Applied Soft Computing 61 (2017) 394–411

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A many-objective evolutionary algorithm based on a projection-assisted intra-family election☆

Zefeng Chen [a], Yuren Zhou [a,b,*], Yi Xiang [a,b]

[a] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China
[b] Collaborative Innovation Center of High Performance Computing, Sun Yat-sen University, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 27 February 2017
Received in revised form 6 July 2017
Accepted 27 July 2017
Available online 4 August 2017

Keywords:
Many-objective optimization
Evolutionary algorithm
Evolution direction
Projection
Angle

ABSTRACT

In recent years, many researchers have put emphasis on the study of how to keep a good balance between convergence and diversity in many-objective optimization. This paper proposes a new many-objective evolutionary algorithm based on a *projection-assisted intra-family election*. In the proposed algorithm, *basic evolution directions* are adaptively generated according to the current population and potential *evolution directions* are excavated in each individual's *family*. Based on these *evolution directions*, a strategy of *intra-family election* is performed in every family and elite individuals are elected as representatives of the specific *family* to join the next stage, which can enhance the convergence of the algorithm. Moreover, a selection procedure based on angles is used to maintain the diversity. The performance of the proposed algorithm is verified and compared with several state-of-the-art many-objective evolutionary algorithms on a variety of well-known benchmark problems ranging from 5 to 20 objectives. Empirical results demonstrate that the proposed algorithm outperforms other peer algorithms in terms of both the diversity and the convergence of the final solutions set on most of the test instances. In particular, our proposed algorithm shows obvious superiority when handling the problems with larger number of objectives.

1. Introduction

In the real world, there are a variety of multi-objective optimization problems (MOPs). An MOP needs to optimize several conflicting objectives simultaneously to achieve a tradeoff among different objectives. Without loss of generality, a standard MOP can be expressed as a minimization problem with n decision variables and m objective functions [1]:

$$\begin{aligned} &\textbf{\textit{minimize}} \quad F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T \\ &\textbf{\textit{subjectto}} \quad \mathbf{x} \in \Omega \end{aligned}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is the n-dimensional decision vector, $f_i(\mathbf{x})$, $i = 1, 2, \ldots, m$ are the m objective functions, and $\Omega \subset R^n$ is the decision space. The image set $S = \{F(\mathbf{x}) | \mathbf{x} \in \Omega\}$ is called the objective space.

Let $\mathbf{y} = (y_1, y_2, \ldots, y_m)^T$ and $\mathbf{z} = (z_1, z_2, \ldots, z_m)^T$ be two objective vectors in R^m, $\mathbf{y}$ is said to Pareto dominate $\mathbf{z}$ (written as $\mathbf{y} \prec \mathbf{z}$) if and only if (1) for all $i \in \{1, 2, \ldots, m\}$, $y_i \le z_i$, and (2) there exists some $j \in \{1, 2, \ldots, m\}$ such that $y_j < z_j$. If neither $\mathbf{y} \prec \mathbf{z}$ nor $\mathbf{z} \prec \mathbf{y}$, then $\mathbf{y}$ and $\mathbf{z}$ are Pareto non-dominated with each other (written as $\mathbf{y} \prec\succ \mathbf{z}$). A decision vector $\mathbf{x}^* \in \Omega$ is called a Pareto optimal solution if its objective vector is not dominated by any other vector in the objective space S. All Pareto optimal solutions constitute the Pareto optimal set. The image of the Pareto optimal set is called the Pareto front (PF for short) [2].

As a class of algorithms inspired by the process of natural evolution [3], evolutionary algorithms (EAs), such as genetic algorithm (GA) [4,5], particle swarm optimization (PSO) [6,7], ant colony optimization (ACO) [8,9], differential evolution (DE) [10,11] and artificial bee colony (ABC) [12,13], are very suitable for multi-objective optimization due to their inherent good properties and the ability to find multiple solutions at once [14,15]. As a matter of fact, since the beginning of 1990s, there has been considerable interest in the study of multi-objective evolutionary algorithms (MOEAs). Indeed, there exist different kinds of classical MOEAs (such as NSGA-II [16], MOEA/D [17] and SPEA2 [18]) which can effectively solve MOPs with 2 or 3 objectives. However, many real-

☆ This work was supported by the National Nature Science Foundation of China [grant numbers 61170081, 61472143, 61673403, 61773410]; and the Scientific Research Special Plan of Guangzhou Science and Technology Programme [grant number 201607010045].

* Corresponding author at: School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China.

E-mail addresses: chzfeng@mail2.sysu.edu.cn (Z. Chen), zhouyuren@mail.sysu.edu.cn (Y. Zhou).

http://dx.doi.org/10.1016/j.asoc.2017.07.052

world scientific and engineering problems involve more than three objectives and are difficult to be handled, for instance, the problem of optimal product selection from feature models [19] and water supply portfolio planning [20]. Thus, we need to pay great attention to the MOPs with four or more objectives, which are known as many-objective optimization problems (MaOPs). For MaOPs, many researchers gradually start to lay stress on designing different kinds of many-objective evolutionary algorithms (MaOEAs), such as NSGA-III [21], HypE [22], GrEA [23] and BiGE [24]. In fact, according to the selection mechanism adopted in the algorithm, the existing MOEAs or MaOEAs can be roughly divided into three main categories, i.e., Pareto-based approach, indicator-based approach and decomposition-based approach.

As we all know, the designing of an MOEA generally needs to pursue a balance between two aspects: one is the convergence (that is, making the final solutions set as close to the true PF as possible), and the other is the diversity (that is, making the final solutions set widely and uniformly distributed along the true PF). However, when dealing with MaOPs, it will become much more challenging for an algorithm to keep a good balance between convergence and diversity. This is because that the increase of the number of objectives will make the problem of deterioration more prevalent [25]. Moreover, it will bring about several thorny problems, such as the significantly increased computational complexity [22], the dominance resistance (DR) phenomenon [26–28], limited solution set size [29,30] and the visualization of trade-off surface [31]. Naturally, different kinds of MOEAs are faced with different types of major difficulties due to their own characteristics [32]. Thus, when designing an algorithm for many-objective optimization, researchers need to utilize different effective techniques to overcome the tough obstacles.

The Pareto-based MOEAs, which rely on the Pareto dominance, usually suffer from the DR phenomenon when dealing with MaOPs. The DR phenomenon is manifested as the phenomenon in which the enormously increasing proportion of non-dominated solutions makes almost all the solutions in the population incomparable [33]. And it will result in the loss of selection pressure in a high-dimensional space. To cope with this problem, some researchers focused on designing effective strategies for diversity maintenance which can remedy the loss of selection pressure in a sense. For example, in 2011, Adra et al. proposed two diversity management mechanisms (i.e., DM1 and DM2) for many-objective optimization [34]. In 2014, Li et al. proposed a shift-based density estimation (SDE) strategy and integrated this strategy into three popular Pareto-based approaches [35]. In 2015, Wang et al. also designed a new L_p-norm-based diversity maintenance scheme in their proposed Two_Arch2 [25]. These strategies for diversity maintenance have been demonstrated by experimental results. On the other hand, some researchers attempted to make solutions more distinguishable through developing novel relationships, such as ϵ-dominance [36], fuzzy Pareto-dominance [37], grid dominance [23] and generalized Pareto-optimality [38]. It is worth noting that the generalized Pareto-optimality proposed in [38] is the first research which attempts to generalize the conventional Pareto-optimality.

Other than the DR phenomenon, most of the Pareto-based approaches carry out a non-dominated sorting procedure, which needs to consume a lot of computation costs when the number of objectives increases. In the past decade, researchers also fastened on how to further improve the efficiency of non-dominated sorting. Some researchers resorted to specific strategies to develop new algorithms for non-dominated sorting, such as Jensen's algorithm [39,40], climbing sort and deductive sort [41]. Meanwhile, others attempted to reduce the cost of non-dominated sorting through proposing new data structures, such as ENS [42], M-front [43] and DDA-NS [44].

The indicator-based MOEAs utilize a certain quality indicator to guide the selection and evolution of individuals. To this end, some indicators have been utilized to develop different indicator-based algorithms. For example, HypE [22] and SMS-EMOA [45] adopt the hypervolume (HV) indicator, while MOMBI [46] and MOMBI-II [47] are based on the R2 indicator. IBEA [48] and Two_Arch2 [25] utilize the binary additive ϵ-indicator $I_{\epsilon+}$. In fact, the HV indicator is commonly adopted due to its established theoretical properties (such as the strict monotonicity with regard to the Pareto dominance). However, the calculating of the HV needs to consume a lot of time. When the number of objectives is large, the computation cost for calculating the HV will become prohibitively expensive. Although some efficient methods for calculating the HV [22,49,50] have been designed, it is still not enough to make the HV-based approaches widely applied in solving the MaOPs efficiently.

The decomposition-based MOEAs usually use a scalarizing function to decompose an MOP into a number of scalar optimization subproblems which will be solved in a collaborative manner. This class of approaches, which are represented by MOEA/D [17], have become increasingly popular for solving many-objective problems [29,51]. Since the proposition of the original MOEA/D in 2007, a lot of studies have been conducted on the improvement of MOEA/D and the designing of new decomposition-based MOEAs or MaOEAs, which can be seen from a recent survey by Trivedi et al. [52]. On the one hand, some researchers devoted themselves to improving different aspects of MOEA/D, such as the method of generating weight vectors, the decomposition approach, the mating selection mechanism and the reproduction operator. On the other hand, some novel MaOEAs are developed based on the framework of MOEA/D. For instance, MOEA/D-M2M [53] and MOEA/D-AM2M [54] adopt a new decomposition strategy which decomposes the original MOP into a number of multi-objective optimization subproblems. Similar to MOEA/D-M2M, NSGA-III [21], MOEA/DD [55] and RVEA [56] also share the thought that a set of predefined reference vectors are utilized to partition the high-dimension objective space into small subspaces. These algorithms belong to the decomposition-based approaches in essence [52].

It is generally known that the decomposition-based approaches need to predefine a set of weight vectors or reference vectors. Consequently, the method of generating weight vectors becomes an important and tough issue for the decomposition-based approaches. The original version of MOEA/D and many of its variants employ Das and Dennis's systematic approach [57] to generate evenly distributed weight vectors. However, as shown in a recent study [58], the performance of the decomposition-based MaOEAs strongly depends on the shapes of the PF. The decomposition-based MaOEAs with predefined even-distributed weight vectors are not suitable to deal with problems with irregular PF. Moreover, how to configure weight vectors in a high-dimension objective space is still an open question [33,52]. Thus, it is necessary to conduct further research on how to achieve the adaptation of weight vectors. Bearing the above considerations in mind, in this paper, we desire to adopt some important techniques to design a new algorithm which is independent of the predefined vector set. Concretely, we propose a many-objective evolutionary algorithm based on a *projection-assisted intra-family election*, called MOEA/PIE. The main characteristics of the proposed MOEA/PIE algorithm are three-fold.

(1) First of all, a novel *projection-assisted intra-family election* mechanism is integrated into the evolution procedure. In this mechanism, the concept of *evolution direction* assisted by projection is utilized to make the population evolve to the true PF. We design a *basic evolution direction* and create a *family* for each individual in the current population with a special mating scheme. Moreover, new potential *evolution directions* for

the current individual are created according to the relationship between the current individual and its mate. Based on the fitness associated with a certain *evolution direction*, a strategy of *intra-family election* is performed in every *family* and elite individuals are elected as representatives of the specific *family* to join the next stage, which can enhance the convergence of the algorithm.

(2) The second characteristic lies in the selection procedure, which can well maintain the diversity of the population based on angles. Using the information of angles among all the pairs of individuals, we can measure the crowding degree of different individuals and maintain the diversity of the population. In the selection procedure, we design a specific scheme for handling *extreme solutions* to retain the convergence when maintaining the diversity. Afterwards, a dynamic method is adopted to maintain the diversity of population. In this method, each time we identify the pair of individuals with the minimum angle and remove the worse one between these two individuals. This method can promote the diversity more efficiently.

(3) The third characteristic is that the proposed MOEA/PIE does not require a set of predefined reference points or weight vectors. Although the role of *evolution directions* in MOEA/PIE is a bit like the reference points in NSGA-III or the weight vectors in MOEA/D, the *evolution directions* are adaptively generated and set according to the current population.

The rest of the paper is organized as follows. Section 2 gives some important concepts, and Section 3 describes the proposed algorithm in details. After the illustration of the experimental setup in Section 4, Section 5 presents the experimental results and discussions. And Section 6 mainly focuses on the parameter study of the proposed algorithm and conducts further discussions. Finally, conclusions are drawn in Section 7.

2. Important concepts

In this section, we firstly introduce some important concepts, which will be used in our proposed MOEA/PIE algorithm.

2.1. Family

For an individual $\mathbf{x}$ in a population, we can utilize a type of mating scheme to select an individual nearby $\mathbf{x}$ (denoted by $\mathbf{x}'$) to serve as a mate for $\mathbf{x}$. Suppose K is an integer constant and $K > 1$, we can apply specific genetic operators $K-1$ times to $\mathbf{x}$ and $\mathbf{x}'$, and obtain $K-1$ offspring (denoted by $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{K-1}$). The parent $\mathbf{x}$ and its offspring are called the *family* of $\mathbf{x}$ (containing K members), which can be regarded as a subpopulation nearby $\mathbf{x}$. That is, the *family* of an individual $\mathbf{x}$ is defined as:

$$S_{\mathbf{x}} = \left\{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{K-1}\right\} \tag{1}$$

2.2. Basic evolution direction

Suppose S is a normalized population. For an individual $\mathbf{x}$ in S, we use $\bar{\mathrm{F}}(\mathbf{x}) = \left(\bar{f}_1(\mathbf{x}), \bar{f}_2(\mathbf{x}), \ldots, \bar{f}_m(\mathbf{x})\right)^T$ to denote its normalized objective vector. After identifying the maximum value of each normalized objective (denoted by $z_i^{\max}$ for the ith objective, $i = 1, 2, \ldots, m$) among all the individuals in S, we can construct the *nadir point* $\mathbf{z}^{\max} = \left(z_1^{\max}, z_2^{\max}, \ldots, z_m^{\max}\right)^T$. Based on the *nadir point*, we define the *basic evolution direction* for an individual $\mathbf{x}$ (denoted as $\mathbf{v}_{\mathbf{x}}$) as the vector pointing from $\mathbf{z}^{max}$ to $\bar{\mathrm{F}}(\mathbf{x})$ in the normalized objective space, i.e.,

$$\mathbf{v}_{\mathbf{x}} = \bar{\mathrm{F}}(\mathbf{x}) - \mathbf{z}^{\max} \tag{2}$$

The reason why we use the name of *evolution direction* is that the objective vector of an individual will draw nearer to the true PF if this individual moves in the direction of $\mathbf{v}_{\mathbf{x}}$.

