ ;
5   end
6   /*  $I$  stores the index of direction
   vectors after sorting  $d_i$  */
7    $[d_i, I] = \text{sort}(d_i), j = 1, \dots, K, j \neq i$ ;
8    $B^i = I(1:T)$ ;
9 end
```

---

**Algorithm 8:** Detection of the Effective Direction Vectors (DETECTION)

**Input** :  $P$ : current population;  
 $DV = \{\lambda^1, \dots, \lambda^K\}$ : direction vectors;  
 $z^*$ : ideal point;  
**Output**: effective direction vectors  $\overline{DV}$ .

/\* Find all dominated solutions \*/

```

1  $P_{dom} = \text{NONDOMINATED}(P)$ ;
2  $[S^1, \dots, S^K] = \text{ASSOCIATION}(P_{dom}, DV, z^*)$ ;
3  $\overline{DV} = \emptyset$ ;
4 for  $k = 1$  to  $K$  do
5   if  $S^k \neq \emptyset$  then
6      $\overline{DV} = \overline{DV} \cup \{\lambda^k\}$ ;
7   end
8   /*  $|\overline{DV}| \leq |DV|$  */
9 end
```

---

exists. The solutions  $S$  located inside the nadir point are first selected.

After that, each solution is associated with its closest direction vector (Algorithm 4). The closeness between a solution  $x$  and a direction vector  $\lambda$  is defined as follows:

$$\arccos\left(\frac{(F(x) - z^*) \cdot \lambda^T}{\|F(x) - z^*\| \|\lambda\|}\right) \quad (4)$$

where  $\|\cdot\|$  calculates the norm.

To maintain a good diversity, a direction vector is allowed to associate with one and only one solution. Nevertheless, if the number of solutions associated with a direction vector exceeds one, the solution with minimum aggregated objective function value  $g^{\text{pbi}}(x|\lambda^k, z^*)$  is kept. If a direction vector has no associated solution, a solution is randomly selected from  $S$  to associate with it.

**E. Adjustment for the Number of Direction Vectors**

At  $t$ th generation, if the relative decrease along  $m$  direction vectors ( $\Delta_t$ ) is lower than a very small value ( $10^{-4}$ ), which indicates all the subproblems have been well-converged, the adjustment for the number of direction vectors is activated,

**Algorithm 9:** Adjustment for the Positions of Ineffective Direction Vectors (DV\_ADJUSTMENT2)

**Input** :  $P$ : current population;  
 $DV = \{\lambda^1, \dots, \lambda^K\}$ : direction vectors;  
 $T$ : neighborhood size;  
**Output**: updated direction vectors  $DV = \{\lambda^1, \dots, \lambda^K\}$ ;  
updated neighborhood index set  $B$ .

/\* Identify the effective direction vectors  $\overline{DV}$  \*/

```

1  $\overline{DV} = \text{DETECTION}(P, DV, z^*)$ ;
2 while  $|\overline{DV}| < K$  do
3   if  $\binom{|\overline{DV}|}{2} \leq (K - |\overline{DV}|)$  then
4     /* Calculate the midpoints for all
     possible pairs in  $\overline{DV}$  */
5     for  $i = 1$  to  $|\overline{DV}|$  do
6       for  $j = i + 1$  to  $|\overline{DV}|$  do
7          $\lambda^* = (\lambda^i + \lambda^j)/2$ ;
8          $\overline{DV} = \overline{DV} \cup \{\lambda^*\}$ ;
9       end
10    end
11  else
12    /* find the distance of each  $\lambda \in \overline{DV}$  to
    its nearest neighbor
    Pair =  $\{(1, 2), \dots, (i, j), \dots, (|\overline{DV}| - 1, |\overline{DV}|)\}$ ;
    d =  $(d_{1,2}, \dots, d_{i,j}, \dots, d_{|\overline{DV}|-1, |\overline{DV}|}) = (0, \dots, 0)$ ;
13    for  $i = 1$  to  $|\overline{DV}|$  do
14       $d_{i,i} = +\infty$ ;
15      for  $j = i + 1$  to  $|\overline{DV}|$  do
16         $d_{i,j} = |\lambda^i - \lambda^j|$ ;
17      end
18       $d_{min}(i) = \min_{1 \leq j \leq |\overline{DV}|} (d_{i,j})$ ;
19    end
20     $d_{max} = \max(d_{min}(1), \dots, d_{min}(|\overline{DV}|))$ ;
21    /* Sort d in an ascending order and
    then sort Pair based on d
    [d, I] = sort(d);
    Pair = Pair(I);
22    /* Select  $K - |\overline{DV}|$  pairs of direction
    vectors
    Mid = find(d == d_max);
    l = min(Mid);
    r = max(Mid);
23    while  $r - l + 1 < K - |\overline{DV}| \wedge l > 1$  do
24      l --;
25    end
26    while  $r - l + 1 < K - |\overline{DV}|$  do
27      r ++;
28    end
29    /* Generate new direction vectors
30    for  $k = l$  to  $r$  do
31       $\lambda^* = (\lambda^{\text{Pair}(k).1} + \lambda^{\text{Pair}(k).2})/2$ ;
32       $\overline{DV} = \overline{DV} \cup \{\lambda^*\}$ ;
33    end
34  end
35 end
36 end
37 end
38  $DV = \overline{DV}$ ;
39  $B = \text{TRACK\_NEIGHBORS}(DV, T)$ ;
```

---

presented in Algorithm 6.  $\Delta_t$  is defined as follows:

$$\Delta_t = \sum_{k=1}^K \left( \left\| \frac{F(x_t^k) - F(x_{t-\phi_1}^k)}{F(x_t^k)} \right\| \right) \quad (5)$$

TABLE I  
SORTED DISTANCE VECTORS  $d$  AND CORRESPONDING PAIR FOR FIG. 3(a), WHERE  $d_{\max}$  IS HIGHLIGHTED IN THE BOLDFACE

<i>index</i>	1	2	3	<b>4</b>	5	6	7	8	9	10	11	12	13	14	15
<i>Pair</i>	(a,b)	(g,h)	(b,c)	<b>(f,g)</b>	(a,c)	(c,f)	(f,h)	(b,f)	(c,g)	(a,f)	(c,h)	(b,g)	(a,g)	(b,h)	(a,h)
<i>d<sub>Pair</sub></i>	0.1838	0.1979	0.2546	<b>0.2970</b> ( $d_{\max}$ )	0.4384	0.4808	0.4949	0.7354	0.7778	0.9192	0.9757	1.0324	1.2162	1.2303	1.4141

TABLE II  
CORRESPONDING  $d_{i,j}$  AND  $d_{\min}(i)$  FOR FIG. 3(a), WHERE  $d_{\max}$  IS HIGHLIGHTED IN THE BOLDFACE

<i>j \ i</i>	a	b	c	f	g	h
a	$+\infty$	0.1838	0.4384	0.9192	1.2162	1.4141
b	0.1838	$+\infty$	0.2546	0.7354	1.0324	1.2303
c	0.4384	0.2546	$+\infty$	0.4808	0.7778	0.9757
f	0.9192	0.7354	0.4808	$+\infty$	0.2970	0.4949
g	1.2162	1.0324	0.7778	0.2970	$+\infty$	0.1979
h	1.4141	1.2303	0.9757	0.4949	0.1979	$+\infty$
$d_{\min}(i)$	0.1838	0.1838	0.2546	<b>0.2970</b> ( $d_{\max}$ )	0.1979	0.1979

TABLE III  
SETUPS OF DTLZ TEST PROBLEMS

Name	Parameter	Number of Objectives (m)	Number of Decision Variables (n)
DTLZ1-7		3	10
DTLZ1	$k = 5$	4, 5, 6, 8, 10	$m + k - 1$
DTLZ2-6	$k = 10$	4, 5, 6, 8, 10	$m + k - 1$
DTLZ7	$k = 20$	4, 5, 6, 8, 10	$m + k - 1$

where  $K$  is the size of the population,  $x_t^k$  denotes the  $k$ th solution at  $t$ th generation.

