



# An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions

Zhun Fan<sup>1</sup> · Wenji Li<sup>1</sup> · Xinye Cai<sup>2</sup> · Han Huang<sup>3</sup> · Yi Fang<sup>1</sup> · Yugen You<sup>1</sup> · Jiajie Mo<sup>2</sup> · Caimin Wei<sup>4</sup> · Erik Goodman<sup>5</sup>

© Springer-Verlag GmbH Germany, part of Springer Nature 2019

## Abstract

This paper proposes an improved epsilon constraint-handling mechanism and combines it with a decomposition-based multi-objective evolutionary algorithm (MOEA/D) to solve constrained multi-objective optimization problems (CMOPs). The proposed constrained multi-objective evolutionary algorithm (CMOEA) is named MOEA/D-IEpsilon. It adjusts the epsilon level dynamically according to the ratio of feasible to total solutions in the current population. In order to evaluate the performance of MOEA/D-IEpsilon, a new set of CMOPs with two and three objectives is designed, having large infeasible regions (relative to the feasible regions), and they are called LIR-CMOPs. Then, the 14 benchmarks, including LIR-CMOP1-14, are used to test MOEA/D-IEpsilon and four other decomposition-based CMOEAs, including MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP and CMOEA/D. The experimental results indicate that MOEA/D-IEpsilon is significantly better than the other four CMOEAs on all of the test instances, which shows that MOEA/D-IEpsilon is more suitable for solving CMOPs with large infeasible regions. Furthermore, a real-world problem, namely the robot gripper optimization problem, is used to test the five CMOEAs. The experimental results demonstrate that MOEA/D-IEpsilon also outperforms the other four CMOEAs on this problem.

**Keywords** Constrained multi-objective evolutionary algorithms · Epsilon constraint handling · Constrained multi-objective optimization · Robot gripper optimization

## 1 Introduction

Real-world optimization problems usually involve the simultaneous optimization of multiple conflicting objectives with

a number of constraints. Without loss of generality, a CMOP considered in this paper is defined as follows (Deb 2001):

$$\begin{cases} \text{minimize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \\ \text{subject to} & g_i(\mathbf{x}) \geq 0, \quad i = 1, \dots, q \\ & h_j(\mathbf{x}) = 0, \quad j = 1, \dots, p \\ & \mathbf{x} \in \mathbb{R}^n \end{cases} \quad (1)$$

Communicated by V. Loia.

✉ Xinye Cai  
Xinye@nuaa.edu.cn

- <sup>1</sup> Department of Electronic Engineering, Shantou University, Shantou 515063, Guangdong, China
- <sup>2</sup> College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, Jiangsu, China
- <sup>3</sup> School of Software Engineering, South China University of Technology, Guangzhou 515063, Guangdong, China
- <sup>4</sup> Department of Mathematics, Shantou University, Shantou 515063, Guangdong, China
- <sup>5</sup> BEACON Center for the Study of Evolution in Action, Michigan State University, East Lansing, MI, USA

where  $F(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x}))^T \in \mathbb{R}^m$  is an  $m$ -dimensional objective vector,  $g_i(\mathbf{x}) \geq 0$  is an inequality constraint and  $h_j(\mathbf{x}) = 0$  is an equality constraint.  $\mathbf{x} \in \mathbb{R}^n$  is an  $n$ -dimensional decision vector. The feasible region  $S$  is defined as the set  $\{\mathbf{x} | g_i(\mathbf{x}) \geq 0, i = 1, \dots, q$  and  $h_j(\mathbf{x}) = 0, j = 1, \dots, p\}$ .

In CMOPs, there is usually more than one constraint. The overall constraint violation is a widely used approach to deal with constraint violations, as it summarizes them into a single scalar, as follows:

$$\phi(\mathbf{x}) = \sum_{i=1}^q |\min(g_i(\mathbf{x}), 0)| + \sum_{j=1}^p |h_j(\mathbf{x})| \quad (2)$$

If  $\phi(\mathbf{x}) = 0$ ,  $\mathbf{x}$  is feasible; otherwise, it is infeasible. Any solution in set  $S$  is feasible, and for any two solutions  $\mathbf{x}^1 \in S$  and  $\mathbf{x}^2 \in S$ ,  $\mathbf{x}^1$  is said to dominate  $\mathbf{x}^2$  if  $f_i(\mathbf{x}^1) \leq f_i(\mathbf{x}^2)$  for each  $i \in \{1, \dots, m\}$  and  $f_j(\mathbf{x}^1) < f_j(\mathbf{x}^2)$  for at least one  $j \in \{1, \dots, m\}$ , denoted as  $\mathbf{x}^1 \preceq \mathbf{x}^2$ . For a solution  $\mathbf{x}^* \in S$ , if there is no other solution in  $S$  dominating  $\mathbf{x}^*$ , then  $\mathbf{x}^*$  is called a Pareto optimal solution. A set including all of the Pareto optimal solutions is called a Pareto optimal set ( $PS$ ). Mapping the  $PS$  into the objective space obtains a set of objective vectors, which is called a Pareto optimal front ( $PF$ ), and  $PF = \{F(\mathbf{x}) | \mathbf{x} \in PS\}$ .

CMOEAs aim to find a representative set of Pareto optimal solutions. They have to tackle the multiple conflicting objectives with a number of constraints simultaneously and to maintain a good balance between convergence and diversity of the achieved solutions. In CMOEAs, there are two basic components: One is the constraint-handling mechanism and the other is the multi-objective evolutionary algorithm (MOEA).

In terms of constraint handling, many methods have been proposed in evolutionary optimization (Cai et al. 2013; Hu et al. 2013). They can be roughly divided into penalty function methods, special representations and operators, repair methods, separation of objectives and constraints and hybrid methods (Coello Coello 2002). The penalty function method is widely used due to its simplicity in the constraint handling (Runarsson and Yao 2005). However, the ideal penalty factors cannot be known in advance for an arbitrary CMOP, and tuning the penalty factors can be a very tedious task.

In recent years, a number of other constraint-handling techniques have had a relatively high impact on evolutionary optimization, including feasibility rules, stochastic ranking,  $\varepsilon$ -constrained method, novel penalty functions, novel special operators, multi-objective concepts and ensemble of constraint-handling techniques (Mezura-Montes and Coello Coello 2011). However, most of them aim to solve constrained scalar optimization problems when they are first proposed.

MOEAs can be classified into three different types according to their selection approaches. The first type is non-dominated-based methods, and representative examples include NSGA-II (Deb et al. 2002), PAES-II (Corne et al. 2001), SPEA-II (Zitzler et al. 2001), NSGA-III (Deb and Jain 2014) and so on. The second type is decomposition-based approaches, and typical examples include MOEA/D (Zhang and Li 2007), MOEA/D-DE (Li and Zhang 2009), EAG-MOEA/D (Cai et al. 2015), MOEA/D-M2M (Liu et al. 2014), MOEA/D-SAS (Cai et al. 2017) and so on. Currently, MOEA/D is a popular algorithm to solve unconstrained

multi-objective optimization problems (MOPs). MOEA/D decomposes a MOP into many scalar optimization subproblems and optimizes them simultaneously in a collaborative way. The last type is indicator-based methods. This type of MOEAs selects solutions based on the improvement of a performance metric. Representative methods include IBEA (Zitzler and Künzli 2004), SMS-EMOA (Beume et al. 2007), HypE (Bader and Zitzler 2011), FV-MOEA (Jiang et al. 2015) and so on.

There are two commonly used test suites of CMOPs, including CTP (Deb 2001) and CF test instances (Zhang et al. 2008). For CTP1-CTP5 and CF1-CF10, the feasible regions are relatively large, and a CMOEA can approximate their PFs without encountering any infeasible obstacles during the entire evolutionary process. Thus, CTP1-5 and CF1-10 are not good test problems to evaluate the performance of constraint-handling mechanisms. For the remaining test problems CTP6-8, the feasible regions are relatively large, and the population of a CMOEA can reach these regions with high probability. Thus, CTP and CF test suites cannot effectively measure the performance of constraint-handling techniques. When solving CTP (Deb 2001) and CF (Zhang et al. 2008) test instances, the constraint dominance principle (CDP) (Deb et al. 2002) is good enough to handle the constraints.