2.3. Fitness associated with basic evolution direction

For an individual $\mathbf{y}$ in the family of $\mathbf{x}$ (i.e., $S_{\mathbf{x}}$), we consider the scalar projection of $\bar{\mathrm{F}}(\mathbf{y}) - \bar{\mathrm{F}}(\mathbf{x})$ onto the *basic evolution direction* $\mathbf{v}_{\mathbf{x}}$ (denoted by $a_1 = scalar\_proj_{\mathbf{v}_{\mathbf{x}}}\left(\bar{\mathrm{F}}(\mathbf{y}) - \bar{\mathrm{F}}(\mathbf{x})\right)$, which is a scalar) and the vector rejection of $\bar{\mathrm{F}}(\mathbf{y}) - \bar{\mathrm{F}}(\mathbf{x})$ onto $\mathbf{v}_{\mathbf{x}}$ (denoted by $\mathbf{a}_2 = rej_{\mathbf{v}_{\mathbf{x}}}\left(\bar{\mathrm{F}}(\mathbf{y}) - \bar{\mathrm{F}}(\mathbf{x})\right)$, which is a vector). Using a_1 and $\mathbf{a}_2$, we construct the *fitness of* $\mathbf{y}$ *associated with* $\mathbf{v}_{\mathbf{x}}$ (denoted by $g(\mathbf{y} \mid \mathbf{v}_{\mathbf{x}})$), which is defined as:

$$g(\mathbf{y} \mid \mathbf{v}_{\mathbf{x}}) = (-a_1) + \theta \cdot \|\mathbf{a}_2\| \tag{3}$$

where $\theta \geq 0$ is a user-defined control parameter. Specifically, the scalar projection a_1 and vector rejection $\mathbf{a}_2$ can be computed respectively as follows:

$$\begin{aligned} a_1 &= scalar\_proj_{\mathbf{v}_{\mathbf{x}}}\left(\bar{\mathrm{F}}(\mathbf{y}) - \bar{\mathrm{F}}(\mathbf{x})\right) \\ &= \frac{\left(\bar{\mathrm{F}}(\mathbf{y}) - \bar{\mathrm{F}}(\mathbf{x})\right)^T \mathbf{v}_{\mathbf{x}}}{\|\mathbf{v}_{\mathbf{x}}\|} \end{aligned} \tag{4}$$

$$\begin{aligned} \mathbf{a}_2 &= rej_{\mathbf{v}_{\mathbf{x}}}\left(\bar{\mathrm{F}}(\mathbf{y}) - \bar{\mathrm{F}}(\mathbf{x})\right) \\ &= \left(\bar{\mathrm{F}}(\mathbf{y}) - \bar{\mathrm{F}}(\mathbf{x})\right) - a_1 \cdot \frac{\mathbf{v}_{\mathbf{x}}}{\|\mathbf{v}_{\mathbf{x}}\|} \end{aligned} \tag{5}$$

In fact, the scalar projection a_1 is the projection distance of $\bar{\mathrm{F}}(\mathbf{y}) - \bar{\mathrm{F}}(\mathbf{x})$ onto $\mathbf{v}_{\mathbf{x}}$ with a plus or minus (depending on whether the angle between $\bar{\mathrm{F}}(\mathbf{y}) - \bar{\mathrm{F}}(\mathbf{x})$ and $\mathbf{v}_{\mathbf{x}}$ is an acute angle or not). And recall that $\mathbf{v}_{\mathbf{x}}$ is a vector pointing from $\mathbf{z}^{max}$ to $\bar{\mathrm{F}}(\mathbf{x})$. If an individual $\mathbf{y}$ moves in the direction of $\mathbf{v}_{\mathbf{x}}$, $\bar{\mathrm{F}}(\mathbf{y})$ will get nearer and nearer to the true PF, and the value of a_1 will become larger and larger. Thus, a_1 can be used to measure the convergence of $\mathbf{y}$ towards the true PF. On the other hand, the length of the vector rejection $\mathbf{a}_2$ (i.e., $||\mathbf{a}_2||$) is the perpendicular distance from $\bar{\mathrm{F}}(\mathbf{y})$ to the line through $\bar{\mathrm{F}}(\mathbf{x})$ and $\mathbf{z}^{max}$. The closer $\bar{\mathrm{F}}(\mathbf{y})$ get to $\bar{\mathrm{F}}(\mathbf{x})$, the smaller the value of $||\mathbf{a}_2||$ will be. In this sense, $||\mathbf{a}_2||$ can be used to measure the diversity of population. Consequently, by adding the value of $||\mathbf{a}_2||$ multiplied by θ to $(-a_1)$, $g(\mathbf{y} \mid \mathbf{v}_{\mathbf{x}})$ serves as a composite measure of $\mathbf{y}$ for both convergence and diversity. What's more, we can control the balance between convergence and diversity by adjusting the value of the control parameter θ in Eq. (3).

2.4. Extreme solution

Suppose the standard basis for $\mathbb{R}^m$ is $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_m\}$, where $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$, $\mathbf{e}_2 = (0, 1, \ldots, 0)^T, \ldots, \mathbf{e}_m = (0, 0, \ldots, 1)^T$. Considering $\mathbf{e}_i$ ($i = 1, 2, \ldots, m$), which is corresponding to the ith objective axis in the objective space, we define individual $\mathbf{x}$ as an *extreme solution* if $\bar{\mathrm{F}}(\mathbf{x})$ has the minimum angle to $\mathbf{e}_i$. It should be noted that there are at most m extreme solutions in a population.

Algorithm 1. General framework of MOEA/PIE.

Input: population size N
Output: final population P

```
1:  P ⟵ InitializePopulation()
2:  while termination criterion is not fulfilled do
3:      Q ⟵ PIE-Evolution(P)
4:      P ⟵ EnvironmentalSelection(Q)
5:  end while
6:  return P
```

Algorithm 2. *PIE-Evolution.*

Input: population P
Output: new population Q
$Q \longleftarrow \emptyset$
for each individual $\mathbf{x}$ in P **do**
  Select a mate for $\mathbf{x}$, denoted by $\mathbf{x}'$
  if $\mathbf{x} \prec \mathbf{x}'$ **then**
    $flag_{\mathbf{x}} \longleftarrow 1$
  elseif $\mathbf{x}' \prec \mathbf{x}$ **then**
    $flag_{\mathbf{x}} \longleftarrow -1$
  else //$\mathbf{x} \prec\succ \mathbf{x}'$
    $flag_{\mathbf{x}} \longleftarrow 0$
  end if
  Apply genetic operators to $\mathbf{x}$ and $\mathbf{x}'$ to generate $K-1$ offspring, denoted by $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{K-1}$
  $S_{\mathbf{x}} \longleftarrow \left\{\mathbf{x}, \mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{K-1}\right\}$ //the *family* of $\mathbf{x}$
end for
$S \longleftarrow \bigcup_{\mathbf{x} \in P} S_{\mathbf{x}}$
Normalize all the individuals in S
Identify the nadir point: $\mathbf{z}^{\max} = \left(z_1^{\max}, z_2^{\max}, \ldots, z_m^{\max}\right)^T$
for each individual $\mathbf{x}$ in P **do**
  if $flag_{\mathbf{x}} = 0$ **then**
    $\mathbf{v}_{\mathbf{x}} = \bar{\mathrm{F}}(\mathbf{x}) - \mathbf{z}^{\max}$
    Calculate *fitness* $g(\mathbf{y} \mid \mathbf{v}_{\mathbf{x}})$ for each $\mathbf{y}$ in $S_{\mathbf{x}}$
    Select the best and second best individuals (denoted by $\mathbf{y}'_1$ and $\mathbf{y}'_2$) from $S_{\mathbf{x}}$
  elseif $flag_{\mathbf{x}} = 1$ **then**
    $\mathbf{v}_{\mathbf{x}} = \bar{\mathrm{F}}(\mathbf{x}) - \mathbf{z}^{\max}$
    $\mathbf{v}_d = \bar{\mathrm{F}}(\mathbf{x}) - \bar{\mathrm{F}}(\mathbf{x}')$
    Calculate *fitness* $g(\mathbf{y} \mid \mathbf{v}_{\mathbf{x}})$ and $g(\mathbf{y} \mid \mathbf{v}_d)$ for each $\mathbf{y}$ in $S_{\mathbf{x}}$
    Select the best individual (denoted by $\mathbf{y}'_1$) in terms of $g(\mathbf{y} \mid \mathbf{v}_{\mathbf{x}})$ from $S_{\mathbf{x}}$
    Select the best individual (denoted by $\mathbf{y}'_2$) in terms of $g(\mathbf{y} \mid \mathbf{v}_d)$ from $S_{\mathbf{x}}$
  elseif //$flag_{\mathbf{x}} = -1$
    $\mathbf{v}_{\mathbf{x}'} = \bar{\mathrm{F}}(\mathbf{x}') - \mathbf{z}^{\max}$
    $\mathbf{v}_{d'} = \bar{\mathrm{F}}(\mathbf{x}') - \bar{\mathrm{F}}(\mathbf{x})$
    Calculate *fitness* $g(\mathbf{y} \mid \mathbf{v}_{\mathbf{x}'})$ and $g(\mathbf{y} \mid \mathbf{v}_{d'})$ for each $\mathbf{y}$ in $S_{\mathbf{x}}$
    Select the best individual (denoted by $\mathbf{y}'_1$) in terms of $g(\mathbf{y} \mid \mathbf{v}_{\mathbf{x}'})$ from $S_{\mathbf{x}}$
    Select the best individual (denoted by $\mathbf{y}'_2$) in terms of $g(\mathbf{y} \mid \mathbf{v}_{d'})$ from $S_{\mathbf{x}}$
  end if
  Add $\mathbf{y}'_1$ and $\mathbf{y}'_2$ into the set Q
end for
return Q

3. Proposed algorithm: MOEA/PIE

3.1. Framework of proposed algorithm

The framework of the proposed MOEA/PIE is described in Algorithm 1. First, the initialization procedure generates a population P with N individuals by randomly sampling from the decision space Ω. Then, the population is evolved by using an evolution procedure based on a *projection-assisted intra-family election* to generate a new population Q with $2N$ individuals (Line 3 in Algorithm 1). Next, a procedure of environmental selection based on angles is applied to Q to maintain a diversified population with N individuals (Line 4 in Algorithm 1). The above two procedures are repeated until the termination criterion is fulfilled. In the following subsections, the important procedures of MOEA/PIE are to be described in details.

3.2. Evolution procedure based on a projection-assisted intra-family election

As shown in Algorithm 2, the evolution procedure in MOEA/PIE adopts a *projection-assisted intra-family election* mechanism (PIE for short). Specifically, the PIE mechanism uses *evolution direction* assisted by projection to perform a strategy of *intra-family election*, which can contribute to the evolution of population to the true PF. In fact, the evolution procedure in MOEA/PIE focuses primarily on the enhancing of the convergence of the whole algorithm. Next, we will illustrate how the PIE mechanism works in details.

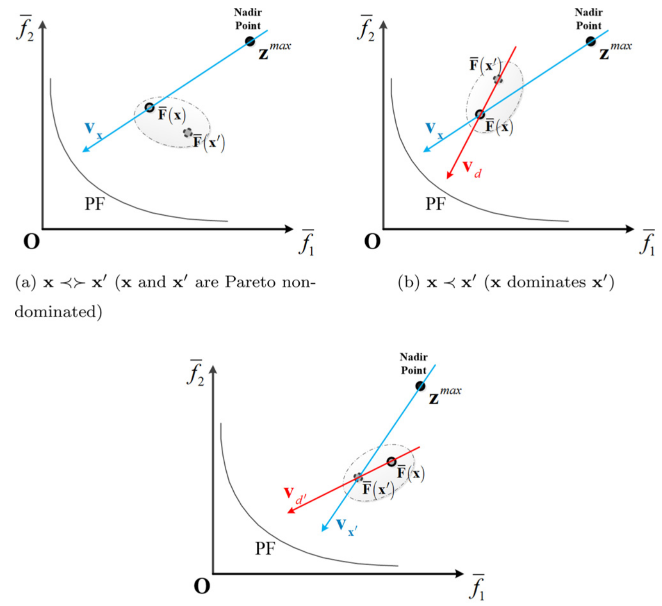

Fig. 1. Evolution directions for different categories of individuals (in a two-dimensional objective space). $\mathbf{x}$ is the current individual and $\mathbf{x}'$ is the mate for $\mathbf{x}$. $\bar{\mathrm{F}}(\mathbf{x})$ and $\bar{\mathrm{F}}(\mathbf{x}')$ are the normalized objective vectors of $\mathbf{x}$ and $\mathbf{x}'$, respectively.

For each individual $\mathbf{x}$ in P, an individual (denoted by $\mathbf{x}'$) is selected from the whole population to serve as a mate for $\mathbf{x}$. The mate for $\mathbf{x}$ (i.e., $\mathbf{x}'$) is determined by using a special mating scheme (see Section 3.3). Then, $K-1$ offspring (denoted by $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_{K-1}$) are generated by applying genetic operators (including crossover and mutation) to $\mathbf{x}$ and $\mathbf{x}'$. The parent $\mathbf{x}$ and its offspring constitute the *family* of $\mathbf{x}$ (containing K members). And all the *families* of all the individuals $\mathbf{x}$ in P constitute a population S, which is in fact a combination of the parent population P and all the newly generated offspring. To eliminate the effect brought from the different scaling of objectives, we normalize the objective vectors of all the individuals in S by adopting the adaptive normalization method as suggested in NSGA-III [21]. Then we identify the nadir point of the normalized population S. In the subsequent processes after normalization, when referring to the objectives of an individual, unless otherwise specified, we always consider the normalized objectives.

As mentioned before, we have created a *family* $S_{\mathbf{x}}$ for each individual $\mathbf{x}$ in P, and the parents are $\mathbf{x}$ and its mate $\mathbf{x}'$. Here, we can directly utilize the *basic evolution direction* and the corresponding *fitness* to conduct a simple strategy to select the elite individuals from $S_{\mathbf{x}}$. However, we design an adaptive method to utilize more potential *evolution directions*. Concretely, as can be seen in Fig. 1, we firstly determine the dominance relationship between $\mathbf{x}$ and $\mathbf{x}'$, and classify $\mathbf{x}$ into the following three categories:

Case 1 ($\mathbf{x}$ and $\mathbf{x}'$ are Pareto non-dominated with each other): In this case, we directly utilize the *basic evolution direction* $\mathbf{v}_{\mathbf{x}}$ (see Fig. 1a),[1] which can serve as a big push for the population to be evolved towards the true PF.

Case 2 ($\mathbf{x}$ Pareto dominates $\mathbf{x}'$): From the dominance relationship $\mathbf{x} \prec \mathbf{x}'$, we can know that the direction $\mathbf{v}_d = \bar{\mathrm{F}}(\mathbf{x}) - \bar{\mathrm{F}}(\mathbf{x}')$ (that is, the vector pointing from $\bar{\mathrm{F}}(\mathbf{x}')$ to $\bar{\mathrm{F}}(\mathbf{x})$) is a potential *evolution direction* which can help push the population towards the true PF. Thus,

[1] For the purpose of selecting elite individuals from $S_{\mathbf{x}}$ which is the *family* of $\mathbf{x}$, we mainly focus on $\mathbf{v}_{\mathbf{x}}$ and do not consider $\mathbf{v}_{\mathbf{x}'}$ which is the *basic evolution direction* of $\mathbf{x}'$.

in this case, as can be seen from Fig. 1b, we utilize two *evolution directions* (i.e., $\mathbf{v_x}$ and $\mathbf{v}_d$).

Case 3 ($\mathbf{x}'$ Pareto dominates $\mathbf{x}$): Compared with the *basic evolution direction* $\mathbf{v_x}$, the direction $v_{x\prime} = \bar{F}(\mathbf{x}\prime) - \mathbf{z}^{\max}$ can be more powerful because $\mathbf{x}'$ is superior to $\mathbf{x}$. Similar to **Case 2**, $\mathbf{v}_{d\prime} = \bar{F}(\mathbf{x}\prime) - \bar{F}(\mathbf{x})$ is also a potential *evolution direction*. Thus, in this case, as can be seen from Fig. 1c, we utilize two *evolution directions* (i.e., $\mathbf{v}_{x'}$ and $\mathbf{v}_{d'}$).

In MOEA/PIE, the strategy of *projection-assisted intra-family election* is performed on every individual $\mathbf{x}$ in the population P (Lines 17–36 in Algorithm 2). For the *family* of $\mathbf{x}$ (i.e., $S_\mathbf{x}$), there exist competitive relationships among all the members (including the parent $\mathbf{x}$ and its $K-1$ offspring) in $S_\mathbf{x}$. To elect elite individuals to join the new population from $S_\mathbf{x}$, we compute the *fitness* values of all the individuals in $S_\mathbf{x}$ along the designed *evolution directions*. As mentioned above, the individuals in P have been classified into three different categories and different *evolution directions* are adaptively set for different categories of individuals to better strengthen the convergence. Thus, there are three ways of selecting elite individuals to join the new population from $S_\mathbf{x}$, which can be listed as follows:

(1) When $\mathbf{x}$ and $\mathbf{x}'$ are Pareto non-dominated with each other ($flag_\mathbf{x} = 0$), the values of $g(\mathbf{y} \mid v_\mathbf{x})$ for every individuals $\mathbf{y}$ in $S_\mathbf{x}$ (that is, the *fitness* of $\mathbf{y}$ associated with $\mathbf{v_x}$) are calculated. Then, the individual with the lowest *fitness* value and the one with the second lowest *fitness* value are identified.
(2) When $\mathbf{x} \prec \mathbf{x}'$ ($flag_\mathbf{x} = 1$), there are two *evolution directions* to be considered. Thus, we need to calculate the values of $g(\mathbf{y} \mid v_\mathbf{x})$ and $g(\mathbf{y} \mid \mathbf{v}_d)$ for every individual $\mathbf{y}$ in $S_\mathbf{x}$ (that is, the *fitness* of $\mathbf{y}$ associated with $\mathbf{v_x}$ and the *fitness* of $\mathbf{y}$ associated with $\mathbf{v}_d$, respectively). Then, we identify the individual with the lowest *fitness* value of $g(\mathbf{y} \mid v_\mathbf{x})$ and the one with the lowest *fitness* value of $g(\mathbf{y} \mid \mathbf{v}_d)$. If the two individuals are identical, we randomly select the individual with the second lowest *fitness* value of $g(\mathbf{y} \mid v_\mathbf{x})$ or $g(\mathbf{y} \mid \mathbf{v}_d)$.
(3) When $\mathbf{x}' \prec \mathbf{x}$ ($flag_\mathbf{x} = -1$), similar to the previous case, we consider the *fitness* values of $g(\mathbf{y} \mid v_{x\prime})$ and $g(\mathbf{y} \mid \mathbf{v}_{d\prime})$, and select two individuals which have the lowest value of the two *fitness* function respectively.