In Algorithm 6, the nadir point  $z^{\text{nad}}$  is first updated (lines 1–3). After that,  $N$  direction vectors are generated by methods used in either [15] or [52] (line 4) and the neighborhood index set  $B^i$  for each direction vector  $\lambda^i$  is computed by selecting its  $T$  closest direction vectors (Algorithm 7, line 5). The newly generated direction vectors (subproblems) are assigned with existing solutions in line 6.

#### F. Adjustment for the Positions of the Ineffective Direction Vectors

After the number of direction vectors  $K$  are expanded from  $m$  to  $N$ , many ineffective direction vectors may exist in MaOPs with irregular PFs. These ineffective direction vectors are detected and adjusted (lines 13–15 of Algorithm 1) by calling Algorithm 9.

In Algorithm 9, the effective and ineffective direction vectors are first detected. The detection of the effective vectors is given in Algorithm 8 by a simple Pareto-dominance-based method as follows. Each nondominated solution is associated with a direction vector (lines 1 and 2). If a direction vector contains no associated nondominated solutions, the subregion such direction vector covers is very likely to contain no Pareto optimal solutions and this direction vector is considered to be ineffective (line 4–8); otherwise it is considered to be effective.

After detecting effective direction vectors  $\overline{DV}$  ( $|\overline{DV}| \leq |DV|$ ), the rest ( $|DV| - |\overline{DV}|$ ) ( $|DV| == K$ ) direction vectors are generated by inserting new ones at the midpoints between every two effective direction vectors. It can be known that

TABLE IV  
POPULATION SIZES FOR DIFFERENT ALGORITHMS ON MAOPs WITH DIFFERENT NUMBER OF OBJECTIVES

Number of Objectives (m)	Population Size (N)		
	GrEA	NSGA-III	decomposition-based MaOEA
3	300	300	300
4	220	220	220
5	212	212	210
6	180	184	182 ( $H_1 = 4, H_2 = 3$ )
8	156	156	156 ( $H_1 = 3, H_2 = 2$ )
10	276	276	275 ( $H_1 = 3, H_2 = 2$ )

The two-layer direction vector generation method [52] is applied for 6-, 8- and 10-objective problems.  $H_1$  is the number of divisions for the boundary layer and  $H_2$  is the numbers of divisions for the inside layer.

there are at most

$$\binom{|\overline{DV}|}{2} = \frac{|\overline{DV}| \times (|\overline{DV}| - 1)}{2} \quad (6)$$

possible pairs of effective direction vectors, whose indexes can be denoted by

$$\text{Pair} = \{(1, 2), \dots, (i, j), \dots, (|\overline{DV}| - 1, |\overline{DV}|)\}. \quad (7)$$

The distance vector of vector pairs can be denoted as

$$d = (d_{1,2}, \dots, d_{i,j}, \dots, d_{|\overline{DV}|-1, |\overline{DV}|}). \quad (8)$$

As  $(i, j)$  is equivalent to  $(j, i)$ , we only have  $i < j$  for convenience.

If  $\binom{|\overline{DV}|}{2}$  is smaller than  $K - |\overline{DV}|$ , all  $\binom{|\overline{DV}|}{2}$  pairs are used to generate new direction vectors (lines 3–9). This process is iterated until  $|\overline{DV}|$  exceeds  $K$  (line 2).

Otherwise,  $(K - |\overline{DV}|)$  pairs are selected from all possible  $\binom{|\overline{DV}|}{2}$  pairs to generate new direction vectors, as follows.

The distance of every  $i$ th direction vector to its nearest neighbor, denoted by  $d_{\min}(i)$ , is computed (lines 13–19). The pair index set, denoted by  $Mid$ , that has the maximum distance  $d_{\max}$  out of all possible  $d_{\min}(i)$ , is selected (lines 20–23). The lower index of  $Mid$  is stored as  $l$  and the upper index of  $Mid$  is stored as  $r$  (lines 24 and 25). The range between  $l$  and  $r$  are expanded, first move  $l$  backwards and then move  $r$  forwards, until the range  $|r - l + 1|$  reaches  $(N - |\overline{DV}|)$  (lines 26–31).  $(K - |\overline{DV}|)$  new direction vectors are generated by inserting midpoints of (*upper*–*lower*) vector pairs in  $\overline{DV}$  (lines 32–35). Lastly, the neighborhood index set  $B$  is updated for the new set of direction vectors by calling Algorithm 7.

#### G. Example for Algorithm 9

In this section, an example of the procedures for Algorithm 9 is given in Fig. 3 for a bi-objective problem. The piecewise lines are the real PF. The population size  $K$  is set to 8. Among all the eight direction vectors, six ( $a, b, c, f, g$ , and  $h$ ) are identified effective (arrowed solid lines) and the

TABLE V  
MEAN VALUES OF IGD, OBTAINED BY MOEA/D-DE, GREa, NSGA-III, MOEA/DD, AND MAOEA/D-2ADV ON DTLZ INSTANCES

Problem	M	MOEA/D-DE	GrEA	NSGA-III	MOEA/DD	MaOEA/D-2ADV	+/-
DTLZ1	3	1.064e-02 (8.32e-06) †	2.294e-02 (4.15e-03) †	1.145e-02 (3.52e-03) †	1.137e-02 (4.67e-06) †	<b>1.095e-02 (4.32e-04)</b>	-2.91%
	4	3.330e-02 (2.64e-04) †	7.754e-01 (2.10e-02) †	3.230e-02 (1.08e-03) †	<b>3.210e-02 (2.89e-06)</b> ‡	3.218e-02 (7.05e-05)	-0.25%
	5	5.690e-02 (6.78e-04) †	3.676e-01 (9.23e-02) †	5.193e-02 (1.78e-04) †	5.201e-02 (2.09e-05) †	<b>5.151e-02 (1.25e-04)</b>	+0.81%
	6	7.411e-02 (7.45e-04) †	3.625e-01 (6.05e-02) †	7.549e-02 (2.86e-02) †	6.797e-02 (2.92e-06) †	<b>6.329e-02 (1.05e-04)</b>	+6.89%
	8	9.471e-02 (7.15e-04) †	1.030e-01 (1.32e-02) †	2.001e-01 (5.15e-02) †	8.836e-02 (2.04e-05) †	<b>8.156e-02 (1.32e-04)</b>	+7.70%
