

# Enhanced $\theta$ dominance and density selection based evolutionary algorithm for many-objective optimization problems

Chong Zhou<sup>1,2</sup> · Guangming Dai<sup>1,2</sup>  · Maocai Wang<sup>1,2</sup>

Published online: 7 August 2017  
© Springer Science+Business Media New York 2017

**Abstract** Many multi-objective evolutionary algorithms (MOEAs) have been developed for many-objective optimization. This paper proposes a new enhanced  $\theta$  dominance and density selection based evolutionary algorithm (called  $\theta$ -EDEA) for many-objective optimization problems. We firstly construct an  $m$ -dimension hyper-plane using the extreme point on the each dimension. Then we replace the distance between the origin point and projection of solution on the reference line of  $\theta$  dominance which recently is proposed in  $\theta$  dominance based evolutionary algorithm ( $\theta$ -DEA), with the perpendicular distance between each solution and the hyper-plane to develop an enhanced  $\theta$  dominance. Finally, in order to maintain better diversity,  $\theta$ -EDEA employs density based selection mechanism to select the solution for the next population in the environment selection phase.  $\theta$ -EDEA still inherits clustering operator and ranking operator of  $\theta$ -DEA to balance diversity and convergence. The performance of  $\theta$ -EDEA is validated and compared with five state-of-the-art algorithms on two

well-known many-objective benchmark problems with three to fifteen objectives. The results show that  $\theta$ -EDEA is capable of obtaining a solution set with better convergence and diversity.

**Keywords** Many-Objective Optimization · Multi-objective evolutionary algorithm ·  $\theta$  dominance ·  $m$ -dimension hyper-plane · Density

## 1 Introduction

Multi-objective evolutionary algorithms (MOEAs) have been widely used in real-world applications [1], e.g., software engineering [2, 3], water distribution systems [4], and industrial scheduling [5, 6]. Hence, many MOEAs have been developed to solve the multi-objective optimization problems. They have shown excellent success on the multi-objective problems with two or three objectives, such as non-dominated sorting genetic Algorithm 2 (NSGA-II) [7], strength Pareto evolutionary Algorithm 2 (SPEA2) [8], multiobjective evolutionary algorithm based on decomposition (MOEA/D) [9], and Pareto envelope based selection Algorithm 2 (PESA-II) [10]. However, they suffer from some challenges in solving the multi-objective optimization problems with four or more objectives [1]. Moreover, many optimization problems having a large number of objectives also are commonly seen in real-world applications [11, 12]. Thus, it is not surprising that tackling a large number of objectives has been one of the hot research topics in evolutionary multiobjective optimization (EMO) community during recent years [13].

Many-objective optimization problems (MaOPs) are defined as problems with four or more objectives [1]. MaOPs pose many difficulties to any MOEAs. First and

---

✉ Guangming Dai  
cugdgm@126.com

Chong Zhou  
zhouchong0315@gmail.com

Maocai Wang  
cugwmc@163.com

<sup>1</sup> School of Computer, China University of Geosciences, Wuhan, 430074, China

<sup>2</sup> Hubei Key Laboratory of Intelligent Geo-Information Processing, China University of Geosciences, Wuhan, 430074, China

foremost, with the number of objectives increasing, almost all the solutions in the population become nondominated, which would lead to a severe loss of selection pressure that drive solutions toward the Pareto Front (PF). This considerably impedes the evolutionary process [14]. Secondly, in high dimension objective space, the conflict between convergence and diversity becomes deteriorative, since most of the diversity-preservation operators (such as the crowding distance operator [7], or  $k$ th nearest distance [8]) prefer selecting the dominance solutions [15]. They cannot enhance the selection pressure toward the PF, and may even put a brake on the evolutionary process to a certain extent. With the increased number of objective space in size, these operators also become a computationally expensive operation. Thirdly, visualization of a high-dimensional front is a challenging task. These difficulties have been highlighted both analytically and experimentally in the early studies [16, 17] on evolutionary many-objective optimization.

To overcome difficulties mentioned above, a variety of algorithms have been proposed, which can be mainly classified into five categories. Firstly, new dominance relation based approaches modify the Pareto dominance to produce fine selection pressure toward PF, such as  $\theta$  dominance [18],  $\epsilon$ -dominance [19, 20], grid dominance [21], fuzzy Pareto dominance [22, 23], preference order ranking [24], and so on [25–27]. These dominance relations extend their dominance area to a certain extent. Secondly, the idea of the decomposition-based approach is that it decomposes an  $m$ -objective problem into a series of single-objective subproblems based on aggregation functions and then solves these subproblems simultaneously. The well-known algorithm is the multi-objective evolutionary algorithm based on decomposition(MOEA/D) [9]. There are many other improved versions of MOEA/D, such as MOEA based on both dominance and decomposition (MOEA/DD) [13], improved decomposition-based evolutionary algorithm(I-DBEA) [28]. Thirdly, indicator based approaches introduce a performance indicator as the fitness value to guide the process of evolution. Hypervolume(called S metric) [29], R2 [30] and other indicators [31] have been used widely in MOEAs for many-objective problems. Some common algorithms contain hypervolume estimation algorithm (HypE) [32] S metric selection EMOA (SMS-EMOA) [33], many-objective metaheuristic based on the R2 indicator (MOMBI) [34] and indicator-based evolutionary algorithm(IBE) [31]. Among these indicators, hypervolume is probably the most popular one due to its good theoretical properties. However, with the increased number of objectives, computational costs become very expensive, which lead to being rarely applied in an application. Fourthly, reference points based MOEAs depend on multiple predefined reference points to make them multi-directional search. This class algorithm mainly has NSGA-III [35],  $\theta$  dominance based

evolutionary algorithm ( $\theta$ -DEA) [18], clustering-ranking evolutionary algorithm (crEA) [36]. Reference points play a crucial role in maintaining both convergence and diversity. Finally, hybrid technique based methods not only enhance pressure of population toward true PF, but also maintain diversity of population during the evolutionary process. The typical algorithms have Two\_Arch2 [37], MOEA/D-U [38] and EFR-RR [38]. Two\_Arch2 divides nondominated solutions into two archives, i.e. the convergence archive(CA) and the diversity archive(DA). CA adopting indicator of IBEA [31] promotes convergence, while DA maintains diversity based on another approach. Others introduce different operations into original algorithms to enhance their convergence or diversity. As shown in [39], the performance of an algorithm can be significantly enhanced by adopting a unified approach and borrowing operations from another algorithm. Generally speaking, the main idea of these algorithms is to balance both convergence and diversity.

Although many algorithms have been proposed to solve many-objective optimization problem, existing state-of-the-art many-objective evolutionary algorithms are still not powerful enough [18]. There are still some drawbacks in these algorithms. A new dominance relation based many-objective evolutionary algorithm, called  $\theta$ -DEA, is proposed to solve the many-objective optimization problem. Although  $\theta$ -DEA considers both convergence and diversity in the environmental selection process, we can find that it does not demonstrate strong competitiveness for its convergence and diversity on some problems with irregular shape, such as WFG1, WFG2, from [18]. In addition, convergence information in the  $\theta$  dominance may be weak, which leads to obtaining poor convergence performance on the problems with regular shape.

Based on the above motivation, we improve the  $\theta$ -DEA to obtain a solution set with better convergence and even distribution. Our contributions are as following:

- (1) An enhanced  $\theta$ -DEA, called  $\theta$ -EDEA, is proposed through improving  $\theta$  dominance to enhance convergence and introducing density based selection mechanism to maintain diversity. Firstly, we construct an  $m$ -dimension hyper-plane using extreme point on each objective. Then, the distance from the origin point to the projection of a solution on the reference line in  $\theta$  dominance is replaced with the perpendicular distance between the solution and the  $m$ -dimension hyper-plane to form the enhanced  $\theta$  dominance.
- (2) Through computational experiments, the results show that using the perpendicular distance is more effective than the distance from the ideal point to the projection of a solution on the reference line, which indicates that the enhanced  $\theta$  dominance has stronger convergence performance.

- (3) The density based selection mechanism is incorporated into the proposed  $\theta$ -EDEA to maintain diversity in the environment selection phase.
- (4) Through investigating to parameter  $\theta$ , the results indicate that when  $\theta = 5.0$ ,  $\theta$ -EDEA can achieve the best performance.
- (5) An extensive comparison between the proposed  $\theta$ -EDEA and five state-of-the-art algorithms on two well-known test suites is provided in this paper. The results show that  $\theta$ -EDEA is a very promising algorithm for many-objective optimization.

The rest of the paper is organized as follows. Section 2 briefly presents the concepts of MOO, and definition of  $\theta$  dominance. In Section 3, the proposed  $\theta$ -EDEA is described in detail. The experimental results and discussions are outlined in Section 4. Finally, the conclusions are presented in Section 5.

## 2 Preliminaries

### 2.1 Multiobjective optimization problem

Multi-objective optimization problem also is called multi-criteria optimization problem. Generally, a minimize multi-objective optimization problem having  $n$  decision variables and  $m$  objective variables can be mathematically described as:

$$\left. \begin{aligned} & \text{Minimize } f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ & \text{Subject to } \mathbf{x} \in \mathbf{X}. \end{aligned} \right\} \quad (1)$$

Where  $\mathbf{x} = (x_1, \dots, x_n)^T \in \mathbf{X} \subseteq R^n$  is  $n$ -dimension decision vector;  $\mathbf{X}$  represents  $n$ -dimension decision space;  $\mathbf{y} = (y_1, \dots, y_m)^T \in \mathbf{Y} \subseteq R^m$  is  $m$ -dimension objective vector;  $\mathbf{Y}$  denotes  $m$ -dimension objective space.  $f : \mathbf{X} \rightarrow \mathbf{Y}$  constitutes  $m$  conflicting objective functions, and is a mapping from  $n$ -dimensional decision space  $\mathbf{X}$  to  $m$ -dimensional objective space  $\mathbf{Y}$ .

Based on the above, a few relevant definitions are defined as follows:

**Definition 1 (Pareto Dominance)** [7, 40]: Let  $\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{X}$ ,  $\mathbf{x}_1 \prec \mathbf{x}_2$  if and only if

$$\forall i \in \{1, 2, \dots, m\} : f_i(\mathbf{x}_1) \leq f_i(\mathbf{x}_2) \wedge \exists j \in \{1, 2, \dots, m\} : f_j(\mathbf{x}_1) < f_j(\mathbf{x}_2)$$

**Definition 2 (Pareto Optimality)** [7, 40]: A vector  $\mathbf{x}^* \in \mathbf{X}$  is Pareto optimal if there is no other  $\mathbf{x} \in \mathbf{X}$ , such that  $\mathbf{x} \prec \mathbf{x}^*$

The set of all Pareto optimal points is called the Pareto set (PS). The corresponding shape of Pareto set in objective space forms the Pareto Front (PF).

**Definition 3 (Ideal Point):** The ideal point  $\mathbf{z}^*$  is a vector  $\mathbf{z}^* = (z_1^*, z_2^*, \dots, z_m^*)^T$ , where  $z_i^*$  is the infimum of each objective  $i \in \{1, 2, \dots, m\}$ .

**Definition 4 (Nadir Point):** The nadir point  $\mathbf{z}^{nad}$  is a vector  $\mathbf{z}^{nad} = (z_1^{nad}, z_2^{nad}, \dots, z_m^{nad})$ , where  $z_i^{nad}$  is the supremum of each objective,  $i \in \{1, 2, \dots, m\}$

### 2.2 $\theta$ Dominance relation

$\theta$  dominance which uses penalty-based boundary intersection (PBI) approach of MOEA/D is proposed in [18].  $\theta$  dominance works in the normalized objective space, where the origin point is the ideal point. Let  $\lambda_i$  be a reference line or weight vector,  $f(\mathbf{x})$  be a solution in the objective space. PBI is described as:

$$\text{Min } F(\mathbf{x}) = d_{i,1}(\mathbf{x}) + \theta \cdot d_{i,2}(\mathbf{x}) \quad (2)$$

Where  $\theta$  is a predefined penalty parameter,  $i$  is an index of a reference line. As shown in Fig. 1,  $d_{i,1}(\mathbf{x})$  denotes the distance from the origin point to the projection of  $f(\mathbf{x})$  on the reference line  $\lambda_i$  and  $d_{i,2}(\mathbf{x})$  is the perpendicular distance between  $f(\mathbf{x})$  and  $\lambda_i$ .  $\theta$  dominance is defined on a set of solution  $S_i$  and a predefined set of reference lines  $\Lambda$ .  $\theta$  dominance relation is determined by two steps. Firstly, each solution in  $S_i$  is assigned to a reference line to form  $N$  different clusters  $C$  according to  $d_{i,2}(\mathbf{x})$ , where  $N$  is size of reference lines. Secondly,  $F(\mathbf{x})$  of each solution is computed and solutions belonging to the same cluster  $C_i$  can be compared by  $F(\mathbf{x})$  to determine  $\theta$  dominance relation. Based on the above, several concepts related to  $\theta$  dominance can be defined as follows.