No matter which way to go, each time we are able to select two individuals (denoted by $\mathbf{y}'_1$ and $\mathbf{y}'_2$) from $S_\mathbf{x}$ and add them into the new population Q. In fact, the two individuals selected from $S_\mathbf{x}$ are the elite individuals along a certain *evolution direction* among all the individuals in $S_\mathbf{x}$, and they can serve as the representatives of a specific *family*. This is also why we use the term "*intra-family election*" to denote the above process of selecting two individuals. After performing the strategy of *projection-assisted intra-family election* on every individual $\mathbf{x}$ in P, we can finally obtain a new population Q with $2N$ individuals.

Algorithm 3. Environmental Selection.

Input: population Q
Output: new population P

$P \longleftarrow \emptyset$
for $i \longleftarrow 1$ **to** m **do**
  for each individual $\mathbf{x}$ in Q **do**
    $\mathbf{e}_i = (0, \ldots, 1, \ldots, 0)^T$ (the i-th element is 1 and others are 0's)
    Calculate its angle to $\mathbf{e}_i$
  end for
  Identify two individuals (denoted by $\mathbf{x}_k$ and $\mathbf{x}_h$) that have the minimum and the second minimum angles to $\mathbf{e}_i$
  Select the individual with better convergence between $\mathbf{x}_k$ and $\mathbf{x}_h$, add it into P and remove it from Q
end for
while $|P| + |Q| > N$ **do**
  Identify the pair of individuals (denoted by $\{\mathbf{x}_r, \mathbf{x}_s\}$) with the minimum angle in Q
  Select the individual with worse secondary-criteria between $\mathbf{x}_r$ and $\mathbf{x}_s$, remove it from Q
end while
$P \longleftarrow P \cup Q$
return P

3.3. Mating scheme

As mentioned in Section 3.2, we need to select a mate $\mathbf{x}'$ for each individual $\mathbf{x}$ in P. The mating scheme adopted in MOEA/PIE is inspired by the similarity-based mating scheme proposed by Ishibuchi et al. [59]. However, instead of directly using Euclidean distance, we use the vector angle to measure the similarity of two individuals.

Specifically speaking, the mating scheme in MOEA/PIE works as follows. Firstly, the binary tournament selection procedure[2] is performed β times to select β candidates from the whole population P. Then, we compute the angles between the β candidates and the current individual $\mathbf{x}$, and choose the most similar one (namely, the one which has the minimum angle to $\mathbf{x}$) to serve as the mate for $\mathbf{x}$.

3.4. Selection procedure based on angles

The pseudo-code of selection procedure is given in Algorithm 3. To maintain the diversity of population, the procedure of selection in MOEA/PIE adopts a strategy based on angles and finally outputs a new population P with N diversified individuals. As shown in Algorithm 3, this procedure composes of two main parts: the first one involves the handling of *extreme solutions*, and the second one is the inclusion of non-extreme solutions.

In MOEA/PIE, the *extreme solutions* are not directly added into the new population. Instead, further considerations are carried out to strengthen the convergence of the whole population. In fact, the specific strategy adopted in MOEA/PIE is to identify two individuals that respectively have the minimum and the second minimum angles to $\mathbf{e}_i$ and then make a choice between them according to a measure of convergence. For the convergence measure of an individual $\mathbf{x}$, here we adopt the distance of $\bar{F}(\mathbf{x})$ to the ideal point $\mathbf{z}^{\min} = \left(z_1^{\min}, z_2^{\min}, \ldots, z_m^{\min}\right)^T$ (where $z_i^{\min}$ is the minimum value of the ith objective, $i = 1, 2, \ldots, m$). Since a procedure of normalization is performed in Algorithm 2, the ideal point $\mathbf{z}^{min}$ has become the origin in the normalized objective space. Hence, the distance of $\bar{F}(\mathbf{x})$ to the ideal point simply equals to the length of $\bar{F}(\mathbf{x})$ (that is, $\|\bar{F}(\mathbf{x})\|$). Concretely, suppose the two individuals that respectively have the minimum and the second minimum angles to $\mathbf{e}_i$ are $\mathbf{x}_k$ and $\mathbf{x}_h$, we compare the values of $\|\bar{F}(\mathbf{x}_k)\|$ and $\|\bar{F}(\mathbf{x}_h)\|$. If $\|\bar{F}(\mathbf{x}_k)\| - \|\bar{F}(\mathbf{x}_h)\| \le \frac{1}{2} \cdot \|\bar{F}(\mathbf{x}_h)\|$, which means that the convergence of the individual $\mathbf{x}_k$ is better than that of $\mathbf{x}_h$ to a certain extent, then the individual $\mathbf{x}_k$ is selected to join the new population P, and it is also removed from Q (for no further considerations). Otherwise, if $\|\bar{F}(\mathbf{x}_k)\| - \|\bar{F}(\mathbf{x}_h)\| > \frac{1}{2} \cdot \|\bar{F}(\mathbf{x}_h)\|$, then $\mathbf{x}_h$ is added into the new population P and removed from Q. In this way, we can avoid adding some *extreme solutions* whose objective vectors may be far away from the true PF. Moreover, it is conducive to retaining the convergence when maintaining the diversity.

After the handling of *extreme solutions*, we need to pick enough individuals from Q to fill the new population P. As for the picking

[2] The binary tournament selection procedure adopted in MOEA/PIE favors the non-dominated individual and the one which has a shorter distance to the ideal point. That is, if an individual Pareto dominates another individual, then the one which Pareto dominates another one wins. Moreover, if two individuals are Pareto non-dominated with each other, then the one with a shorter distance to the ideal point wins.

of the remaining individuals, a dynamic method is adopted to promote the diversity of population. Concretely, we first identify the pair of individuals (denoted by $\{\mathbf{x}_r, \mathbf{x}_s\}$) with the minimum angle among all the pairs of individuals in Q. And then the individual with worse secondary-criteria between $\mathbf{x}_r$ and $\mathbf{x}_s$ is removed from Q for no further considerations. The secondary-criteria can be either the aforementioned distance of $\bar{\mathrm{F}}(\mathbf{x})$ to the ideal point or the second minimum angle to the other individuals. Here, we adopt the former one to balance the convergence and diversity. It should be noted that there may exist several pairs of individuals which share the same minimum angle. When this case occurs, we choose the pair with the minimum sum of all the normalized objective values. The aforementioned procedure is repeated until the sum of the size of Q and the size of P decreases to N for the first time. Then the remaining individuals in Q are added into P such that the size of P will become N.

3.5. Discussions

After describing the details of the proposed MOEA/PIE, this subsection attempts to give some discussions on the similarities and differences between MOEA/PIE and other related algorithms.

3.5.1. Similarities and differences between MOEA/PIE and MOEA/D

Unlike the MOEA/D, the MOEA/PIE algorithm does not require a set of predefined weight vectors. Although the role of the *evolution directions* in the MOEA/PIE algorithm is a bit like the weight vectors in the MOEA/D algorithm, the *evolution directions* are adaptively generated and set according to the current population, which can dynamically change the search directions during the evolutionary process. The dynamic nature lies in two aspects. One is that every individual has a *basic evolution direction*, and the other is the adaptive method for setting different *evolution directions* based on the dominance relationship.

In the MOEA/PIE algorithm, a fitness function as shown in Eq. (3) is defined based on the *basic evolution direction* $\mathbf{v_x}$. The fitness function is a bit like the penalty-based boundary intersection (PBI) function used in the MOEA/D algorithm. In fact, Eq. (3) together with Eqs. (4) and (5) can be converted to two PBI functions as follows:

$$\begin{aligned} g\left(\mathbf{y} \mid \mathbf{v_x}\right) &= (-a_1) + \theta \cdot \|\mathbf{a}_2\| \\ &= g^{pbi}\left(\mathbf{y} \mid \mathbf{v_x}, \mathbf{z}^{\min}\right) - g^{pbi}\left(\mathbf{x} \mid \mathbf{v_x}, \mathbf{z}^{\min}\right) \end{aligned} \tag{6}$$

3.5.2. Similarities and differences between MOEA/PIE and VaEA

Both of the two algorithms mainly utilize the angle between the objective vectors of two individuals to measure the closeness between a pair of individuals, and use a type of dynamic method for diversity promoting. However, there are two main differences between them.

(1) First of all, the handling method of *extreme solutions* in these two algorithms are different. The VaEA algorithm identifies these individuals that have the minimum angles to $\mathbf{e}_i$ (i = 1, 2, . . ., m), and directly adds these *extreme solutions* into the new population. However, in the MOEA/PIE algorithm, for every i (i = 1, 2, . . ., m), two individuals that respectively have the minimum and the second minimum angles to $\mathbf{e}_i$ are found and a choice will be made between them according to a measure of convergence.
(2) Secondly, the MOEA/PIE algorithm adopts a dynamic deleting method while the VaEA algorithm adopts a dynamic adding method. The VaEA algorithm utilizes the so-called *maximum-vector-angle-first* principle to select individuals from the last non-dominated front one by one, and uses the *worse-elimination* principle to allow individuals with poor convergence to be replaced by other individuals. However, the method adopted in the MOEA/PIE algorithm is different from that in the VaEA algorithm. In the MOEA/PIE algorithm, the pair of individuals with the minimum angle among all the pairs of individuals are identified, and then the individual with worse secondary-criteria between this pair of individuals is removed. The method adopted in the MOEA/PIE algorithm can be called a dynamic deleting method.

3.5.3. Similarities and differences between MOEA/PIE and RVEA

Both of MOEA/PIE and RVEA involve some identical or similar concepts. However, they are different in the concrete ideas adopted in the algorithm, which can be concluded as follows:

(1) Both of the two algorithms utilize the angle information (that is, the angle between any two vectors). The RVEA algorithm mainly utilizes the angles between the objective vectors and reference vectors to associate each individual with its closest reference vector. However, the MOEA/PIE algorithm utilizes the angle between the objective vectors of two individuals to measure the closeness between a pair of individuals.
(2) Both of the two algorithms involve a normalization procedure. The RVEA algorithm focuses on the normalization of angles, whereas the MOEA/PIE algorithm conducts the normalization of objective vectors during the evolution process.
(3) Although the RVEA algorithm adopts a reference vector adaptation strategy which can adapt the reference vectors according to the ranges of different objective functions, it still requires a set of predefined reference vectors as the input. By contrast, the MOEA/PIE algorithm does not require a set of predefined weight vectors or reference vectors.

4. Experimental setup

In this section, we attempt to empirically investigate the performance of the proposed MOEA/PIE algorithm. Firstly, we describe the benchmark problems and performance metrics used in our experiments. Then, we briefly introduce four state-of-the-art algorithms employed for comparisons and verifying the effectiveness of the proposed MOEA/PIE. Finally, we provide the general parameter settings adopted in our empirical studies.

4.1. Benchmark problems

In the empirical experiments, we adopt two well-known test suites: DTLZ [60] and WFG [61]. Concretely, we choose DTLZ1 to DTLZ4 from the former test suite and WFG1 to WFG9 from the latter test suite. Besides, we also adopt two scaled test problems including ScaledDTLZ1 and ScaledDTLZ2 [21]. From the construction of the two test suites, we can know that all these test problems can be scaled to any number of objectives and decision variables. In practice, we set the number of objectives for all the problems to different values (m = 5, 10, 15, 20). As suggested in [60], the number of decision variables for DTLZ test problems is set to $n = m + k - 1$, where k = 5 for DTLZ1 and ScaledDTLZ1 and k = 10 for DTLZ2, ScaledDTLZ1, DTLZ3 and DTLZ4. As for WFG test problems, the number of decision variables is set to $n = 2 \times (m - 1) + l$, where l = 20 as suggested in [61]. ScaledDTLZ1 and ScaledDTLZ2 are the modified versions of DTLZ1 and DTLZ2 respectively. Both of them involve a scaling factor s^i for the ith objective. According to [21], when the number of objectives is set to different values (m = 5, 10, 15, 20), the corresponding s is also set to different values (s = 10, 2, 1.2, 1.2 for ScaledDTLZ1, and s = 10, 3, 2, 1.2 for ScaledDTLZ2).

In fact, the aforementioned benchmark problems possess a wide variety of sophisticated characteristics, such as multi-modality, non-separability, biased parameters, convex or concave PF and mixed PF geometries [55,62]. These different characteristics make the benchmark problems capable of challenging the performance of an MOEA in terms of both convergence and diversity.

4.2. Performance metrics

In our empirical studies, two widely used performance metrics are chosen to evaluate the performance of each algorithm, including the Inverse Generational Distance (IGD) [63,64] and generalized SPREAD [65].

1. ***IGD:*** Let P be the set of final non-dominated points obtained by an MOEA, and P^* be a set of points uniformly sampled over the true Pareto front. Then the IGD of P is defined as:

$$IGD(P) = \frac{1}{|P^*|} \sum_{\mathbf{x}^* \in P^*} d(\mathbf{x}^*, P) \tag{7}$$

where $d(\mathbf{x}^*, P)$ is the Euclidean distance between a point $\mathbf{x}^* \in P^*$ and its nearest neighbor in P, and $|P^*|$ is the cardinality of P^*. In brief, IGD measures the convergence and diversity of the obtained solutions. The smaller the IGD value is, the better the quality of the final non-dominated points obtained by an MOEA is.

- ***Generalized SPREAD:*** Let P be a set of solutions obtained by an MOEA, and P^* be a set of known Pareto-optimal solutions. $\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_m$ are m extreme solutions in P^*. Then the generalized SPREAD (denoted by Δ) of P is defined as:

$$\Delta(P) = \frac{\sum_{i=1}^{m} d(\mathbf{e}_i, P) + \sum_{\mathbf{x}^* \in P^*} \left| d(\mathbf{x}^*, S) - \bar{d} \right|}{\sum_{i=1}^{m} d(\mathbf{e}_i, P) + |P^*| \cdot \bar{d}} \tag{8}$$

where $d(\mathbf{x}^*, P) = \min_{\mathbf{y} \in P, \mathbf{y} \neq \mathbf{x}^*} \|\mathbf{F}(\mathbf{x}^*) - \mathbf{F}(\mathbf{y})\|^2$ and $\bar{d} = \frac{1}{|P^*|} \sum_{\mathbf{x}^* \in P^*} d(\mathbf{x}^*, P)$. In brief, Δ measures the diversity of the obtained solutions. The smaller the value of Δ is, the better the diversity of the final non-dominated points obtained by an MOEA is. From the definitions of IGD and Δ (see Eqs. (7) and (8)), we can know that the calculations of the two metrics need to first determine a set of known Pareto-optimal solutions. Actually, it involves the issue of sampling points from the true PF. In practice, we adopt the sampling method as suggested in [66].

4.3. Algorithms for comparisons

In the empirical experiments, our proposed MOEA/PIE is compared with four state-of-the-art many-objective evolutionary algorithms, including NSGA-III [21], MOEA/D [17], MOEA/DD [55] and VaEA [66]. These algorithms are representative algorithms for multi-/many- objective optimization and have been demonstrated to be capable of effectively handling MaOPs. Thus, they can be employed to provide a rigorous performance baseline for validating the performance of the new algorithm.