10	9.681e-02 (7.15e-04) †	1.556e-01 (6.06e-02) †	2.318e-01 (4.40e-02) †	9.099e-02 (2.19e-04) †	<b>8.946e-02 (3.13e-04)</b>	+1.68%	
DTLZ2	3	3.463e-02 (2.14e-04) †	7.141e-02 (1.69e-03) †	<b>2.886e-02 (1.22e-04)</b> ‡	2.948e-02 (9.71e-07) ‡	3.272e-02 (4.48e-05)	-13.37%
	4	1.025e-01 (1.78e-03) †	3.771e-01 (1.77e-02) †	8.743e-02 (3.73e-05) ‡	<b>8.740e-02 (8.34e-07)</b> ‡	8.759e-02 (2.80e-04)	-0.22%
	5	1.762e-01 (5.16e-03) †	1.462e-01 (3.51e-03) †	<b>1.332e-01 (2.69e-04)</b> ‡	<b>1.332e-01 (2.05e-06)</b> ‡	1.333e-01 (1.62e-04)	-0.08%
	6	2.888e-01 (5.78e-03) †	2.543e-01 (1.96e-03) †	2.315e-01 (1.59e-04) †	<b>2.314e-01 (2.55e-06)</b> ‡	2.317e-01 (3.00e-05)	-0.13%
	8	4.467e-01 (8.44e-03) †	<b>3.585e-01 (3.25e-03)</b> †	5.535e-01 (2.36e-01) †	3.646e-01 (1.47e-05) ‡	3.650e-01 (8.63e-05)	-1.81%
10	4.937e-01 (6.42e-03) †	4.578e-01 (1.41e-02) †	7.467e-01 (2.24e-01) †	<b>4.106e-01 (2.60e-04)</b> ‡	4.121e-01 (1.87e-04)	-0.37%	
DTLZ3	3	3.883e-02 (2.01e-03) †	7.343e-02 (1.59e-02) †	3.260e-02 (4.75e-05) ‡	<b>3.217e-02 (3.13e-05)</b> ‡	3.538e-02 (1.63e-03)	-9.98%
	4	1.212e-01 (1.09e-02) †	3.696e-01 (1.66e-02) †	8.808e-02 (3.04e-03) ‡	<b>8.743e-02 (2.53e-05)</b> ‡	9.014e-02 (2.38e-03)	-3.10%
	5	4.319e-01 (8.55e-01) †	8.510e-01 (3.29e-01) †	1.341e-01 (3.25e-03) ‡	<b>1.333e-01 (1.46e-04)</b> ‡	1.374e-01 (2.45e-03)	-3.08%
	6	1.394e+00 (3.17e-00) †	9.452e-01 (1.92e-01) †	2.346e-01 (1.17e-02) †	<b>2.315e-01 (2.95e-05)</b> ‡	2.337e-01 (9.54e-04)	-0.95%
	8	1.671e+00 (6.73e+00) †	8.892e-01 (2.54e-01) †	6.850e-01 (2.48e-01) †	<b>3.652e-01 (5.40e-04)</b> ‡	3.679e-01 (8.54e-04)	-0.74%
10	4.941e-1 (6.43e-03) †	9.136e-01 (1.86e-01) †	8.784e-01 (9.81e-02) †	<b>4.123e-01 (6.69e-04)</b> ‡	4.161e-01 (1.65e-03)	-0.92%	
DTLZ4	3	3.775e-02 (5.77e-04) †	7.512e-02 (1.89e-03) †	3.255e-02 (3.89e-06) ‡	<b>3.214e-02 (1.07e-06)</b> ‡	3.280e-02 (6.22e-05)	-2.05%
	4	1.065e-01 (4.54e-04) †	3.777e-01 (1.52e-02) †	8.740e-02 (6.40e-06) ‡	<b>8.740e-02 (6.63e-07)</b> ‡	8.775e-02 (3.48e-04)	-0.40%
	5	1.628e-01 (2.37e-02) †	1.440e-01 (2.72e-03) †	<b>1.331e-01 (1.29e-04)</b> ‡	1.332e-01 (1.97e-06) ‡	1.334e-01 (1.54e-04)	-0.23%
	6	3.183e-01 (1.96e-02) †	2.570e-01 (2.31e-03) †	<b>2.313e-01 (1.79e-04)</b> ‡	2.314e-01 (3.59e-06) ‡	2.321e-01 (1.24e-04)	-0.35%
	8	4.836e-01 (1.16e-02) †	3.667e-01 (3.57e-03) †	3.654e-01 (1.26e-03) †	<b>3.646e-01 (3.20e-05)</b> ‡	3.720e-01 (3.73e-03)	-2.03%
10	5.265e-01 (5.96e-03) †	4.810e-01 (1.04e-02) †	4.159e-01 (2.87e-03) †	<b>4.109e-01 (4.97e-04)</b> ‡	4.405e-01 (5.76e-03)	-7.20%	
DTLZ5	3	1.820e-02 (3.65e-05) †	9.211e-03 (4.15e-04) †	4.427e-03 (5.42e-04) †	9.211e-03 (4.15e-04) †	<b>1.845E-03 (1.79E-04)</b>	+58.32%
	4	1.597e-02 (4.98e-05) †	4.394e-02 (1.14e-02) †	2.774e-02 (4.17e-03) †	6.556e-02 (7.08e-03) †	<b>6.992E-03 (8.05E-04)</b>	+56.32%
	5	1.995e-02 (1.05e-04) †	3.174e-02 (4.00e-03) †	9.550e-02 (2.82e-02) †	1.238e-01 (6.82e-03) †	<b>1.527E-02 (1.83E-03)</b>	+23.46%
	6	1.820e-02 (1.72e-04) †	1.151e-01 (2.08e-02) †	1.168e-01 (3.75e-02) †	1.194e-01 (1.78e-02) †	<b>1.434E-02 (2.99E-03)</b>	+21.21%
	8	2.618e-02 (4.65e-04) †	2.097e-01 (3.83e-02) †	3.256e-01 (1.15e-01) †	1.711e-01 (1.82e-02) †	<b>1.368E-02 (6.42E-03)</b>	+47.75%
10	2.043e-02 (4.04e-04) †	2.934e-01 (6.94e-02) †	4.465e-01 (1.11e-01) †	1.460e-01 (1.61e-02) †	<b>1.154E-02 (5.57E-03)</b>	+43.51%	
DTLZ6	3	1.836e-02 (2.90e-06) †	5.683e-02 (2.05e-02) †	3.936e-02 (2.26e-02) †	5.683e-02 (2.05e-02) †	<b>2.219E-03 (1.87E-04)</b>	+87.91%
	4	1.600e-02 (1.25e-04) †	4.092e-02 (1.48e-02) †	9.072e-02 (2.00e-02) †	1.164e-01 (2.45e-02) †	<b>5.888E-03 (5.67E-04)</b>	+63.20%
	5	1.982e-02 (3.27e-05) †	1.652e-01 (2.19e-02) †	1.364e+00 (9.48e-02) †	1.622e-01 (1.85e-02) †	<b>1.297E-02 (1.50E-03)</b>	+34.56%
	6	1.666e-02 (7.86e-04) †	9.195e-01 (6.38e-01) †	2.929e+00 (1.62e-01) †	2.863e-01 (1.85e-02) †	<b>1.545E-02 (2.87E-03)</b>	+7.26%
	8	2.536e-02 (2.35e-04) †	8.848e-01 (6.34e-01) †	4.014e+00 (4.49e-01) †	3.686e-01 (1.82e-02) †	<b>1.350E-02 (3.22E-03)</b>	+46.77%
10	1.997e-02 (1.99e-04) †	2.153e+00 (1.18e+00) †	4.961e+00 (4.73e-01) †	6.477e-01 (2.01e-02) †	<b>1.492E-02 (1.89E-03)</b>	+25.29%	
DTLZ7	3	1.479e-01 (2.20e-01) †	3.452e+00 (1.95e-01) †	3.771e-02 (9.35e-04) †	7.754e-02 (1.38e-03) †	<b>3.538e-02 (1.72e-03)</b>	+6.18%
	4	3.719e-01 (1.47e-01) †	2.512e+00 (7.66e-01) †	1.565e-01 (6.34e-03) †	3.223e-01 (1.12e-02) †	<b>1.543e-01 (7.38e-03)</b>	+1.41%
	5	7.633e-01 (6.15e-02) †	1.152e+00 (3.53e-02) †	4.116e-01 (1.28e-01) †	5.519e-01 (2.45e-02) †	<b>3.457e-01 (6.06e-03)</b>	+16.01%
	6	1.485e+00 (3.12e-02) †	<b>4.692e-01 (1.12e-02)</b> †	6.528e-01 (2.41e-02) †	9.463e-01 (8.79e-02) †	5.573e-01 (6.59e-02) †	-18.78%
	8	1.695e+00 (6.80e-02) †	7.907e-01 (6.67e-02) †	2.286e+00 (3.60e-01) †	1.559e+00 (1.81e-01) †	<b>7.425e-01 (1.19e-02)</b>	+6.10%
10	1.985e+00 (9.04e-02) †	<b>8.561e-01 (4.21e-02)</b> †	4.572e+00 (7.64e-01) †	1.881e+00 (1.90e-01) †	9.552e-01 (1.11e-01) †	-11.58%	

Wilcoxon's rank sum test at a 0.05 significance level is performed between MaOEA/D-2ADV and each of the other competing algorithms. † and ‡ denotes that the performance of the corresponding algorithm is significantly worse than or better than that of MaOEA/D-2ADV, respectively. The best mean IGD values are highlighted in boldface.