To overcome the shortcomings of the CTP and CF test suites discussed above, we propose a set of new CMOPs (named LIR-CMOP1-14). Each of them has a number of large infeasible regions, and the feasible regions are relatively small. The population of a CMOEA cannot easily discover these small feasible regions, which brings new challenges to the existing CMOEAs. In fact, many real-world optimization problems also have this characteristic. For example, the robot gripper optimization problem considered in this paper has large infeasible regions as illustrated in Sect. 6. Thus, it has important significance in practice to design specific mechanisms for solving CMOPs with large infeasible regions.

In this paper, we propose an improved  $\varepsilon$ -constrained version of MOEA/D to deal with CMOPs. Compared with the original  $\varepsilon$ -constrained method (Takahama and Sakai 2006), the proposed method can keep a good balance in the search between the feasible and infeasible regions (Črepinšek et al. 2013). It uses the information of the ratio of feasible to total solutions of the current population to dynamically balance the exploration between the feasible regions and infeasible regions.

The remainder of the paper is organized as follows. Section 2 introduces the related work on MOEA/D and the existing CMOEAs based on MOEA/D. Section 3 illustrates the improved epsilon constraint-handling method as here embedded in MOEA/D. Section 4 designs a set of new CMOPs (LIR-CMOPs) with large infeasible regions. Section 5 describes a comprehensive set of experiments to com-

pare the proposed CMOEA (MOEA/D-IEpsilon) with four other CMOEAs, including MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP and CMOEA/D. In Sect. 6, a robot gripper optimization problem is used to test MOEA/D-IEpsilon and the other four CMOEAs. Finally, Sect. 7 presents the conclusions.

## 2 Related work

### 2.1 MOEA/D

MOEA/D (Zhang and Li 2007) decomposes a MOP into a number of scalar optimization subproblems and optimizes them simultaneously in a collaborative way. Each subproblem is defined by a decomposition function with a weight vector  $\lambda^i$ . In MOEA/D, a set of  $N$  uniformly spread weight vectors  $\lambda^1, \dots, \lambda^N$  are adopted to formulate  $N$  subproblems. The weight vectors  $\lambda^i$  satisfy  $\sum_{k=1}^m \lambda_k^i = 1$  and  $\lambda_k^i \geq 0$  for each  $k \in \{1, \dots, m\}$ . In terms of decomposition methods, there are three commonly used approaches, including weighted sum (Miettinen 1999), Tchebycheff (Miettinen 1999) and boundary intersection approaches (Zhang and Li 2007).

In the weighted sum approach, each subproblem is defined by summing each objective weighted by a different weight. The  $j$ th subproblem with the weighted sum decomposition method is defined as follows:

$$\begin{aligned} \text{minimize } g^{te}(\mathbf{x}|\lambda) &= \sum_{i=1}^m \lambda_i^j f_i(\mathbf{x}) \\ \text{subject to } \mathbf{x} &\in S \end{aligned} \tag{3}$$

For a minimizing MOP, in the case of a convex PF, the weighted sum approach can work well. However, if the PF is non-convex, only a part of PF can be found by this approach.

In the Tchebycheff decomposition method, the  $j$ th subproblem is defined as follows:

$$\begin{aligned} \text{minimize } g^{te}(\mathbf{x}|\lambda, z^*) &= \max_{1 \leq i \leq m} \{\lambda_i^j |f_i(\mathbf{x}) - z_i^*|\} \\ \text{subject to } \mathbf{x} &\in S \end{aligned} \tag{4}$$

where  $z^* = (z_1^*, \dots, z_m^*)$  is the ideal point, and  $z_i^* = \min\{f_i(\mathbf{x}) | \mathbf{x} \in S\}$ . The Tchebycheff method is a widely used decomposition approach. It can approximate both concave and convex parts of PFs.

In the boundary intersection approach, two distances  $d_1$  and  $d_2$  are defined to evaluate the convergence and diversity, respectively. The  $j$ th subproblem is defined as follows:

$$\begin{aligned} \text{minimize } g^{pbi}(\mathbf{x}|\lambda^j, z^*) &= d_1 + \theta d_2 \\ \text{subject to } \mathbf{x} &\in S \\ \text{where } d_1 &= \frac{\|(F(\mathbf{x}) - z^*)^T \lambda^j\|}{\|\lambda^j\|} \\ d_2 &= \left\| (F(\mathbf{x}) - z^*) - d_1 \left( \frac{\lambda^j}{\|\lambda^j\|} \right) \right\| \end{aligned} \tag{5}$$

The boundary intersection method is able to solve MOPs with any shape of PFs. However, the penalty factor  $\theta$  must be set in advance.

### 2.2 Decomposition-based CMOEAs

In decomposition-based CMOEAs, a CMOP is decomposed into a set of constrained scalar optimization subproblems, and these subproblems are solved in a collaborative way simultaneously. Representative methods include CMOEA/D (Asafuddoula et al. 2012), MOEA/D-Epsilon (Yang et al. 2014), MOEA/D-CDP (Jan and Khanum 2013) and MOEA/D-SR (Jan and Khanum 2013).

CMOEA/D (Asafuddoula et al. 2012) embeds an epsilon constraint-handling approach into MOEA/D, and the epsilon value is set adaptively. To be more specific, the epsilon level is set to  $CV_{mean} * RFS$ .  $CV_{mean}$  denotes the mean value of the overall constraint violation in the current population, and  $RFS = \frac{\text{Number of feasible solutions}}{\text{Population size}}$  denotes the ratio of feasible to total solutions in the current population. For two solutions, if their overall constraint violations are both less than  $CV_{mean} * RFS$  or their overall constraint violations are equal, the one with the better aggregation value is selected. Otherwise, the one with the smaller overall constraint violation is selected.

MOEA/D-Epsilon (Yang et al. 2014) also adopts the epsilon method to handle constraints. Unlike CMOEA/D, the epsilon value in MOEA/D-Epsilon is set dynamically with the increase of generation counter  $K$ . The detailed setting of the epsilon value can be found in Takahama and Sakai (2006).

MOEA/D-CDP (Jan and Khanum 2013) adopts CDP (Deb et al. 2002) to deal with constraints in the framework of MOEA/D. There are three basic rules to select solutions. For two feasible solutions, the one with the better aggregation value is selected. For two infeasible solutions, the one with the smaller overall constraint violation is selected. For a feasible and an infeasible solution, the feasible one is selected.

MOEA/D-SR (Jan and Khanum 2013) embeds the stochastic ranking method (SR) (Runarsson and Yao 2000) in MOEA/D to deal with constraints. A parameter  $p_f \in [0, 1]$  is set to balance the selection between the objectives and the constraints in MOEA/D-SR. For two solutions, if a random number is less than  $p_f$ , the one with the better aggregation value is selected into the next generation. If the random number is greater than  $p_f$ , the solutions selection is similar to that

of MOEA/D-CDP. In the case of  $p_f = 0$ , MOEA/D-SR is equivalent to MOEA/D-CDP.

In summary, CMOEA/D and MOEA/D-Epsilon both adopt the epsilon constraint-handling approach to solve CMOPs. To get across large infeasible regions,  $\varepsilon$  should be increased at sometimes and be greater than the maximum overall constraint violation in the current population. However, in CMOEA/D,  $\varepsilon$  is always less than or equal to  $CV_{mean}$ , and in MOEA/D-Epsilon,  $\varepsilon$  is always decreasing during the evolutionary process. In MOEA/D-CDP, feasible solutions are always better than infeasible solutions. Thus, the infeasible solutions which can help to get across large infeasible regions are difficult to survive. MOEA/D-SR applies a parameter  $p_f$  to balance the searching between the feasible and infeasible regions. In order to get across large infeasible regions,  $p_f$  should be set dynamically. However,  $p_f$  is a static parameter in MOEA/D-SR. To overcome the shortcomings of the four decomposition-based CMOEAs discussed above, an improved epsilon constraint-handling method embedded in MOEA/D is proposed.

### 3 The proposed method

In this section, the concept of epsilon level comparison, the original epsilon level setting method and the improved epsilon level setting approach are described.