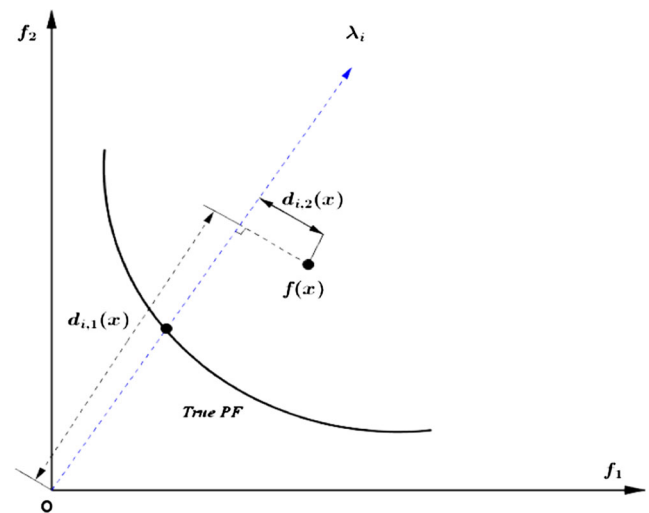


Fig. 1 Illustration of distances  $d_{i,1}(\mathbf{x})$ ,  $d_{i,2}(\mathbf{x})$

**Definition 5** ( $\theta$  dominance) [18]: Let  $\mathbf{x}, \mathbf{y} \in S_t$ ,  $\mathbf{x}$  is said to  $\theta$ -dominance  $\mathbf{y}$ , denoted by  $\mathbf{x} \prec_{\theta} \mathbf{y}$ , if and only if  $\mathbf{x} \in C_i, \mathbf{y} \in C_i$  and  $F_i(\mathbf{x}) < F_i(\mathbf{y})$  where  $i \in \{1, 2, \dots, N\}$

**Definition 6** ( $\theta$  optimality) [18]: A solution  $\mathbf{x}^* \in S_t$  is called as  $\theta$  optimality iff there is no other solution  $\mathbf{x} \in S_t$  such that  $\mathbf{x} \prec_{\theta} \mathbf{x}^*$

A set of all  $\theta$ -optimal solutions is referred to as the  $\theta$ -optimal set ( $\theta$ -OS) and each solution of  $\theta$ -OS is mapped correspondingly in the objective space to form the  $\theta$ -optimal front [18].

$\theta$  dominance aim to gain nondominated solution set that minimizes the distance to the PF and spreads well over the PF.

---

**Algorithm 1** Framework of the proposed  $\theta$ -EDEA

---

```

1:  $\Lambda \leftarrow \text{GenerateReferencePoints}()$ 
2:  $P_0 \leftarrow \text{InitializePopulation}()$ 
3:  $z^* \leftarrow \text{InitializeIdealPoint}()$ 
4:  $t \leftarrow 0$ 
5: while stopping criteria is not reach do
6:  $Q_t \leftarrow \text{RecombinationAndMutation}(P_t)$ 
7:  $R_t \leftarrow P_t \cup Q_t$ 
8:  $S_t \leftarrow \text{GetParetoNondominationLevels}(R_t)$ 
9:  $z^* \leftarrow \text{UpdateIdealPoint}(S_t)$ 
10:  $L \leftarrow \text{ConstructHyperplane}(S_t)$ 
11:  $C \leftarrow \text{Clustering}(S_t, \Lambda)$ 
12:  $\{F'_1, F'_2, \dots, F'_N\} \leftarrow \text{Ranking}(C)$ 
13:  $P_t \leftarrow \text{DensityBasedSelection}()$ 
14:  $t \leftarrow t+1$ 
15: end while

```

---

### 3 The proposed algorithm

#### 3.1 Framework of the proposed $\theta$ -EDEA

Framework of the proposed  $\theta$ -EDEA is presented in Algorithm 1. Firstly reference points  $\Lambda = \{\lambda_1, \lambda_2, \dots, \lambda_N\}$ , which are considered as center of clusters for clustering operator and maintain diversity, are generated using Das and Dennis's method [41]. Then, the initial population  $P_0$ , with  $N$  individuals is randomly produced. The ideal point  $z^*$  is initialized in step 3. Steps 5–15 are iterated in evolution procedure until stopping criteria is met. Step 6 adopts SBX operator with a large distribution index and polynomial mutation to produce offspring population  $Q_t$ . After union  $R_t$  of  $Q_t$  and current population  $P_t$ , nondominated sorting is used to classify  $R_t$  into different nondominated levels ( $F_1, F_2$  and so on). Population,  $S_t = \cup_{i=1}^l F_i$ , is achieved, where  $F_l$  satisfies size of  $S_t$  exceeding  $N$  for the first time.

Then,  $S_t$  is applied to update the ideal point. In step 10, an  $m$ -dimension hyper-plane is constructed using the extreme points. Clustering operator is employed to split the members in  $S_t$  into  $N$  clusters,  $C = \{C_1, C_2, \dots, C_N\}$ . Ranking operator is manipulated to sort the member in  $C_i$  in terms of enhanced  $\theta$  dominance. This produces different layers,  $F^{\wedge} = \{F'_1, F'_2, \dots, F'_N\}$ . Finally, the density based selection mechanism is used to select the remaining individuals from  $F'_i$  to fill slots in  $P_{t+1}$

#### 3.2 Reference points generation

A set of reference points is generated using Das and Dennis's method [41] that can create a set of uniformly distributed points in the objective space with a predefined integer  $p$  which controls the divisions along each axis. In this approach, reference points are sampled from a unit simplex. Reference lines which pass through the origin point and reference points not only are used to maintain diversity of obtained solutions, but also are considered as the center of each cluster produced by clustering operator. The number of reference points ( $H$ ) can be calculated as

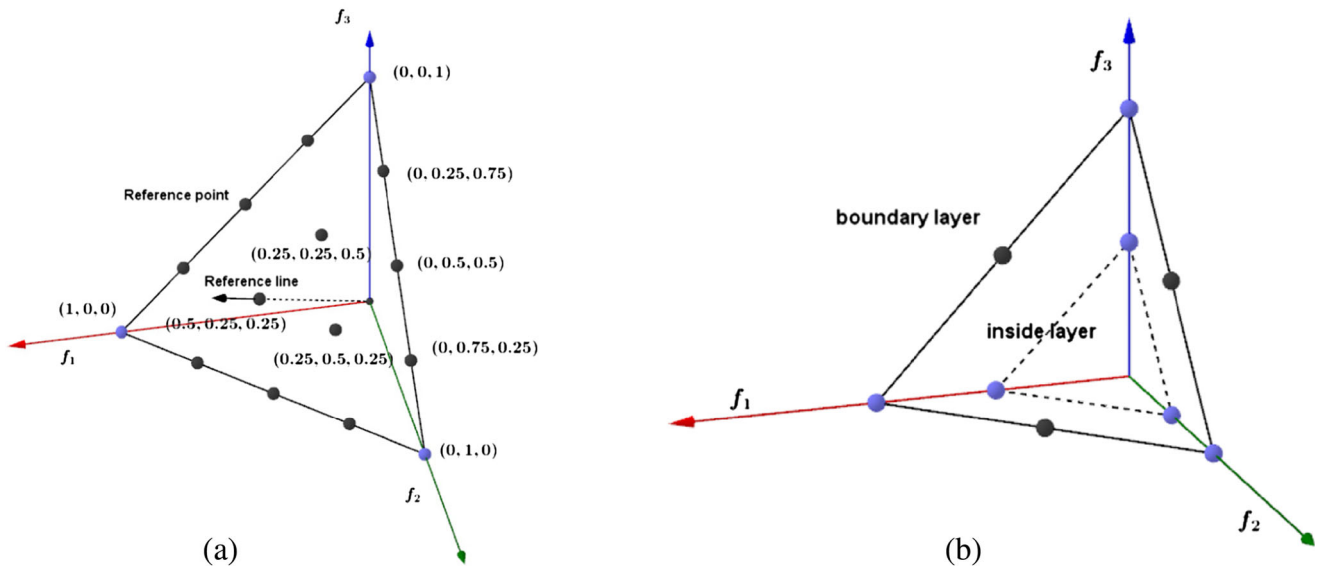
$$H = \binom{p+m-1}{p} \quad (3)$$

From the above formulation, the number of reference points depends on the dimension of objective space  $m$  and user-defined integer  $p$ . For example, for three objective problem, if  $p$  is set to 4, total number of reference points is  $H = 15$ . For clarity, the distribution of reference points or reference lines is illustrated in Fig. 2a on a three objective problem.

When  $p \geq m$ , number of reference points will be intermediate. However, for high-dimensional objective problem, even if we use  $p = m$ , it would lead to a huge number of reference points, which apparently increases the computational burden. For example, when  $m = 7, p = 7$  will require 1716 reference points. On the other hand, if we simply address this issue by lowering  $p$ , it will make all reference points sparsely lie along the boundary of the simplex, which is obviously harmful to the solution diversity. To solve the issue, we utilize two-layered reference points with relatively small values of  $p$  as presented in [35]. Let  $p_1$  and  $p_2$  denote divisions of boundary layer and inside layer respectively, then number of reference points is computed as:

$$H = \binom{p_1+m-1}{p_1} + \binom{p_2+m-1}{p_2} \quad (4)$$

Figure 2b illustrates the distribution of two-layered reference points, with  $p_1 = 2$  for the boundary layer and  $p_2 = 1$  for the inside layer on a three objective problem.



**Fig. 2** a Structured reference points in the three-objective problem with  $p = 4$ . b Two-layer structured reference points with  $p_1 = 2$  for the boundary layer and  $p_2 = 1$  for the inside layer

### 3.3 Constructing m-dimensional hyper-plane

In the proposed algorithm, we need  $m$  extreme points for constructing an  $m$ -dimension hyper-plane. Therefore, the extreme points ( $z_i^e$ ) in the  $i$ th objective axis are identified by finding the solution ( $\mathbf{x} \in S_f$ ) that requires the following achievement scalarizing function minimum.

---

#### Algorithm 2 ConstructHyperplane

---

**Input:**  $S_f$  (nondominated population),  $Z^*$  (ideal point)  
**Output:**  $\mathbf{a}$  (intercept vector)

- 1:  $Z^e \leftarrow \{-\infty, -\infty, \dots, -\infty\}$
  - 2:  $Z^e \leftarrow \text{ComputeExtremePoint}(S_f, Z^*)$
  - 3: **if**  $Z^e$  don't contain duplicate extreme points **do**
  - 4:      $\mathbf{a} \leftarrow \text{ComputeIntercept}(Z^e, Z^*)$
  - 5: **else**
  - 6:      $\mathbf{a} \leftarrow \text{FindMaxPoint}(S_f)$
  - 7: **end if**
- 

$$ASF(\mathbf{x}, \mathbf{w}_j) = \max_{i=1}^m \left\{ \frac{f_i(\mathbf{x}) - z_i^*}{w_{j,i}} \right\} \tag{5}$$

Where  $\mathbf{w}_j = (w_{j,1}, w_{j,2}, \dots, w_{j,m})$  is the axis direction of the  $j$ th objective axis and  $w_{j,i}$  subjects to the following condition.

$$w_{j,i} = \begin{cases} 0, & j \neq i, \\ 1, & j = i. \end{cases} \tag{6}$$

Note that, 0 should be replaced with a small number  $10^{-6}$ . Thus, these  $m$  extreme points are used to construct an  $m$

dimension hyper-plane. Let  $\mathbf{a} = \{a_1, a_2, \dots, a_m\}$  denote intercept  $a_i$  of the  $i$ th objective axis. So the equation of the  $m$  dimension hyper-plane is expressed as following

$$\frac{x_1}{a_1} + \frac{x_2}{a_2} + \dots + \frac{x_m}{a_m} = 1. \tag{7}$$

Given a matrix  $A = (z_1^e - z^*, z_2^e - z^*, \dots, z_m^e - z^*)$  and a vector  $\mathbf{b} = (1, 1, \dots, 1)$ , a vector  $\mathbf{x}$  is generated by  $A\mathbf{x} = \mathbf{b}$ . Each intercept  $a_i$  can be computed by following equation.

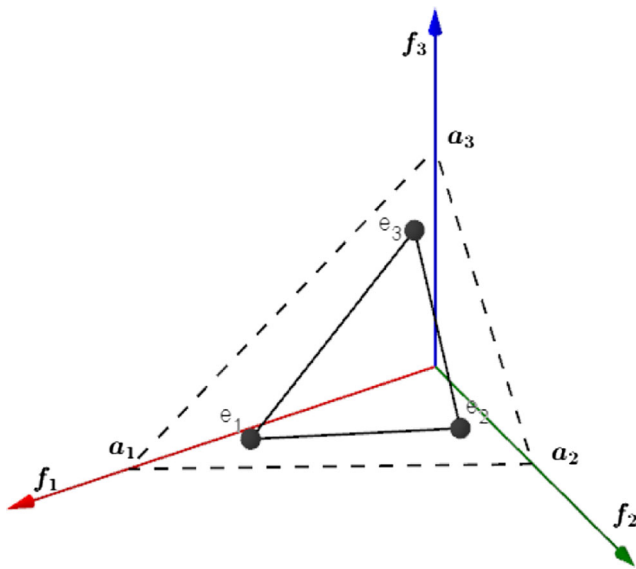
$$a_i = \frac{1}{x_i} + z_i^* \quad i \in \{1, 2, \dots, m\}. \tag{8}$$

Note that, if all extreme points contain duplicate extreme points, intercepts can be replaced with the maximum value ( $z_i^{\max}$ ) for each objective axis  $i = 1, 2, \dots, m$ . Procedure of calculating intercepts and constituting the hyper-plane using extreme points is showed in Fig. 3 on a three objective problem. The main process of constructing hyper-plane is showed in Algorithm 2.

### 3.4 Clustering operator

Clustering operator is an important step in the proposed algorithm. The clustering operator works in the objective space not in the decision space, and the ideal point is considered as the origin point. However, the clustering operator adapts different reference line as center of cluster so that each solution in  $S_f$  is associated with a certain reference line.