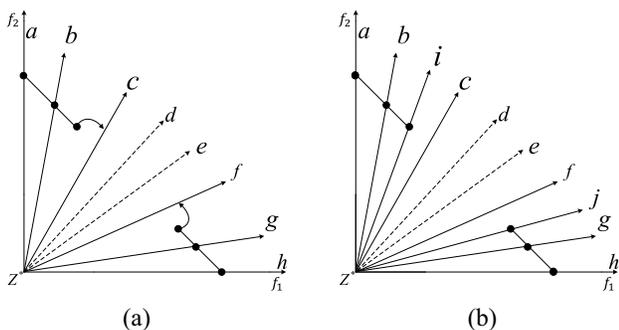


Fig. 3. Example of adjusting the positions of direction vectors (Algorithm 9) for a bi-objective problem. In (a), out of eight direction vectors, six (arrowed solid lines) are effective and two (arrowed dashed lines) are ineffective. In (b), one new direction vectors ( $i$ ) is inserted in the midpoint between  $b$  and  $c$  and another one ( $j$ ) is inserted in the midpoint between  $f$  and  $g$ .

other two are ineffective (arrowed dashed lines), as shown in Fig. 3(a). It is worth noting that direction vectors  $d$  and  $e$  are effective as there exist Pareto optimal solutions associated with them.

Based on the six effective direction vectors, the distance vector  $d_{i,j}$  of all the possible vector pairs is calculated, as well as the distance of every  $i$ th direction vector to its nearest neighbor  $d_{\min}(i)$  (lines 13–19). The maximum distance  $d_{\max}$  out of all possible  $d_{\min}(i)$  is calculated (line 20). The values of  $d_{i,j}$ ,  $d_{\min}(i)$  and  $d_{\max}$  are presented in Table II, where the value of  $d_{\max} = 0.2970$  is highlighted in boldface.

After that, all the distance vectors  $d$  are sorted in an ascending order (line 21) and all the vector pairs, Pair, are sorted based on  $d$  (line 22). The sorted distance vector,  $d = (d_{a,b}, d_{g,h}, \dots, d_{a,h}) = (0.1838, 0.1979, \dots, 1.4141)$  and its corresponding pair vector, Pair =  $\{(a, b), (g, h), \dots, (a, h)\}$  are presented in Table I. Then,  $d_{\max}$  is tracked in  $d$ , whose index vector  $Mid$  is 4 (only one variable in  $Mid$ , also  $l = r = 4$ ) (lines 23–25). After two while loops (lines 26–31), the value of  $l$  is 3 and the value of  $r$  is 4. The corresponding vector pair Pair(3) [i.e.,  $(b, c)$ ] and Pair(4) [i.e.,  $(f, g)$ ] are selected. Two new direction vectors ( $i$  and  $j$ ) are inserted in the midpoints of the selected pairs (lines 32–35), as shown in Fig. 3(b). After the adjustment, the effective direction vectors are  $a, b, g, h, i$ , and  $j$  and the direction vectors  $c$  and  $f$  become

TABLE VI  
MEAN VALUES OF IGD, OBTAINED BY MOEA/D-AWA, RVEA, AND MAOEA/D-2ADV ON DTLZ INSTANCES

Problem	M	MOEA/D-AWA	RVEA	MaOEA/D-2ADV	+/-
DTLZ1	3	1.487e-02 (3.26e-04) †	1.477e-02 (1.86e-04) †	<b>1.095e-02 (4.32e-04)</b>	+25.86%
	4	5.692e-02 (3.42e-03) †	1.444e-01 (2.91e-02) †	<b>3.218e-02 (7.05e-05)</b>	+43.46%
	5	1.010e-01 (9.75e-03) †	1.671e-01 (1.52e-02) †	<b>5.151e-02 (1.25e-04)</b>	+49.00%
	6	8.922e-02 (2.29e-03) †	6.807e-02 (5.63e-05) †	<b>6.329e-02 (1.05e-04)</b>	+7.02%
	8	1.212e-01 (7.21e-03) †	8.860e-02 (2.19e-04) †	<b>8.156e-02 (1.32e-04)</b>	+7.95%
	10	1.167e-01 (2.27e-03) †	9.171e-02 (2.49e-04) †	<b>8.946e-02 (3.13e-04)</b>	+2.45%
DTLZ2	3	4.522e-02 (6.94e-04) †	3.920e-02 (6.78e-04) †	<b>3.272e-02 (4.48e-05)</b>	+16.53%
	4	1.313e-01 (6.95e-03) †	1.767e-01 (1.12e-03) †	<b>8.759e-02 (2.80e-04)</b>	+33.29%
	5	2.014e-01 (9.96e-03) †	2.674e-01 (1.01e-03) †	<b>1.333e-01 (1.62e-04)</b>	+33.81%
	6	3.491e-01 (9.47e-03) †	<b>2.317e-01 (7.01e-05)</b>	<b>2.317e-01 (3.00e-05)</b>	0.00%
	8	5.298e-01 (1.87e-02) †	3.654e-01 (2.96e-04) †	<b>3.650e-01 (8.63e-05)</b>	+0.11%
	10	5.801e-01 (1.67e-02) †	4.151e-01 (1.09e-03) †	<b>4.121e-01 (1.87e-04)</b>	+0.72%
DTLZ3	3	4.725e-02 (2.12e-03) †	4.076e-02 (2.41e-03) †	<b>3.538e-02 (1.63e-03)</b>	+13.20%
	4	1.615e-01 (1.46e-02) †	4.054e-01 (3.77e-01) †	<b>9.014e-02 (2.38e-03)</b>	+44.19%
	5	2.438e+00 (1.94e+00) †	6.400e+00 (4.57e+00) †	<b>1.374e-01 (2.45e-03)</b>	+94.36%
	6	4.394e-01 (5.56e-00) †	<b>2.321e-01 (3.92e-04)</b> ‡	2.337e-01 (9.54e-04)	-0.69%
	8	2.296e-01 (3.36e-01) †	<b>3.668e-01 (1.04e-03)</b> ‡	3.679e-01 (8.54e-04)	-0.30%
	10	1.715e-01 (1.98e+00) †	4.222e-01 (1.84e-03) †	<b>4.161e-01 (1.65e-03)</b>	+1.44%
DTLZ4	3	4.430e-02 (7.73e-04) †	4.197e-02 (1.15e-03) †	<b>3.280e-02 (6.22e-05)</b>	+21.85%
	4	1.391e-01 (6.41e-03) †	1.768e-01 (1.16e-03) †	<b>8.775e-02 (3.48e-04)</b>	+36.92%
	5	2.136e-01 (7.89e-03) †	2.681e-01 (2.40e-03) †	<b>1.334e-01 (1.54e-04)</b>	+37.55%
	6	3.124e-01 (1.39e-02) †	<b>2.320e-01 (1.38e-04)</b> ‡	2.321e-01 (1.24e-04)	-0.04%
	8	4.804e-01 (1.44e-02) †	<b>3.667e-01 (8.06e-04)</b> ‡	3.720e-01 (3.73e-03)	-1.45%
	10	5.400e-01 (8.67e-03) †	<b>4.209e-01 (9.19e-03)</b> ‡	4.405e-01 (5.76e-03)	-4.66%
