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

MOST real-world optimization problems involve 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 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 (MaOEAs) have emerged. These MaOEAs 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 subcategories.

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 ε-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 dominance and convergence approaches [17], [28], [31], [52], [62]. For instance, Deb and Jain [17], [41] proposed a reference-point-based MaOEAs (NSGA-III) as an extension of NSGA-II [19]. A MaOEAs 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 MaOEAs.  

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 MaOEAs 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, MaOEAs/D-2ADV, is detailed in Section IV. In Section V, the experimental results are analyzed and discussed in detail. The effects of components in MaOEAs/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

\[
\text{minimize } F(x) = (f_1(x), \ldots, f_m(x))^T \\
\text{subject to } x \in \Omega
\]  

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

Let \( u, v \in \mathbb{R}^m \), \( u \) is said to **dominate** \( v \), denoted by \( u < v \), if and only if \( u_j \leq v_j \) for every \( j \in \{1, \ldots, m\} \) and \( u_k < v_k \) for at least one index \( k \in \{1, \ldots, m\} \). Given a set \( S \) in \( \mathbb{R}^m \), 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].

\( ^{1} \)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 $z^* = (z^*_1, \ldots, z^*_m)^T$, where $z^*_j = \min_{x \in \Omega} f_j(x), j \in \{1, \ldots, m\}$. The nadir objective vector $z^{\text{nad}}$ is a vector $z^{\text{nad}} = (z^{\text{nad}}_1, \ldots, z^{\text{nad}}_m)^T$, where $z^{\text{nad}}_j = \max_{x \in \Omega} f_j(x), i \in \{1, \ldots, m\}$.

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

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

(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 non-dominated solutions to approximate the PF in [29] and [30]. However, as it is difficult to give a method of estimating the sparsity of both working population and non-dominated 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 non-dominated 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 non-dominated solutions, stored in an external archive. The extra computational time is needed for density estimation and non-dominated 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, 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 non-dominated 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.
and 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 ■). The nadir point \(z_{\text{nadir}}\), 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 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.\(^2\) 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.

\(^2\)Assume PBI is used with a large \(\theta\) value for decomposition.

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 MaOEAs. 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
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 \(DV = \{\lambda^1, \ldots, \lambda^K\}\), 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.

3) The neighborhood index sets \(B = \{B^1, \ldots, B^K\}\), 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 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 \([P, DV, B, z^*, z^{nadir}] = \text{INITIALIZATION}(K, m);\)
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 \(P = \text{ASSOCIATION}_\text{BASED}_\text{SELECTION}(P \cup Q, DV, z^*, z^{nadir}, N);\)
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 \([P, DV, B, z^{nadir}] = \text{DV\_ADJUSTMENT}_1(P, DV, N, T, m);\)
10 \(K = N;\)
11 end
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 \([DV, B] = \text{DV\_ADJUSTMENT}_2(P, DV, T);\)
15 end
16 \(t = t + 1;\)
17 end

### Algorithm 2: Initialization (INITIALIZATION)

**Input:** \(K\): the population size or number of direction vectors;  
\(m\): the number of objectives;  
\(N\): maximum population size;  
\(m\): the number of objectives;

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

1 Randomly generate an initial population, \(P = \{x^1, \ldots, x^K\}\);
2 for \(j = 1\) to \(m\) do

3 \(\lambda^j_1 = 1, \lambda^j_i = 0, i = 1, \ldots, m, i \neq j;\)
4 \(B^j = \emptyset;\)
5 \(z^* = \min_{y \in P} f(y);\)
6 \(z^{nadir} = +\infty;\)
7 end

### Algorithm 3: Offspring Generation (VARIATION)

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

**Output:** an offspring population \(Q = \{y^1, \ldots, y^K\}, z^*\).

1 \(Q = \emptyset;\)
2 foreach \(x^i \in P, i = 1, 2, \ldots, K\) do

3 /* Selection of neighborhood index set */

4 \(D = \begin{cases} \emptyset & K = m \\ B^k & K = N \text{ and } \text{rand}(\) \(< \delta \\ {1, \ldots, K} & \text{otherwise} \end{cases} \) (3)

5 /* Reproduction */

6 if \(D = \emptyset\) then

7 Perform a mutation operator on \(x^i\) with probability \(p_m\) to produce a new solution \(y^i;\)
8 else

9 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;
10 end
11 for \(j = 1\) to \(m\) do

12 if \(z^* > f_j(y^i)\) then

13 \(z^* = f_j(y^i)\);
14 end
15 end
16 \(Q = Q \cup \{y^i\};\)
17 end
Algorithm 4: Association (ASSOCIATION)

Input: $P$: current population;
\( z^* \): ideal point;
\( DV = \{ \lambda^1, \ldots, \lambda^K \} \): direction vectors.