All the algorithms (including the proposed MOEA/PIE and four comparison algorithms) are implemented in the jMetal framework [67], and executed on a PC, which runs Microsoft Windows 7 SP1 64-bit operating system on 3.30 GHz Intel(R) Core(TM) Dual CPUs with 4 GB RAM.

4.4. General parameter settings

The parameter settings in our empirical studies are listed as follows:

Table 1
Settings of *MFE*.

m	*MFE*				
	DTLZ1	DTLZ2	DTLZ3	DTLZ4	Others
5	212×600	212×350	212×1000	212×1000	212×1250
10	276×1000	276×750	276×1500	276×2000	276×2000
15	136×1500	136×1000	136×2000	136×3000	136×3000
20	250×1500	250×1000	250×2000	250×3000	250×3000

Table 2
Settings of *N*.

m	H	N	
		NSGA-III &VaEA &MOEA/PIE	MOEA/D &MOEA/DD
5	210 ($p = 12$)	212	210
10	275 ($p_1 = 3, p_2 = 2$)	276	275
15	135 ($p_1 = 2, p_2 = 1$)	136	135
20	250 ($p_1 = 2, p_2 = 1$)	250	250

Table 3
Parameter settings for crossover and mutation.

Parameter	Value/form
Crossover operator	SBX
Mutation operator	Polynomial
Crossover probability (p_c)	1.0
Mutation probability (p_m)	$1/n$
Distribution index for crossover (η_c)	30
Distribution index for mutation (η_m)	20

1. ***Number of runs:*** Each algorithm is independently run 20 times on each test instance.
2. ***Termination criterion:*** The termination criterion of an algorithm is a predefined maximum function evaluations (*MFE* for short). The settings of *MFE* for different numbers of objectives and different problems are summarized in Table 1.
3. ***Population size:*** It is worth noting that the three comparison algorithms (NSGA-III, MOEA/D and MOEA/DD) need to provide a set of reference points or weight vectors beforehand. And the size of population (N) should be specified according to the number of reference points/weight vectors (H). As for the generation method of reference points/weight vectors, we adopt Das and Dennis's systematic approach [57]. In practice, one-layer method (depending on the number of divisions p) is applied for 5-objective problems, and two-layer method (depending on the number of divisions for the boundary and inside layer, denoted by p_1 and p_2 respectively) is applied for 10-, 15- and 20-objective problems. As suggested in the original literatures of different algorithms [17,55,21,66], the population size in MOEA/D and MOEA/DD is directly set to H, while the population size in NSGA-III is set to the smallest multiple of four larger than H. As for the proposed MOEA/PIE and VaEA, we use the same population size as in NSGA-III. Concretely, the settings of the population size N for different numbers of objectives and different algorithms are summarized in Table 2.
4. ***Parameters for crossover and mutation:*** For the genetic operators in all the considered algorithms, we use a polynomial mutation operator [68] and a simulated binary crossover (SBX) operator [69] to generate offspring. And the specific settings for crossover and mutation are shown in Table 3.
5. ***Additional parameters in each algorithm:*** In MOEA/D, MOEA/DD and VaEA, there are some additional parameters which need to be specified. In practice, we follow the suggestions given by the developers of these algorithms, and keep the settings of these additional parameters the same as in the original

Table 4
Median of IGD on WFG test suite. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	m	MOEA/PIE	NSGA-III	MOEA/D	MOEA/DD	VaEA
WFG1	5	3.320E+00	3.028E+00◦	1.905E+00◦	3.328E+00‡	2.659E+00◦
	10	7.269E+00	7.043E+00◦	7.034E+00◦	9.008E+00•	7.232E+00‡
	15	9.640E+00	9.060E+00◦	5.743E+00◦	1.214E+01•	9.152E+00◦
	20	1.818E+01	1.697E+01◦	2.244E+01•	2.334E+01•	1.706E+01◦
WFG2	5	5.164E−01	3.214E−01◦	1.293E+00•	7.798E−01•	4.183E−01◦
	10	1.140E+00	9.830E−01◦	1.471E+00•	1.611E+00•	1.093E+00‡
	15	2.123E+00	3.683E+00•	3.332E+00•	4.060E+00•	2.365E+00•
	20	2.282E+00	2.738E+00•	3.104E+00•	3.085E+00•	2.294E+00•
WFG3	5	5.734E−01	5.433E−01◦	9.106E−01•	5.688E−01‡	4.826E−01◦
	10	2.050E+00	2.225E+00•	3.569E+00•	1.858E+00◦	1.831E+00◦
	15	2.832E+00	3.970E+00•	7.978E+00•	3.783E+00•	2.855E+00•
	20	3.592E+00	5.401E+00•	1.285E+01•	7.244E+00•	3.819E+00•
WFG4	5	9.540E−01	9.907E−01•	2.992E+00•	1.009E+00•	9.439E−01◦
	10	4.087E+00	4.514E+00•	6.900E+00•	6.417E+00•	4.007E+00‡
	15	8.981E+00	9.877E+00•	1.309E+01•	9.824E+00•	8.312E+00◦
	20	1.194E+01	1.162E+01◦	2.206E+01‡	1.182E+01‡	1.113E+01◦
WFG5	5	9.634E−01	9.785E−01•	2.727E+00•	9.905E−01•	9.528E−01◦
	10	4.029E+00	4.517E+00•	6.438E+00•	6.392E+00•	4.004E+00◦
	15	9.063E+00	9.881E+00•	1.576E+01•	1.401E+01•	8.257E+00◦
	20	1.123E+01	1.170E+01•	2.505E+01•	1.609E+01•	1.076E+01◦
WFG6	5	9.608E−01	9.862E−01•	2.062E+00•	1.003E+00•	9.732E−01•
	10	4.216E+00	4.529E+00•	7.913E+00•	6.248E+00•	4.055E+00◦
	15	9.907E+00	9.893E+00‡	1.417E+01•	1.401E+01•	8.315E+00◦
	20	1.209E+01	1.173E+01◦	1.981E+01•	1.638E+01•	1.062E+01◦
WFG7	5	9.627E−01	1.000E+00•	2.820E+00•	1.011E+00•	9.621E−01◦
	10	4.118E+00	4.538E+00•	6.875E+00•	5.771E+00•	4.013E+00◦
	15	8.610E+00	9.910E+00•	1.733E+01•	1.273E+01•	8.276E+00◦
	20	1.139E+01	1.169E+01•	2.514E+01•	1.136E+01‡	1.083E+01◦
WFG8	5	1.049E+00	1.006E+00◦	2.996E+00•	1.031E+00◦	1.084E+00•
	10	4.365E+00	4.289E+00◦	7.256E+00•	6.171E+00•	4.173E+00◦
	15	9.427E+00	9.651E+00‡	1.757E+01•	1.317E+01•	8.824E+00◦
	20	1.243E+01	1.229E+01◦	2.624E+01‡	1.119E+01◦	1.239E+01‡
WFG9	5	9.985E−01	9.326E−01◦	2.595E+00•	1.004E+00‡	9.850E−01◦
	10	3.894E+00	4.366E+00•	6.744E+00•	5.640E+00•	3.922E+00•
	15	8.428E+00	9.161E+00•	1.710E+01•	1.071E+01•	8.274E+00◦
	20	1.184E+01	1.087E+01◦	2.525E+01•	1.664E+01•	1.080E+01◦

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

literatures [17,55,66]. As for the proposed MOEA/PIE, the settings of three additional parameters[3] are as follows: $K = 3$, $\beta = 5$, $\theta = m$.

5. Experimental results

In this section, we will present in detail the experimental results of the comparative study between the proposed MOEA/PIE and other four prominent algorithms (NSGA-III, MOEA/D, MOEA/DD and VaEA).

5.1. Performance comparisons on WFG test problems

Table 4 shows the comparison results of MOEA/PIE and other four algorithms on WFG test problems in terms of the IGD metric. Here, the value of the IGD metric is expressed in the form of "*median*", where "*median*" refers to the median value of 20 runs. In different cases, the best metric values among all the algorithms are highlighted with grey shade while the second best ones are highlighted with light grey shade.

As can be clearly seen from Table 4, in terms of IGD, MOEA/PIE and VaEA clearly outperform NSGA-III, MOEA/D and MOEA/DD. More specifically, out of the 36 test instances, MOEA/PIE obtains 19 best and second best IGD results while VaEA obtains 32 ones. MOEA/PIE performs best on WFG3, WFG4, WFG5 and WFG7, while the performance of NSGA-III is competitive with that of MOEA/PIE on WFG2, WFG6 and WFG9. Besides, VaEA performs better than MOEA/PIE on WFG4, WFG5, WFG6 and WFG7.

Now we focus on the detailed comparisons between MOEA/PIE and other peer algorithms. As can be seen from Table 4, the Wilcoxon's rank sum test [70] is performed to determine the significance of difference between MOEA/PIE and each peer algorithm. Relying on a preset significance level (that is, $\alpha = 0.05$), we can observe that significant difference can be detected on the vast majority of the test instances. Particularly, the proportion of the test instances where MOEA/PIE significantly outperforms NSGA-III, MOEA/D, MOEA/DD and VaEA is 20/36, 32/36, 28/36 and 7/36, respectively. On the contrary, the proportion of the test instances where MOEA/PIE is beaten out by NSGA-III, MOEA/D, MOEA/DD and VaEA is 14/36, 3/36, 3/36 and 25/36, respectively.

The comparison results of different algorithms on WFG test problems in terms of the Δ metric are presented in Table 5. The key observation from Table 5 is that in terms of Δ, the performances of MOEA/PIE and VaEA are far better than the other three algorithms on most of the test instances. More specifically, out of the 36 test instances, MOEA/PIE obtains 19 best results, while VaEA obtains 15 ones. MOEA/PIE can obtain a good distribution of solutions, especially on WFG3-9. In short, the whole performance of MOEA/PIE in

[3] The effect of these additional parameter on the performance of the proposed MOEA/PIE algorithm will be shown in Section 6.

Table 5
Median of Δ on WFG test suite. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	m	MOEA/PIE	NSGA-III	MOEA/D	MOEA/DD	VaEA
WFG1	5	5.916E − 01	5.981E − 01‡	1.291E + 00•	4.380E − 01∘	4.913E − 01∘
	10	7.597E − 01	7.254E − 01∘	1.107E + 00•	8.035E − 01•	6.921E − 01∘
	15	9.310E − 01	9.065E − 01∘	1.070E + 00•	9.578E − 01•	9.049E − 01∘
	20	9.768E − 01	1.013E + 00•	1.014E + 00•	1.016E + 00•	1.041E + 00•
WFG2	5	4.198E − 01	2.075E − 01∘	1.397E + 00•	3.507E − 01∘	4.151E − 01∘
	10	6.085E − 01	6.044E − 01‡	1.085E + 00•	6.219E − 01•	4.918E − 01∘
	15	8.216E − 01	8.386E − 01‡	1.078E + 00•	9.571E − 01•	7.169E − 01∘
	20	8.241E − 01	8.538E − 01‡	1.042E + 00•	1.002E + 00•	7.086E − 01∘
WFG3	5	2.294E − 01	5.596E − 01•	1.048E + 00•	3.323E − 01•	2.621E − 01•
	10	2.207E − 01	7.352E − 01•	1.268E + 00•	4.557E − 01•	2.119E − 01∘
	15	3.619E − 01	1.022E + 00•	1.111E + 00•	7.145E − 01•	2.981E − 01∘
	20	3.006E − 01	1.137E + 00•	1.109E + 00•	6.213E − 01•	2.233E − 01∘
WFG4	5	9.476E − 02	1.502E − 01•	1.650E + 00•	3.175E − 01•	1.128E − 01•
	10	1.061E − 01	1.529E − 01•	1.272E + 00•	4.814E − 01•	1.082E − 01•
	15	2.193E − 01	3.078E − 01•	1.136E + 00•	6.153E − 01•	2.586E − 01•
	20	1.499E − 01	2.550E − 01•	1.171E + 00•	7.666E − 01•	1.924E − 01•
WFG5	5	1.118E − 01	1.564E − 01•	1.352E + 00•	3.203E − 01•	1.119E − 01•
	10	1.025E − 01	1.545E − 01•	1.237E + 00•	4.856E − 01•	1.069E − 01•
	15	2.367E − 01	3.005E − 01•	1.113E + 00•	7.504E − 01•	2.418E − 01•
	20	1.856E − 01	2.544E − 01•	1.102E + 00•	7.589E − 01•	1.781E − 01∘
WFG6	5	9.868E − 02	1.497E − 01•	5.649E − 01•	3.184E − 01•	1.202E − 01•
	10	1.078E − 01	1.551E − 01•	8.705E − 01•	4.691E − 01•	1.073E − 01∘
	15	2.269E − 01	3.105E − 01•	9.001E − 01•	7.537E − 01•	2.831E − 01•
	20	1.663E − 01	2.459E − 01•	8.683E − 01•	7.679E − 01•	2.103E − 01•
WFG7	5	9.861E − 02	1.480E − 01•	1.718E + 00•	3.166E − 01•	1.118E − 01•
	10	1.049E − 01	1.557E − 01•	1.334E + 00•	4.684E − 01•	1.093E − 01•
	15	2.400E − 01	3.032E − 01•	1.144E + 00•	7.193E − 01•	3.085E − 01•
	20	1.286E − 01	3.372E − 01•	1.174E + 00•	5.794E − 01•	1.971E − 01•
WFG8	5	1.523E − 01	2.307E − 01•	1.588E + 00‡	3.591E − 01•	1.436E − 01∘
	10	2.419E − 01	3.596E − 01•	1.361E + 00•	4.783E − 01•	1.955E − 01∘
	15	4.776E − 01	4.477E − 01‡	1.149E + 00•	7.823E − 01•	3.632E − 01∘
	20	2.320E − 01	6.041E − 01•	1.158E + 00•	4.976E − 01•	2.282E − 01∘
WFG9	5	1.029E − 01	1.788E − 01•	1.312E + 00•	3.604E − 01•	1.135E − 01•
	10	1.229E − 01	2.031E − 01•	1.287E + 00•	4.894E − 01•	1.145E − 01∘
	15	1.861E − 01	3.164E − 01•	1.127E + 00•	8.849E − 01•	2.671E − 01•
	20	9.359E − 02	2.893E − 01•	1.155E + 00•	8.742E − 01•	1.865E − 01•

Table 6
Median of IGD on DTLZ test suite. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	m	MOEA/PIE	NSGA-III	MOEA/D	MOEA/DD	VaEA
DTLZ1	5	5.960E − 02	5.298E − 02∘	1.664E − 01•	5.293E − 02∘	5.422E − 02∘
	10	1.087E − 01	1.094E − 01•	1.522E − 01•	1.088E − 01‡	1.097E − 01•
	15	1.490E − 01	1.759E − 01•	1.954E − 01•	1.710E − 01•	1.598E − 01•
	20	1.533E − 01	1.929E − 01•	2.003E − 01•	1.764E − 01•	1.621E − 01•
DTLZ2	5	1.660E − 01	1.660E − 01‡	3.147E − 01•	1.659E − 01‡	1.666E − 01‡
	10	4.013E − 01	4.216E − 01•	6.443E − 01•	4.212E − 01•	4.154E − 01•
	15	5.867E − 01	6.204E − 01•	9.137E − 01•	6.194E − 01•	6.001E − 01•
	20	6.087E − 01	6.351E − 01•	9.681E − 01•	6.340E − 01•	6.283E − 01•
DTLZ3	5	1.725E − 01	1.662E − 01∘	8.561E − 01•	1.661E − 01∘	1.657E − 01∘
	10	4.074E − 01	4.215E − 01•	6.509E − 01•	4.214E − 01•	4.326E − 01•
	15	5.929E − 01	6.222E − 01•	9.143E − 01•	6.200E − 01•	7.532E − 01•
	20	6.220E − 01	6.370E − 01•	9.876E − 01•	6.344E − 01‡	7.910E − 01•
DTLZ4	5	1.668E − 01	1.659E − 01∘	3.553E − 01•	1.660E − 01∘	1.662E − 01‡
	10	4.007E − 01	4.206E − 01•	6.905E − 01•	4.218E − 01•	4.120E − 01•
	15	5.868E − 01	6.184E − 01•	8.906E − 01•	6.176E − 01•	5.985E − 01•
	20	6.101E − 01	6.342E − 01•	9.594E − 01•	6.343E − 01•	6.202E − 01•
ScaledDTLZ1	5	5.784E + 01	1.799E + 02•	4.919E + 02•	1.035E + 03•	5.238E + 01∘
	10	1.299E + 01	1.338E + 01‡	3.648E + 01•	3.701E + 01•	7.335E + 00∘
	15	7.794E − 01	8.927E − 01•	1.193E + 00•	7.942E − 01‡	6.350E − 01∘
	20	1.189E + 00	1.176E + 00‡	1.715E + 00•	1.094E + 00∘	1.129E + 00∘
ScaledDTLZ2	5	1.690E + 02	2.570E + 02•	1.312E + 03•	3.441E + 03•	1.426E + 02∘
	10	8.032E + 02	1.064E + 03•	4.966E + 03•	4.971E + 03•	5.750E + 02∘
	15	1.247E + 03	1.751E + 03•	3.595E + 03•	3.594E + 03•	7.214E + 02∘
	20	2.783E + 00	2.518E + 00∘	6.323E + 00•	4.704E + 00•	2.665E + 00∘

terms of Δ is significantly superior to that of NSGA-III, MOEA/D and MOEA/DD, and is slightly better than that of VaEA.