Let us consider a reference line  $\lambda_i$  pass through the origin with a reference point.  $f(\mathbf{x})$  presents a solution in the objective space.  $d_{i,1}(\mathbf{x})$  is the distance between the



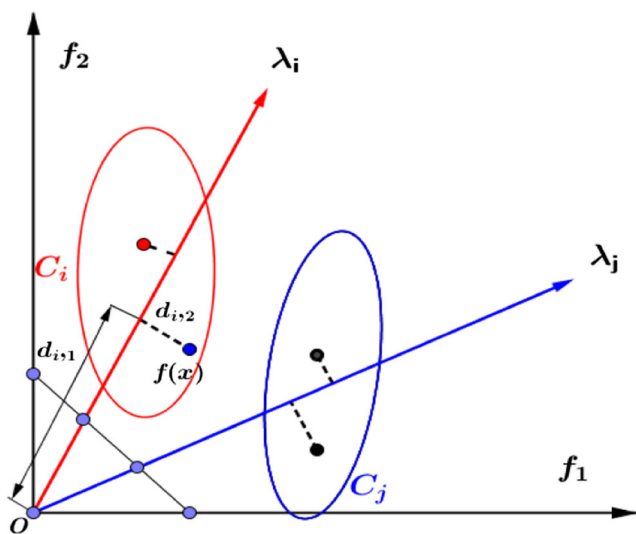
**Fig. 3** Constructing the hyper-plane from extreme points and computing intercepts are shown for a three-objective problem

ideal point and the projection of  $f(\mathbf{x})$  on  $\lambda_i$ ,  $d_{i,2}(\mathbf{x})$  is the perpendicular distance between  $f(\mathbf{x})$  and  $\lambda_i$ . These are shown in Fig. 4, They can be computed respectively as

$$d_{i,1}(\mathbf{x}) = \|f(\mathbf{x})\lambda_i / \|\lambda_i\|\| \tag{9}$$

$$d_{i,2}(\mathbf{x}) = \|f(\mathbf{x}) - d_{i,1}(\mathbf{x})(\lambda_j / \|\lambda_j\|)\|. \tag{10}$$

Procedure of clustering operator is that the solution  $\mathbf{x}$  with the closest distance ( $d_{i,2}(\mathbf{x})$ ) to  $i$ th reference line can be categorized to the cluster  $C_i$ , as be described in Algorithm 3.



**Fig. 4** Procedure of clustering operator

**Algorithm 3** Clustering operator

**Input:**  $S_t$ (population),  $\Lambda$ (reference point)

**Output:**  $C_1, C_2, \dots, C_N$

```

1:  $N \leftarrow |\Lambda|$ 
2:  $\{C_1, C_2, \dots, C_N\} \leftarrow \{\emptyset, \emptyset, \dots, \emptyset\}$ 
3: foreach  $s \in S_t$  do
4:    $n \leftarrow 1$ 
5:    $\min \leftarrow \infty$ 
6:   for  $j = 1:N$  do
7:     if  $d_{j,2}(s) < \min$  then
8:        $\min \leftarrow d_{j,2}(s)$ 
9:        $n = j$ 
10:    end if
11:  end for
12:   $C_n \leftarrow C_n \cup \{s\}$ 
13: end for
    
```

**3.5 Enhanced  $\theta$  dominance and ranking operator**

Enhanced  $\theta$  dominance is similar to  $\theta$  dominance in [18], but the difference between them is that  $d_{i,1}(\mathbf{x})$  in the  $\theta$  dominance is replaced. Enhanced  $\theta$  dominance also is defined on the population and the predefined set of reference line that pass through the origin point and reference points. Each solution can be divided into a cluster produced by the clustering operator. Solutions in the same cluster can compare with others using enhanced  $\theta$  dominance. In order to illustrate the enhanced  $\theta$  dominance, we explain it in two-objective space. Suppose that  $f(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}))^T$  is a solution in a two objective problem, line  $L$  is constructed by Algorithm 2. Let  $d_L(\mathbf{x})$  be the perpendicular distance between  $f(\mathbf{x})$  and line  $L$ , and  $d_i(\mathbf{x})$  be the perpendicular distance between  $f(\mathbf{x})$  and reference line  $\lambda_i$ . These distances are demonstrated in the two-objective space for convex problem in Fig. 5. They can be calculated respectively by (11) and (10)

$$d_L(\mathbf{x}) = \frac{|f_1(\mathbf{x})/a_1 + f_2(\mathbf{x})/a_2 - 1|}{\sqrt{(1/a_1)^2 + (1/a_2)^2}} \tag{11}$$

The larger  $d_L(\mathbf{x})$  value means better convergence while the smaller  $d_{i,2}(\mathbf{x})$  value can maintain better diversity. The  $F(\mathbf{x})$  of enhanced  $\theta$  dominance is defined as following:

$$\mathbf{Min} \quad F(\mathbf{x}) = \theta * d_{i,2}(\mathbf{x}) + (-d_L(\mathbf{x})) \tag{12}$$

Where  $\theta$  is a supplied penalty parameter. Since  $d_{i,2}(\mathbf{x})$  and  $d_L(\mathbf{x})$  are mutual conflict,  $d_L(\mathbf{x})$  need be set negative value so that they can simultaneously minimize.

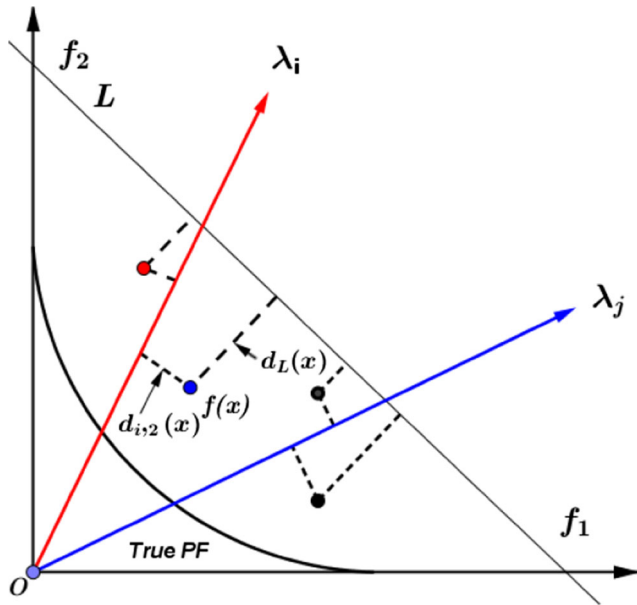


Fig. 5 Illustration of  $d_{i,2}(x)$  and  $d_L(x)$

**Definition 7** (enhanced  $\theta$  dominance): Let  $\mathbf{x}, \mathbf{y}$  be two solutions.  $\mathbf{x} <_E \mathbf{y}$ , if only and if  $\mathbf{x} \in C_i, \mathbf{y} \in C_i$ , and  $F(\mathbf{x}) < F(\mathbf{y})$ , where  $i \in \{1, 2, \dots, N\}$

Due to enhanced  $\theta$  dominance balancing both convergence and diversity, each solution in  $S_t$  is pushed fast and evenly toward the Pareto front. Meanwhile, with the extreme points on the each objective updating, line  $L$  also is reconstructed, which makes the line  $L$  close to Pareto Front.

However, for a concave problem, it is possible to occur that some solutions locate above line  $L$  with the evolution running. Figure 6 illustrates the process on the two objective problem. In the early stage of the evolutionary process, all solutions in the population locate below the line  $L$ , as shown in Fig. 6a. Then, some solutions possible locate above the line  $L$ , as shown in Fig. 6b. In the end stage of the evolutionary process, all solutions in  $S_t$  move to above the line  $L$ , as shown in Fig. 6c. The last two cases lead to the result that  $d_L(\mathbf{x})$  with smaller value is better. To tackle the two cases, the distance can be amended as follows.

$$d_L(\mathbf{x}) = \begin{cases} d_L(\mathbf{x}), & \text{if } f_1(x_1)/a_1 + f_2(x_2)/a_2 + f_3(x_3)/a_3 - 1 < 0 \\ -d_L(\mathbf{x}), & \text{otherwise.} \end{cases} \tag{13}$$

$F(\mathbf{x})$  of each solution is calculated by (11), and solutions in each cluster are assigned a layer using ranking operator which sorts the solutions in ascending order by their  $F(\mathbf{x})$ . The solution with smaller  $F(\mathbf{x})$  value is front in each cluster. The first solution in each cluster is selected to consist of layer  $F'_1$ . Then we use the same steps to produce  $F'_2, F'_3, \dots, F'_l$ . Finally, we can make each solution be assigned into a layer.

### 3.6 Density based selection mechanism

To maintain diversity, we compute density of solution based on clustering operator. After clustering operator, each solution is split into a cluster. Density value  $\sigma$  of a solution is defined as number of solution in the cluster to which the solution belongs. Solution with smaller density value is better. The density based selection mechanism firstly selects the lowest layer solution into new population  $P_{t+1}$  until the number of selected solutions equals the population size. However, solutions in the last layer are selected based on its density value. The pseudo-code of the selection procedure is given in Algorithm 4

**Algorithm 4** Selection based on density

**Input:**  $\{F'_1, F'_2, \dots, F'_l\}$  (different layer),  $N$  (population size)  
**Output:**  $P_{t+1}$  (new population)

```

1:  $P_{t+1} \leftarrow \emptyset$ 
2: for  $i = 1:l$  do
3:   if  $|P_{t+1}| + |F'_i| \leq N$  then
4:      $P_{t+1} \leftarrow P_{t+1} \cup F'_i$ 
5:   else
6:     break
7:   end if
8: end for
9:  $\text{ComputeDensity}(F'_i)$ 
10:  $P_{t+1} \leftarrow P_{t+1} \cup \text{DensityBasedSelection}(F'_l, N - |P_{t+1}|)$ 

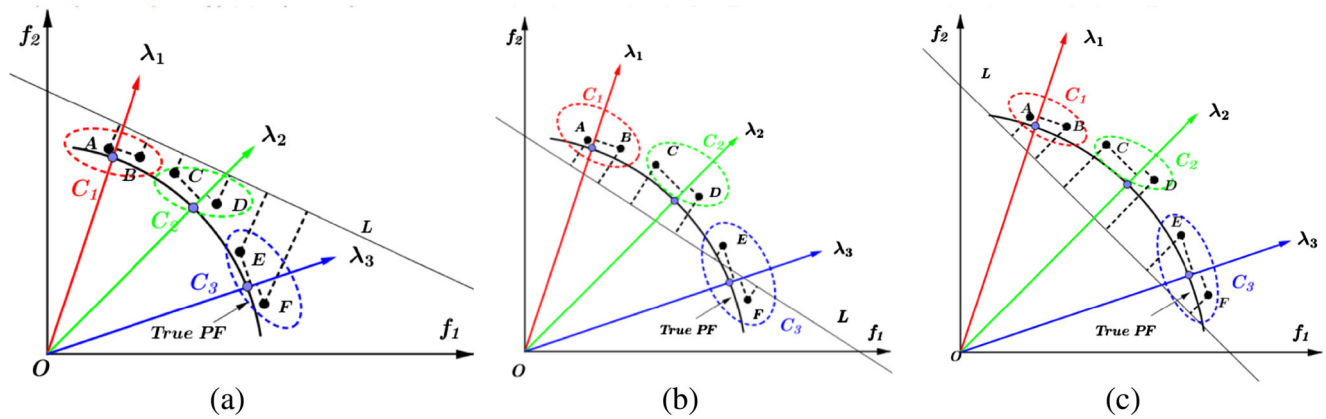
```

### 3.7 Computational complexity analysis

In the proposed  $\theta$ -EDEA, the main computational complexity depends on four components: *GetParetoNondominationLevels*, *ConstructHyperplane*, *Clustering Operator* and *Ranking Operator* from Algorithm 1. Given a MaOP with  $m$  objectives and population size  $N$  as an example, *GetParetoNondominationLevels* requires  $O(mN^2)$  computations. *ConstructHyperplane* in Algorithm 2 requires  $O(mN)$  computations. Computation of *Clustering Operator* in Algorithm 3 is same as  $\theta$ -DEA, requiring  $O(mN^2)$ . In the enhanced  $\theta$  dominance, distance of each  $\mathbf{x}$  in  $S_t$  to hyper-plane is computed in  $O(m)$  computations. So total complexity is  $O(m|S_t|)$ . The worst case of *Ranking Operator* requires  $O(|S_t| \log |S_t|)$  computations. Since  $|S_t| < 2N$ , total computational complexity of the proposed algorithm is  $O(mN^2)$ , which is same as that of  $\theta$ -DEA [18].

## 4 Experiment and results

This section devotes to the experimental design and result analysis for demonstrating the performance of the proposed



**Fig. 6** Changes of location between line  $L$  and solutions with the evolution process, on the two-objective problem

algorithm  $\theta$ -EEDA. We compare the proposed algorithm with five other state-of-the-art algorithms: NSGA-III [35]<sup>1</sup>, GrEA [21]<sup>2</sup>, MOEA/D [9]<sup>3</sup> HypE [32]<sup>4</sup>  $\theta$ -DEA [18]<sup>5</sup>. Test problems and performance metrics used in our experiments are briefly introduced in Section 4.1 and Section 4.2 respectively. Parameter setting is given in Section 4.3. Section 4.4 is significance test adopted in the paper. Finally, result and discussion are provided.