#### 3.1 Epsilon level comparison

In the epsilon constraint handling approach (Takahama and Sakai 2006), the relaxation of constraints is controlled by the epsilon level  $\varepsilon$ . For two solutions  $\mathbf{x}^1$  and  $\mathbf{x}^2$ , their overall constraint violations are  $\phi^1$  and  $\phi^2$ . Then, for any  $\varepsilon$  satisfying  $\varepsilon \geq 0$ , the epsilon level comparison  $\preceq_\varepsilon$  is defined as follows:

$$(\mathbf{x}^1, \phi^1) \preceq_\varepsilon (\mathbf{x}^2, \phi^2) \Leftrightarrow \begin{cases} \mathbf{x}^1 \preceq \mathbf{x}^2, & \text{if } \phi^1, \phi^2 \leq \varepsilon \\ \mathbf{x}^1 \preceq \mathbf{x}^2, & \text{if } \phi^1 = \phi^2 \\ \phi^1 < \phi^2, & \text{otherwise} \end{cases} \quad (6)$$

In Eq. (6), the epsilon comparison approach is equivalent to CDP (Deb et al. 2002) when  $\varepsilon = 0$ . In the case of  $\varepsilon = \infty$ , it does not consider any constraints. In other words, the comparison between any two solutions is based on their non-dominated ranks on objectives when  $\varepsilon = \infty$ .

#### 3.2 Epsilon Level Setting

In the epsilon constraint-handling method, the setting of  $\varepsilon$  is quite critical. In Takahama and Sakai (2006), an epsilon level setting method is suggested as follows:

$$\varepsilon(k) = \begin{cases} \varepsilon(0)(1 - \frac{k}{T_c})^{cp}, & 0 < k < T_c, \varepsilon(0) = \phi(\mathbf{x}^\theta) \\ 0, & k \geq T_c \end{cases} \quad (7)$$

where  $\mathbf{x}^\theta$  is the top  $\theta$ th individual of the initial population sorted by overall constraint violations in a descending order.  $cp$  is to control the speed of reducing relaxation of constraints.  $\varepsilon(k)$  is updated until the generation counter  $k$  reaches the control generation  $T_c$ . When  $k \geq T_c$ ,  $\varepsilon(k) = 0$ . The recommended parameter ranges in Takahama and Sakai (2006) are listed as follows:  $\theta = (0.05 * N)$ ,  $cp \in [2, 10]$  and  $T_c \in [0.1T_{max}, 0.8T_{max}]$ .  $N$  denotes the population size, and  $T_{max}$  represents the maximum evolutionary generation.

#### 3.3 Improved epsilon level setting

The setting of  $\varepsilon(k)$  in Eq.(7) is always decreasing during the evolutionary process, which may not be suitable to solve CMOPs with large infeasible regions. To overcome this problem, an improved epsilon setting approach is suggested as follows:

$$\varepsilon(k) = \begin{cases} \text{rule1: } \phi(\mathbf{x}^\theta), & \text{if } k = 0 \\ \text{rule2: } (1 - \tau)\varepsilon(k - 1), & \text{if } r_k < \alpha \text{ and } k < T_c \\ \text{rule3: } (1 + \tau)\phi_{max}, & \text{if } r_k \geq \alpha \text{ and } k < T_c \\ \text{rule4: } 0, & \text{if } k \geq T_c \end{cases} \quad (8)$$

where  $\phi_k(\mathbf{x}^\theta)$  is the overall constraint violation of the top  $\theta$ th individual in the initial population and  $r_k$  is the ratio of feasible to total solutions in the  $k$ th generation.  $\tau$  ranges between 0 and 1 and has two functions. One is to control the speed of reducing the relaxation of constraints, and the other is to control the scale factor multiplied by the maximum overall constraint violation.  $\alpha$  is to control the searching preference between the feasible and infeasible regions, and  $\alpha \in [0, 1]$ .  $\phi_{max}$  is the maximum overall constraint violation found so far.

The  $\varepsilon(0)$  setting method in Eq. 8 is sometimes the same as that in Eq. 7. If  $\varepsilon(0) = 0$ ,  $\varepsilon(k)$  in Eq. 7 is identically equal to zero, which tends to hinder a CMOEA’s exploration of the infeasible regions. However,  $\varepsilon(k)$  in Eq. 8 is not identically equal to zero when  $\varepsilon(0) = 0$  according to the third rule of the proposed epsilon setting approach.

In the case  $k > 0$ , three rules are adopted to control the value of  $\varepsilon$  in Eq. 8. *Rule2* is adopted to strengthen the searching in the feasible regions. *Rule3* is used to strengthen the exploration in the infeasible regions. The last *rule4* is same as in the CDP (Deb et al. 2002) constraint-handling method.

Two parameters  $k$  and  $r_k$  are applied to choose the right control rule for  $\varepsilon(k)$ . If  $k < T_c$  and  $r_k < \alpha$ , *rule2* for setting  $\varepsilon(k)$  is adopted. In this circumstance,  $\varepsilon(k)$  is set to  $(1 - \tau)\varepsilon(k - 1)$ , which has an exponential decreasing rate.

It has a faster descent rate than the epsilon setting in Eq. (7), which can help to enhance the searching in the feasible regions more effectively. If  $k < T_c$  and  $r_k \geq \alpha$ , *rule3* for setting  $\varepsilon(k)$  is applied. In this situation, most solutions are feasible. Thus, strengthening the exploration in the infeasible regions may help a CMOEAs to get across a number of large infeasible regions. In *rule3*,  $\varepsilon(k) = (1 + \tau)\phi_{max}$ , which strengthens the exploration in the infeasible regions. Thus, the improved epsilon method has the balanced ability to explore the feasible and infeasible regions simultaneously.

$\alpha$  is a critical parameter to balance the searching between the feasible and infeasible regions. If the RFS  $r_k$  is less than  $\alpha$ , *rule2* is adopted to enhance the exploration in the feasible regions. Otherwise, *rule3* is applied to enhance the exploration in the infeasible regions. Thus, the proposed epsilon constraint method can keep a good balance of exploration between the feasible and infeasible regions. It utilizes the RFS to dynamically balance the exploration between the feasible regions and infeasible regions.

Compared with the  $\varepsilon$  setting in Eq. (7), the proposed method in Eq. (8) has the ability to increase  $\varepsilon(k)$  during the evolutionary process, which can help to solve CMOPs with large infeasible regions.

In the case of  $k \geq T_c$ , *rule4* is applied. In this situation,  $\varepsilon(k) = 0$ , and the epsilon constraint-handling method exerts the highest selection pressure toward the feasible regions.

### 3.4 Embedding the improved epsilon method in MOEA/D

The proposed MOEA/D-IEpsilon integrates the improved epsilon constraint-handling method in Eq. 8 into the framework of MOEA/D. In MOEA/D-IEpsilon, a CMOP is decomposed into a number of constrained scalar subproblems, and these subproblems are optimized simultaneously in a collaborative way. In our experimental studies, the Tchebycheff approach is adopted, and its detailed definition is listed in Eq. (4).

For a given weight vector  $\lambda$ , there exists an optimal solution of Eq. (4), and this optimal solution is also a Pareto optimal solution of Eq. (1). Therefore, we can achieve different Pareto optimal solutions of Eq. (1) by setting different weight vectors.

The pseudocode of MOEA/D-IEpsilon is listed in Algorithm 1. It is almost the same as that of MOEA/D, except for the method of subproblem updating. Lines 1–6 initialize a number of parameters in MOEA/D-IEpsilon. First, a CMOP is decomposed into  $N$  subproblems which are associated with  $\lambda^1, \dots, \lambda^N$ . Then, the population  $P$ , the initial epsilon value  $\varepsilon(0)$ , the proportion of feasible solutions in the  $k$ th generation  $r_k$ , the maximum overall constraint violation  $\phi_{max}$  found so far, the ideal point  $z^*$  and the neighbor indexes  $B(i)$  are initialized.