DTLZ5	3	3.930e-03 (2.75e-04) †	3.759e-02 (1.30e-03) †	<b>1.845e-03 (1.79e-04)</b>	+53.05%
	4	2.676e-02 (5.28e-03) †	9.341e-02 (1.54e-02) †	<b>6.992e-03 (8.05e-04)</b>	+73.87%
	5	2.860e-02 (4.04e-03) †	1.786e-01 (2.42e-02) †	<b>1.527e-02 (1.83e-03)</b>	+46.61%
	6	<b>1.386e-02 (1.59e-03)</b> ‡	3.805e-01 (2.25e-01) †	1.434e-02 (2.99e-03)	-3.46%
	8	1.870e-02 (3.59e-03) †	4.458e-01 (2.26e-01) †	<b>1.368e-02 (6.42e-03)</b>	+26.84%
	10	<b>8.609e-03 (1.25e-03)</b> ‡	2.317e-01 (1.27e-01) †	1.154e-02 (5.57e-03)	-34.05%
DTLZ6	3	3.440e-04 (2.79e-04) †	5.031e-02 (7.05e-03) †	<b>2.219e-03 (5.67e-04)</b>	+35.49%
	4	2.276e-03 (1.04e-03) †	2.382e-01 (4.46e-02) †	<b>5.888e-03 (5.67e-04)</b>	+74.13%
	5	2.459e-02 (3.21e-03) †	3.550e-01 (4.82e-02) †	<b>1.297e-02 (1.50e-03)</b>	+47.25%
	6	<b>1.258e-02 (2.37e-03)</b> ‡	3.496e-01 (9.57e-02) †	1.545e-02 (2.87e-03)	-22.81%
	8	2.014e-02 (2.19e-03) †	4.165e-01 (1.72e-01) †	<b>1.350e-02 (3.22e-03)</b>	+32.97%
	10	<b>1.053e-02 (3.22e-03)</b> ‡	4.611e-01 (1.40e-01) †	1.492e-02 (1.89e-03)	-41.69%
DTLZ7	3	7.847e-02 (3.20e-03) †	7.386e-02 (2.77e-03) †	<b>3.538e-02 (1.72e-03)</b>	+52.10%
	4	4.270e-01 (1.92e-02) †	<b>1.363e-01 (4.42e-02)</b> ‡	1.543e-01 (7.38e-03)	-13.21%
	5	7.234e-01 (1.39e-01) †	<b>3.008e-01 (6.17e-02)</b> ‡	3.457e-01 (6.06e-03)	-14.93%
	6	8.347e-01 (6.33e-02) †	<b>5.446e-01 (2.24e-01)</b> ‡	5.573e-01 (6.59e-03)	-2.33%
	8	1.324e-00 (9.49e-02) †	1.177e+00 (5.53e-01) †	<b>7.425e-01 (1.19e-02)</b>	+36.92%
	10	1.445e+00 (1.26e-01) †	1.257e+00 (5.10e-01) †	<b>9.552e-01 (1.11e-01)</b>	+24.01%

Wilcoxon's rank sum test at a 0.05 significance level is performed between MaOEA/D-2ADV and each of the other competing algorithms. † and ‡ denotes that the performance of the corresponding algorithm is significantly worse than or better than that of MaOEA/D-2ADV, respectively. The best mean *IGD* values are highlighted in boldface.

ineffective as no Pareto optimal solutions are associated with them anymore.

### H. Computational Complexity

In MaOEA/D-2ADV (Algorithm 1), the initialization procedure (Algorithm 2) requires  $O(m)$  computations, where  $m$  is the number of objectives. The generation of offspring (Algorithm 3) needs  $O(K)$  computations, where  $K = m$  or  $N$  is the current population size. The computational complexity of the association procedure (Algorithm 4) is  $O(mK^2)$ . The procedure of the association-based selection (Algorithm 5) includes the space reduction, which needs  $O(mK)$  computations and the selection, which needs  $O(K)$  computations.

For two types of adjustment for direction vectors in MaOEA/D-2ADV, the adjustment for the number of the direction vectors (Algorithm 6) is carried out only once, which needs  $O(mK^2)$  computations. The adjustment for the positions of the ineffective direction vectors (Algorithm 9) includes two steps: 1) the detection of the effective direction vector (Algorithm 8) and 2) the subsequential adjustments. The former step needs  $O(mK^2)$  computations and the latter step needs

$O(mL^2)$  computations, where  $L$  is the number of the effective direction vectors.

In summary, the worst computational complexity of MaOEA/D-2ADV (Algorithm 1) within one generation is  $O(mN^2)$ .

## V. EXPERIMENTAL STUDIES AND DISCUSSION

DTLZ [21] test suite is used as the benchmark problems in our experiments. Its setups are listed in Table III. In the experiments, each algorithm was run 30 times independently for each benchmark problem. A maximum of 300 000 function evaluations is given to all the compared algorithms. The population sizes of all the compared algorithm for MaOPs with different number of objectives are listed in Table IV. To make fair comparisons, PBI is used in all the decomposition-based algorithms, except for RVEA [10]. The angle-penalized distance [10] remains being adopted in RVEA. Inverted generational distance (IGD) [12] metric is used to measure the performance of compared algorithms. Generation interval parameter  $\phi_1$  is set to 500 and  $\phi_2$  is set to 50 in all the experimental studies.

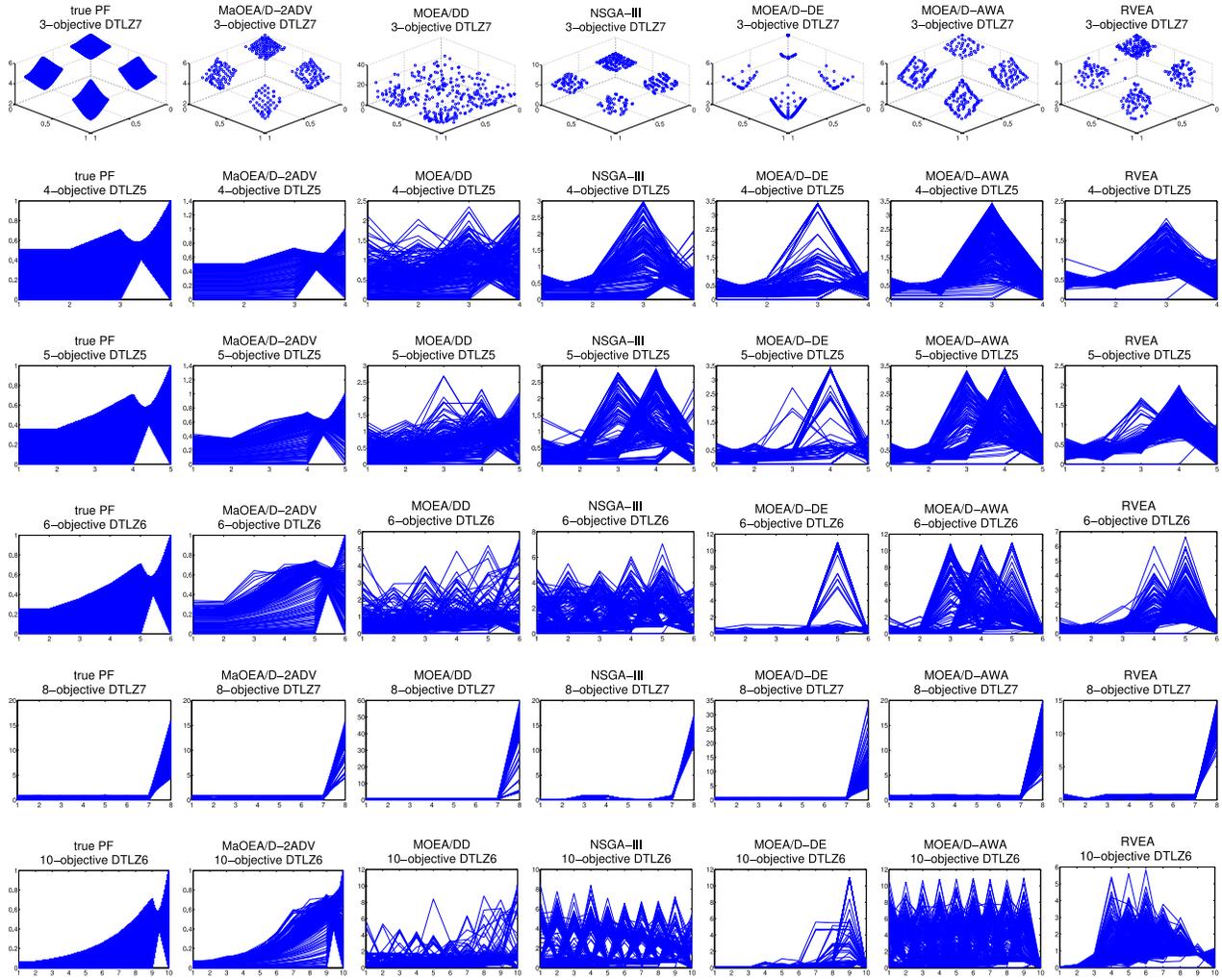


Fig. 4. Parallel coordinate plots for true PFs and PF approximations obtained by MaOEA/D-2ADV, MOEA/DD, NSGA-III, MOEA/D-DE, MOEA/D-AWA, and RVEA on MaOPs with irregular PFs (DTLZ5–DTLZ7).