Output: Solutions associated with direction vectors \( S^1, \ldots, S^K \).

\[ S^k = \emptyset \quad \text{for all} \quad k = 1, \ldots, K; \]

\[ \text{foreach} \quad x^i \in P \quad \text{do} \]
\[ \quad \text{for} \quad j = 1 \quad \text{to} \quad m \quad \text{do} \]
\[ \quad \quad x^i_j = x^i_j - z^*_j; \]
\[ \quad \text{end} \]
\[ \quad /* \text{Refer to (4)} */ \]
\[ \quad k = \arg\min_{x^i \in DV} \arccos \left( \frac{x^i \cdot \lambda^k}{||x^i|| \cdot ||\lambda^k||} \right); \]
\[ S^k = S^k \cup \{ x^i \}; \]
\[ \text{end} \]

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 \( (1, 0, \ldots, 0) \)] 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 \( z^\text{nad} \) 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 \( \{1, \ldots, K\} \) based on a small probability \( \delta \). The differential evolution (DE) \cite{55} with polynomial mutation operator \cite{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) 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 5: Association-Based Selection (ASSOCIATION_BASED_SELECTION)

Input: \( P \): current population;
\( DV = \{ \lambda^1, \ldots, \lambda^K \} \): direction vectors;
\( z^* \): the initial ideal point;
\( z^\text{nad} \): the initial nadir point;
\( N \): maximum population size.

Output: the selected population \( Q \).

\[ S = P, Q = \emptyset; \]
\[ /* \text{Space reduction} */ \]
\[ S = \emptyset; \]
\[ /* \text{Selection} */ \]
\[ \text{for} \quad k = 1 \quad \text{to} \quad K \quad \text{do} \]
\[ \quad \text{if} \quad S^k = \emptyset \quad \text{then} \]
\[ \quad \quad \text{randomly select a solution from} \quad S \quad \text{and add to} \quad Q; \]
\[ \quad \text{end} \]
\[ \quad \text{if} \quad |S^k| > 1 \quad \text{then} \]
\[ \quad \quad x = \arg\min_{x \in S^k} \text{f}_j(x|\lambda^k, z^*); \]
\[ \quad \quad Q = Q \cup \{ x \}; \]
\[ \quad \text{end} \]
\[ \text{end} \]
\[ [S^1, \ldots, S^K] = \text{ASSOCIATION}(S, DV, z^*); \]

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 \( Q \);

\( B \) is the updated neighborhood index set;
\( z^\text{nad} \) is the nadir point.

\[ \text{for} \quad k = 1 \quad \text{to} \quad m \quad \text{do} \]
\[ \quad \bar{z}^\text{nad} = \max_{x \in P} \text{f}_j(x); \]
\[ \text{end} \]
\[ 4 \] \text{Generate} \( N \) \text{ direction vectors} \( DV = \{\lambda^1, \ldots, \lambda^N\} \);
\[ 5 \] \( B = \text{TRACK_NEIGHBORS}(DV, T) \);
\[ 6 \] \( Q = \text{ASSOCIATION_BASED_SELECTION}(P, DV) \);
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:

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

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 $\sum_{i=1}^{K} g_i(x) |\lambda^k|$, $z^*$ 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 $\theta$th generation, if the relative decrease along $m$ direction vectors ($\Delta_t$) is lower than a very small value ($10^{-3}$), which indicates all the subproblems have been well-converged, the adjustment for the number of direction vectors is activated, presented in Algorithm 6. $\Delta_t$ is defined as follows:

$$\Delta_t = \sum_{k=1}^{K} \left( \frac{F(x^k_t) - F(x^k_{t-\phi_1})}{F(x^k_t)} \right)$$

Algorithm 7: Track Neighborhood Index Set $B$

Input: $DV = \{\lambda^1, \ldots, \lambda^K\}$: direction vectors; $T$: neighborhood size.

Output: neighborhood index set $B$.

for $i = 1$ to $K$

di$_i$ = $+\infty$;
end

for $j = i + 1$ to $K$

di$_{ij} = |\lambda^i - \lambda^j|$;
end

$J$ stores the index of direction vectors after sorting $d_i$ /*

$[d_i, I] = \text{sort}(d_i)$, $j = 1, \ldots, K$, $j \neq i$;

$B^i = I(1:T)$;

Algorithm 8: Detection of the Effective Direction Vectors

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

Output: effective direction vectors $DV$.