### 4.1 Test problems

Deb–Thiele–Laumanns–Zitzler (DTLZ) [38, 42] and Walking Fish Group (WFG) [43, 44] are two popular scalable test suites for many-objective optimization. To test performance of the proposed algorithm, we use DTLZ1-4 and all 9 WFG test functions from three to fifteen objective problems. DTLZ1 PF is satisfied with  $\sum_{i=1}^m f_i(x) = 0.5$  while other DTLZ problems need to satisfy  $\sum_{i=1}^m f_i(x)^2 = 1$ . The number of decision variables is set as  $n = m + r - 1$  for DTLZ test cases. In our experiment,  $r$  is set as 5 for the DTLZ1 and is set as 10 for the DTLZ2-4. For the WFG test suite, according to [18], the number of decision variables suggests to  $n = 24$ , where the position-related variable  $k = m - 1$  and the distance-related variable  $l = n - k$ . The features of these problems are described as Table 1.

<sup>1</sup>The code of NSGA-III is from <http://learntsinghuaeducn:8080/2012310563/ManyEAsrar>.

<sup>2</sup>The code of GrEA is from <http://www.tech.dmu.ac.uk/~syang/publications.html>.

<sup>3</sup>The code of MOEA/D is from <http://dces.essex.ac.uk/staff/zhang/webofmoead.htm>.

<sup>4</sup>The code of HypE is from <http://www.tik.ee.ethz.ch/pisa>.

<sup>5</sup>The code of  $\theta$ -DEA is from <http://learntsinghuaeducn:8080/2012310563/ManyEAsrar>.

### 4.2 Performance metrics

To evaluate the solution set obtained by the proposed algorithm, the inverted generational distance (IGD) [41] that is one of the most widely used metric is adopted in our experience. IGD provides a combined information about convergence and diversity of a solution set. To compute IGD value, a set of uniformly distributed points over the true PF is required. However, for many-objective optimization, a larger number of studies failed to point out how they sampled those points along the PF. Since reference points or reference directions are predefined in  $\theta$ -DEA, NSGA-III and MOEA/D algorithms, respectively and true Pareto-optimal surface of most test problems are known we can locate the intersection point of the Pareto-optimal surface with each reference line. Therefore, according to

**Table 1** Features of problems

Problems	Features
DTLZ1	Linear, Multi-modal
DTLZ2	Concave
DTLZ3	Concave, Multi-modal
DTLZ4	Concave, Biased
WFG1	Mixed, Biased, Scaled
WFG2	Convex, Disconnected, Multi-modal, Non-separable, Scaled
WFG3	Linear, Degenerate, Non-separable, Scaled
WFG4	Concave, Multi-modal, Scaled
WFG5	Concave, Deceptive, Scaled
WFG6	Concave, Non-separable, Scaled
WFG7	Concave, Biased, Scaled
WFG8	Concave, Biased, Non-separable, Scaled
WFG9	Concave, Biased, Non-separable, Scaled, Deceptive



intersection points, a new way to compute IGD is proposed by Deb and Jain [35]. The way is defined as following

$$IGD(\mathbf{A}, \mathbf{Z}) = \frac{1}{|\mathbf{Z}|} \sum_{i=1}^{|\mathbf{Z}|} \min_{j \in \mathbf{A}} d(a_j, z_i) \tag{14}$$

Where  $d(a_j, z_i) = \|a_j - z_i\|_2$ .  $\mathbf{A}$  presents the set of final nondominated solutions obtained in the objective space. The set  $\mathbf{Z}$  consists of intersection points of the true PF with each reference line. The set  $\mathbf{A}$  with smaller IGD values is better.

However, MOEAs without supplying reference points/directions fail to compute the IGD value. To compare with these MOEAs, we adopt another popular metric i.e., hypervolume(HV) [45, 46]. HV denotes the volume of the objective space between the obtained solution set and a reference point and also gives the solution set a comprehensive assessment with respect to convergence and diversity. The compute way can be described as:

$$HV(A, \mathbf{r}) = volume \left( \bigcup_{f \in A} [f_1(\mathbf{x}), r_1] \times \dots \times [f_m(\mathbf{x}), r_m] \right) \tag{15}$$

Where  $A$  is the set of nondominated solutions in the objective space.  $\mathbf{r} = (r_1, r_2, \dots, r_m)$  denotes a reference point in the objective space which is dominated by all points in the set  $A$ .

To compute HV value, we need normalize the objective values of solutions in  $A$  according to the range of the problem's Pareto front, due to test problems with different scaling of the search space. As the recommendation in [47], the reference point is set to 1.1 times the upper bound of the Pareto front (i.e.,  $r = 11^m$ )

In addition, for the problems with no more than ten objectives, we calculate HV exactly using the recently proposed WFG algorithm [29]. For problems with 15 objectives, we approximate the HV by the Monte Carlo simulation proposed in [32], and 10 000 000 sampling points are used to ensure the accuracy.

### 4.3 Parameter setting

In this section, we set all parameters utilized by these algorithms. For common parameters, we set the same value for the fair comparison. For other parameters, we set the best value that can obtain the best nondominated solution set.

- 1) Population size: for NSGA-III,  $\theta$ -DEA,  $\theta$ -DEA\*, MOEA/D algorithms, their population size  $N$  is controlled by division  $p$ . So we set parameter  $p$  to determine population size  $N$ . For other algorithms, we adopt a same parameter value to fairly compare. Table 2 lists the population sizes used in this paper for the problem with different objectives.

**Table 2** Setting of Population Size

No. of Objectives(m)	Divisions(H)	Population Size(N)
3	12	91
5	6	210
8	3,2	156
10	3,2	275
15	2,1	135

- 2) Parameters of Crossover and Mutation: For GrEA, MOEA/D, HypE, SBX probability is  $pc = 10$  and its distribution index is  $\eta_c = 2$  and Polynomial mutation probability is  $p_m = 1/n$  and its distribution index is  $\eta_m = 20$ . As for NSGA-III,  $\theta$ -DEA,  $\theta$  EDEA, the settings are only a bit different according to [29], where  $\eta_c$  is set to 30.
- 3) Stopping Condition and Number of Runs: an algorithm for each run reach to a specified number of generations. Each algorithm in this paper is independently run 20 times on each test instance.
- 4) Penalty Parameter of MOEA/D-PBI and  $\theta$ -DEA:  $\theta = 5.0$
- 5) Neighborhood Size in MOEA/D:  $T = 20$ .
- 6) Grid Division ( $div$ ) in GrEA: the settings of  $div$  are summarized in Table 3
- 7) Number of Points in Monte Carlo Sampling: it is set to 10 000 according to [28].

### 4.4 Significance test

To test the different statistical significance we use the Wilcoxon signed-rank test [48–50] at the 5% significance level.

**Table 3** Setting of grid division GrEA

Problem	No. of Objective(m)	Grid Division(div)
DTLZ1	3, 5, 8, 10, 15	14, 17, 12, 17, 28
DTLZ2	3, 5, 8, 10, 15	15, 12, 12, 12, 12
DTLZ3	3, 5, 8, 10, 15	17, 19, 19, 19, 33
DTLZ4	3, 5, 8, 10, 15	16, 11, 12, 17, 18
WFG1	3, 5, 8, 10, 15	6, 14, 13, 13, 13
WFG2	3, 5, 8, 10, 15	18, 16, 15, 19, 17
WFG3	3, 5, 8, 10, 15	16, 14, 12, 14, 16
WFG4	3, 5, 8, 10, 15	10, 13, 10, 11, 11
WFG5	3, 5, 8, 10, 15	10, 11, 12, 14, 14
WFG6	3, 5, 8, 10, 15	10, 12, 12, 14, 15
WFG7	3, 5, 8, 10, 15	10, 11, 12, 14, 14
WFG8	3, 5, 8, 10, 15	10, 12, 10, 14, 11
WFG9	3, 5, 8, 10, 15	9, 12, 12, 15, 13

## 4.5 Results and discussion

In this paper, we compare the proposed algorithm with six MOEA algorithms on the DTLZ and WFG test suits. According to two performance metrics IGD and HV comparison results are shown in Tables 4–6. Then we discuss and analyze the results.

### 4.5.1 Comparison with NSGA-III/MOEA/D and $\theta$ -DEA

In this section, we discuss IGD results on the DTLZ test problems. The way to compute IGD and experimental settings are same as those in the original NSGA-III [35] and  $\theta$ -DEA study [18]. Since our proposed algorithm, NSGA-III,  $\theta$ -DEA and MOEA/D all can obtain a same set of intersection point of true PF with reference lines or weight vectors for IGD computation, we only compare these algorithms on three-objective to fifteen-objective DTLZ1–4 problems.  $\theta$ -DEA\* that is another version of  $\theta$ -DEA with normalization and will also be involved in the comparison. Note that the IGD results of NSGA-III, MOEA/D-PBI and two versions of  $\theta$ -DEA used to compare with those of  $\theta$  EDEA are taken from [18, 35].

The results on the four DTLZ test problems are given in Table 4, with both best, median and worst of the IGD values over 20 independent runs being reported for the five compared MOEAs, where the best performance among the five compared algorithms is highlighted in bold. From Table 4, we can find that MOEA/D-PBI performs well on DTLZ2 test problems with all objectives and DTLZ3 test problems with 8, 10, 15 objectives. On the DTLZ3 problem with 8, 10, 15 objectives, MOEA/D-PBI can obtain the smallest IGD value. The DTLZ2 problem is a relatively easy problem with a spherical PF. The DTLZ3 problem adds a huge number of local PFs paralleling to the global PF based on DTLZ2, which poses a stiff challenge for algorithms to converge to the global PF. However, MOEA/D-PBI does not work well on DTLZ4 with any number of objectives. The main reason is that DTLZ4 problem introduces a biased density of points along the PF increasing challenge to maintain the diversity in the objective space. MOEA/D-PBI is not able to find a widely distributed set of solutions.

NSGA-III don't achieve best IGD value on all the test problems. But for all DTLZ1 and DTLZ4 with more than three objectives, NSGA-III obtains smaller IGD value than MOEA/D except for DTLZ1 with five-objective problem. However, for all problems,  $\theta$ -DEA performs consistently better than NSGA-III. We think that the reason is that  $\theta$  dominance used in  $\theta$ -DEA\* simultaneously maintain convergence and diversity. Although  $\theta$ -DEA performs consistently better than NSGA-III on all the problems except DTLZ1

with three, five objectives and DTLZ2 with three objectives,  $\theta$ -DEA\* achieved a smaller IGD value than  $\theta$ -DEA on almost all problems except DTLZ4 with fifteen objectives. For all the DTLZ3 and DTLZ4 problems,  $\theta$ -DEA\* obtains a much better IGD value than  $\theta$ -DEA on DTLZ3, but it obtains similar IGD value as  $\theta$ -DEA on DTLZ4.

From Table 4, the performance of  $\theta$ -EDEA is very promising on the four DTLZ test problems with more than three objectives. For all DTLZ1 instances, DTLZ3 with three, five objectives and DTLZ4 with three, five, eight objectives, our proposed  $\theta$ -EDEA can obtain the smallest IGD value than other algorithms. For all DTLZ2 instances, DTLZ3 and DTLZ4 high objective instances, performance of  $\theta$ -EDEA is slightly poor than  $\theta$ -DEA\* on these instances, while it performs better than  $\theta$ -DEA on DTLZ2 and DTLZ3 with ten, fifteen objectives. For DTLZ3 and DTLZ4 with more than eight objectives, IGD values obtained by  $\theta$ -EDEA are a marginally worse than IGD values obtained by  $\theta$ -DEA\*. We suspect that the reason is that  $\theta$ -EDEA sometimes fails to find the extreme point in high-dimensional objective space and constructs a wrong hyper-plane

From the 20 test instances of the DTLZ test suite presented in Table 4, we can find that  $\theta$ -EDEA achieves 9 the smallest IGD values, while NSGA-III achieves 0, MOEA/D-PBI achieves 6,  $\theta$ -DEA\* achieves 1, and  $\theta$ -DEA achieves 1. From these results, we can conclude that  $\theta$ -EDEA outperforms NSGA-III, MOEA/D-PBI,  $\theta$ -DEA\*, and  $\theta$ -DEA on DTLZ test problems in terms of IGD. Based on the above comparisons, it can be concluded that the proposed  $\theta$ -EDEA can generally maintain a good balance between convergence and diversity assisted by structured reference points and hyper-plane.

### 4.5.2 Comparison with state-of-the-art algorithms

In this section, we compare the proposed  $\theta$ -EDEA with other algorithms on the DTLZ and WFG test suits in terms of HV value. Tables 5–6 show the average HV for DTLZ1–4 and WFG1–9 problems over 20 independent runs, where the best average results of HV indicator are shown in bold-face. Moreover, we also perform the pair wise comparison in terms of the reported average values of HV indicator between  $\theta$ -EDEA and other algorithms via the Wilcoxon signed-rank test as shown in Tables 5–6, where symbol “\*” denotes that the outcome is significantly outperformed by  $\theta$ -EDEA.