#### Algorithm 1: MOEA/D-IEpsilon

```

Input:
 $N$ : the number of subproblems.
 $FES_{max}$ : the maximum function evaluations.
 $T_{max}$ : the maximum generation,  $T_{max} = FES_{max}/N$ 
 $N$  weight vectors:  $\lambda^1, \dots, \lambda^N$ .
 $T$ : the size of the neighborhood.
 $\delta$ : the selecting probability from neighbors.
 $n_r$ : the maximal number of solutions replaced by a child.
Parameters  $\tau, \alpha, T_c$  and  $\theta$  in Eq. (8).
Output:
 $NS$ : a set of feasible non-dominated solutions
1 Decompose a CMOP into  $N$  subproblems associated with  $\lambda^1, \dots, \lambda^N$ .
2 Generate an initial population  $P = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ .
3 Initialize  $\varepsilon(0), r_k$  and  $\phi_{max}$  according to Eq. (8).
4 Initialize the ideal point  $z^* = (z_1, \dots, z_m)$ .
5 For each  $i = 1, \dots, N$ , set  $B(i) = \{i_1, \dots, i_T\}$ , where  $\lambda^{i_1}, \dots, \lambda^{i_T}$  are the  $T$  closest weight vectors to  $\lambda^i$ .
6 Set  $FES = 0$  and  $k = 0$ .
7 while  $FES \leq FES_{max}$  do
8   if  $k > 0$  then
9      $\varepsilon(k) = \text{SetEpsilon}(\tau, r_k, \alpha, \phi_{max}, T_c, k)$ 
10  end
11  Generate a random permutation  $rp$  from  $\{1, \dots, N\}$ .
12  for  $i \leftarrow 1$  to  $N$  do
13    Generate a random number  $r \in [0, 1]$ 
14    Set  $j = rp(i)$ .
15    if  $r < \delta$  then
16       $S = B(j)$ 
17    else
18       $S = \{1, \dots, N\}$ 
19    end
20    Generate  $\mathbf{y}^j$  through the DE operator.
21    Perform polynomial mutation on  $\mathbf{y}^j$ .
22    Evaluate the newly generated solution  $\mathbf{y}^j$ .
23     $FES = FES + 1$ ; update  $\phi_{max}$ 
24    for  $t \leftarrow 1$  to  $m$  do
25      if  $z_t^* > f_t(\mathbf{y}^j)$  then  $z_t^* = f_t(\mathbf{y}^j)$ ;
26    end
27    Set  $c = 0$ .
28    while  $c < n_r$  or  $S \neq \emptyset$  do
29      select an index  $j$  from  $S$  randomly.
30       $result = \text{UpdateSubproblems}(\mathbf{x}^j, \mathbf{y}^j, \varepsilon(k))$ 
31      if  $result == true$  then  $c = c + 1$ ;
32       $S = S \setminus \{j\}$ 
33    end
34  end
35   $k = k + 1$ ; update  $r_k$ 
36   $NS = \text{NondominatedSelect}(NS \cup P)$ 
37 end

```

Lines 8–10 set the value of  $\varepsilon(k)$  according to Algorithm 3. Lines 11–26 generate a set of new solutions and update  $z^*$ ,  $FES$  and  $\phi_{max}$ . To be more specific, a set of solutions which may be updated by a newly generated solution  $\mathbf{y}^j$  is selected (lines 11–19). In line 20, the differential evolution (DE) crossover is adopted to generate a new solution  $\mathbf{y}^j$ . The polynomial mutation operator is executed to mutate  $\mathbf{y}^j$  in line 21. In line 22, the solution  $\mathbf{y}^j$  is evaluated.  $FES$  and  $\phi_{max}$

are updated in line 23. The ideal point  $z^*$  is updated (lines 24–26).

Lines 27–33 implement the updating process of subproblems. In line 30, the subproblems are updated based on the improved epsilon constraint-handling approach, and the detailed procedures are listed in Algorithm 2. Finally, a set of non-dominated solutions ( $NS$ ) is selected based on the non-dominated sorting in line 36.

---

#### Algorithm 2: Subproblem Update

---

```

1 Function result = UpdateSubproblems( $\mathbf{x}^j, \mathbf{y}^j, \varepsilon(k)$ )
2   result = false
3   if  $\phi(\mathbf{y}^j) \leq \varepsilon(k)$  and  $\phi(\mathbf{x}^j) \leq \varepsilon(k)$  then
4     if  $g^{te}(\mathbf{y}^j | \lambda^j, z^*) \leq g^{te}(\mathbf{x}^j | \lambda^j, z^*)$  then
5        $\mathbf{x}^j = \mathbf{y}^j$ 
6       result = true
7     end
8   else if  $\phi(\mathbf{y}^j) == \phi(\mathbf{x}^j)$  then
9     if  $g^{te}(\mathbf{y}^j | \lambda^j, z^*) \leq g^{te}(\mathbf{x}^j | \lambda^j, z^*)$  then
10       $\mathbf{x}^j = \mathbf{y}^j$ 
11      result = true
12    end
13  else if  $\phi(\mathbf{y}^j) < \phi(\mathbf{x}^j)$  then
14     $\mathbf{x}^j = \mathbf{y}^j$ 
15    result = true
16  end
17  return result
18 end

```

---

In Algorithm 2, there are three basic rules to update a subproblem. For two solutions  $\mathbf{x}^j$  and  $\mathbf{y}^j$ , if their overall constraint violations are less than or equal to  $\varepsilon(k)$ , and  $\mathbf{y}^j$  has a smaller aggregation value (the value of the decomposition function) than that of  $\mathbf{x}^j$ , then  $\mathbf{x}^j$  is replaced by  $\mathbf{y}^j$  (lines 3–7). If  $\mathbf{x}^j$  and  $\mathbf{y}^j$  have the same overall constraint violation, and  $\mathbf{y}^j$  has a smaller aggregation value than that of  $\mathbf{x}^j$ , then  $\mathbf{x}^j$  is replaced by  $\mathbf{y}^j$  (lines 8–12). Otherwise, if  $\mathbf{y}^j$  has a smaller overall constraint violation than that of  $\mathbf{x}^j$ , then  $\mathbf{x}^j$  is replaced by  $\mathbf{y}^j$  (lines 13–14). When the subproblem is updated, the function  $UpdateSubproblems(\mathbf{x}^j, \mathbf{y}^j, \varepsilon(k))$  returns *true*; otherwise, it returns *false*.

---

#### Algorithm 3: Set Epsilon

---

```

1 Function  $\varepsilon(k) = SetEpsilon(\tau, r_k, \alpha, \phi_{max}, T_c, k)$ 
2   if  $r_k < \alpha$  and  $k < T_c$  then
3      $\varepsilon(k) = (1 - \tau)\varepsilon(k - 1)$ 
4   else if  $r_k \geq \alpha$  and  $k < T_c$  then
5      $\varepsilon(k) = (1 + \tau)\phi_{max}$ 
6   else if  $k \geq T_c$  then
7      $\varepsilon(k) = 0$ 
8   end
9   return  $\varepsilon(k)$ 
10 end

```

---

In Algorithm 3, the value of  $\varepsilon(k)$  is set according to the last three rules in Eq. (8). If the proportion of feasible solutions  $r_k$  is less than the predefined parameter  $\alpha$ , and the generation counter  $k$  is less than the control generation  $T_c$ ,  $\varepsilon(k)$  is set to  $(1 - \tau)\varepsilon(k - 1)$  (lines 2–3). In lines 4–5, if the proportion of feasible solutions  $r_k$  is equal to or greater than the predefined parameter  $\alpha$ , and the generation counter  $k$  is less than the control generation  $T_c$ ,  $\varepsilon(k)$  is set to  $(1 + \tau)\phi_{max}$ , which corresponds to the *rule3* in Eq. (8). When the generation counter  $k$  is equal to or greater than the control generation  $T_c$ ,  $\varepsilon(k)$  is set to zero. This is realized in lines 6–7, which reflects the *rule4* in Eq. (8).

## 4 Test instances

To evaluate the performance of the proposed MOEA/D-Iepsilon, a set of new CMOPs with large infeasible regions (named LIR-CMOPs) is designed according to our previous work (Fan et al. 2016). In terms of constraint functions, all of them have large infeasible regions. In term of objective functions, there are two components: shape functions and distance functions (Huband et al. 2006).