/* Find all dominated solutions */

$P_{\text{dom}} = \text{Nondominated}(P)$;

$\{S^1, \ldots, S^K\} = \text{Association}(P_{\text{dom}}, DV, z^*)$;

$DV = \emptyset$;

for $k = 1$ to $K$

if $S^k \neq \emptyset$ then

$DV = DV \cup \{\lambda^k\}$;
end

$|DV| \leq |DV|$ /*

end

Algorithm 9: Adjustment for the Positions of Ineffective Direction Vectors ($DV_{\text{ADJUSTMENT2}}$)

Input: $P$: current population; $DV = \{\lambda^1, \ldots, \lambda^K\}$: direction vectors; $T$: neighborhood size.

Output: updated direction vectors $DV = \{\lambda^1, \ldots, \lambda^K\}$; updated neighborhood index set $B$.

// Identify the effective direction vectors $DV$

if $|DV| < K$ then

$\text{Pair}(P, DV, z^*)$;

for $i = 1$ to $|DV|$ do

$\lambda^i = (\lambda^i + \lambda^j)/2$;
$DV = DV \cup \{\lambda^i\}$;
end

else

/* find the distance of each $\lambda \in DV$ to its nearest neighbor */

$\text{Pair} = \{(1, 2), \ldots, (i, j), \ldots, (|DV| - 1, |DV|)\}$;
$\text{d} = (d_1, d_2, \ldots, d_{|DV| - 1, |DV|}) = (0, \ldots, 0)$;

for $i = 1$ to $|DV|$ do

$d_{i,i} = +\infty$;

for $j = i + 1$ to $|DV|$ do

$d_{ij} = |\lambda^i - \lambda^j|$;
end

$d_{\min}(i) = \min_{1 \leq j \leq |DV|} (d_{ij})$;
end

$d_{\max} = \max(d_{\min}(1), \ldots, d_{\min}(|DV|))$;

/* Select $K - |DV|$ pairs of direction vectors */

$Mid = \text{find}(d === d_{\max})$;
$l = \min(Mid)$;
$r = \max(Mid)$;

while $r - l + 1 < K - |DV|$ and $l > 1$ do

$l = -l$;
end

while $r - l + 1 < K - |DV|$ do

$r = r + 1$;
end

/* Generate new direction vectors */

for $k = 1$ to $K$ do

$\lambda^* = (\lambda_{\text{Pair}(1,1)} + \lambda_{\text{Pair}(2,2)})/2$;
$DV = DV \cup \{\lambda^*\}$;
end

$DV = DV$;

$B = \text{Track Neighbors}(DV, T)$;
where \( K \) is the size of the population, \( x^k \) 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 \( |\text{DV}| \leq |\text{DV}| \), the rest (\(|\text{DV}| - |\text{DV}| \)) direction vectors are generated by inserting new ones at the midpoints between every two effective direction vectors. It can be known that there are at most

\[
\left( |\text{DV}| \right) = |\text{DV}| \times \left( |\text{DV}| - 1 \right)
\]

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

\[
\text{Pair} = \{(1, 2), \ldots, (i, j), \ldots, (|\text{DV}| - 1, |\text{DV}|)\}
\]

The distance vector of vector pairs can be denoted as

\[
d = \left( d_{1,2}, \ldots, d_{i,j}, \ldots, d_{|\text{DV}|-1,|\text{DV}|} \right)
\]

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

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

Otherwise, \((K - |\text{DV}|)\) pairs are selected from all possible \( \left( \frac{|\text{DV}|}{2} \right) \) pairs to generate new direction vectors, as follows.

The distance of every \( i \)th direction vector to its nearest neighbor, denoted by \( d_{\text{min}}(i) \), is computed (lines 13–19). The pair index set, denoted by \( \text{Mid} \), that has the maximum distance \( d_{\text{max}} \) out of all possible \( d_{\text{min}}(i) \), is selected (lines 20–23). The lower index of \( \text{Mid} \) is stored as \( l \) and the upper index of \( \text{Mid} \) 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 \( (N - |\text{DV}|) \) (lines 26–31). \((K - |\text{DV}|)\) new direction vectors are generated by inserting midpoints of \((\text{upper} - \text{lower})\) vector pairs in \( \text{DV} \) (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, \((a, b, c, f, j, g, h)\) 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