To describe the distribution of obtained solutions on the test instances, we provide illustrations of three-objective DTLZ test instances in low-dimensional objective space and fifteen-objective WFG test instance in high-dimensional

**Table 4** Best, median, and worst IGD values obtained for NSGA-III/MOEA/D-PBI, two versions of  $\theta$ -DEA and  $\theta$ -EDEA on  $M$  objective DTLZ1, DTLZ2, DTLZ3, and DTLZ4 problems

Problem	M	MaxGen	NSGA-III	MOEA/D-PBI	$\theta$ -DEA*	$\theta$ -DEA	$\theta$ -EDEA
DTLZ1	3	400	4.880E-04	4.095E-04	5.655E-04	3.006E-04	<b>2.940E-04</b>
			1.308E-03	1.495E-03	1.307E-03	9.511E-04	<b>9.295E-04</b>
			4.880E-03	4.743E-03	9.449E-03	2.718E-03	<b>2.605E-03</b>
	5	600	5.116E-04	3.179E-04	4.432E-04	3.612E-04	<b>3.164E-04</b>
			9.799E-04	6.372E-04	7.328E-04	4.259E-04	<b>3.561E-04</b>
			1.979E-03	1.635E-03	2.138E-03	5.797E-04	<b>4.522E-04</b>
	8	750	2.044E-03	3.914E-03	1.982E-03	1.869E-03	<b>1.585E-03</b>
			3.979E-03	6.106E-03	2.704E-03	2.061E-03	<b>1.931E-03</b>
			8.721E-03	8.537E-03	4.620E-03	2.337E-03	<b>2.325E-03</b>
	10	1000	2.215E-03	3.872E-03	2.099E-03	1.999E-03	<b>1.484E-03</b>
			3.462E-03	5.073E-03	2.448E-03	2.268E-03	<b>2.055E-03</b>
			6.869E-03	6.130E-03	3.935E-03	2.425E-03	<b>2.373E-03</b>
	15	1500	2.649E-03	1.236E-02	2.442E-03	2.884E-03	<b>1.520E-03</b>
			5.063E-03	1.431E-02	8.152E-03	3.504E-03	<b>2.366E-03</b>
			1.123E-02	1.692E-02	2.236E-01	3.992E-03	<b>3.784E-03</b>
DTLZ2	3	250	1.262E-03	<b>5.432E-04</b>	1.042E-03	7.567E-04	7.054E-04
			1.357E-03	<b>6.406E-04</b>	1.569E-03	9.736E-04	9.973E-04
			2.114E-03	<b>8.006E-04</b>	5.497E-03	1.130E-03	1.179E-03
	5	350	4.254E-03	<b>1.219E-03</b>	2.720E-03	1.863E-03	1.931E-03
			4.982E-03	<b>1.437E-03</b>	3.252E-03	2.146E-03	2.363E-03
			5.862E-03	<b>1.727E-03</b>	5.333E-03	2.288E-03	2.539E-03
	8	500	1.371E-02	<b>3.097E-03</b>	7.786E-03	6.120E-03	6.123E-03
			1.571E-02	<b>3.763E-03</b>	8.990E-03	6.750E-03	7.995E-03
			1.811E-02	<b>5.198E-03</b>	1.140E-02	7.781E-03	9.683E-03
	10	750	1.350E-02	<b>2.474E-03</b>	7.558E-03	6.111E-03	6.845E-03
			1.528E-02	<b>2.778E-03</b>	8.809E-03	6.546E-03	7.578E-03
			1.697E-02	<b>3.235E-03</b>	1.020E-02	7.069E-03	8.981E-03
	15	1000	1.360E-02	<b>5.254E-03</b>	8.819E-03	7.269E-03	7.444E-03
			1.726E-02	<b>6.005E-03</b>	1.133E-02	8.264E-03	1.137E-02
			2.114E-02	<b>9.409E-03</b>	1.484E-02	9.137E-03	1.335E-02
DTLZ3	3	1000	9.751E-04	9.773E-04	1.343E-03	8.575E-04	<b>5.873E-04</b>
			4.007E-03	3.426E-03	3.541E-03	3.077E-03	<b>2.921E-03</b>
			6.665E-03	9.113E-03	<b>5.528E-03</b>	5.603E-03	8.909E-03
	5	1000	3.086E-03	1.129E-03	1.982E-03	8.738E-04	<b>7.203E-04</b>
			5.960E-03	2.213E-03	4.272E-03	1.971E-03	<b>1.806E-03</b>
			1.960E-03	6.147E-03	1.911E-02	4.340E-03	<b>3.534E-03</b>
	8	1000	1.244E-02	<b>6.459E-03</b>	8.769E-03	6.493E-03	8.956E-03
			2.375E-02	1.948E-02	1.535E-02	<b>1.036E-02</b>	1.394E-02
			9.649E-02	1.123E+00	3.826E-02	<b>1.549E-02</b>	2.016E-02
	10	1500	8.849E-03	<b>2.791E-03</b>	5.970E-03	5.074E-03	5.608E-03
			1.188E-02	<b>4.319E-03</b>	7.244E-03	6.121E-03	8.079E-03
			2.083E-02	1.010E+00	2.323E-02	<b>7.243E-03</b>	9.678E-03
	15	2000	1.401E-02	<b>4.360E-03</b>	9.834E-03	7.892E-03	9.366E-03
			2.145E-02	1.664E-02	1.917E-02	<b>9.924E-03</b>	1.268E-02
			4.195E-02	1.260E+00	6.210E-01	<b>1.434E-02</b>	1.695E-02

**Table 4** (continued)

Problem	M	MaxGen	NSGA-III	MOEA/D-PBI	$\theta$ -DEA*	$\theta$ -DEA	$\theta$ -EDEA
DTLZ4	3	600	2.915E-04	2.929E-01	1.866E-04	1.408E-04	<b>1.405E-04</b>
			5.970E-04	4.280E-01	2.506E-04	<b>1.918E-04</b>	2.092E-04
			4.286E-01	<b>5.234E-01</b>	5.320E-01	5.321E-01	5.320E-01
	5	1000	9.849E-04	1.080E-01	2.616E-04	2.780E-04	<b>2.554E-04</b>
			1.255E-03	5.787E-01	3.790E-04	3.142E-04	<b>3.139E-04</b>
			1.721E-03	7.348E-01	4.114E-04	<b>3.586E-04</b>	4.497E-04
	8	1250	5.079E-03	5.298E-01	2.780E-03	2.323E-03	<b>2.202E-03</b>
			7.054E-03	8.816E-01	<b>3.098E-03</b>	3.172E-03	3.712E-03
			6.051E-01	9.723E-01	<b>3.569E-03</b>	3.635E-03	4.596E-03
	10	2000	5.694E-03	3.966E-01	2.746E-03	<b>2.715E-03</b>	3.090E-03
			6.337E-03	9.203E-01	3.341E-03	<b>3.216E-03</b>	3.930E-03
			1.076E-01	1.077E+00	3.914E-03	<b>3.711E-03</b>	4.835E-03
	15	3000	7.110E-03	5.890E-01	<b>4.143E-03</b>	4.182E-03	5.107E-03
			3.431E-01	1.133E+00	5.904E-03	<b>5.633E-03</b>	8.327E-03
			1.073E+00	1.249E+00	7.680E-03	<b>6.562E-03</b>	9.104E-03

Best performance is shown in bold

**Table 5** Average HV on the DTLZ1-4 problems

Problem	m	MaxGen	GrEA	NSGA*III	MOEA/D	HypE	$\theta$ -DEA	$\theta$ -EDEA
DTLZ1	3	400	1.082739*	1.115466*	1.114239*	1.117363*	1.118932*	<b>1.120385</b>
	5	600	1.519459*	1.576958*	1.577202*	1.454109*	1.577901*	<b>1.578002</b>
	8	750	2.110453*	2.137026*	2.136825*	1.989577*	2.138167*	<b>2.138317</b>
	10	1000	2.558587*	2.592386*	2.591796*	2.526151*	2.592568*	<b>2.593868</b>
	15	1500	3.912006*	4.173867*	4.170569*	3.853399*	4.175940*	<b>4.177080</b>
DTLZ2	3	250	0.730874*	0.739818*	0.744239*	<b>0.754543</b>	0.744635*	0.744661
	5	350	1.304378*	1.302634*	1.307512*	0.765242*	1.307572*	<b>1.308034</b>
	8	500	<b>1.997173</b>	1.968493*	1.978216*	0.929827*	1.978721*	1.979740
	10	750	<b>2.515879</b>	2.509008*	2.514912	1.165486*	2.514398*	2.514908
	15	1000	4.061970*	4.134103*	4.137689*	1.524265*	4.137862*	<b>4.138910</b>
DTLZ3	3	1000	0.693546*	0.739503*	0.736892*	<b>0.755781</b>	0.739674*	0.744583
	5	1000	1.161970*	1.302303*	1.303509*	0.793805*	1.307257*	<b>1.308333</b>
	8	1000	1.717030*	1.954336*	1.295804*	0.993805*	1.969218*	<b>1.979488</b>
	10	1500	2.392539*	2.508985*	2.430164*	1.165778*	2.514406*	<b>2.514908</b>
	15	2000	3.355547*	4.127026*	3.092941*	1.538516*	4.136242*	<b>4.137585</b>
DTLZ4	3	600	0.564057*	<b>0.749818</b>	0.498067*	0.558772*	0.602951*	0.650585
	5	1000	1.307255*	1.307596*	1.284505*	1.013612*	1.309439*	<b>1.309772</b>
	8	1250	<b>1.988538</b>	1.981936	1.882543*	0.969695*	1.976632*	1.979322
	10	2000	2.506069*	2.514950*	2.454102*	1.227722*	2.514963*	<b>2.515473</b>
	15	3000	4.129253*	4.138694*	3.984783*	2.142551*	4.138223*	<b>4.138335</b>

The best value of each instance is highlighted in boldface

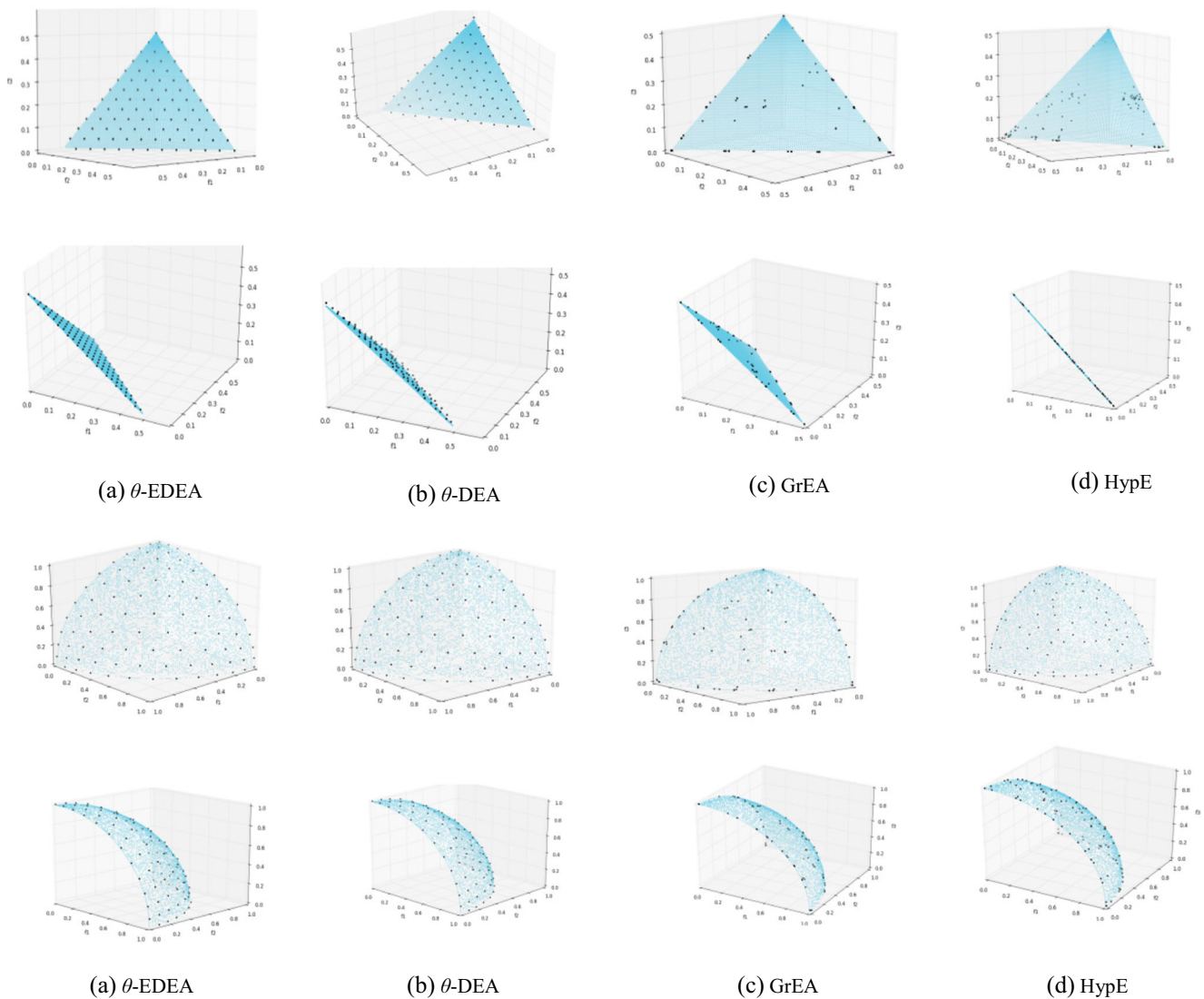
\*The outcome is significantly outperformed by  $\theta$ -EDEA