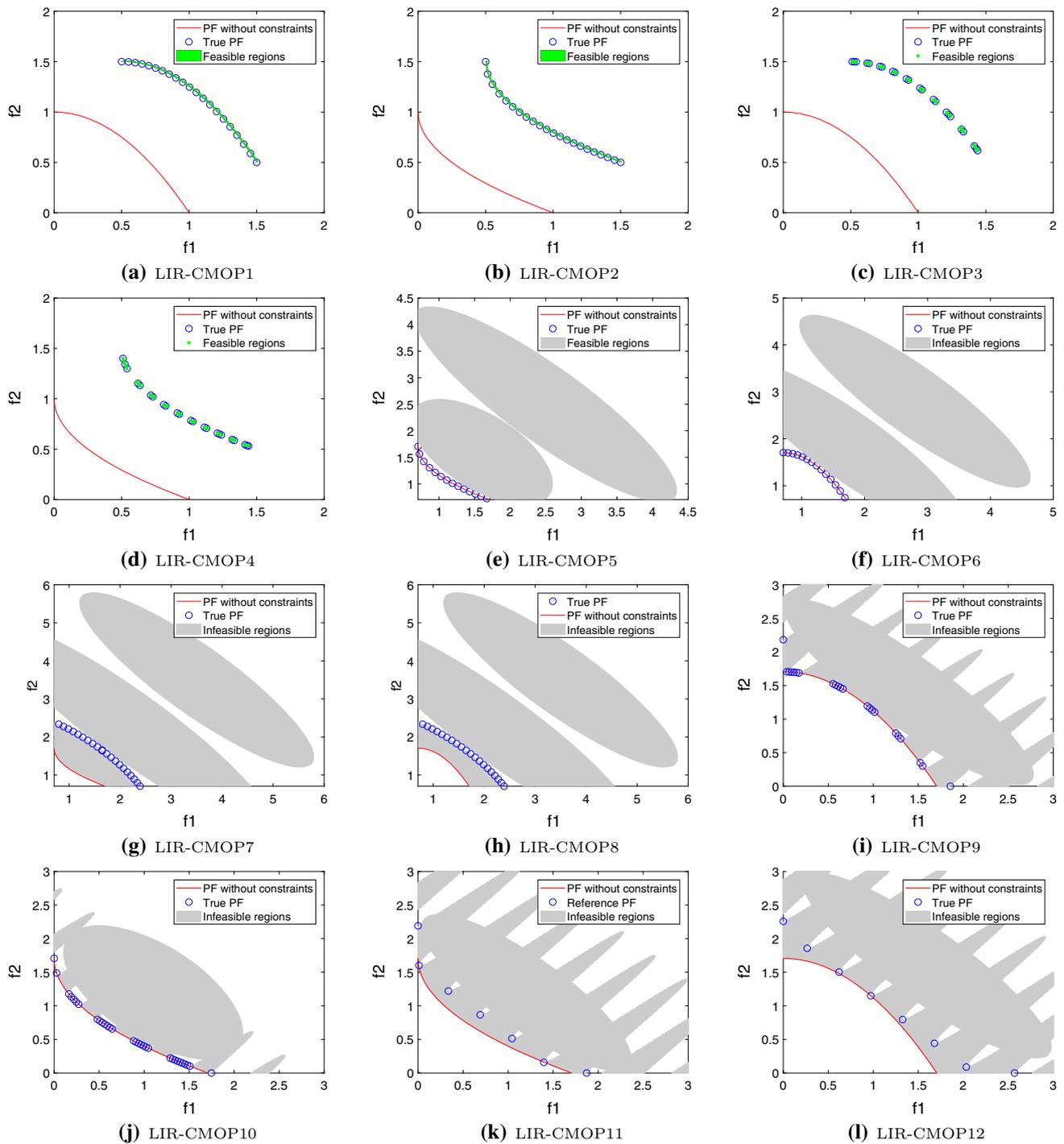
The shape functions are applied to set the shape of the PFs. In the LIR-CMOP test suite, two types of shape functions, including both convex and concave shapes, are designed. Distance functions are adopted that test the convergence performance of a CMOP. In LIR-CMOP5-14, the distance functions are multiplied by a scale factor, which increases difficulty of convergence. The detailed definitions of LIR-CMOPs are listed in “Appendix.”

In this test suite, four test problems, including LIR-CMOP1-4, have large infeasible regions. Figure 1a–d plots the feasible regions of LIR-CMOP1-4, respectively. It can be seen that the feasible regions of these test instances are very small. In other words, there are a number of large infeasible regions.

LIR-CMOP5 and LIR-CMOP6 have convex and concave PFs, respectively, as shown in Fig. 1e–f, and their PFs are the same as those of their unconstrained counterparts. The PFs of LIR-CMOP5 and LIR-CMOP6 can be achieved by a MOEA without any constraint-handling mechanisms.

In order to expand the test scope, LIR-CMOP7 and LIR-CMOP8 are designed. For these two test instances, their unconstrained PFs are located in the infeasible regions, and their PFs are situated on their constraint boundaries. Thus, a MOEA without constraint-handling methods cannot find the real PFs for LIR-CMOP7 and LIR-CMOP8, which are shown in Fig. 1g, h.

LIR-CMOP9-12 have two different types of constraints. The first type creates large infeasible regions as shown in the black ellipses in Fig. 1i–l. The second type creates difficulty in the entire objective space, as it divides the PFs of LIR-



**Fig. 1** Illustrations of the feasible and infeasible regions of LIR-CMOP1-12

CMOP9-12 into a number of disconnected segments. For LIR-CMOP9-10, their PFs are a part of their unconstrained PFs, and for LIR-CMOP11-12, their PFs are situated on their constraint boundaries.

In the LIR-CMOP test suite, CMOPs with three objectives are also designed. Two CMOPs, including LIR-CMOP13 and

LIR-CMOP14, have three objectives as shown in Fig. 2a, b. The PF of LIR-CMOP13 is the same as that of its unconstrained counterpart. The PF of LIR-CMOP14 is located on the boundaries of its constraints.

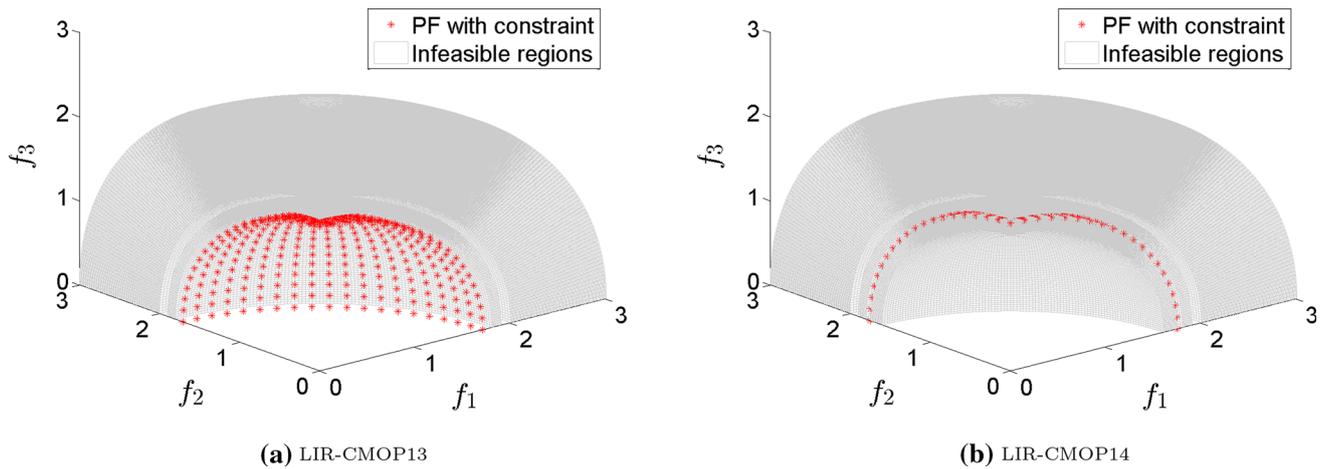


Fig. 2 Illustrations of the infeasible regions of LIR-CMOP13-14

## 5 Experimental study

### 5.1 Experimental settings

To evaluate the performance of the proposed MOEA/D-Epsilon, four other CMOEAs (MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP and CMOEA/D), with differential evolution (DE) crossover, are tested on LIR-CMOP1-14. The detailed parameters of these five CMOEAs are listed as follows:

1. Mutation probability  $Pm = 1/n$  ( $n$  is the number of decision variables) and its distribution index is set to 20.  $CR = 1.0$ ,  $f = 0.5$ .
2. Population size:  $N = 300$ .
3. Neighborhood size:  $T = 30$ .
4. For two objective CMOPs, the  $i$ th weight vector  $\lambda^i = (\frac{i-1}{N-1}, \frac{N-i}{N-1})$ . For three objective CMOPs, the  $N$  weight vectors are defined in the file W3D\_300.dat in the Web site.<sup>1</sup>
5. Stopping condition: Each algorithm runs for 30 times independently and stops when 300,000 function evaluations are reached.
6. Probability of selecting individuals in the neighborhood:  $\delta = 0.9$ .
7. The maximal number of solutions replaced by a child:  $n_r = 2$ .
8. Parameter setting in MOEA/D-Epsilon:  $T_c = 800$ ,  $\alpha = 0.95$ ,  $\tau = 0.1$  and  $\theta = 0.05N$ .
9. Parameter setting in MOEA/D-Epsilon:  $T_c = 800$ ,  $cp = 2$ , and  $\theta = 0.05N$ .
10. Parameter setting in MOEA/D-SR:  $S_r = 0.05$ .