<table>
<thead>
<tr>
<th>Problem</th>
<th>MOEA/D-DE</th>
<th>GREA</th>
<th>NSGA-III</th>
<th>MOEA/DD</th>
<th>MAOEA/D-2ADV</th>
<th>v/</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>3</td>
<td>1.044 ± 0.002</td>
<td>2.294 ± 0.002</td>
<td>1.145 ± 0.002</td>
<td>1.137 ± 0.002</td>
<td>1.095 ± 0.002</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>3.330 ± 0.002</td>
<td>7.731 ± 0.002</td>
<td>3.210 ± 0.002</td>
<td>3.210 ± 0.002</td>
<td>0.958 ± 0.002</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5.900 ± 0.002</td>
<td>3.676 ± 0.002</td>
<td>5.192 ± 0.002</td>
<td>5.201 ± 0.002</td>
<td>5.181 ± 0.002</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>7.411 ± 0.002</td>
<td>3.625 ± 0.002</td>
<td>7.549 ± 0.002</td>
<td>6.797 ± 0.002</td>
<td>6.329 ± 0.002</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>9.471 ± 0.002</td>
<td>1.310 ± 0.002</td>
<td>2.001 ± 0.001</td>
<td>8.833 ± 0.002</td>
<td>8.156 ± 0.002</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>9.684 ± 0.002</td>
<td>1.556 ± 0.002</td>
<td>2.316 ± 0.002</td>
<td>9.099 ± 0.002</td>
<td>8.946 ± 0.002</td>
</tr>
</tbody>
</table>

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

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 † denote 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](image-url)

(a) 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 vector (i) is inserted in the midpoint between \( b \) and \( c \) and another one (j) is inserted in the midpoint between \( f \) and \( g \).
TABLE VI
MEAN VALUES OF IGD, OBTAINED BY MOEA/D-AWA, RVEA, AND MAOEA/D-2ADV ON DTLZ INSTANCES