**Table 6** Average HV on the WFG1-9 problems

Problem	$m$	MaxGen	GrEA	NSGAIII	MOEA/D	HypE	$\theta$ -DEA	$\theta$ -EDEA
WFG1	3	400	0.895068*	0.689467*	0.675222*	0.988430*	0.968445*	<b>1.044409</b>
	5	750	1.289914*	0.905452*	1.349140*	1.016691*	1.428698*	<b>1.514039</b>
	8	1500	1.789506*	1.438081*	1.758656*	1.543206*	<b>1.889775</b>	1.863716
	10	2000	2.349430*	2.224545*	1.818101*	2.194333*	2.138866*	<b>2.410439</b>
	15	3000	3.813874	<b>4.096943</b>	1.729792*	3.972482	3.867407	3.629010
WFG2	3	400	1.229894	1.236114	1.111475*	<b>1.241143</b>	1.116515*	1.204943
	5	750	1.571387*	1.597623*	1.533952*	1.597188*	1.500357*	<b>1.599362</b>
	8	1500	2.091851*	<b>2.141299</b>	2.028409*	2.105278	1.972818*	2.122803
	10	2000	2.581640*	<b>2.592674</b>	2.488966*	2.567307*	2.393355*	2.577545
	15	3000	4.100339*	<b>4.167543</b>	3.967803*	4.164007*	3.961793*	4.133155
WFG3	3	400	0.796657	0.692841	0.771185	<b>0.814962</b>	0.703137	0.698698
	5	750	<b>1.061229</b>	1.007997*	0.818806*	1.006524*	1.029293*	1.059608
	8	1500	1.269627	1.216728*	0.767703*	<b>1.329387</b>	1.237738	1.231836
	10	2000	1.587147	1.586258	0.533817*	<b>1.754860</b>	1.366208*	1.375443
	15	3000	2.694943	2.510223	0.585506*	<b>2.893723</b>	0.904082*	1.517330
WFG4	3	400	0.706897*	0.724183*	0.683321*	<b>0.755720</b>	0.725280*	0.731983
	5	750	1.288774*	<b>1.290371</b>	1.170129*	0.863721*	1.268144*	1.271631
	8	1500	1.941171	<b>1.974878</b>	1.29033*	1.170046*	1.865674*	1.919393
	10	2000	2.478209	<b>2.510336</b>	1.448645*	1.511532*	2.222351*	2.466326
	15	3000	3.946541*	<b>4.136393</b>	1.728177*	2.441369*	2.990774*	4.078850
WFG5	3	400	0.701313*	0.693657*	0.664493*	0.714197*	0.704382*	<b>0.733420</b>
	5	750	1.221282*	1.237322*	1.129326*	0.853723*	1.253552*	<b>1.274434</b>
	8	1500	1.865096*	1.879006*	1.276512*	1.176945*	1.860502*	<b>1.921456</b>
	10	2000	2.334575*	2.361469*	1.549924*	1.591378*	2.235619*	<b>2.467227</b>
	15	3000	3.508628*	3.833242*	1.864379*	2.925016*	2.719208*	<b>4.077490</b>
WFG6	3	400	0.680906*	0.691486*	0.654956*	0.710370*	0.702309*	<b>0.732276</b>
	5	750	1.214910*	1.212372*	1.050044*	0.523099*	1.250184*	<b>1.272797</b>
	8	1500	1.853571*	1.877105*	0.701298*	0.594473*	1.840666*	<b>1.920959</b>
	10	2000	2.344692*	2.343912*	0.794119*	0.696925*	2.212380*	<b>2.466218</b>
	15	3000	3.610583*	3.717982*	0.583321*	1.217393*	2.936836*	<b>4.078140</b>
WFG7	3	400	0.718109*	0.736025	0.677115*	<b>0.743787</b>	0.723335	0.728889
	5	750	1.290140	<b>1.298653</b>	1.121068*	0.599807*	1.266145*	1.268587
	8	1500	<b>1.994678</b>	1.975418	0.874136*	0.575521*	1.854709*	1.914270
	10	2000	2.497715	<b>2.509460</b>	1.051292*	1.510454*	2.265434*	2.465931
	15	3000	3.900431*	<b>4.134418</b>	0.801271*	2.137169*	3.474231*	4.078020
WFG8	3	400	0.662997*	0.684537*	0.632906*	0.684505*	0.671341*	<b>0.713981</b>
	5	750	1.180906*	1.209515*	0.970442*	0.206996*	1.180849*	<b>1.233336</b>
	8	1500	1.745758*	1.759951*	0.327348*	0.807302*	1.824686*	<b>1.834345</b>
	10	2000	2.353431*	2.280276*	0.252241*	1.308194*	2.284630*	<b>2.362661</b>
	15	3000	3.653792*	3.815520*	0.705129*	1.878598*	2.655804*	<b>3.932465</b>
WFG9	3	400	0.685212*	0.706173*	0.575197*	<b>0.738415</b>	0.699731*	0.725614
	5	750	1.229112*	1.216387*	1.006993*	1.117802*	1.216973*	<b>1.241202</b>
	8	1500	<b>1.865343</b>	1.803989*	0.920784*	1.452032*	1.757462*	1.842188
	10	2000	2.367534*	2.354544*	1.115323*	2.161329*	2.082889*	<b>2.370702</b>
	15	3000	3.537925*	3.801860*	1.012731*	2.842365*	2.674073*	<b>3.897450</b>

The best value of each instance is highlighted in boldface

\*The outcome is significantly outperformed by  $\theta$ -EDEA



**Fig. 7** Pareto-fronts produced by (a)  $\theta$ -EDEA, (b)  $\theta$ -DEA, (c) GrEA and (d) HypE on the DTLZ1 and DTLZ3 problems with three objectives

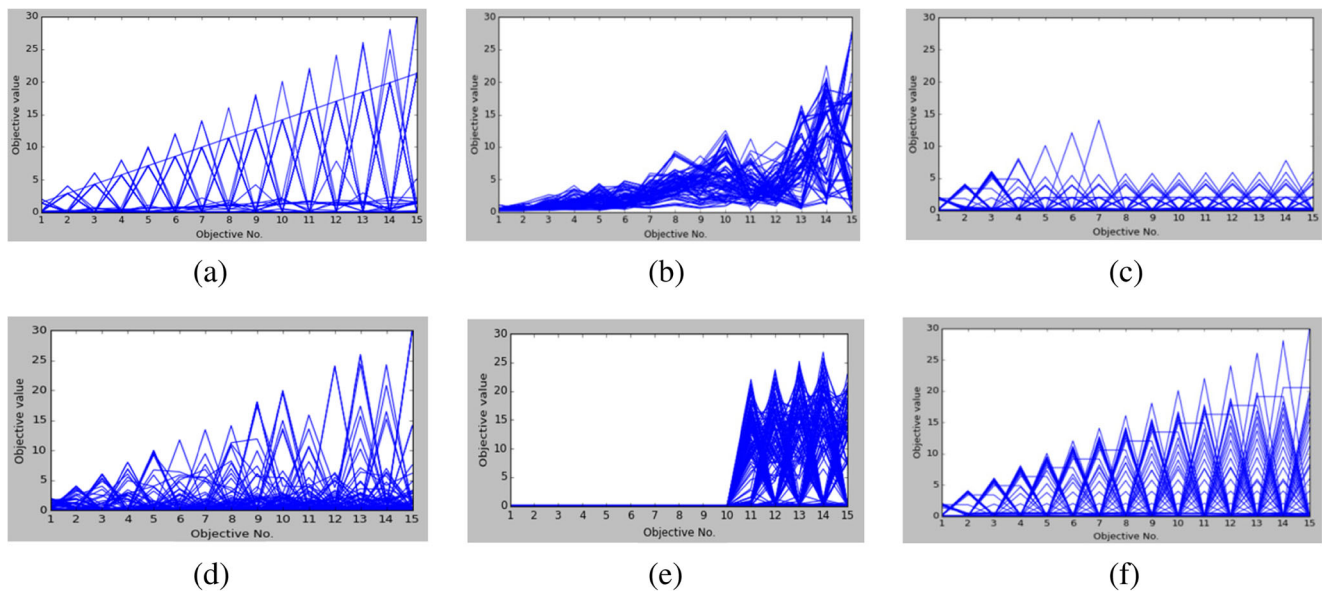
objective space. Distributions of final solutions obtained by competitive algorithms in a single run on the DTLZ1 and DTLZ3 with three objectives are shown in Fig. 7 by three-dimensional coordinates, while final solutions in a single run on the WFG6 with fifteen objectives are shown in Fig. 8 by parallel coordinates. It is clearly seen from Figs. 7 and 8 that our proposed  $\theta$ -EDEA is able to find a solution set with good convergence and diversity.

To quantify how well each algorithm performs overall, the performance score [32] used to rank the algorithms is introduced. Suppose there are  $l$  algorithms  $Alg_1, Alg_2, \dots, Alg_l$  involved in the comparison, let  $\delta_{i,j}$  be 1, if  $Alg_i$  is significantly better than  $Alg_j$  according to HV, and 0

otherwise. Then, for each algorithm  $Alg_i$ , the performance score  $P(Alg_i)$  is computed as

$$P(Alg_i) = \sum_{\substack{j=1 \\ j \neq i}}^l \delta_{i,j} \tag{16}$$

The performance score demonstrates how many other algorithms significantly outperform the corresponding algorithm on the test case considered. So the algorithm with a lower values is better. Figure 9a demonstrates the average performance score for different test problems and Fig. 9b



**Fig. 8** Final solution sets produced by **a** NSGA-III, **b** GrEA, **c**  $\theta$ -DEA, **d** MOEA/D, **e** HypE and **f**  $\theta$ -EDEA on the fifteen-objective WFG6 problem, is shown by parallel coordinates

demonstrates the average performance score for different numbers of objectives.

Based on the above results, we can obtain some observations for each algorithm. GrEA obtains the best HV value on DTLZ2 DTLZ3, WFG7 and WFG9 with eight objectives, while it was slightly worse on the other problems. From Fig. 8b, GrEA cannot converge the best value on each dimension and distribution of the final solution set obtained by GrEA is dense on the WFG6 with fifteen objectives. But GrEA works better on the three objective DTLZ instances. This is mainly because that GrEA divides each dimension of the objective space into the same number of divisions. So the performance of GrEA is influenced by the parameter *div*.

NSGA-III shows the closest performance to the proposed  $\theta$ -EDEA over a widely range of WFG test problems, especially for WFG4 and WFG7, and it also can perform better than  $\theta$ -DEA. But NSGA-III is slightly worse than  $\theta$ -DEA on the DTLZ problem from Fig. 9a. It is interesting to find that from Fig. 9b, it takes the second place on the five-objective and fifteen-objective instances.

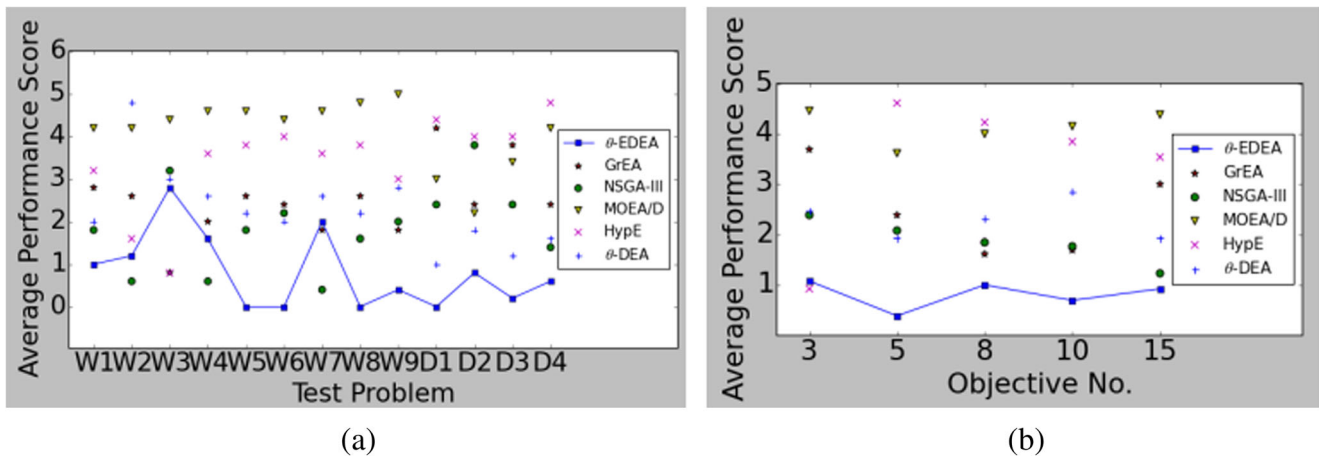
For solving normalized test problems, MOEA/D achieves satisfactory performance except DTLZ4. However, it does not perform well on the WFG test problems. The main reason may be that MOEA/D don't employ normalization on the WFG problems. This is why it ranks poorly in Fig. 9.

From Fig. 9b, HypE achieves a good performance on all test problems with three objectives according to HV. Indeed,

it wins 7 out of the 13 instances with three objectives. However, it does not show advantages over the other algorithms on instances with more than three objectives except for WFG3 with eight, ten, fifteen objectives. On these instances, HypE obtains the best HV values. HypE need to compute exactly the hypervolume as its fitness values when  $m < 3$ , otherwise it computes hypervolume using Monte Carlo simulation to accelerate the speed. So inaccurate fitness value may lead to its poor performance on problems with many objectives. Moreover, HypE does not maintain evenly distributed solution set on all test instances, as shown in Figs. 7d and 8e.

Similar to MOEA/D,  $\theta$ -DEA without normalization achieves a better performance  $\theta$  on DTLZ test problems than NSGA-III, but it does not behave quite well on WFG test problems, especially on WFG2. From Fig. 8c, it does not maintain a good convergence on the fifteen-objective WFG6 problem. However,  $\theta$ -DEA obtained a good convergence and diversity on three-objective DTLZ test problems, as shown in Fig. 7b. Due to worse performance on WFG test problems, its performance is poorer than NSGA-III on all test problems.