To investigate the performance and understand the behavior of MaOEA/D-2ADV, the experimental studies are conducted on the following.

- 1) Comparisons of MaOEA/D-2ADV with many-objective optimizers (MOEA/D-DE [51], GrEA [69], NSGA-III [17], and MOEA/DD [52]).
- 2) Comparisons of MaOEA/D-2ADV with decomposition-based many-objective optimizers with the adjustments of direction vectors (MOEA/D-AWA [58] and RVEA [10]).
- 3) Sensitivity analysis of generational interval parameters  $\phi_1$  and  $\phi_2$ .
- 4) The effects of fast convergence (see Section I in the supplementary material).
- 5) The effects of adjustments for the positions of the directions vectors (see Sections II and III in the supplementary material).

#### A. Comparison With State-of-the-Art MaOEs

In this section, MaOEA/D-2ADV is compared with four many-objective optimizers: 1) MOEA/D-DE [51];

2) GrEA [69]; 3) NSGA-III [17]; and 4) MOEA/DD [52]. The performances of all the compared algorithms, in terms of IGD, is presented in Table V. Wilcoxon's rank sum test at a 0.05 significance level is performed between the MaOEA/D-2ADV and each of the other competing algorithms. The best mean IGD values are highlighted in boldface. A positive number in last column of Table V indicates the degree of IGD value obtained by MaOEA/D-2ADV over that obtained by the second best algorithm while a negative value indicates the degree of IGD value obtained by the best algorithm over that obtained by MaOEA/D-2ADV.

It can be observed that MaOEA/D-2ADV is significantly better than MOEA/D-DE, on all the test problems. Meanwhile, MaOEA/D-2ADV is significantly better than all the four compared algorithms on DTLZ1, DTLZ5, and DTLZ6, except for four-objective DTLZ1. MOEA/DD achieves very good performance on DTLZ2–DTLZ4, although its IGD values are actually very close to that of MaOEA/D-2ADV on these problems. Furthermore, MaOEA/D-2ADV is always significantly better than all the compared algorithms on DTLZ5 and DTLZ6 which have degenerate PFs.

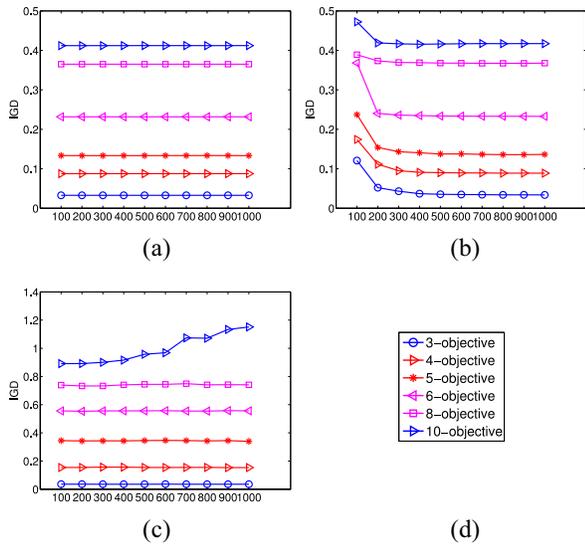


Fig. 5. Mean IGD values obtained by MaOEA/D-2ADV with different values of interval parameter  $\phi_1$  on (a) DTLZ2, (b) DTLZ3, and (c) DTLZ7 over 30 runs. (d) Legend.

### B. Comparisons With State-of-the-Art Decomposition-Based MOEAs With the Adjustment of Direction Vectors

In this section, MaOEA/D-2ADV is compared with MOEA/D-AWA [58] and RVEA [10], two state-of-the-art decomposition-based MOEAs that also use the adjustments of the direction vectors to address MaOPs with irregular PFs. The results of three compared algorithms are presented in Table VI. It can be observed that MaOEA/D-2ADV has significantly better performance than that of MOEA/D-AWA and RVEA on most test problems. It is also worth to note that MaOEA/D-2ADV achieves remarkable improvements over MOEA/D-AWA and RVEA on the performance of DTLZ5 and DTLZ6, whose PFs are degenerate.

To visualize the performance of the six compared algorithms on MaOPs with irregular PFs [degenerate ones (DTLZ5 and DTLZ6) or disconnected one (DTLZ7)], the parallel coordinate plots for true PFs and PF approximations obtained by MaOEA/D-2ADV, MOEA/DD, NSGA-III, MOEA/D, MOEA/D-AWA, and RVEA on different test problems are shown in Fig. 4. It can be observed that the PF approximations obtained by MaOEA/D-2ADV is the closest to the true PFs, in terms of both convergence and diversity. These observations indicate that MaOEA/D-2ADV performs the best among all the compared algorithms on MaOPs with irregular PFs, which is consistent with our motivations in Section II.

### C. Sensitivity Analysis of $\phi_1$ and $\phi_2$

$\phi_1$  is a generational interval parameter for adjusting the number of direction vectors and  $\phi_2$  is a generational interval parameter for adjusting the positions of the ineffective direction vectors. In this section, the sensitivity analysis of  $\phi_1$  and  $\phi_2$  is conducted as follows.

Fig. 5 shows the performance, in terms of IGD, of MaOEA/D-2ADV with different  $\phi_1$  values (100–1000) on

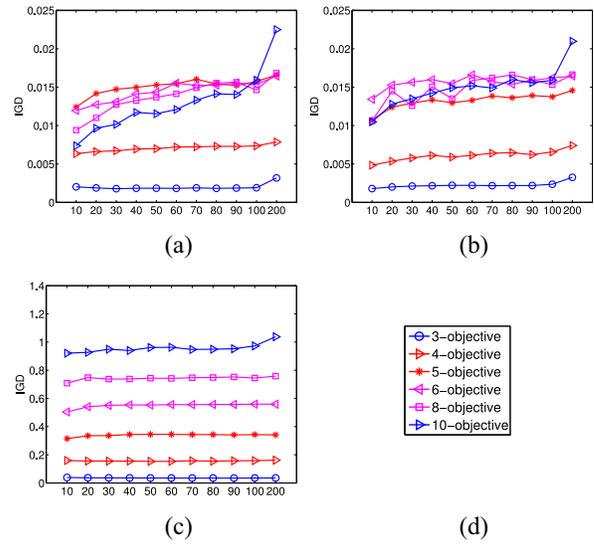


Fig. 6. Mean IGD values obtained by MaOEA/D-2ADV with different values of interval parameter  $\phi_2$  on DTLZ5–DTLZ7 over 30 runs. (a) DTLZ5. (b) DTLZ6. (c) DTLZ7. (d) Legend.

DTLZ2, DTLZ3, and DTLZ7 over 30 independent runs. It can be observed that, in general, MaOEA/D-2ADV is very robust with regard to  $\phi_1$ . However, the optimal value of  $\phi_1$  is problem-dependent.  $\phi_1 = 400$  or  $500$  may be the best choice for most problems.