<table>
<thead>
<tr>
<th>Problem</th>
<th>MOEA/D-AWA</th>
<th>RVEA</th>
<th>MAOEA/D-2ADV</th>
<th>+/-</th>
</tr>
</thead>
<tbody>
<tr>
<td>DTLZ1</td>
<td>1.487e-02</td>
<td>1.477e-02</td>
<td>1.095e-02</td>
<td>+25.86%</td>
</tr>
<tr>
<td></td>
<td>5.692e-02</td>
<td>1.444e-01</td>
<td>3.218e-02</td>
<td>+20.14%</td>
</tr>
<tr>
<td></td>
<td>1.010e-01</td>
<td>1.671e-01</td>
<td>5.151e-01</td>
<td>+49.00%</td>
</tr>
<tr>
<td></td>
<td>8.922e-02</td>
<td>6.807e-02</td>
<td>6.329e-02</td>
<td>+7.02%</td>
</tr>
<tr>
<td></td>
<td>8.121e-01</td>
<td>8.860e-02</td>
<td>8.156e-02</td>
<td>+7.95%</td>
</tr>
<tr>
<td></td>
<td>1.167e-01</td>
<td>9.171e-02</td>
<td>8.946e-02</td>
<td>+2.55%</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>4.522e-02</td>
<td>3.920e-02</td>
<td>3.722e-02</td>
<td>+16.53%</td>
</tr>
<tr>
<td></td>
<td>1.313e-01</td>
<td>1.767e-01</td>
<td>8.759e-02</td>
<td>+33.29%</td>
</tr>
<tr>
<td></td>
<td>2.014e-01</td>
<td>2.674e-01</td>
<td>1.333e-01</td>
<td>+33.81%</td>
</tr>
<tr>
<td></td>
<td>3.491e-01</td>
<td>2.317e-01</td>
<td>2.317e-01</td>
<td>0.00%</td>
</tr>
<tr>
<td></td>
<td>5.298e-01</td>
<td>3.654e-01</td>
<td>3.650e-01</td>
<td>+0.11%</td>
</tr>
<tr>
<td></td>
<td>5.801e-01</td>
<td>4.151e-01</td>
<td>4.121e-01</td>
<td>+0.72%</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>4.752e-02</td>
<td>4.076e-02</td>
<td>3.538e-02</td>
<td>+13.30%</td>
</tr>
<tr>
<td></td>
<td>1.615e-01</td>
<td>4.054e-01</td>
<td>9.014e-01</td>
<td>+44.19%</td>
</tr>
<tr>
<td></td>
<td>2.438e-00</td>
<td>6.400e-00</td>
<td>1.374e-01</td>
<td>+94.36%</td>
</tr>
<tr>
<td></td>
<td>4.394e-01</td>
<td>2.321e-01</td>
<td>2.357e-01</td>
<td>-0.69%</td>
</tr>
<tr>
<td></td>
<td>2.296e-01</td>
<td>3.668e-01</td>
<td>3.679e-01</td>
<td>-0.30%</td>
</tr>
<tr>
<td></td>
<td>1.715e-01</td>
<td>4.222e-01</td>
<td>4.161e-01</td>
<td>+1.44%</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>4.430e-02</td>
<td>4.197e-02</td>
<td>3.280e-02</td>
<td>+21.85%</td>
</tr>
<tr>
<td></td>
<td>1.391e-01</td>
<td>1.768e-01</td>
<td>8.775e-02</td>
<td>+36.92%</td>
</tr>
<tr>
<td></td>
<td>2.136e-01</td>
<td>2.681e-01</td>
<td>1.354e-01</td>
<td>+37.55%</td>
</tr>
<tr>
<td></td>
<td>3.152e-01</td>
<td>2.320e-01</td>
<td>3.313e-01</td>
<td>-0.04%</td>
</tr>
<tr>
<td></td>
<td>4.804e-01</td>
<td>3.667e-01</td>
<td>3.720e-01</td>
<td>-1.45%</td>
</tr>
<tr>
<td></td>
<td>5.400e-01</td>
<td>4.209e-01</td>
<td>4.405e-01</td>
<td>-4.66%</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>3.930e-03</td>
<td>3.759e-02</td>
<td>1.845e-03</td>
<td>+53.05%</td>
</tr>
<tr>
<td></td>
<td>2.676e-02</td>
<td>9.341e-02</td>
<td>6.992e-03</td>
<td>+73.87%</td>
</tr>
<tr>
<td></td>
<td>2.860e-02</td>
<td>1.786e-01</td>
<td>1.527e-02</td>
<td>+46.11%</td>
</tr>
<tr>
<td></td>
<td>1.386e-02</td>
<td>3.805e-01</td>
<td>1.434e-02</td>
<td>-3.46%</td>
</tr>
<tr>
<td></td>
<td>1.870e-02</td>
<td>4.458e-01</td>
<td>1.368e-02</td>
<td>+26.84%</td>
</tr>
<tr>
<td></td>
<td>8.609e-03</td>
<td>2.317e-01</td>
<td>1.154e-02</td>
<td>-34.05%</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>3.440e-04</td>
<td>5.031e-02</td>
<td>2.219e-03</td>
<td>+35.49%</td>
</tr>
<tr>
<td></td>
<td>2.256e-03</td>
<td>2.382e-01</td>
<td>5.888e-03</td>
<td>+24.13%</td>
</tr>
<tr>
<td></td>
<td>2.459e-02</td>
<td>3.550e-02</td>
<td>1.297e-02</td>
<td>+47.25%</td>
</tr>
<tr>
<td></td>
<td>1.258e-02</td>
<td>3.496e-01</td>
<td>1.545e-02</td>
<td>-22.81%</td>
</tr>
<tr>
<td></td>
<td>2.014e-02</td>
<td>4.165e-01</td>
<td>1.358e-02</td>
<td>+32.32%</td>
</tr>
<tr>
<td></td>
<td>1.053e-02</td>
<td>4.611e-01</td>
<td>1.492e-02</td>
<td>-41.69%</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>3.847e-02</td>
<td>7.386e-02</td>
<td>3.538e-02</td>
<td>+52.10%</td>
</tr>
<tr>
<td></td>
<td>4.270e-01</td>
<td>1.363e-01</td>
<td>1.543e-01</td>
<td>-13.31%</td>
</tr>
<tr>
<td></td>
<td>7.234e-01</td>
<td>3.008e-01</td>
<td>3.457e-01</td>
<td>-14.95%</td>
</tr>
<tr>
<td></td>
<td>8.347e-01</td>
<td>5.446e-01</td>
<td>5.573e-01</td>
<td>-2.33%</td>
</tr>
<tr>
<td></td>
<td>1.324e-00</td>
<td>1.177e+00</td>
<td>7.425e-01</td>
<td>+36.92%</td>
</tr>
<tr>
<td></td>
<td>1.445e+00</td>
<td>1.257e+00</td>
<td>9.552e-01</td>
<td>+24.01%</td>
</tr>
</tbody>
</table>

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

H. Computational Complexity

In MaOEAD-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 MaOEAD-2ADV, the adjustment for the renumber 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 MaOEAD-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.
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 MaOEAs

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.
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 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 MaOEAs and two MaOEAs 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