To intuitively understand the distribution of the final solutions in high-dimensional objective space, we take the 15-objective WFG3 and 15-objective WFG9 for an example. In fact, WFG3 problem has a degenerate and linear PF, while WFG9 problem has a multi-modal, biased and deceptive search space and a concave PF. These two problems pose a significant challenge for MOEAs to obtain a solution set with good convergence and good diversity. This is why we choose them to deeply investigate the distribution of the solutions set.

Figs. 2 and 3 show the parallel coordinates of final solutions obtained by five algorithms on 15-objective WFG9 and 15-objective WFG3, respectively. These plots are corresponding to the particular run in which the IGD value is closest to the median value over 20 runs.

From Fig. 2, we can observe that the final solutions of MOEA/PIE, NSGA-III and VaEA are well distributed along the true PF of WFG9, which means that these three algorithms are capable of finding a good approximation and coverage of the PF. The solutions obtained by MOEA/D can only converge to a part of the PF, while those obtained by MOEA/DD fail to cover the intermediate values in every objective.

As can be seen from Fig. 3, the distribution of the final solutions obtained by MOEA/D and MOEA/DD on 15-objective WFG3 are similar to that of the final solutions obtained by the two algorithms on

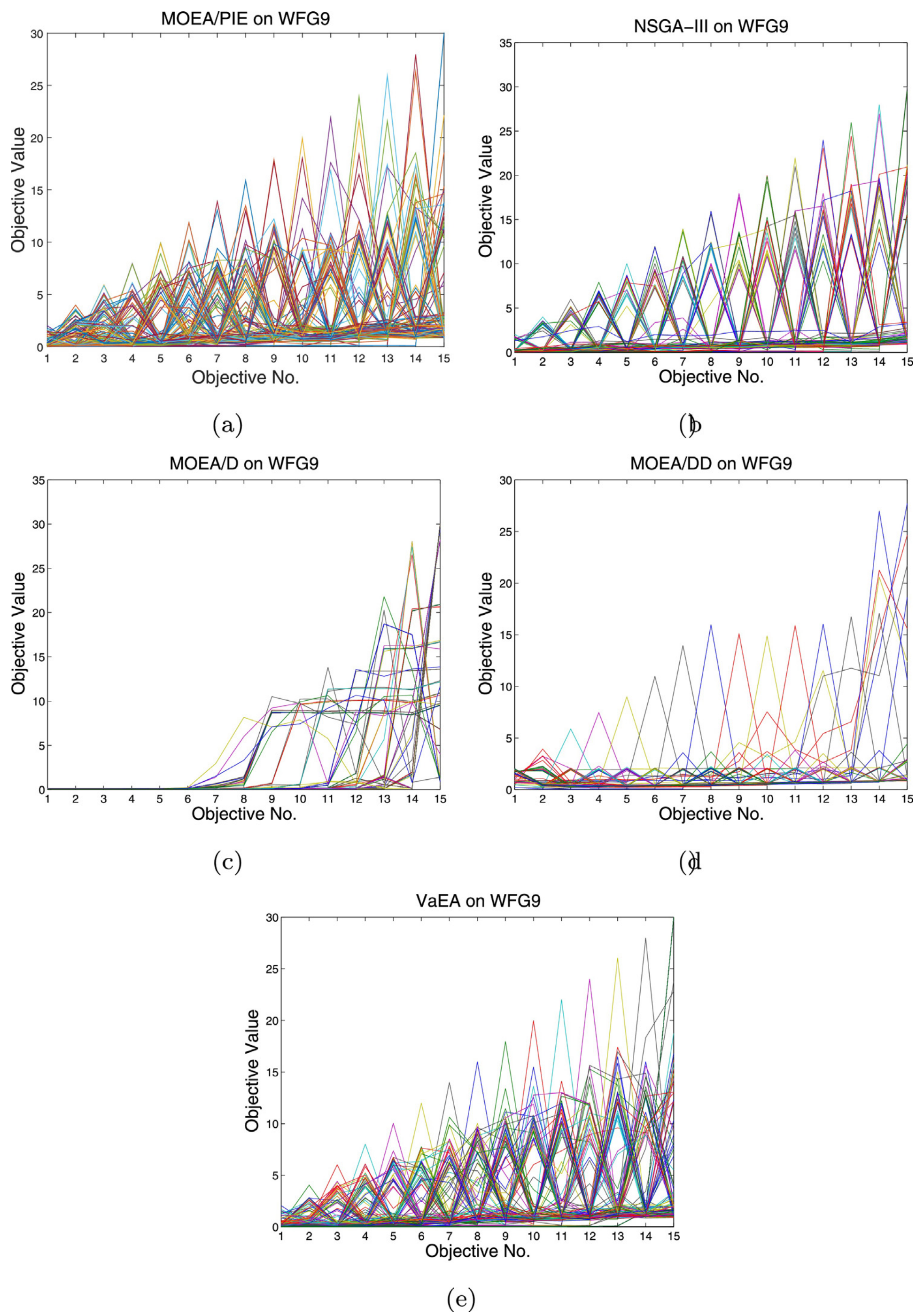

Fig. 2. Parallel coordinates of final solutions obtained by five algorithms on 15-objective WFG9.

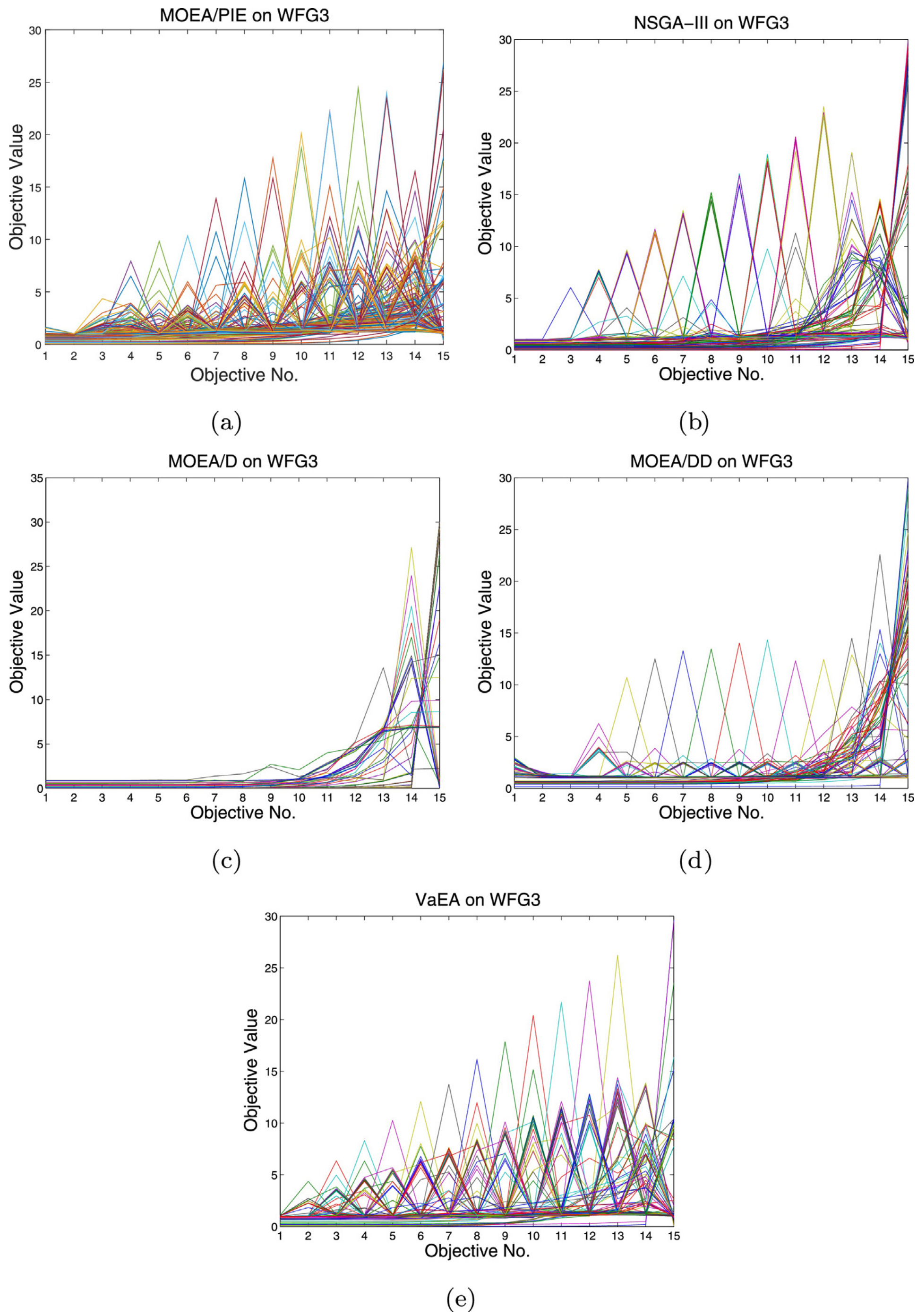

Fig. 3. Parallel coordinates of final solutions obtained by five algorithms on 15-objective WFG3.

15-objective WFG9, respectively. As for MOEA/PIE, NSGA-III and VaEA, the final solutions of the three algorithms are well distributed along the true PF of WFG3. The solutions obtained by MOEA/PIE and VaEA can cover more values in every objective than the solutions obtained by NSGA-III. However, these three algorithms both fail to converge to a certain part of the PF (that is, the boundaries on the 13-th and 14-th objective).

5.2. Performance comparisons on DTLZ test problems

The comparison results of different algorithms on DTLZ test problems in terms of IGD and Δ are presented in Tables 6 and 7, respectively.

The main observation from Table 6 is that in terms of IGD, MOEA/PIE has the best performance on most of the test instances while the whole performance of VaEA ranks second. Out of the 24 test instances, the number of the best IGD results MOEA/PIE obtains is 12. The proportion of the test instances where MOEA/PIE significantly outperforms NSGA-III, MOEA/D, MOEA/DD and VaEA is 17/24, 24/24, 16/24 and 12/24, respectively. On the contrary, the proportion of the test instances where MOEA/PIE is beaten out by NSGA-III, MOEA/D, MOEA/DD and VaEA is 4/24, 0/24, 4/24 and 9/24, respectively.

As can be seen from Table 7, in terms of Δ, the whole performance of MOEA/PIE is the best. For DTLZ3 and DTLZ4, MOEA/PIE performs best. For DTLZ2 and ScaledDTLZ2, MOEA/PIE is quite com-

Table 7
Median of Δ on DTLZ test suite. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	m	MOEA/PIE	NSGA-III	MOEA/D	MOEA/DD	VaEA
DTLZ1	5	1.891E − 01	9.800E − 03∘	1.407E + 00•	1.430E − 03∘	1.978E − 01•
	10	2.161E − 01	6.351E − 02∘	6.166E − 01•	5.719E − 02∘	3.074E − 01•
	15	4.085E − 01	3.843E − 02∘	8.837E − 01•	1.539E − 02∘	5.259E − 01•
	20	3.814E − 01	2.145E − 02∘	9.200E − 01•	1.750E + 00•	6.639E − 01•
DTLZ2	5	1.057E − 01	1.476E − 01•	1.745E + 00•	1.474E − 01•	1.103E − 01•
	10	1.092E − 01	1.500E − 01•	1.014E + 00•	1.455E − 01•	1.043E − 01∘
	15	2.544E − 01	3.063E − 01•	1.095E + 00•	2.899E − 01•	2.631E − 01•
	20	1.890E − 01	2.468E − 01•	1.090E + 00•	2.976E − 01•	1.768E − 01∘
DTLZ3	5	1.158E − 01	1.469E − 01•	1.381E + 00•	1.475E − 01•	1.195E − 01•
	10	1.146E − 01	1.517E − 01•	9.692E − 01•	1.459E − 01•	2.647E − 01•
	15	2.765E − 01	3.082E − 01•	1.042E + 00•	2.897E − 01•	1.158E + 00•
	20	2.085E − 01	2.514E − 01•	1.108E + 00•	2.967E − 01•	7.656E − 01•
DTLZ4	5	1.118E − 01	1.474E − 01•	1.834E + 00•	1.475E − 01•	1.128E − 01•
	10	1.010E − 01	1.486E − 01•	1.360E + 00•	1.474E − 01•	1.182E − 01•
	15	2.166E − 01	3.011E − 01•	1.156E + 00•	2.816E − 01•	3.235E − 01•
	20	1.300E − 01	2.503E − 01•	1.182E + 00•	3.709E − 01•	2.353E − 01•
ScaledDTLZ1	5	1.833E − 01	1.194E − 02∘	1.297E + 00•	9.936E − 01•	2.499E − 01•
	10	2.599E − 01	7.432E − 02∘	1.651E + 00•	9.404E − 01•	3.462E − 01•
	15	3.799E − 01	3.102E − 02∘	8.582E − 01•	6.243E − 01•	4.662E − 01•
	20	6.422E − 01	9.145E − 01•	9.558E − 01•	1.814E + 00•	7.256E − 01•
ScaledDTLZ2	5	1.071E − 01	1.472E − 01•	1.501E + 00•	1.047E + 00•	1.099E − 01•
	10	1.151E − 01	1.520E − 01•	1.205E + 00•	1.078E + 00•	1.044E − 01∘
	15	2.808E − 01	2.995E − 01‡	1.087E + 00•	1.082E + 00•	2.439E − 01∘
	20	1.834E − 01	2.468E − 01•	9.811E − 01•	7.920E − 01•	1.731E − 01∘

Table 8
Median of IGD on 5-objective problems. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	$\theta = 0.0$	$\theta = 5.0$	$\theta = 10.0$	$\theta = 15.0$	$\theta = 20.0$
DTLZ1	6.035E − 02	5.911E − 02	5.873E − 02	5.900E − 02	5.897E − 02
DTLZ2	2.048E − 01	1.657E − 01	1.663E − 01	1.664E − 01	1.665E − 01
DTLZ3	1.724E − 01	1.723E − 01	1.719E − 01	1.701E − 01	1.727E − 01
DTLZ4	2.093E − 01	1.669E − 01	1.670E − 01	1.666E − 01	1.668E − 01
ScaledDTLZ1	1.450E + 02	5.788E + 01	5.647E + 01	5.942E + 01	5.895E + 01
ScaledDTLZ2	1.654E + 02	1.609E + 02	1.697E + 02	1.662E + 02	1.771E + 02
WFG1	3.361E + 00	3.330E + 00	3.339E + 00	3.336E + 00	3.343E + 00
WFG2	3.060E − 01	2.466E − 01	2.505E − 01	2.521E − 01	2.480E − 01
WFG3	5.833E − 01	3.738E − 01	3.746E − 01	3.722E − 01	3.768E − 01
WFG4	9.482E − 01	9.492E − 01	9.548E − 01	9.523E − 01	9.550E − 01
WFG5	9.569E − 01	9.632E − 01	9.667E − 01	9.639E − 01	9.626E − 01
WFG6	9.533E − 01	9.609E − 01	9.624E − 01	9.637E − 01	9.630E − 01
WFG7	9.791E − 01	9.658E − 01	9.574E − 01	9.579E − 01	9.687E − 01
WFG8	1.058E + 00	1.047E + 00	1.043E + 00	1.045E + 00	1.044E + 00
WFG9	9.664E − 01	9.912E − 01	1.004E + 00	9.991E − 01	9.926E − 01

petitive with VaEA. For ScaledDTLZ1, the performance of MOEA/PIE is competitive with NSGA-III.