The performance of proposed  $\theta$ EDEA is the best than other compared algorithms from Fig. 7 on the two problems. However, it doesn't show consistently excellent performances on all WFG test problems, especially on WFG3. We suspect that geometrical shapes of WFG3' PFs are degenerate and linear, which results in that uniform ref-



**Fig. 9** **a** Average performance score over all objective dimensions for different test problems, namely DTLZ (Dx) and WFG (Wx). **b** Average performance score over all test problems for different number of

objectives. The smaller the score, the better the PF approximation in terms of HV. The values of  $\theta$ -EDEA are connected by a solid line to easier assess the score

erence points have opposite effect. As shown in [43], all the algorithms with uniform reference points don't perform well on the WFG3 problem. In general, from all comparison in terms of HV, our proposed  $\theta$ -EDEA gets a better

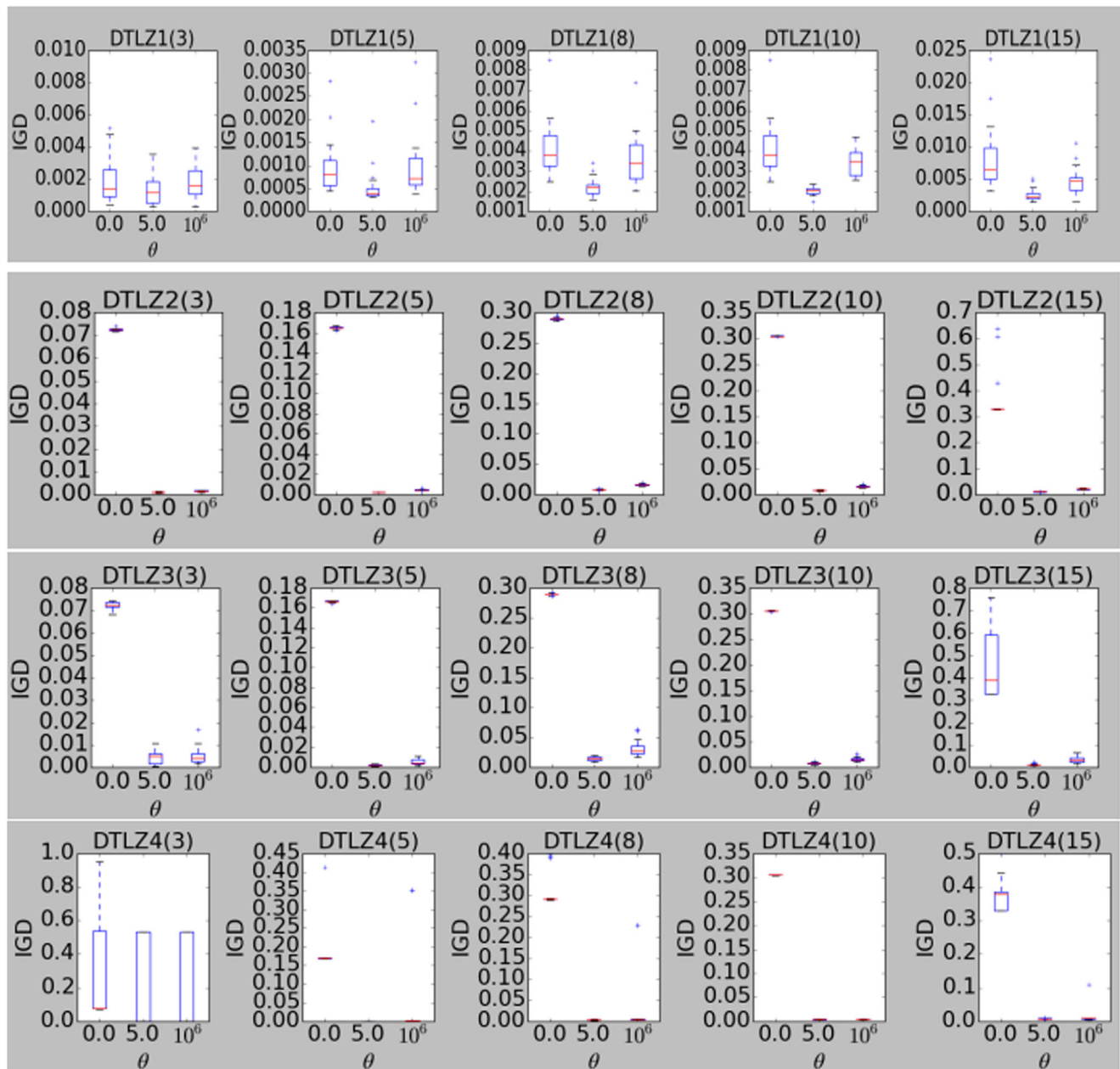
performance on test problems with more than three objectives. We think that the reason may be that the hyper-plane constructed is nearer and nearer to the true PF with the evolution running, and the closest solution to true PF can

**Table 7** Average IGD and HV values obtained by  $\theta$ -EDEA\* ( $\theta = 0.0$ ) and  $\theta$ -DEA ( $\theta = 0.0$ ) on the DTLZ1-4 problems

Problem	$m$	MaxGen	IGD		HV	
			$\theta$ -EDEA* ( $\theta = 0.0$ )	$\theta$ -DEA ( $\theta = 0.0$ )	$\theta$ -EDEA* ( $\theta = 0.0$ )	$\theta$ -DEA ( $\theta = 0.0$ )
DTLZ1	3	400	<b>0.001892</b>	0.031534	<b>1.117392</b>	1.059386
	5	600	<b>0.000963</b>	0.084348	<b>1.577674</b>	1.253205
	8	750	<b>0.004202</b>	0.138724	<b>2.138022</b>	2.012146
	10	1000	<b>0.003556</b>	0.147053	<b>2.592827</b>	2.508532
	15	1500	<b>0.008137</b>	0.297560	<b>4.176743</b>	3.290382
DTLZ2	3	250	<b>0.072470</b>	0.085894	<b>0.729786</b>	0.676301
	5	350	<b>0.165131</b>	0.261315	<b>1.266188</b>	0.829019
	8	500	<b>0.290134</b>	0.454144	<b>1.934898</b>	1.592977
	10	750	<b>0.304797</b>	0.503104	<b>2.483959</b>	2.106326
	15	1000	<b>0.363108</b>	0.935090	<b>3.791441</b>	1.299609
DTLZ3	3	1000	<b>0.072351</b>	0.085968	<b>0.723960</b>	0.674149
	5	1000	<b>0.166572</b>	0.262791	<b>1.261848</b>	0.815090
	8	1000	<b>0.290001</b>	0.453196	<b>1.931257</b>	1.614756
	10	1500	<b>0.305365</b>	0.503241	<b>2.480466</b>	2.135813
	15	2000	<b>0.468710</b>	0.912567	<b>3.697852</b>	1.521037
DTLZ4	3	600	<b>0.324778</b>	0.354339	<b>0.560231</b>	0.521130
	5	1000	<b>0.181351</b>	0.263465	<b>1.255655</b>	0.814275
	8	1250	<b>0.311729</b>	0.470214	<b>1.929659</b>	1.570027
	10	2000	<b>0.305477</b>	0.511248	<b>2.490012</b>	2.123995
	15	3000	<b>0.372665</b>	0.687801	<b>3.854801</b>	3.168345

The best value of each instance is highlighted in boldface

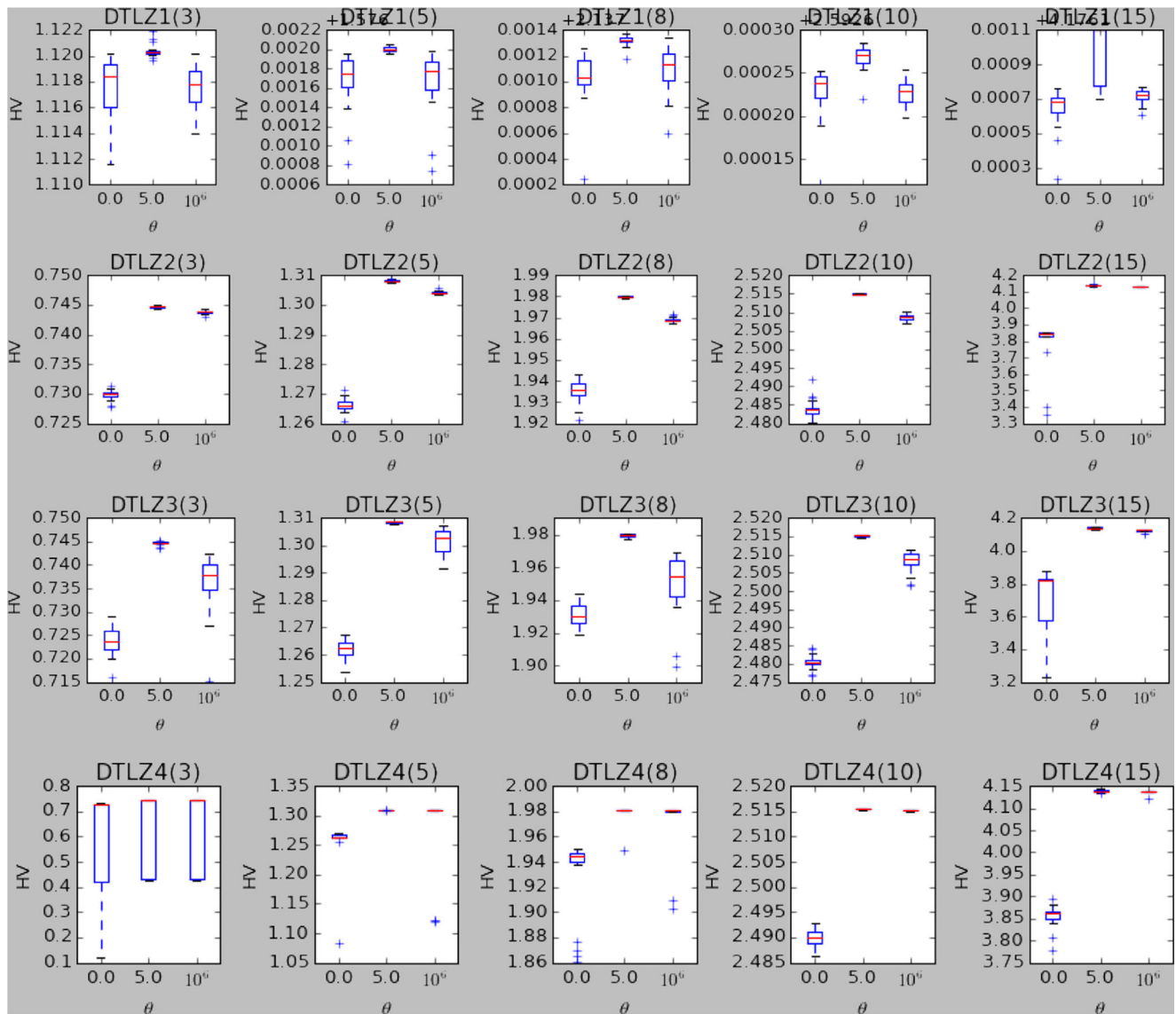




**Fig. 10** Examination of the influence of  $\theta$  on IGD of  $\theta$ -EEDA for DTLZ1–4 problems with varying number of objectives  $m$ . The box plots show the IGD of 20 independent runs each on (1)  $\theta = 0$ , (2)  $\theta = 5.0$ , (3)  $\theta = 10^6$ . Number of objectives is in brackets

be found through the perpendicular distance between the solution and the hyper-plane. From Fig. 8d, solution set obtained by  $\theta$ -EEDA has a better convergence and diversity than other algorithms. From the test problem, our proposed  $\theta$ -EEDA achieves the best score on all test problems except for WFG2-3, WFG4 and WFG7 in Fig. 9a. From the number of objective,  $\theta$ -EEDA shows obvious advantages on the MaOPs, as shown in Fig. 9b except for three objectives.

In summary,  $\theta$ -EEDA is able to obtain better or similar performance with other compared state-of-the-art algorithms on MaOPs. However,  $\theta$ -EEDA still doesn't achieve better performance on the WFG3, due to distinctive features of the problem. HypE has a better performance than  $\theta$ -EEDA on the problems with three objectives, while it doesn't maintain better diversity. NSGA-III performs the best on WFG2-4 and WFG7 problems. Compared with  $\theta$ -DEA, the proposed algorithm in this paper obtains better or simi-



**Fig. 11** Examination of the influence of  $\theta$  on HV of  $\theta$ -EEDA for DTLZ1–4 problems with varying number of objectives  $m$ . The box plots show the HV of 20 independent runs each on (1)  $\theta = 0$ , (2)  $\theta = 5.0$ , (3)  $\theta = 10^6$ . Number of objectives is in brackets

lar performance in the all test instances. So our proposed methods can effectively handle many-objective problems.

#### 4.5.3 Distance influence analysis

In this section, we inspect the effectiveness of the perpendicular distance between a solution and the  $m$ -dimension hyper-plane in the enhanced  $\theta$  dominance. In the  $\theta$ -DEA and  $\theta$ -EEDA,  $\theta$  takes a value of 0.0, so that they only depend on a single distance during the evolutionary process. From Table 7, we can see that the proposed algorithm can achieve a better performance than  $\theta$ -DEA on the all test problems in terms of two indicators.  $\theta$ -EEDA\* performs random selection rather than density based selection in the last layer for fair comparison. Therefore, the results

indicate that the perpendicular distance between a solution and the  $m$ -dimension hyper-plane is more effective in guiding the search.