Fig. 6 shows the performance, in terms of IGD, of MaOEA/D-2ADV with different  $\phi_2$  values (10–200) on three irregular benchmark problems (DTLZ5–DTLZ7). It is clear to see that MaOEA/D-2ADV has better performance when decreasing the value of  $\phi_2$ . This observation indicates that better performance of MaOEA/D-2ADV can be achieved by increasing the frequency of the adjustments for the positions of the ineffective direction vectors. However, there is an obvious tradeoff between the number of adjustments and the computational cost.

## VI. CONCLUSION

In this paper, we propose a decomposition-based many-objective evolutionary algorithm with two types of adjustments for the direction vectors (MaOEA/D-2ADV). The first type aims to expand the number of direction vectors after the fast convergence along the boundary direction vectors, for approximating more complete PFs and the second type changes the positions of the ineffective direction vectors by iteratively deleting ineffective ones and inserting new ones between effective direction vectors for MaOPs with irregular PFs (e.g., disconnected and degenerate PFs). In addition, a simple Pareto-dominance-based approach is proposed to detect the effectiveness of each direction vector. MaOEA/D-2ADV is compared with four state-of-the-art MaOEA and two MaOEA with the adjustments of the direction vectors. The experimental studies show that MaOEA/D-2ADV outperforms other algorithms on most test problems and it is especially effective on MaOPs with irregular PFs.

## REFERENCES

- [1] S. F. Adra and P. J. Fleming, "Diversity management in evolutionary many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 15, no. 2, pp. 183–195, Apr. 2011.
- [2] H. Aguirre, A. Oyama, and K. Tanaka, "Adaptive  $\varepsilon$ -sampling and  $\varepsilon$ -hood for evolutionary many-objective optimization," in *Evolutionary Multi-Criterion Optimization*. Heidelberg, Germany: Springer, 2013, pp. 322–336.
- [3] H. Aguirre and K. Tanaka, "A hybrid scalarization and adaptive  $\varepsilon$ -ranking strategy for many-objective optimization," in *Parallel Problem Solving From Nature, PPSN XI*. Heidelberg, Germany: Springer, 2010, pp. 11–20.
- [4] M. Asafuddoula, T. Ray, and R. A. Sarker, "A decomposition-based evolutionary algorithm for many objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 3, pp. 445–460, Jun. 2015.
- [5] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, 2011.
- [6] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *Eur. J. Oper. Res.*, vol. 181, no. 3, pp. 1653–1669, 2007.
- [7] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 174–188, Apr. 2003.
- [8] D. Brockhoff and E. Zitzler, "Offline and online objective reduction in evolutionary multiobjective optimization based on objective conflicts," Comput. Eng. Netw. Lab., ETH Zurich, Zürich, Switzerland, TIK Tech. Rep. 269, 2007.
- [9] Q. Cai, M. Gong, S. Ruan, Q. Miao, and H. Du, "Network structural balance based on evolutionary multiobjective optimization: A two-step approach," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 903–916, Dec. 2015.
- [10] R. Cheng, Y. Jin, M. Olhofer, and B. Sendhoff, "A reference vector guided evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 5, pp. 773–791, Oct. 2016.
- [11] C. A. C. Coello, "Evolutionary multi-objective optimization: A historical view of the field," *IEEE Comput. Intell. Mag.*, vol. 1, no. 1, pp. 28–36, Feb. 2006.
- [12] C. A. C. Coello and N. C. Cortés, "Solving multiobjective optimization problems using an artificial immune system," *Genet. Program. Evol. Mach.*, vol. 6, no. 2, pp. 163–190, 2005.
- [13] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, vol. 5. New York, NY, USA: Springer, 2007.
- [14] C. A. C. Coello and C. S. P. Zacetenco, "20 years of evolutionary multi-objective optimization: What has been done and what remains to be done," in *Computational Intelligence: Principles and Practice*. Piscataway, NJ, USA: IEEE Comput. Intell. Soc., 2006, pp. 73–88.
- [15] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems," *SIAM J. Optim.*, vol. 8, no. 3, pp. 631–657, 1998.
- [16] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: Wiley, 2001.
- [17] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 577–601, Aug. 2014.
- [18] K. Deb, M. Mohan, and S. Mishra, "Evaluating the  $\varepsilon$ -domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evol. Comput.*, vol. 13, no. 4, pp. 501–525, 2005.
- [19] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [20] K. Deb and D. Saxena, "Searching for Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," in *Proc. World Congr. Comput. Intell. (WCCI)*, 2006, pp. 3352–3360.
- [21] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multiobjective Optimization*. London, U.K.: Springer, 2005.
- [22] J. J. Durillo, A. J. Nebro, F. Luna, and E. Alba, "On the effect of the steady-state selection scheme in multi-objective genetic algorithms," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, Nantes, France, 2009, pp. 183–197.
- [23] M. Emmerich, N. Beume, and B. Naujoks, "An EMO algorithm using the hypervolume measure as selection criterion," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, Guanajuato, Mexico, 2005, pp. 62–76.
- [24] M. G. Fabre, G. T. Pulido, and C. A. C. Coello, "Alternative fitness assignment methods for many-objective optimization problems," in *Artificial Evolution*. Heidelberg, Germany: Springer, 2009, pp. 146–157.
- [25] C. M. Fonseca and P. J. Fleming, "An overview of evolutionary algorithms in multiobjective optimization," *Evol. Comput.*, vol. 3, no. 1, pp. 1–16, 1995.
- [26] C. M. Fonseca and P. J. Fleming, "Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 28, no. 1, pp. 26–37, Jan. 1998.
- [27] D. Gong, J. Sun, and X. Ji, "Evolutionary algorithms with preference polyhedron for interval multi-objective optimization problems," *Inf. Sci.*, vol. 233, pp. 141–161, Jun. 2013.
- [28] D. Gong, G. Wang, and X. Sun, "Set-based genetic algorithms for solving many-objective optimization problems," in *Proc. Comput. Intell.*, Guildford, U.K., 2013, pp. 96–103.
- [29] F.-Q. Gu and H.-L. Liu, "A novel weight design in multi-objective evolutionary algorithm," in *Proc. Int. Conf. Comput. Intell. Security*, 2010, pp. 137–141.
- [30] F. Gu, H.-L. Liu, and K. C. Tan, "A multiobjective evolutionary algorithm using dynamic weight design method," *Int. J. Innov. Comput. Inf. Control*, vol. 8, no. 5, pp. 3677–3688, 2012.
- [31] Z. He and G. G. Yen, "Many-objective evolutionary algorithm: Objective space reduction and diversity improvement," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 145–160, Feb. 2016.
- [32] E. J. Hughes, "Multiple single objective Pareto sampling," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 4. Canberra, ACT, Australia, 2003, pp. 2678–2684.
- [33] E. J. Hughes, "MSOPS-II: A general-purpose many-objective optimizer," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Singapore, 2007, pp. 3944–3951.
- [34] H. Ishibuchi, N. Akedo, and Y. Nojima, "Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems," *IEEE Trans. Evol. Comput.*, vol. 19, no. 2, pp. 264–283, Apr. 2015.
- [35] H. Ishibuchi, Y. Hitotsuyanagi, N. Tsukamoto, and Y. Nojima, "Use of biased neighborhood structures in multiobjective memetic algorithms," *Soft Comput.*, vol. 13, nos. 8–9, pp. 795–810, 2009.
- [36] H. Ishibuchi, Y. Setoguchi, H. Masuda, and Y. Nojima, "Performance of decomposition-based many-objective algorithms strongly depends on Pareto front shapes," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 169–190, Apr. 2016.
- [37] H. Ishibuchi, N. Tsukamoto, Y. Hitotsuyanagi, and Y. Nojima, "Effectiveness of scalability improvement attempts on the performance of NSGA-II for many-objective problems," in *Proc. 10th Annu. Conf. Genet. Evol. Comput.*, Atlanta, GA, USA, 2008, pp. 649–656.