To intuitively understand the distribution of the final solutions in high-dimensional objective space, we take the 15-objective DTLZ4 and 15-objective DTLZ3 for an example. The parallel coordinates of final solutions obtained by five algorithms on 15-objective DTLZ4 and 15-objective DTLZ3 are plotted in Figs. 5 and 4, respectively.

Obviously, from Fig. 4, the final solutions of MOEA/PIE are well distributed along the true PF. Moreover, the solutions obtained by MOEA/PIE can cover more values in every objective than that of NSGA-III, MOEA/DD and VaEA. The solutions obtained by NSGA-III and MOEA/DD are similar, both of which mainly cover three values in every objective. However, according to the newest literature [71], we can only conclude that the performance of these three algorithms are similar. As can be seen from Fig. 5, in terms of the distribution of final solutions, the performances of MOEA/PIE, NSGA-III and MOEA/DD on 15-objective DTLZ3 are similar to the performances of these three algorithms on 15-objective DTLZ4, respectively. However, for this problem, VaEA obtains some extremely high values on several objectives.

6. Further discussions

In the proposed MOEA/PIE algorithm, there are three additional parameters (K, β and θ), which need to be specified by the users. Thus, in this section, we attempt to empirically investigate how different parameter settings produce an effect on the performance of the proposed MOEA/PIE algorithm. In addition to presenting the experimental results of the parameter study, we also try to provide recommendations for how to set the parameters in MOEA/PIE. What's more, we also empirically investigate how the choice of the *basic evolution direction* affects the proposed algorithm.

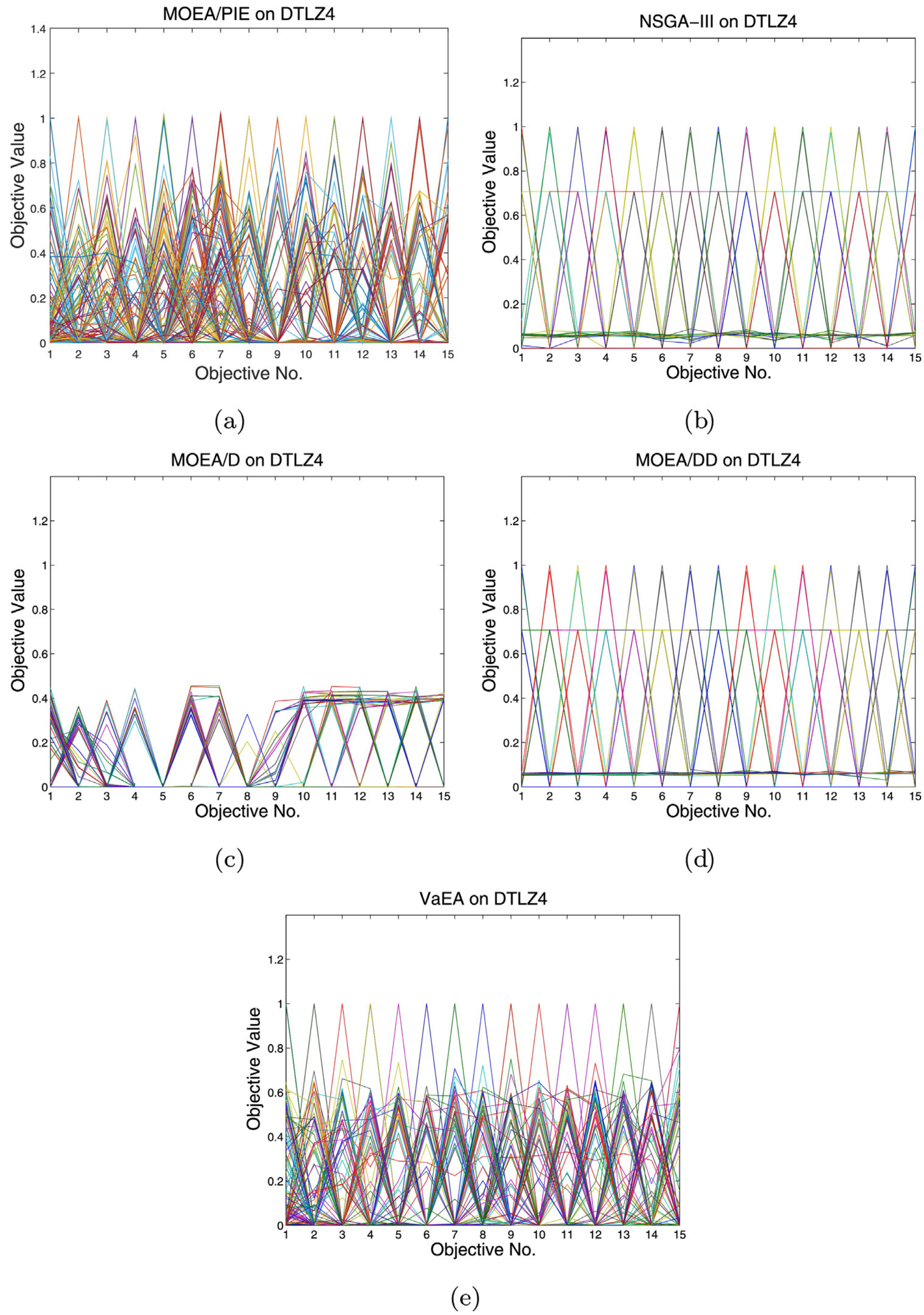

Fig. 4. Parallel coordinates of final solutions obtained by five algorithms on 15-objective DTLZ4.

6.1. On the effect of parameter θ

θ is a user-defined control parameter used in the definition of the fitness $g(\mathbf{y} \mid \mathbf{v_x})$. By adjusting the value of θ, we can control the balance between convergence and diversity to a certain extent.

In the empirical experiments, we choose DTLZ1-4, ScaledDTLZ1-2 and WFG1-9 as the test problems, and set the number of objectives for all the problems to different values (m = 5, 10, 15, 20). To investigate the effect of parameter θ on the performance of MOEA/PIE, we keep all the other parameters unchanged and let θ vary from 0 to 20 with an increment of 5 at a time (i.e., θ = 0, 5, 10, 15, 20).

The IGD values obtained by the MOEA/PIE with different settings of θ on the test problems with different numbers of objectives are shown in Tables 8–11. As can be seen from these tables, different settings of θ make MOEA/PIE have distinct performances on problems with different numbers of objectives. Thus, we may easily have an intuitive sense that we can set θ to different values when handling MaOPs with different numbers of objectives. It is worth noting that no matter how the number of objectives changes, the MOEA/PIE with θ = 0 always performs worse in terms of the IGD metric. This phenomenon also demonstrates that when designing the fitness $g(\mathbf{y} \mid \mathbf{v_x})$, adding a term which reflects the diversity is necessary and conducive to improving the performance of the whole algorithm.

Table 9
Median of IGD on 10-objective problems. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	$\theta = 0.0$	$\theta = 5.0$	$\theta = 10.0$	$\theta = 15.0$	$\theta = 20.0$
DTLZ1	$1.105E-01$	$1.085E-01$	$1.092E-01$	$1.086E-01$	$1.092E-01$
DTLZ2	$4.097E-01$	$4.014E-01$	$4.006E-01$	$4.009E-01$	$4.022E-01$
DTLZ3	$4.077E-01$	$4.095E-01$	$4.111E-01$	$4.091E-01$	$4.113E-01$
DTLZ4	$4.196E-01$	$4.014E-01$	$4.011E-01$	$4.014E-01$	$4.019E-01$
ScaledDTLZ1	$1.430E+01$	$1.352E+01$	$1.493E+01$	$1.349E+01$	$1.414E+01$
ScaledDTLZ2	$8.428E+02$	$8.060E+02$	$7.722E+02$	$7.933E+02$	$7.890E+02$
WFG1	$7.253E+00$	$7.265E+00$	$7.250E+00$	$7.248E+00$	$7.241E+00$
WFG2	$5.186E-01$	$4.779E-01$	$5.024E-01$	$5.045E-01$	$4.979E-01$
WFG3	$5.463E+00$	$1.499E+00$	$1.567E+00$	$1.513E+00$	$1.538E+00$
WFG4	$4.684E+00$	$4.119E+00$	$4.070E+00$	$4.113E+00$	$4.079E+00$
WFG5	$4.192E+00$	$4.027E+00$	$4.030E+00$	$4.053E+00$	$4.038E+00$
WFG6	$4.233E+00$	$4.228E+00$	$4.220E+00$	$4.212E+00$	$4.246E+00$
WFG7	$4.377E+00$	$4.087E+00$	$4.105E+00$	$4.102E+00$	$4.066E+00$
WFG8	$5.186E+00$	$4.333E+00$	$4.390E+00$	$4.382E+00$	$4.403E+00$
WFG9	$3.968E+00$	$3.895E+00$	$3.931E+00$	$3.909E+00$	$3.927E+00$

Table 10
Median of IGD on 15-objective problems. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	$\theta = 0.0$	$\theta = 5.0$	$\theta = 10.0$	$\theta = 15.0$	$\theta = 20.0$
DTLZ1	$1.747E-01$	$1.492E-01$	$1.490E-01$	$1.490E-01$	$1.490E-01$
DTLZ2	$7.046E-01$	$5.870E-01$	$5.884E-01$	$5.891E-01$	$5.856E-01$
DTLZ3	$6.564E-01$	$5.944E-01$	$5.975E-01$	$5.962E-01$	$5.976E-01$
DTLZ4	$5.957E-01$	$5.863E-01$	$5.869E-01$	$5.871E-01$	$5.874E-01$
ScaledDTLZ1	$1.041E+00$	$7.772E-01$	$7.816E-01$	$7.580E-01$	$7.807E-01$
ScaledDTLZ2	$1.201E+03$	$1.295E+03$	$1.280E+03$	$1.236E+03$	$1.245E+03$
WFG1	$9.655E+00$	$9.572E+00$	$9.697E+00$	$9.678E+00$	$9.662E+00$
WFG2	$9.886E-01$	$8.588E-01$	$8.808E-01$	$8.603E-01$	$9.594E-01$
WFG3	$9.235E+00$	$2.657E+00$	$2.682E+00$	$2.637E+00$	$2.720E+00$
WFG4	$1.165E+01$	$8.873E+00$	$8.841E+00$	$8.960E+00$	$8.911E+00$
WFG5	$1.092E+01$	$9.111E+00$	$9.061E+00$	$9.059E+00$	$8.968E+00$
WFG6	$9.697E+00$	$9.742E+00$	$1.005E+01$	$9.720E+00$	$1.003E+01$
WFG7	$1.129E+01$	$8.677E+00$	$8.518E+00$	$8.580E+00$	$8.568E+00$
WFG8	$1.183E+01$	$9.521E+00$	$9.534E+00$	$9.611E+00$	$9.532E+00$
WFG9	$9.497E+00$	$8.393E+00$	$8.467E+00$	$8.389E+00$	$8.516E+00$

Table 11
Median of IGD on 20-objective problems. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	$\theta = 0.0$	$\theta = 5.0$	$\theta = 10.0$	$\theta = 15.0$	$\theta = 20.0$
DTLZ1	$1.797E-01$	$1.535E-01$	$1.536E-01$	$1.536E-01$	$1.536E-01$
DTLZ2	$7.577E-01$	$6.054E-01$	$6.096E-01$	$6.071E-01$	$6.091E-01$
DTLZ3	$7.890E-01$	$6.294E-01$	$6.275E-01$	$6.190E-01$	$6.192E-01$
DTLZ4	$6.001E-01$	$6.098E-01$	$6.088E-01$	$6.097E-01$	$6.098E-01$
ScaledDTLZ1	$7.681E-01$	$5.033E-01$	$4.923E-01$	$5.030E-01$	$5.000E-01$
ScaledDTLZ2	$2.402E+00$	$2.341E+00$	$2.356E+00$	$2.348E+00$	$2.387E+00$
WFG1	$1.850E+01$	$1.840E+01$	$1.842E+01$	$1.870E+01$	$1.857E+01$
WFG2	$1.005E+00$	$9.558E-01$	$9.587E-01$	$9.338E-01$	$8.967E-01$
WFG3	$1.167E+01$	$3.531E+00$	$3.456E+00$	$3.435E+00$	$3.442E+00$
WFG4	$1.712E+01$	$1.193E+01$	$1.190E+01$	$1.187E+01$	$1.199E+01$
WFG5	$1.600E+01$	$1.135E+01$	$1.131E+01$	$1.128E+01$	$1.119E+01$
WFG6	$1.350E+01$	$1.209E+01$	$1.226E+01$	$1.247E+01$	$1.239E+01$
WFG7	$1.589E+01$	$1.129E+01$	$1.131E+01$	$1.137E+01$	$1.134E+01$
WFG8	$1.689E+01$	$1.246E+01$	$1.234E+01$	$1.251E+01$	$1.233E+01$
WFG9	$1.391E+01$	$1.194E+01$	$1.188E+01$	$1.186E+01$	$1.177E+01$

Table 12
Average rankings of MOEA/PIE with different settings of θ on test problems with different numbers of objectives according to the IGD metric.

m	$\theta = 0.0$	$\theta = 5.0$	$\theta = 10.0$	$\theta = 15.0$	$\theta = 20.0$
5	3.7777777777777763	**2.3888888888888893**	2.8888888888888884	2.8333333333333335	3.111111111111111
10	4.499999999999999	**2.2222222222222223**	3.0	2.5	2.7777777777777772
15	4.111111111111111	2.5	3.0555555555555545	**2.3888888888888884**	2.944444444444445
20	4.722222222222221	2.4999999999999996	2.5	3.0	**2.2777777777777777**

Table 13
Median and IQR of IGD on 5-objective problems. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	$K = 3$	$K = 5$	$K = 7$	$K = 9$
DTLZ1	5.937E − 02 (1.2E − 03)	5.885E − 02 (1.2E − 03)	6.009E − 02 (2.3E − 03)	6.108E − 02 (3.5E − 03)
DTLZ2	1.665E − 01 (3.7E − 03)	1.664E − 01 (2.8E − 03)	1.682E − 01 (2.2E − 03)	1.693E − 01 (2.5E − 03)
DTLZ3	1.707E − 01 (4.2E − 03)	1.702E − 01 (2.3E − 03)	1.701E − 01 (4.2E − 03)	1.701E − 01 (3.7E − 03)
DTLZ4	1.666E − 01 (2.2E − 03)	1.677E − 01 (1.1E − 03)	1.687E − 01 (2.7E − 03)	1.697E − 01 (2.1E − 03)
ScaledDTLZ1	5.989E + 01 (2.4E + 01)	5.697E + 01 (4.0E + 01)	7.752E + 01 (3.5E + 01)	8.296E + 01 (1.6E + 01)
ScaledDTLZ2	1.725E + 02 (5.5E + 01)	1.700E + 02 (3.0E + 01)	1.626E + 02 (2.4E + 01)	1.503E + 02 (2.4E + 01)
WFG1	3.332E + 00 (4.9E − 02)	3.317E + 00 (4.7E − 02)	3.304E + 00 (4.0E − 02)	3.382E + 00 (1.2E − 01)
WFG2	2.541E − 01 (1.5E − 02)	2.620E − 01 (3.1E − 02)	2.572E − 01 (2.8E − 02)	2.633E − 01 (3.6E − 02)
WFG3	3.416E − 01 (1.6E − 02)	3.272E − 01 (6.1E − 03)	3.398E − 01 (1.8E − 02)	3.630E − 01 (3.5E − 02)
WFG4	9.521E − 01 (1.5E − 02)	9.548E − 01 (1.4E − 02)	9.639E − 01 (2.7E − 02)	9.800E − 01 (3.1E − 02)
WFG5	9.680E − 01 (1.8E − 02)	9.749E − 01 (1.8E − 02)	9.768E − 01 (1.9E − 02)	9.816E − 01 (1.6E − 02)
WFG6	9.634E − 01 (1.5E − 02)	9.650E − 01 (1.2E − 02)	9.692E − 01 (2.0E − 02)	9.617E − 01 (1.1E − 02)
WFG7	9.634E − 01 (2.4E − 02)	9.798E − 01 (2.2E − 02)	9.784E − 01 (1.9E − 02)	9.775E − 01 (2.6E − 02)
WFG8	1.045E + 00 (8.8E − 03)	1.050E + 00 (1.5E − 02)	1.060E + 00 (2.4E − 02)	1.060E + 00 (1.4E − 02)
WFG9	1.001E + 00 (1.6E − 02)	1.007E + 00 (2.5E − 02)	1.024E + 00 (4.1E − 02)	1.039E + 00 (3.1E − 02)