<sup>1</sup> [http://imagelab.stu.edu.cn/Content.aspx?type=content&Content\\_ID=238](http://imagelab.stu.edu.cn/Content.aspx?type=content&Content_ID=238).

### 5.2 Performance metric

To measure the performance of MOEA/D-Epsilon, CMOEA/D, MOEA/D-CDP, MOEA/D-SR and MOEA/D-Epsilon, two commonly used metrics—the inverted generation distance (IGD) (Bosman and Thierens 2003) and the hypervolume (Zitzler and Thiele 1999)—are adopted. The definition of IGD is shown next.

#### – Inverted Generational Distance (IGD):

The IGD metric reflects the performance regarding convergence and diversity simultaneously. The detailed definition is as follows:

$$\begin{cases} IGD(P^*, A) = \frac{\sum_{y^* \in P^*} d(y^*, A)}{|P^*|} \\ d(y^*, A) = \min_{y \in A} \left\{ \sqrt{\sum_{i=1}^m (y_i^* - y_i)^2} \right\} \end{cases} \quad (9)$$

where  $P^*$  is a set of representative solutions in the ideal PF and  $A$  is an approximate PF achieved by a CMOEA. The value of IGD denotes the distance between  $P^*$  and  $A$ . For CMOPs with two objectives, 1000 points are sampled uniformly from the true PF to construct  $P^*$ . (Note that this measure cannot be used if the true Pareto front is unknown, so it is used primarily for benchmarking purposes.) For CMOPs with three objectives, 10,000 points are sampled uniformly from the PF to constitute  $P^*$ . It is worth noting that a smaller value of IGD represents better performance with regard to both diversity and convergence.

### – Hypervolume ( $HV$ ):

$HV$  reflects the closeness of the non-dominated set achieved by a CMOEA to the real PF. The larger  $HV$  means that the corresponding non-dominated set is closer to the true PF.

$$HV(S) = VOL \left( \bigcup_{x \in S} [f_1(x), z_1^r] \times \cdots \times [f_m(x), z_m^r] \right) \quad (10)$$

where  $VOL(\cdot)$  is the Lebesgue measure,  $\mathbf{z}^r = (z_1^r, \dots, z_m^r)^T$  is a reference point in the objective space. For a LIR-CMOP, the reference point is placed at 1.2 times the distance to the nadir point of the true PF. It is worth noting that a larger value of  $HV$  represents better performance regarding both diversity and convergence.

## 5.3 Significance test methods

Two significance test methods, including Wilcoxon rank sum test and the Friedman aligned test, are used to check whether the difference between the proposed MOEA/D-IEpsilon and the compared CMOEAs is statistically significant. By using the Friedman aligned test, we aim to compare all the five CMOEAs, while the Wilcoxon test allows us to compare MOEA/D-IEpsilon with the other four CMOEAs individually. Each test is carried out with a 0.05 significance level. To control the family-wise error rate (FWER), a set of post hoc procedures, including the Bonferroni–Dunn procedure (Dunn 1961), the Holm procedure (Holm 1979), the Holland procedure (Holland and Copenhaver 1987), the Finner procedure (Finner 1993), the Hochberg procedure (Hochberg 1988), the Hommel procedure (Hommel 1988), the Rom procedure (Rom 1990) and the Li procedure (Li 2008), are used according to Derrac et al. (2011).

## 5.4 Discussion of experiments

### 5.4.1 Performance comparison on LIR-CMOP test suite

The statistical results of the IGD values on LIR-CMOP1-14 achieved by five CMOEAs in 30 independent runs are listed in Table 1. The Friedman aligned test indicates that MOEA/D-IEpsilon ranks the highest among the five CMOEAs, as shown in the last row of Table 1. The  $p$ -value computed through the statistics of the Friedman aligned test is  $1.055\text{E}-07$ , which strongly suggests the existence of significant differences among the five tested CMOEAs. To check whether the difference between MOEA/D-IEpsilon and the compared CMOEAs is statistically significant, a set of post hoc procedures are carried out as shown in Table 2. Since each  $p$ -value in Table 2 is less than the preset significance level 0.05, we can conclude that MOEA/D-IEpsilon performs significantly better than the other four CMOEAs in terms of IGD metric.

Table 3 shows the results of the HV values of LIR-CMOP1-14 achieved by five CMOEAs in 30 independent runs. From the last row of this table, we can observe that MOEA/D-IEpsilon ranks the highest among the five CMOEAs according to the Friedman aligned test. The  $p$  value computed through the statistics of the Friedman aligned test is  $8.864\text{E}-08$ , which strongly suggests the existence of significant differences among the algorithms considered. To check whether the difference between MOEA/D-IEpsilon and the compared CMOEAs is statistically significant, a family of post hoc tests is carried out as shown in Table 4. Since each  $p$ -value in Table 4 is less than the preset significance level 0.05, we can also conclude that MOEA/D-IEpsilon performs significantly better than the other four CMOEAs in terms of HV metric (Table 3).

LIR-CMOP1-4 have large infeasible regions in the entire search space, as discussed in Sect. 4. For these four test instances, MOEA/D-IEpsilon is significantly better than the other four tested CMOEAs in terms of the IGD metric. Figure 3a, b shows the final populations achieved by each CMOEA with the best IGD values during the 30 runs on LIR-CMOP1 and LIR-CMOP4. It is clear that MOEA/D-IEpsilon has the best performance regarding diversity among the five CMOEAs under test.

LIR-CMOP5-12 have large infeasible regions, as discussed in Sect. 4. It can be observed that MOEA/D-IEpsilon is significantly better than the other four tested CMOEAs on LIR-CMOP5-12. The final populations achieved by each CMOEA on LIR-CMOP9 and LIR-CMOP11 with the best IGD values are plotted in Fig. 3c, d. For LIR-CMOP9, MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP and CMOEA/D only achieve a part of the real PF. However, MOEA/D-IEpsilon can obtain the whole real PF. Thus, MOEA/D-IEpsilon performs better than the other four CMOEAs in terms of diversity. For LIR-CMOP11, the proposed method MOEA/D-IEpsilon can achieve the whole PF. However, the other four CMOEAs do not converge to the whole PF. Thus, MOEA/D-IEpsilon has better convergence performance than the other four CMOEAs. For three-objective test instances (LIR-CMOP13 and LIR-CMOP14), MOEA/D-IEpsilon is also significantly better than the other four CMOEAs.

From the above analysis, It is clear that MOEA/D-IEpsilon is significantly better than the other four CMOEAs on all of the 14 test instances.

### 5.4.2 Investigation on searching balance

To investigate the searching balance of MOEA/D-IEpsilon, we plot the ratios of feasible to total solutions of the working populations (RFSWP) of MOEA/D-IEpsilon and MOEA/D-Epsilon on LIR-CMOP1-14 at each generation, as shown in Fig. 4. For LIR-CMOP1-4, the RFSWP of MOEA/D-