- [38] H. Ishibuchi, N. Tsukamoto, and Y. Nojima, "Evolutionary many-objective optimization: A short review," in *Proc. IEEE World Congr. Comput. Intell. Evol. (CEC)*, Hong Kong, 2008, pp. 2419–2426.
- [39] A. L. Jaimes and C. A. C. Coello, "Study of preference relations in many-objective optimization," in *Proc. 11th Annu. Conf. Genet. Evol. Comput.*, Montreal, QC, Canada, 2009, pp. 611–618.
- [40] A. L. Jaimes, C. A. C. Coello, and J. E. U. Barrientos, "Online objective reduction to deal with many-objective problems," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, Nantes, France, 2009, pp. 423–437.
- [41] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach, part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.
- [42] S. Jiang, J. Zhang, and Y. S. Ong, "Asymmetric Pareto-adaptive scheme for multiobjective optimization," in *Proc. Aust. Joint Conf. Artif. Intell.*, Perth, WA, Australia, 2011, pp. 351–360.
- [43] Y. Jin and B. Sendhoff, "Connectedness, regularity and the success of local search in evolutionary multi-objective optimization," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 3. Canberra, ACT, Australia, 2003, pp. 1910–1917.
- [44] S. Kalbousi, S. Bechikh, M. Kessentini, and L. B. Said, "Preference-based many-objective evolutionary testing generates harder test cases for autonomous agents," in *Search Based Software Engineering*. Heidelberg, Germany: Springer, 2013, pp. 245–250.
- [45] J. Knowles and D. Corne, "Quantifying the effects of objective space dimension in evolutionary multiobjective optimization," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, Matsushima, Japan, 2007, pp. 757–771.

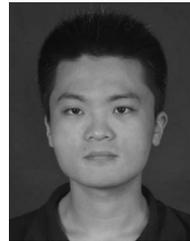
- [46] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining convergence and diversity in evolutionary multiobjective optimization," *Evol. Comput.*, vol. 10, no. 3, pp. 263–282, 2002.
- [47] K. Le, D. Landa-Silva, and H. Li, "An improved version of volume dominance for multi-objective optimisation," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, Nantes, France, 2009, pp. 231–245.
- [48] K.-B. Lee and J.-H. Kim, "Multi-objective particle swarm optimization with preference-based sorting," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, New Orleans, LA, USA, 2011, pp. 2506–2513.
- [49] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms: A survey," *ACM Comput. Surveys*, vol. 48, no. 1, p. 13, 2015.
- [50] H. Li and D. Landa-Silva, "An adaptive evolutionary multi-objective approach based on simulated annealing," *Evol. Comput.*, vol. 19, no. 4, pp. 561–595, 2011.
- [51] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 284–302, Apr. 2009.
- [52] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.
- [53] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for Pareto-based algorithms in many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 348–365, Jun. 2014.
- [54] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, MA, USA: Kluwer Acad., 1999.
- [55] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. Springer, 2006.
- [56] R. C. Purshouse and P. J. Fleming, "Evolutionary many-objective optimisation: An exploratory analysis," in *Proc. Congr. Evol. Comput. (CEC)*, Canberra, ACT, Australia, 2004, pp. 2066–2073, vol. 3.
- [57] R. C. Purshouse and P. J. Fleming, "On the evolutionary optimization of many conflicting objectives," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 770–784, Dec. 2007.
- [58] Y. Qi *et al.*, "MOEA/D with adaptive weight adjustment," *Evol. Comput.*, vol. 22, no. 2, pp. 231–264, 2014.
- [59] H. Sato, H. Aguirre, and K. Tanaka, "Controlling dominance area of solutions in multiobjective evolutionary algorithms and performance analysis on multiobjective 0/1 knapsack problems," *Inf. Media Technol.*, vol. 2, no. 4, pp. 1113–1128, 2007.
- [60] H. Sato, H. E. Aguirre, and K. Tanaka, "Controlling dominance area of solutions and its impact on the performance of MOEAs," in *Proc. Int. Conf. Evol. Multi Criterion Optim.*, Matsushima, Japan, 2007, pp. 5–20.
- [61] D. K. Saxena, J. A. Duro, A. Tiwari, K. Deb, and Q. Zhang, "Objective reduction in many-objective optimization: Linear and nonlinear algorithms," *IEEE Trans. Evol. Comput.*, vol. 17, no. 1, pp. 77–99, Feb. 2013.
- [62] H. K. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 15, no. 4, pp. 539–556, Aug. 2011.
- [63] K. C. Tan, E. F. Khor, and T. H. Lee, *Multiobjective Evolutionary Algorithms and Applications*. Springer, 2006.
- [64] H. Wang and X. Yao, "Corner sort for Pareto-based many-objective optimization," *IEEE Trans. Cybern.*, vol. 44, no. 1, pp. 92–102, Jan. 2014.
- [65] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired coevolutionary algorithms for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 4, pp. 474–494, Aug. 2013.
- [66] R. Wang, R. C. Purshouse, and P. J. Fleming, "Preference-inspired co-evolutionary algorithms using weight vectors," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 423–441, 2015.
- [67] R. Wang, R. C. Purshouse, I. Giagkiozis, and P. J. Fleming, "The iPCEA-g: A new hybrid evolutionary multi-criteria decision making approach using the brushing technique," *Eur. J. Oper. Res.*, vol. 243, no. 2, pp. 442–453, 2015.
- [68] Z. Xie, J. Zheng, and J. Zou, "A-dominance: Angle based preference multi-objective evolutionary algorithm," *Int. J. Adv. Comput. Technol.*, vol. 4, no. 10, pp. 265–273, 2012.
- [69] S. Yang, M. Li, X. Liu, and J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 17, no. 5, pp. 721–736, Oct. 2013.
- [70] Y. Yuan, H. Xu, B. Wang, and X. Yao, "A new dominance relation-based evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 20, no. 1, pp. 16–37, Feb. 2016.
- [71] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.
- [72] X. Zhang, Y. Tian, and Y. Jin, "A knee point-driven evolutionary algorithm for many-objective optimization," *IEEE Trans. Evol. Comput.*, vol. 19, no. 6, pp. 761–776, Dec. 2015.
- [73] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Proc. Int. Conf. Parallel Problem Solving Nat.*, Birmingham, U.K., 2004, pp. 832–842.
- [74] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach," *IEEE Trans. Evol. Comput.*, vol. 3, no. 4, pp. 257–271, Nov. 1999.



**Xinye Cai** (M'10) received the B.Sc. degree in information engineering from the Huazhong University of Science and Technology, Wuhan, China, in 2004, the M.Sc. degree in electronic engineering from the University of York, York, U.K., in 2006, and the Ph.D. degree in electrical engineering from Kansas State University, Manhattan, KS, USA, in 2009.

He is an Associate Professor with the Department of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, where he is currently leading the Intelligent Optimization Research Group. His current research interests include evolutionary computation, optimization, data mining, and their applications.

Dr. Cai was a recipient of the Evolutionary Many-Objective Optimization Competition at the Congress of Evolutionary Computation 2017.



**Zhiwei Mei** received the master's degree in computer science from the College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2017.

He is currently an Engineer with MeiTuan, Beijing, China. His current research interests include evolutionary computation and multiobjective optimization.



**Zhun Fan** (SM'14) received the B.S. and M.S. degrees in control engineering from the Huazhong University of Science and Technology, Wuhan, China, in 1995 and 2000, respectively, and the Ph.D. degree in electrical and computer engineering from Michigan State University, East Lansing, MI, USA, in 2004.

He is the Head of the Department of Electronic Engineering and a Full Professor with Shantou University, Shantou, China. Prior to joining Shantou University, he was an Assistant Professor, and then an Associate Professor with the Technical University of Denmark, Kongens Lyngby, Denmark. He was with the BEACON Center for Study of Evolution in Action, Michigan State University. His current research interests include intelligent control and robotic systems, robot vision and cognition, MEMS, computational intelligence, design automation, and optimization of mechatronic systems