Table 14
Median and IQR of IGD on 10-objective problems. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	$K = 3$	$K = 5$	$K = 7$	$K = 9$
DTLZ1	1.090E − 01 (1.2E − 03)	1.102E − 01 (7.1E − 04)	1.113E − 01 (9.0E − 04)	1.133E − 01 (1.7E − 03)
DTLZ2	4.024E − 01 (2.5E − 03)	4.101E − 01 (4.8E − 03)	4.170E − 01 (1.4E − 02)	4.151E − 01 (7.3E − 03)
DTLZ3	4.083E − 01 (6.5E − 03)	4.076E − 01 (5.5E − 03)	4.111E − 01 (6.8E − 03)	4.165E − 01 (2.1E − 02)
DTLZ4	4.016E − 01 (3.2E − 03)	4.030E − 01 (3.5E − 03)	4.043E − 01 (3.5E − 03)	4.042E − 01 (4.8E − 03)
ScaledDTLZ1	1.334E + 01 (2.3E + 00)	1.191E + 01 (3.9E + 00)	1.145E + 01 (2.2E + 00)	1.140E + 01 (2.1E + 00)
ScaledDTLZ2	8.175E + 02 (1.0E + 02)	8.244E + 02 (1.2E + 02)	7.862E + 02 (5.7E + 01)	6.953E + 02 (1.3E + 02)
WFG1	7.246E + 00 (5.0E − 02)	7.270E + 00 (5.5E − 02)	7.445E + 00 (2.8E − 01)	7.801E + 00 (6.4E − 01)
WFG2	5.011E − 01 (4.6E − 02)	4.694E − 01 (4.6E − 02)	4.876E − 01 (6.5E − 02)	5.250E − 01 (8.7E − 02)
WFG3	1.473E + 00 (1.5E − 01)	1.331E + 00 (4.3E − 02)	1.279E + 00 (5.6E − 02)	1.484E + 00 (1.9E − 01)
WFG4	4.097E + 00 (1.0E − 01)	4.328E + 00 (7.9E − 02)	4.639E + 00 (2.7E − 01)	4.420E + 00 (2.5E − 01)
WFG5	4.033E + 00 (6.7E − 02)	4.120E + 00 (1.5E − 01)	4.318E + 00 (1.9E − 01)	4.544E + 00 (3.7E − 01)
WFG6	4.182E + 00 (1.1E − 01)	4.400E + 00 (1.7E − 01)	4.297E + 00 (1.8E − 01)	4.212E + 00 (6.9E − 02)
WFG7	4.096E + 00 (5.3E − 02)	4.197E + 00 (8.0E − 02)	4.258E + 00 (6.8E − 02)	4.041E + 00 (1.8E − 01)
WFG8	4.360E + 00 (9.6E − 02)	4.628E + 00 (3.5E − 01)	4.756E + 00 (3.1E − 01)	4.763E + 00 (4.2E − 01)
WFG9	3.911E + 00 (4.6E − 02)	4.019E + 00 (7.9E − 02)	4.098E + 00 (1.3E − 01)	4.250E + 00 (1.9E − 01)

Table 15
Median and IQR of IGD on 15-objective problems. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	$K = 3$	$K = 5$	$K = 7$	$K = 9$
DTLZ1	1.491E − 01 (1.9E − 03)	1.518E − 01 (4.5E − 03)	1.592E − 01 (6.1E − 03)	1.645E − 01 (7.1E − 03)
DTLZ2	5.895E − 01 (6.8E − 03)	6.055E − 01 (1.3E − 02)	5.913E − 01 (1.3E − 02)	5.980E − 01 (2.5E − 02)
DTLZ3	5.959E − 01 (1.7E − 02)	5.974E − 01 (6.6E − 03)	6.076E − 01 (9.3E − 03)	6.110E − 01 (1.4E − 02)
DTLZ4	5.852E − 01 (2.6E − 03)	5.928E − 01 (5.8E − 03)	5.981E − 01 (7.0E − 03)	6.017E − 01 (9.4E − 03)
ScaledDTLZ1	7.697E − 01 (4.0E − 02)	7.697E − 01 (4.5E − 02)	7.827E − 01 (3.7E − 02)	7.904E − 01 (3.4E − 02)
ScaledDTLZ2	1.226E + 03 (1.3E + 02)	1.196E + 03 (1.3E + 02)	1.194E + 03 (1.7E + 02)	1.092E + 03 (1.3E + 02)
WFG1	9.725E + 00 (2.8E − 01)	9.659E + 00 (3.0E − 01)	9.762E + 00 (4.2E − 01)	1.017E + 01 (6.5E − 01)
WFG2	1.008E + 00 (3.2E + 00)	1.000E + 00 (2.2E − 01)	9.236E − 01 (2.6E − 01)	9.030E − 01 (1.5E − 01)
WFG3	2.717E + 00 (1.0E + 00)	2.261E + 00 (6.1E − 02)	2.266E + 00 (8.9E − 02)	2.554E + 00 (1.1E − 01)
WFG4	8.947E + 00 (4.4E − 01)	9.023E + 00 (3.8E − 01)	9.575E + 00 (3.8E − 01)	1.022E + 01 (3.4E − 01)
WFG5	8.978E + 00 (2.2E − 01)	9.226E + 00 (2.4E − 01)	9.579E + 00 (1.7E − 01)	9.161E + 00 (5.5E − 01)
WFG6	9.830E + 00 (4.1E − 01)	1.029E + 01 (2.6E − 01)	9.752E + 00 (7.0E − 01)	9.216E + 00 (3.8E − 01)
WFG7	8.533E + 00 (2.0E − 01)	8.704E + 00 (1.5E − 01)	9.144E + 00 (1.5E − 01)	9.406E + 00 (4.7E − 01)
WFG8	9.461E + 00 (6.9E − 01)	9.783E + 00 (7.0E − 01)	9.299E + 00 (1.2E − 01)	9.552E + 00 (3.1E − 01)
WFG9	8.461E + 00 (4.4E − 01)	8.914E + 00 (2.7E − 01)	9.129E + 00 (2.2E − 01)	9.286E + 00 (5.1E − 01)

Table 16
Median and IQR of IGD on 20-objective problems. The best performance metrics are highlighted with bold font and grey shade. The second best ones are highlighted with light grey shade.

Problem	$K = 3$	$K = 5$	$K = 7$	$K = 9$
DTLZ1	1.539E − 01 (2.7E − 03)	1.591E − 01 (2.2E − 03)	1.666E − 01 (4.6E − 03)	1.766E − 01 (4.4E − 03)
DTLZ2	6.061E − 01 (1.2E − 02)	6.190E − 01 (1.2E − 02)	6.251E − 01 (9.5E − 03)	6.526E − 01 (2.1E − 02)
DTLZ3	6.214E − 01 (3.1E − 02)	6.186E − 01 (1.8E − 02)	6.342E − 01 (1.4E − 02)	6.349E − 01 (2.1E − 02)
DTLZ4	6.094E − 01 (3.3E − 03)	6.150E − 01 (5.1E − 03)	6.185E − 01 (1.3E − 02)	6.287E − 01 (1.4E − 02)
ScaledDTLZ1	6.802E − 01 (6.7E − 02)	5.660E − 01 (1.5E − 02)	5.534E − 01 (1.1E − 02)	6.340E − 01 (8.0E − 02)
ScaledDTLZ2	1.850E + 00 (1.2E − 01)	1.920E + 00 (2.3E − 01)	1.861E + 00 (3.0E − 01)	1.739E + 00 (1.0E − 01)
WFG1	1.806E + 01 (1.2E + 00)	1.901E + 01 (2.3E + 00)	1.835E + 01 (2.2E + 00)	1.818E + 01 (1.0E + 00)
WFG2	1.026E + 00 (1.6E − 01)	8.962E − 01 (1.6E − 01)	8.976E − 01 (1.3E − 01)	9.145E − 01 (9.3E − 02)
WFG3	3.283E + 00 (3.8E − 01)	2.914E + 00 (7.8E − 02)	3.078E + 00 (1.1E − 01)	3.276E + 00 (2.4E − 01)
WFG4	1.194E + 01 (3.8E − 01)	1.303E + 01 (3.7E − 01)	1.447E + 01 (6.1E − 01)	1.555E + 01 (6.2E − 01)
WFG5	1.133E + 01 (2.0E − 01)	1.198E + 01 (7.6E − 01)	1.251E + 01 (1.0E + 00)	1.333E + 01 (4.7E − 01)
WFG6	1.230E + 01 (4.4E − 01)	1.262E + 01 (9.2E − 01)	1.146E + 01 (8.5E − 01)	1.196E + 01 (8.1E − 01)
WFG7	1.136E + 01 (1.6E − 01)	1.207E + 01 (3.0E − 01)	1.315E + 01 (5.1E − 01)	1.408E + 01 (4.4E − 01)
WFG8	1.246E + 01 (6.1E − 01)	1.245E + 01 (5.9E − 01)	1.318E + 01 (3.6E − 01)	1.419E + 01 (7.6E − 01)
WFG9	1.179E + 01 (2.7E − 01)	1.248E + 01 (3.9E − 01)	1.304E + 01 (6.7E − 01)	1.370E + 01 (9.4E − 01)

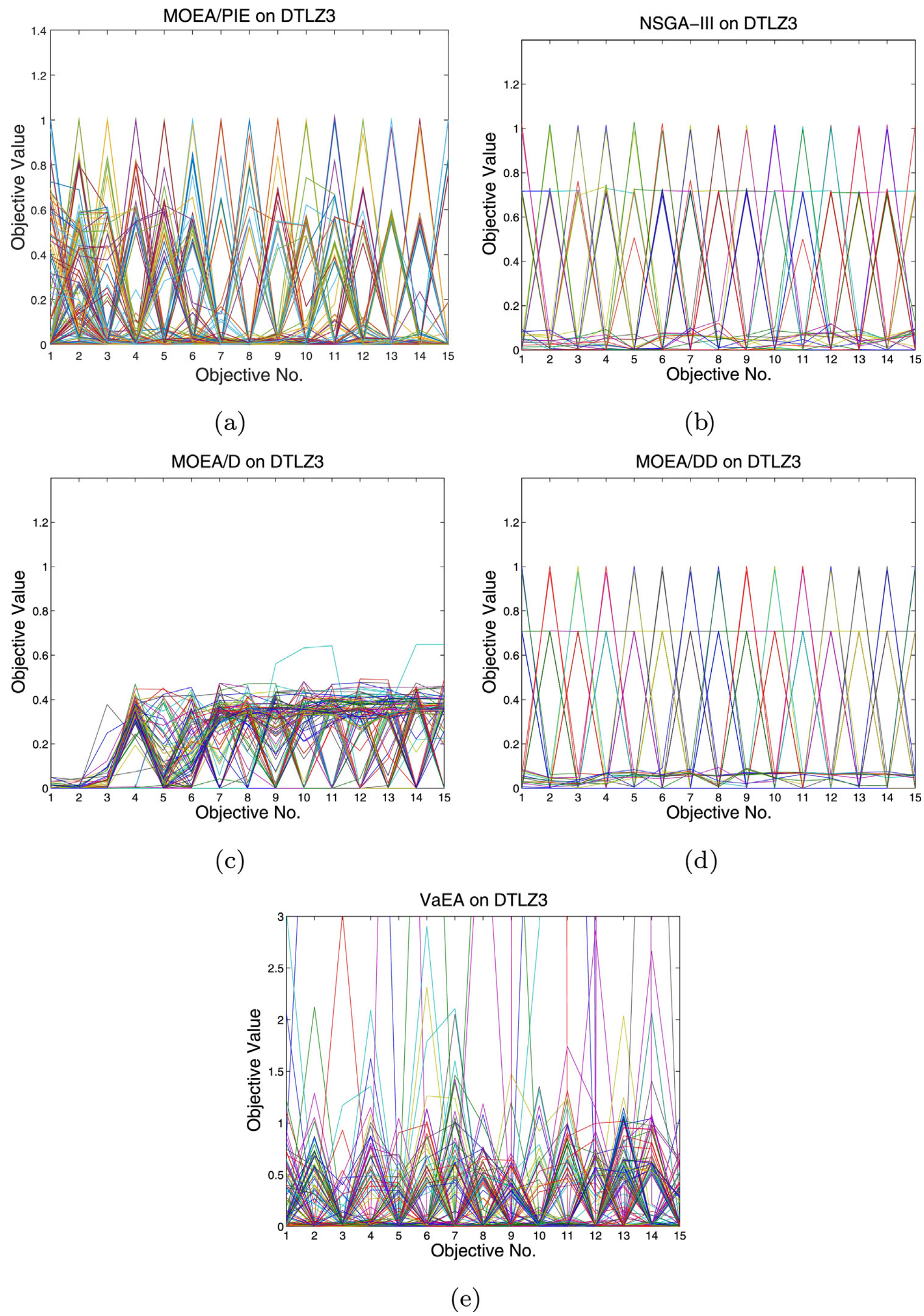

Fig. 5. Parallel coordinates of final solutions obtained by five algorithms on 15-objective DTLZ3.

To further explore the relation between the value of θ and the number of objectives, we shall make a summing up of the results in Tables 8–11. Concretely, for a kind of test problems with a certain number of objectives, we adopt the Friedman test to obtain average rankings of the MOEA/PIE with different settings of θ. As shown in Table 12, we can know that $\theta = 5$ obtains the best ranking when $m = 5$ and $m = 10$. When $m = 15$, the MOEA/PIE with $\theta = 15$ obtains the best ranking. And $\theta = 20$ obtains the best ranking when $m = 20$. Thus, for simplicity, we can set θ to the number of objectives (that is, $\theta = m$) in practice.

6.2. *On the effect of parameter* K

To investigate the effect of parameter K on the performance of MOEA/PIE, we keep all the other parameters unchanged and let K vary from 3 to 9 with an increment of 2 at a time (i.e., $K = 3, 5, 7, 9$).

The IGD values obtained by the MOEA/PIE with different settings of K on the test problems with different numbers of objectives are shown in Tables 13–16. As can be seen from these tables, $K = 3$ and $K = 5$ perform better than other values of K. Thus, when using the proposed MOEA/PIE algorithm in practice, it is recommended that the value of K is set to 3 (the default value) or 5.

Table 17
Median of Δ on 15-objective and 20-objective problems. The best performance metrics are highlighted with grey shade.