**Table 1** IGD results of MOEA/D-IEpsilon and the other four CMOEAs on LIR-CMOP1-14 test instances

Test instances	MOEA/D-IEpsilon	MOEA/D-Epsilon	MOEA/D-SR	MOEA/D-CDP	CMOEA/D
<i>LIR-CMOP1</i>					
Mean	<b>7.213E-03</b>	7.432E-02 <sup>†</sup>	1.719E-02 <sup>†</sup>	1.163E-01 <sup>†</sup>	1.290E-01 <sup>†</sup>
SD	2.425E-03	3.538E-02	1.554E-02	7.265E-02	8.055E-02
<i>LIR-CMOP2</i>					
Mean	<b>5.461E-03</b>	6.407E-02 <sup>†</sup>	9.274E-03 <sup>†</sup>	1.244E-01 <sup>†</sup>	1.627E-01 <sup>†</sup>
SD	1.520E-03	3.869E-02	9.723E-03	5.492E-02	5.819E-02
<i>LIR-CMOP3</i>					
Mean	<b>1.117E-02</b>	9.570E-02 <sup>†</sup>	1.792E-01 <sup>†</sup>	2.460E-01 <sup>†</sup>	2.751E-01 <sup>†</sup>
SD	6.856E-03	4.529E-02	7.306E-02	4.444E-02	3.895E-02
<i>LIR-CMOP4</i>					
Mean	<b>4.859E-03</b>	6.141E-02 <sup>†</sup>	2.034E-01 <sup>†</sup>	2.486E-01 <sup>†</sup>	2.631E-01 <sup>†</sup>
SD	1.591E-03	4.127E-02	6.038E-02	3.858E-02	3.331E-02
<i>LIR-CMOP5</i>					
Mean	<b>2.107E-03</b>	9.455E-01 <sup>†</sup>	1.041E+00 <sup>†</sup>	9.827E-01 <sup>†</sup>	8.637E-01 <sup>†</sup>
SD	2.616E-04	4.705E-01	3.833E-01	4.140E-01	5.071E-01
<i>LIR-CMOP6</i>					
Mean	<b>2.058E-01</b>	1.177E+00 <sup>†</sup>	8.699E-01 <sup>†</sup>	1.224E+00 <sup>†</sup>	1.277E+00 <sup>†</sup>
SD	4.587E-01	4.376E-01	5.992E-01	3.726E-01	2.587E-01
<i>LIR-CMOP7</i>					
Mean	<b>4.598E-02</b>	1.475E+00 <sup>†</sup>	1.074E+00 <sup>†</sup>	1.402E+00 <sup>†</sup>	1.511E+00 <sup>†</sup>
SD	6.855E-02	5.309E-01	7.606E-01	6.226E-01	5.032E-01
<i>LIR-CMOP8</i>					
Mean	<b>3.445E-02</b>	1.522E+00 <sup>†</sup>	1.253E+00 <sup>†</sup>	1.361E+00 <sup>†</sup>	1.575E+00 <sup>†</sup>
SD	6.002E-02	4.716E-01	6.597E-01	5.888E-01	3.849E-01
<i>LIR-CMOP9</i>					
Mean	<b>1.290E-02</b>	4.902E-01 <sup>†</sup>	4.883E-01 <sup>†</sup>	4.994E-01 <sup>†</sup>	4.902E-01 <sup>†</sup>
SD	3.300E-02	4.221E-02	4.130E-02	2.526E-02	4.221E-02
<i>LIR-CMOP10</i>					
Mean	<b>2.143E-03</b>	2.202E-01 <sup>†</sup>	1.898E-01 <sup>†</sup>	2.042E-01 <sup>†</sup>	2.114E-01 <sup>†</sup>
SD	1.261E-04	3.589E-02	6.277E-02	6.573E-02	5.641E-02
<i>LIR-CMOP11</i>					
Mean	<b>4.713E-02</b>	3.809E-01 <sup>†</sup>	2.911E-01 <sup>†</sup>	3.221E-01 <sup>†</sup>	3.321E-01 <sup>†</sup>
SD	5.410E-02	1.131E-01	3.525E-02	7.723E-02	7.109E-02
<i>LIR-CMOP12</i>					
Mean	<b>4.711E-02</b>	2.574E-01 <sup>†</sup>	2.045E-01 <sup>†</sup>	2.289E-01 <sup>†</sup>	2.472E-01 <sup>†</sup>
SD	5.662E-02	8.768E-02	6.771E-02	7.823E-02	8.883E-02
<i>LIR-CMOP13</i>					
Mean	<b>6.447E-02</b>	1.239E+00 <sup>†</sup>	1.059E+00 <sup>†</sup>	1.190E+00 <sup>†</sup>	1.215E+00 <sup>†</sup>
SD	1.844E-03	2.555E-01	5.033E-01	3.290E-01	3.140E-01
<i>LIR-CMOP14</i>					
Mean	<b>6.502E-02</b>	1.172E+00 <sup>†</sup>	9.005E-01 <sup>†</sup>	1.204E+00 <sup>†</sup>	1.054E+00 <sup>†</sup>
SD	1.635E-03	3.043E-01	5.455E-01	2.244E-01	4.515E-01
Friedman aligned test	7.9286	43.75	30.4286	46.0714	49.3214

The last row provides the results of the Friedman aligned test. The  $p$  value computed by the aligned Friedman test is 1.055E-07. Wilcoxon rank sum test at a 0.05 significance level is also performed between MOEA/D-IEpsilon and each of the other four CMOEAs

The best mean is highlighted in boldface

<sup>†</sup>, <sup>‡</sup>Performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-IEpsilon, respectively

**Table 2** Adjusted  $p$ -values for the Friedman aligned test in terms of IGD metric (MOEA/D-IEpsilon is the control method)

Friedman aligned	Unadjusted	Bonferroni	Holm	Hochberg	Hommel	Holland	Rom	Finner	Li
CMOEA/D	0	0	0	0	0	0	0	0	0
MOEA/D-CDP	0.000001	0.000003	0.000002	0.000002	0.000002	0.000002	0.000002	0.000001	0.000001
MOEA/D-Epsilon	0.000003	0.000013	0.000006	0.000006	0.000006	0.000006	0.000006	0.000004	0.000003
MOEA/D-SR	0.003443	0.013773	0.003443	0.003443	0.003443	0.003443	0.003443	0.003443	0.003443

**Table 3** HV results of MOEA/D-IEpsilon and the other four CMOEAs on LIR-CMOP1-14 test instances

Test instances	MOEA/D-IEpsilon	MOEA/D-Epsilon	MOEA/D-SR	MOEA/D-CDP	CMOEA/D
<i>LIR-CMOP1</i>					
Mean	<b>1.015E+00</b>	9.413E-01 <sup>†</sup>	9.840E-01 <sup>†</sup>	7.499E-01 <sup>†</sup>	7.344E-01 <sup>†</sup>
SD	1.490E-03	3.751E-02	4.630E-02	1.202E-01	1.269E-01
<i>LIR-CMOP2</i>					
Mean	<b>1.348E+00</b>	1.267E+00 <sup>†</sup>	1.337E+00 <sup>†</sup>	1.093E+00 <sup>†</sup>	1.033E+00 <sup>†</sup>
SD	1.717E-03	5.526E-02	2.252E-02	1.016E-01	9.522E-02
<i>LIR-CMOP3</i>					
Mean	<b>8.686E-01</b>	7.964E-01 <sup>†</sup>	5.892E-01 <sup>†</sup>	5.034E-01 <sup>†</sup>	4.715E-01 <sup>†</sup>
SD	3.373E-03	3.618E-02	1.105E-01	5.141E-02	3.786E-02
<i>LIR-CMOP4</i>					
Mean	<b>1.093E+00</b>	1.024E+00 <sup>†</sup>	8.048E-01 <sup>†</sup>	7.397E-01 <sup>†</sup>	7.203E-01 <sup>†</sup>
SD	1.910E-03	5.903E-02	8.956E-02	5.264E-02	4.480E-02
<i>LIR-CMOP5</i>					
Mean	<b>1.461E+00</b>	2.833E-01 <sup>†</sup>	1.773E-01 <sup>†</sup>	2.428E-01 <sup>†</sup>	3.870E-01 <sup>†</sup>
SD	9.488E-04	5.766E-01	4.619E-01	5.031E-01	6.151E-01
<i>LIR-CMOP6</i>					
Mean	<b>9.412E-01</b>	1.255E-01 <sup>†</sup>	3.341E-01 <sup>†</sup>	8.582E-02 <sup>†</sup>	3.750E-02 <sup>†</sup>
SD	3.848E-01	3.325E-01	4.458E-01	2.707E-01	1.446E-01
<i>LIR-CMOP7</i>					
Mean	<b>2.847E+00</b>	3.516E-01 <sup>†</sup>	9.943E-01 <sup>†</sup>	4.811E-01 <sup>†</sup>	2.933E-01 <sup>†</sup>
SD	2.205E-01	9.304E-01	1.268E+00	1.083E+00	8.776E-01
<i>LIR-CMOP8</i>					
Mean	<b>2.905E+00</b>	2.690E-01 <sup>†</sup>	7.043E-01 <sup>†</sup>	5.223E-01 <sup>†</sup>	1.788E-01 <sup>†</sup>
SD	2.103E-01	8.100E-01	1.094E+00	9.669E-01	6.669E-01
<i>LIR-CMOP9</i>					
Mean	<b>3.692E+00</b>	2.737E+00 <sup>†</sup>	2.733E+00 <sup>†</sup>	2.705E+00 <sup>†</sup>	2.737E+00 <sup>†</sup>
SD	6.318E-02	1.484E-01	1.284E-01	8.883E-02	1.483E-01
<i>LIR-CMOP10</i>					
Mean	<b>3.241E+00</b>	2.874E+00 <sup>†</sup>	2.929E+00 <sup>†</sup>	2.899E+00 <sup>†</sup>	2.886E+00 <sup>†</sup>
SD	3.537E-04	7.851E-02	1.064E-01	1.207E-01	1.126E-01
<i>LIR-CMOP11</i>					
Mean	<b>4.263E+00</b>	3.218E+00 <sup>†</sup>	3.479E+00 <sup>†</sup>	3.406E+00 <sup>†</sup>	3.386E+00 <sup>†</sup>
SD	1.685E-01	3.542E-01	1.252E-01	2.135E-01	1.831E-01
<i>LIR-CMOP12</i>					
Mean	<b>5.552E+00</b>	4.858E+00 <sup>†</sup>	5.059E+00 <sup>†</sup>	4.972E+00 <sup>†</sup>	4.902E+00 <sup>†</sup>
SD	1.730E-01	3.280E-01	2.103E-01	2.596E-01	3.233E-01
<i>LIR-CMOP13</i>					
Mean	<b>5.710E+00</b>	3.097E-01 <sup>†</sup>	1.184E+00 <sup>†</sup>	5.320E-01 <sup>†</sup>	4.642E-01 <sup>†</sup>
SD	1.084E-02	1.048E+00	2.250E+00	1.442E+00	1.426E+00