#### 4.5.4 Parameter $\theta$ sensitivity analysis

In this section, we investigate the influence of parameter  $\theta$  on the performance of the proposed algorithm. Variation of  $\theta$  would affect selection of  $\theta$ -EEDA in the environmental selection phase. So in our experiments, we are to show influence of parameter  $\theta$  by running  $\theta$ -EEDA on the DTLZ test problem in terms of IGD and HV indicators. According to the observation in [18],  $\theta = 5.0$  can achieve better performance on almost all test problems. So we only consider three situations: (1)  $\theta = 0$  (2)  $\theta = 5.0$  (3)  $\theta = 10^6$

Figures 10 and 11 present how the performance of  $\theta$ -EDEA varies with different value of parameter  $\theta$  on DTLZ1-4 problems according to IGD and HV for 20 independent runs. From these figures, we can find that when  $\theta$  is set 5.0,  $\theta$ -EDEA has a clear advantage on the DTLZ1-3 with all objectives in terms of two indicators: IGD and HV. However, on the only DTLZ4 problem with all objectives,  $\theta = 5.0$  and  $\theta = 10^6$  achieve a slightly better or similar performance in terms of two indicators. The same observation can also be found in study [18]. In addition,  $\theta = 0$  and  $\theta = 10^6$  are two extreme cases. In the environmental selection phase of  $\theta$ -EDEA, if  $\theta = 10^6$  the search behavior of  $\theta$ -EDEA is more like NSGA-III. If  $\theta = 0$ , the search behavior of  $\theta$ -EDEA only depends on perpendicular distance between a solution and the hyper-plane and density selection. Therefore, the results show that the proposed method plays a vital role in the search process. And it can be summarized that  $\theta = 5.0$  can achieve a better performance on all test problems

## 5 Conclusion and future work

On the MaOPs, one of the challenges is how to balance convergence and diversity. In this paper, we improve  $\theta$  dominance which can emphasize both convergence and diversity and propose a new many-objective evolutionary algorithm called  $\theta$ -EDEA. Our proposed  $\theta$ -EDEA is expected to enhance the convergence and diversity of  $\theta$ -DEA in high-dimension objective space by introducing the perpendicular distance between a solution and the hyper-plane constructed using extreme point on each dimension and inheriting clustering operation and ranking operation.

To investigate strong competitiveness of  $\theta$ -EDEA, we have performed an extensive experimental comparison of  $\theta$ -EDEA with five state-of-the-art algorithms, which cover grid division, reference point, decomposition, indicator and new dominance relation based approach, on two widely used DTLZ and WFG test suits. In terms of two performance indicators IGD and HV, our proposed  $\theta$ -EDEA performs significantly better than peer algorithms on most test problems and it is compared favorably with five state-of-the-art many-objective optimizers. To demonstrate clearly this, we also present some visual evidence of superior performance in terms of performance score and diversity. To verify effectiveness of enhanced  $\theta$  dominance, we compare it with  $\theta$  dominance in the same condition through computational experiments. Moreover, we show the influence of parameter  $\theta$  with different value in this paper.

Although many MOEAs have been developed to solve both MOPs and MaOPs, more studies are to be carried out in the future. First, some fundamental studies should be paid more attention. For instance, in single-objective

optimization field, the study in [51] emphasizes that one optimization algorithm can be enhanced by another algorithm by understanding an algorithmic linking between them and then borrowing important operators from one to the other. In multi-objective optimization field, the fundamental studies [39, 51] also should be investigated. Second,  $\theta$ -EDEA should be utilized to solve the constrained optimization problem. Some constraint-handling strategies, such as inverse parabolic confined method and inverse parabolic spread method that they both base on parent-centric and inverse parabolic probability (IP) distribution in [40], can be incorporated to  $\theta$ -EDEA for solving constrained many-objective optimization problem. Finally, we can introduce a new strategy to  $\theta$ -EDEA, like model based estimation of distribution method [52, 53], heuristic search method [54, 55], so as to further improve its performance.

**Acknowledgments** This work is supported by Natural Science Foundation of China under Grant No. 61472375 and No. 41571403, Joint Funds of Equipment Pre-Research and Ministry of Education of China under Grant No. 6141A02022320.

## References

1. Li B, Li J, Tang K, Yao X (2015) Many-objective evolutionary algorithms: a survey. *ACM Comput Surv* 48(1):35. Article 13
2. Amarjeet, Chhabra JK (2015) Improving package structure of object-oriented software using multi-objective optimization and weightedclass connections. *Journal of King Saud University – Computer and Information Sciences*
3. Mkaouer W, Kessentini M, Shaout A, Koligheu P, Bechikh S, Deband K, Ouni A (2015) Many-objective software modularization using NSGA-III. *ACM Trans Softw Eng Methodol* 24(3):45. Article 17
4. Fu G, Kapelan Z, Kasprzyk JR, Reed P (2013) Optimal design of water distribution systems using many-objective visual analytics. *J Water Resour Plan Manage* 139(6):624–633
5. Süllflow A, Drechsler N, Drechsler R (2007) Robust multi-objective optimization in high dimensional spaces. In: *Proceeding of the evolutionary multi-criterion optimization*. Matsushima, Japan, pp 715–726
6. Yuan Y, Xu H (2015) Multiobjective flexible job shop scheduling using memetic algorithms. *IEEE Trans Autom Sci Eng* 12(1):336–353
7. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput* 6(2):182–197
8. Zitzler E, Laumanns M, Thiele L (2002) SPEA2: Improving the strength Pareto evolutionary algorithm. In: *Proceedings of the evolutionary methods design optimization control application of industrial problem*. Athens, Greece, pp 95–100
9. Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
10. Corne DW, Jerram NR, Knowles JD, Oates MJ (2001) PESA-II: Region-based selection in evolutionary multiobjective optimization. In: *Proceedings of the 3rd annual conference on genetic and evolutionary computation*. San Francisco, CA, USA, pp 283–290

11. Narukawa K, Rodemann T (2012) Examining the performance of evolutionary many-objective optimization algorithms on areal-world application. In: Proceedings of the 6th international conference on genetic evolutionary computation. Kitakyushu, Japan, pp 316–319
12. Lygoe RJ, Cary M, Fleming PJ (2013) A real-world application of a many-objective optimisation complexity reduction process. In: Proceedings of the 7th international conference on evolutionary multi-criterion optimization. Sheffield, U.K., pp 641–655
13. Li K, Deb K, Zhang Q (2015) An evolutionary many-objective optimization algorithm based on dominance and decomposition. *IEEE Trans Evol Comput* 19(5):694–716
14. Deb K, Saxena D Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In: Proceedings of the WCCI-2006, pp. 3352–3360
15. Purshouse RC, Fleming PJ (2007) On the evolutionary optimization of many conflicting objectives. *IEEE Trans Evol Comput* 11(6):770–784
16. Khare V, Yao X, Deb K (2003) Performance scaling of multi-objective evolutionary algorithms. In: Proceeding of the evolutionary multi-criterion optimization. Faro, Portugal, pp 376–390
17. Purshouse RC, Fleming PJ (2003) Evolutionary many-objective optimization: An exploratory analysis. In: Proceedings of the IEEE congress on evolutionary computation, vol. 3. Canberra, ACT, Australia, pp 2066–2073
18. Yuan Y, Xu H, Yao X (2016) A new dominance relation-based evolutionary algorithm for many-objective optimization. *IEEE Trans Evol Comput* 20(1):16–37
19. Laumanns M, Thiele L, Deb K, Zitzler E (2002) Combining convergence and diversity in evolutionary multiobjective optimization. *Evol Comput* 10(3):263–282
20. Hadka D, Reed P (2013) Borg: An auto-adaptive many-objective evolutionary computing framework. *IEEE Evol Comput* 21(2):231–259
21. Yang S, Li M, Liu X, Zheng J (2013) A grid-based evolutionary algorithm for many objective optimization. *IEEE Trans Evol Comput* 17(5):721–736
22. Wang G, Jiang H (2007) Fuzzy-dominance and its application in evolutionary many objective optimization. In: Proceedings of the international conference computer intelligent security workshops. Harbin, China, pp 195–198
23. He Z, Yen GG, Zhang J (2014) Fuzzy-based Pareto optimality for many-objective evolutionary algorithms. *IEEE Trans Evol Comput* 18(2):269–285
24. di Pierro F, Khu S-T, Savic DA (2007) An investigation on preference order ranking scheme for multiobjective evolutionary optimization. *IEEE Trans Evol Comput* 11(1):17–45
25. Li M, Zheng J, Li K, Yuan Q, Shen R (2010) Enhancing diversity for average ranking method in evolutionary many-objective optimization. In: Proceedings of the 11th international conference on parallel problem solving from nature (PPSN). Kraków, Poland, pp 647–656
26. Zou X, Chen Y, Liu M, Kang L (2008) A new evolutionary algorithm for solving many-objective optimization problems. *IEEE Trans Syst Man Cybern B Cybern* 38(5):1402–1412
27. Kukkonen S, Lampinen J (2007) Ranking-dominance and many-objective optimization. In: Proceedings of the IEEE congress on evolutionary computation (CEC). Singapore, pp 3983–3990
28. Asafuddoula M, Tapabrata R, Sarkera R (2015) Decomposition-based evolutionary algorithm for many objective optimization. *IEEE Trans Evol Comput* 19(3):445–460
29. While L, Bradstreet L, Barone L (2012) A fast way of calculating exact hyper volumes. *IEEE Trans Evol Comput* 16(1):86–95
30. Brockhoff D, Wagner T, Trautmann H (2012) On the properties of the R2 indicator. In: Proceedings of the 14th annual conference on genetic and evolutionary computation. Philadelphia, PA, USA, pp 465–472
31. Zitzler E, Künzli S (2004) Indicator-based selection in multi-objective search. In: Proceeding of the parallel problem solving from nature. Springer-Verlag, pp 832–842
32. Bader J, Zitzler E (2011) HypE: an algorithm for fast hypervolume-based many-objective optimization. *IEEE Trans Evol Comput* 19(1):45–76
33. Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA Multi objectives election based on dominated hypervolume. *Eur J Oper Res* 181(3):1653–1669
34. Gomez RH, Coello CAC (2013) MOMBI: a new metaheuristic for many-objective optimization based on the R2 indicator. In: Proceedings of the IEEE congress on evolutionary computation. Cancún, Mexico, pp 2488–2495
35. Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints. *IEEE Trans Evol Comput* 18(4):577–601
36. Cai L, Qu S, Yuan Y, Yao X (2015) A clustering-ranking method for many-objective optimization. *Appl Soft Comput* 35:681–694
37. Wang H, Jiao L, Yao X (2015) Two\_arch2 An improved two-archive algorithm for many-objective optimization. *IEEE Trans Evol Comput* 19(4):524–541
38. Yuan Y, Xu H, Wang B, Zhang B, Yao X (2016) Balancing convergence and diversity in decomposition-based many-objective optimizers. *IEEE Trans Evol Comput* 20(2):180–198
39. Padhye N, Bhardawaj P, Deb K (2013) Improving differential evolution through a unified approach. *J Global Optim* 55(4):771–799
40. Padhye N, Mittal P, Deb K (2015) Feasibility preserving constraint-handling strategies for real parameter evolutionary optimization. *Comput Optim Appl* 62(3):851–890
41. Das I, Dennis J (1998) Normal-boundary intersection: a new method for generating the Pareto surface in nonlinear multicriteria optimization problems. *SIAM J Optimization* 8(3):631–657
42. Deb K, Thiele L, Laumanns M, Zitzler E (2001) Scalable multi-objective optimization test problems. *Inst Commun Inf Technol, ETH Zurich, Zurich, Switzerland, TIK Tech Rep* 112
43. Ishibuchi H, Setoguchi Y, Masuda H, Nojima Y (2016) Performance of decomposition-based many-objective algorithms strongly depends on pareto front shapes. *IEEE Trans Evol Comput* PP(99):1–1
44. Huband S, Hingston P, Barone L, While L (2006) A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Trans Evol Comput* 10(5):477–506
45. Russo LM, Francisco AP (2014) Quick hypervolume. *IEEE Trans Evol Comput* 18(4):481–502
46. Zitzler E, Thiele L (1999) Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput* 3(4):257–271
47. Ishibuchi H, Hitotsuyanagi Y, Tsukamoto N, Nojima Y (2010) Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space. In: Proceedings of the international conference on parallel problem solving from nature PPSN, pp 91–100
48. García S, Molina D, Lozano M, Herrera F (2009) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. *J Heuristics* 15(3):617–644
49. Zitzler E, Thiele L, Laumanns M, Fonseca CM, da Fonseca VG (2003) Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans Evol Comput* 7(2):117–132

50. Wilcoxon F (1945) Individual comparisons by ranking methods. *Biom Bull* 80–83
51. Deb K, Padhye N (2014) Enhancing performance of particle swarm optimization through an algorithmic link with genetic algorithms. *Comput Optim Appl* 57(3):761–794
52. Campigotto P, Passerini A, Battiti R (2014) Active learning of Pareto fronts. *IEEE Trans Neural Netw Learn Syst* 25(3):506–519
53. Cheng R, Jin Y, Narukawa K, Sendhoff B (2015) A multiobjective evolutionary algorithm using gaussian process-based inverse modeling. *IEEE Trans Evol Comput* 19(6):838–856
54. Rakshit P, Konar A (2015) Extending multi-objective differential evolution for optimization in presence of noise. *Inf Sci* 305:56–76
55. Figueiredo EMN, Ludermir TB, Bastos-Filho CJA (2016) Many objective particle swarm optimization. *Inf Sci* 374:115–134