Problem	$m = 15$		$m = 20$	
	MOEA/PIE	MOEA/PIE-ideal	MOEA/PIE	MOEA/PIE-ideal
DTLZ1	4.085E − 01	4.143E − 01	3.814E − 01	3.870E − 01
DTLZ2	2.544E − 01	2.792E − 01	1.890E − 01	1.969E − 01
DTLZ3	2.765E − 01	2.605E − 01	2.085E − 01	2.037E − 01
DTLZ4	2.166E − 01	2.188E − 01	1.300E − 01	1.338E − 01
ScaledDTLZ1	3.799E − 01	4.338E − 01	4.103E − 01	5.265E − 01
ScaledDTLZ2	2.808E − 01	2.741E − 01	1.841E − 01	1.805E − 01
WFG1	9.310E − 01	9.324E − 01	9.768E − 01	9.753E − 01
WFG2	8.216E − 01	5.759E − 01	5.023E − 01	5.425E − 01
WFG3	3.619E − 01	3.300E − 01	2.813E − 01	2.422E − 01
WFG4	2.193E − 01	2.195E − 01	1.499E − 01	1.637E − 01
WFG5	2.367E − 01	2.080E − 01	1.856E − 01	1.673E − 01
WFG6	2.269E − 01	2.342E − 01	1.663E − 01	1.829E − 01
WFG7	2.400E − 01	2.066E − 01	1.286E − 01	1.581E − 01
WFG8	4.776E − 01	4.538E − 01	2.320E − 01	4.087E − 01
WFG9	1.861E − 01	1.824E − 01	9.359E − 02	9.544E − 02

6.3. On the choice of basic evolution direction

Recall from the description of the proposed MOEA/PIE in Section 3, we use the *nadir point* $\mathbf{z}^{max}$ to define the *basic evolution direction* for each individual $\mathbf{x}$. However, does this manner of constructing *basic evolution direction* really contribute to the diversity promoting of the algorithm? Aim to answer this question, we carry out an extra experiment to demonstrate the effect of using the *nadir point* to define the *basic evolution direction*. First of all, we define a variant of MOEA/PIE (named MOEA/PIE-ideal) which uses the *ideal point* $\mathbf{z}^{min}$ to define the *basic evolution direction* for each individual. That is, in MOEA/PIE-ideal, the *basic evolution direction* for each individual $\mathbf{x}$ in P is defined as:

$$\mathbf{v}_{\mathbf{x}} = \mathbf{z}^{\min} - \bar{\mathrm{F}}(\mathbf{x}) \tag{9}$$

In this experiment, we choose problems with 15 and 20 objectives to compare the performance of MOEA/PIE and MOEA/PIE-ideal. The median values of Δ obtained by MOEA/PIE and MOEA/PIE-ideal are shown in Table 17. From Table 17, we can observe that MOEA/PIE and MOEA/PIE-ideal have similar performance on 15-objective problems. As for 20-objective problems, the median values of Δ obtained by MOEA/PIE are better than the median values obtained by MOEA/PIE-ideal on most of the test instances. To conclude, the diversity of the solution sets obtained by MOEA/PIE are better than that of MOEA/PIE-ideal on most problems. Thus, compared with using the *ideal point*, the manner of using the *nadir point* to construct *basic evolution direction* can indeed promote the diversity, especially when handling these problems with larger number of objectives.

7. Conclusions

In this paper, we have proposed a new many-objective evolutionary algorithm (namely, MOEA/PIE). It uses a *projection-assisted intra-family election* strategy to make the population evolve to the true Pareto front and guarantee the convergence, and adopts an angle-based strategy to maintain the diversity.

We have empirically verified the performance of the proposed MOEA/PIE algorithm through a plenty of different experiments on DTLZ and WFG test problems with up to 20 objectives. By empirical comparisons of MOEA/PIE and other state-of-the-art algorithms, we have demonstrated that MOEA/PIE significantly outperforms NSGA-III, MOEA/D and MOEA/DD on the vast majority of the test instances, and MOEA/PIE is better than VaEA in terms of the diversity of the solutions. In particular, MOEA/PIE shows obvious superiority and can obtain a good distribution of solutions, especially when handling the problems with larger number of objectives.

As mentioned above, the proposed MOEA/PIE algorithm has been shown to have superior performance when handling with unconstrained many-objective optimization problems. Therefore, in the future research, it is necessary to investigate how to apply this algorithm to the practical problems and constrained many-objective optimization problems. Besides, further investigation on the parameter influence should be carried out.

References

[1] S. Mardle, Nonlinear Multiobjective Optimization by K.M. Miettinen, Birkhaser Verlag, 2001.
[2] P.L. Yu, Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives, J. Optim. Theory Appl. 14 (3) (1974) 319–377.
[3] J.H. Holland, Adaptation in Natural and Artificial Systems, MIT Press, 2015.
[4] D.E. Golberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, 1989.
[5] N. Srinivas, K. Deb, Multiobjective function optimization using nondominated sorting genetic algorithms, Evolut. Comput. 2 (3) (1995) 221–248.
[6] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings IEEE International Conference on Neural Networks, 1995, vol. 4, 1995, pp. 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968.
[7] C.A.C. Coello, M.S. Lechuga, MOPSO: a proposal for multiple objective particle swarm optimization, in: Proceedings of the 2002 Congress on Evolutionary Computation, 2002, CEC '02, vol. 2, 2002, pp. 1051–1056, http://dx.doi.org/10.1109/CEC.2002.1004388.
[8] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.
[9] K.F. Doerner, W.J. Gutjahr, R.F. Hartl, C. Strauss, C. Stummer, Pareto ant colony optimization with ILP preprocessing in multiobjective project portfolio selection, Eur. J. Oper. Res. 171 (3) (2006) 830–841.
[10] R. Storn, K. Price, Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[11] T. Robič, B. Filipič, DEMO differential evolution for multiobjective optimization Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, vol. 3410, Springer Berlin Heidelberg, 2005, pp. 520–533, http://dx.doi.org/10.1007/978-3-540-31880-4_36.
[12] D. Karaboga, An idea based on honey bee swarm for numerical optimization, Technical report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
[13] Y. Xiang, Y. Zhou, H. Liu, An elitism based multi-objective artificial bee colony algorithm, Eur. J. Oper. Res. 245 (1) (2015) 168–193.
[14] T. Bäck, Evolutionary Algorithms in Theory and Practice, Oxford University Press, 1996.
[15] Z. Michalewicz, Genetic Algorithms + Data Structures = Evolution Programs, Springer Science & Business Media, 1996.

[16] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolut. Comput. 6 (2) (2002) 182–197, http://dx.doi.org/10.1109/4235.996017.
[17] Q. Zhang, H. Li, MOEA/D: a multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evolut. Comput. 11 (6) (2007) 712–731, http://dx.doi.org/10.1109/TEVC.2007.892759.
[18] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: improving the strength pareto evolutionary algorithm for multiobjective optimization, Fifth Conference on Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems (2001) 95–100.
[19] R.M. Hierons, M. Li, X. Liu, S. Segura, W. Zheng, SIP:optimal product selection from feature models using many-objective evolutionary optimization, ACM Trans. Softw. Eng. Methodol. 25 (2) (2016) 1–39.
[20] J.R. Kasprzyk, P.M. Reed, G.W. Characklis, B.R. Kirsch, Many-objective de novo water supply portfolio planning under deep uncertainty, Environ. Model. Softw. 34 (6) (2012) 87–104.
[21] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints, IEEE Trans. Evolut. Comput. 18 (4) (2014) 577–601, http://dx.doi.org/10.1109/TEVC.2013.2281535.
[22] J. Bader, E. Zitzler, HypE: an algorithm for fast hypervolume-based many-objective optimization, Evolut. Comput. 19 (1) (2011) 45–76.
[23] S. Yang, M. Li, X. Liu, J. Zheng, A grid-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evolut. Comput. 17 (5) (2013) 721–736, http://dx.doi.org/10.1109/TEVC.2012.2227145.
[24] M. Li, S. Yang, X. Liu, Bi-goal evolution for many-objective optimization problems, Artif. Intell. 228 (C) (2015) 45–65.
[25] H. Wang, L. Jiao, X. Yao, Two-Arch2: an improved two-archive algorithm for many-objective optimization, IEEE Trans. Evolut. Comput. 19 (4) (2015) 524–541.
[26] C.M. Fonseca, P.J. Fleming, Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation, IEEE Trans. Syst., Man, Cybern. – Part A: Syst. Hum. 28 (1) (1998) 26–37, http://dx.doi.org/10.1109/3468.650319.
[27] R.C. Purshouse, P.J. Fleming, On the evolutionary optimization of many conflicting objectives, IEEE Trans. Evolut. Comput. 11 (6) (2007) 770–784, http://dx.doi.org/10.1109/TEVC.2007.910138.
[28] J. Knowles, D. Corne, Quantifying the Effects of Objective Space Dimension in Evolutionary Multiobjective Optimization, Springer Berlin Heidelberg, 2007.
[29] H. Ishibuchi, N. Akedo, Y. Nojima, Behavior of multiobjective evolutionary algorithms on many-objective knapsack problems, IEEE Trans. Evolut. Comput. 19 (2) (2015) 264–283.
[30] Y. Jin, B. Sendhoff, Connectedness, regularity and the success of local search in evolutionary multi-objective optimization, in: The 2003 Congress on Evolutionary Computation, 2003, CEC '03, vol. 3, 2003, pp. 1910–1917, http://dx.doi.org/10.1109/CEC.2003.1299907.
[31] D.J. Walker, R. Everson, J.E. Fieldsend, Visualizing mutually nondominating solution sets in many-objective optimization, IEEE Trans. Evolut. Comput. 17 (2) (2013) 165–184.
[32] M. Li, S. Yang, X. Liu, R. Shen, A Comparative Study on Evolutionary Algorithms for Many-Objective Optimization, Springer Berlin Heidelberg, 2013.
[33] B. Li, J. Li, K. Tang, X. Yao, Many-objective evolutionary algorithms: a survey, ACM Comput. Surv. 48 (1) (2015) 1–35.
[34] S.F. Adra, P.J. Fleming, Diversity management in evolutionary many-objective optimization, IEEE Trans. Evolut. Comput. 15 (2) (2011) 183–195.
[35] M. Li, S. Yang, X. Liu, Shift-based density estimation for pareto-based algorithms in many-objective optimization, IEEE Trans. Evolut. Comput. 18 (3) (2014) 348–365.
[36] D. Hadka, P. Reed, Borg: An auto-adaptive many-objective evolutionary computing framework, Evolut. Comput. 21 (2) (2013) 231–259.
[37] Z. He, G.G. Yen, J. Zhang, Fuzzy-based pareto optimality for many-objective evolutionary algorithms, IEEE Trans. Evolut. Comput. 18 (2) (2014) 269–285.
[38] C. Zhu, L. Xu, E.D. Goodman, Generalization of pareto-optimality for many-objective evolutionary optimization, IEEE Trans. Evolut. Comput. 20 (2) (2016) 299–315.
[39] M.T. Jensen, Reducing the run-time complexity of multiobjective EAs: the NSGA-II and other algorithms, IEEE Trans. Evolut. Comput. 7 (5) (2003) 503–515.
[40] F.-A. Fortin, S. Grenier, M. Parizeau, Generalizing the improved run-time complexity algorithm for non-dominated sorting, in: Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation, ACM, 2013, pp. 615–622.
[41] K. McClymont, E. Keedwell, Deductive sort and climbing sort: new methods for non-dominated sorting, Evolut. Comput. 20 (1) (2012) 1–26, http://dx.doi.org/10.1162/EVCO_a_00041.
[42] X. Zhang, Y. Tian, R. Cheng, Y. Jin, An efficient approach to nondominated sorting for evolutionary multiobjective optimization, IEEE Trans. Evolut. Comput. 19 (2) (2015) 201–213.
[43] M. Drozdik, Y. Akimoto, H. Aguirre, K. Tanaka, Computational cost reduction of nondominated sorting using the M-front, IEEE Trans. Evolut. Comput. 19 (5) (2015) 659–678, http://dx.doi.org/10.1109/TEVC.2014.2366498.
[44] Y. Zhou, Z. Chen, J. Zhang, Ranking vectors by means of the dominance degree matrix, IEEE Trans. Evolut. Comput. 21 (1) (2017) 34–51, http://dx.doi.org/10.1109/TEVC.2016.2567648.
[45] M. Emmerich, N. Beume, B. Naujoks, An EMO algorithm using the hypervolume measure as selection criterion, in: International Conference on Evolutionary Multi-Criterion Optimization, 2005, pp. 62–76.
[46] R.H. Gómez, C.A.C. Coello, MOMBI: A new metaheuristic for many-objective optimization based on the R2 indicator, in: 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 2488–2495, http://dx.doi.org/10.1109/CEC.2013.6557868.
[47] R. Hernández Gómez, C.A. Coello Coello, Improved metaheuristic based on the R2 indicator for many-objective optimization, in: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, GECCO '15, ACM, New York, NY, USA, 2015, pp. 679–686, http://dx.doi.org/10.1145/2739480.2754776.
[48] E. Zitzler, S. Künzli, Indicator-based selection in multiobjective search, Lect. Notes Comput. Sci. 3242 (2004) 832–842.
[49] N. Beume, C.M. Fonseca, M. López-Ibáñez, L. Paquete, J. Vahrenhold, On the complexity of computing the hypervolume indicator, IEEE Trans. Evolut. Comput. 13 (5) (2009) 1075–1082.
[50] L. While, L. Bradstreet, L. Barone, A fast way of calculating exact hypervolumes, IEEE Trans. Evolut. Comput. 16 (1) (2012) 86–95.
[51] H. Ishibuchi, Y. Sakane, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization by NSGA-II and MOEA/D with large populations, in: IEEE International Conference on Systems, Man and Cybernetics, 2009, pp. 1758–1763.
[52] A. Trivedi, D. Srinivasan, K. Sanyal, A. Ghosh, A survey of multiobjective evolutionary algorithms based on decomposition, IEEE Trans. Evolut. Comput 21 (3) (2017) 440–462, http://dx.doi.org/10.1109/TEVC.2016.2608507.
[53] H.L. Liu, F. Gu, Q. Zhang, Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems, IEEE Trans. Evolut. Comput. 18 (3) (2014) 450–455.
[54] H.L. Liu, L. Chen, Q. Zhang, K. Deb, An evolutionary many-objective optimisation algorithm with adaptive region decomposition, in: 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 4763–4769, http://dx.doi.org/10.1109/CEC.2016.7744399.
[55] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evolut. Comput. 19 (5) (2015) 694–716, http://dx.doi.org/10.1109/TEVC.2014.2373386.
[56] R. Cheng, Y. Jin, M. Olhofer, B. Sendhoff, A reference vector guided evolutionary algorithm for many-objective optimization, IEEE Trans. Evolut. Comput. 20 (5) (2016) 773–791.
[57] I. Das, J.E. Dennis, Normal-boundary intersection: a new method for generating the pareto surface in nonlinear multicriteria optimization problems, SIAM J. Optim. 8 (3) (2000) 631–657.
[58] H. Ishibuchi, Y. Setoguchi, H. Masuda, Y. Nojima, Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes, IEEE Trans. Evolut. Comput. 21 (2) (2017) 169–190, http://dx.doi.org/10.1109/TEVC.2016.2587749.
[59] H. Ishibuchi, K. Narukawa, An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization, Eur. J. Oper. Res. 188 (1) (2008) 57–75.
[60] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable multi-objective optimization test problems, in: Proceedings of the 2002 Congress on Evolutionary Computation, 2002, CEC '02, vol. 1, 2002, pp. 825–830, http://dx.doi.org/10.1109/CEC.2002.1007032.
[61] S. Huband, L. Barone, L. While, P. Hingston, A scalable multi-objective test problem toolkit, Lect. Notes Comput. Sci. 3410 (2005) 280–295.
[62] Y. Yuan, H. Xu, B. Wang, X. Yao, A new dominance relation-based evolutionary algorithm for many-objective optimization, IEEE Trans. Evolut. Comput. 20 (1) (2016) 16–37, http://dx.doi.org/10.1109/TEVC.2015.2420112.
[63] Q. Zhang, A. Zhou, S. Zhao, P.N. Suganthan, W. Liu, S. Tiwari, Multiobjective Optimization Test Instances for the CEC 2009 Special Session and Competition, University of Essex, 2008.
[64] D.A. Van, V. Gary, B. Lamont, Multiobjective evolutionary algorithm research: a history and analysis, Evolut. Comput. 8 (2) (1999) 125–147.
[65] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion, in: IEEE Congress on Evolutionary Computation, 2006, CEC 2006, 2006, pp. 3234–3241.
[66] Y. Xiang, Y. Zhou, M. Li, Z. Chen, A vector angle based evolutionary algorithm for unconstrained many-objective optimization, IEEE Trans. Evolut. Comput. PP (99) (2016), 1–1.
[67] J.J. Durillo, A.J. Nebro, jMetal: a Java framework for multi-objective optimization, Adv. Eng. Softw. 42 (2011) 760–771.
[68] M.G. Kalyanmoy Deb, A combined genetic adaptive search (geneas) for engineering design, Comput. Sci. Inf. 26 (1999) 30–45.
[69] K. Deb, R. Agrawal, Simulated binary crossover for continuous search space, Complex Syst. 9 (1995) 115–148.
[70] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics 1 (6) (1945).
[71] M. Li, L. Zhen, X. Yao, How to Read Many-Objective Solution Sets in Parallel Coordinates, 2017.