**Table 3** continued

Test instances	MOEA/D-IEpsilon	MOEA/D-Epsilon	MOEA/D-SR	MOEA/D-CDP	CMOEAs
<i>LIR-CMOP14</i>					
Mean	<b>6.184E+00</b>	5.617E−01 <sup>†</sup>	1.912E+00 <sup>†</sup>	4.032E−01 <sup>†</sup>	1.162E+00 <sup>†</sup>
SD	1.053E−02	1.540E+00	2.705E+00	1.127E+00	2.281E+00
Friedman aligned test	7.7857	43.1786	32.3571	46.2143	47.9643

The last row provides the results of the Friedman aligned test. The  $p$  value computed by the aligned Friedman test is 8.864E−08. Wilcoxon rank sum test at a 0.05 significance level is also performed between MOEA/D-IEpsilon and each of the other four CMOEAs

The best mean is highlighted in boldface

<sup>†</sup>, <sup>‡</sup>Performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-IEpsilon, respectively

**Table 4** Adjusted  $p$ -values for the Friedman aligned test in terms of HV metric (MOEA/D-IEpsilon is the control method)

Friedman aligned	Unadjusted	Bonferroni	Holm	Hochberg	Hommel	Holland	Rom	Finner	Li
CMOEAs	0	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0.000001	0
MOEA/D-CDP	0.000001	0.000002	0.000002	0.000002	0.000002	0.000002	0.000002	0.000001	0.000001
MOEA/D-Epsilon	0.000004	0.000017	0.000008	0.000008	0.000008	0.000008	0.000008	0.000006	0.000004
MOEA/D=SR	0.001401	0.005605	0.001401	0.001401	0.001401	0.001401	0.001401	0.001401	0.001401

Epsilon are all equal to zero from generation 0 to  $T_c$  ( $T_c = 800$ ). When the evolutionary generation is greater than  $T_c$ , the RFSWP of MOEA/D-Epsilon quickly increase to one. However, the RFSWP of MOEA/D-IEpsilon on these test problems are dynamically changing from generation 0 to  $T_c$ , which indicates that MOEA/D-IEpsilon keeps a better balance of searching between infeasible and feasible regions compared to MOEA/D-Epsilon. For LIR-CMOP5-14, the RFSWP of MOEA/D-Epsilon are all equal to one in the whole evolutionary process. On the contrary, the RFSWP of MOEA/D-IEpsilon are dynamically changing. Thus, MOEA/D-IEpsilon can maintain a better balance of the searching between infeasible and feasible regions compared with MOEA/D-Epsilon on LIR-CMOPs.

### 5.4.3 Analysis of experimental results

From the above performance comparison on the 14 test instances LIR-CMOP1-14, it is clear that MOEA/D-IEpsilon has better diversity and convergence performance than the other four decomposition-based CMOEAs on these 14 test instances. A common feature of these test instances is that each of them has a number of large infeasible regions, which demonstrates that the proposed epsilon constraint-handling method can deal with the large infeasible regions very well using its automatic adjustment of the epsilon level.

## 6 Robot gripper optimization

To verify the capability of MOEA/D-IEpsilon to solve real-world optimization problems, a robot gripper optimization

problem with two conflicting objectives and eight constraints is explored.

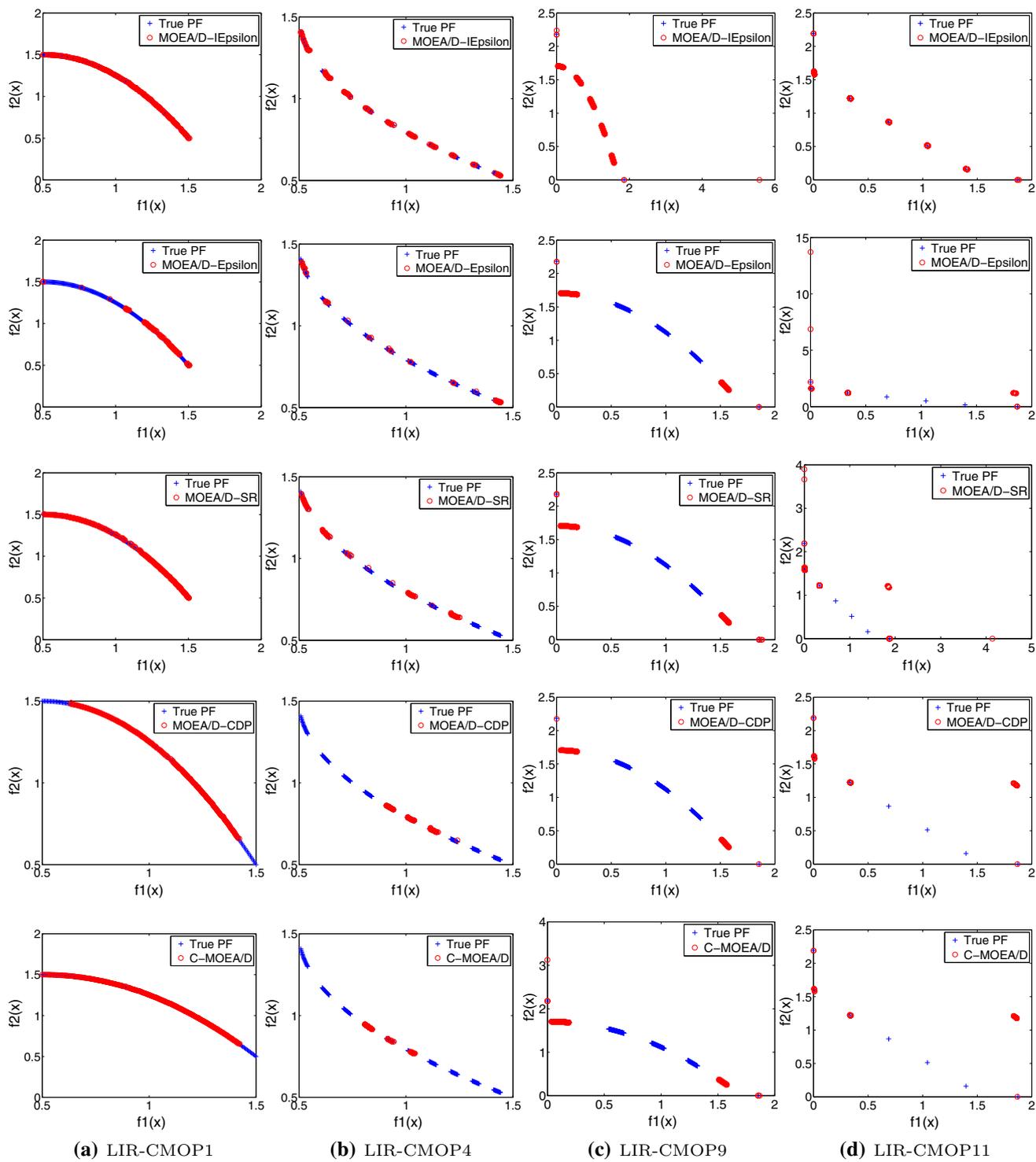
### 6.1 Definition of the robot gripper optimization

The robot gripper optimization problem is defined in Saravanan et al. (2009) and Datta and Deb (2011). Five objectives are formulated in these papers. The robot gripper optimization problem considered in this paper has two conflicting objectives and eight constraints. The geometrical structure of the gripper is plotted in Fig. 5.

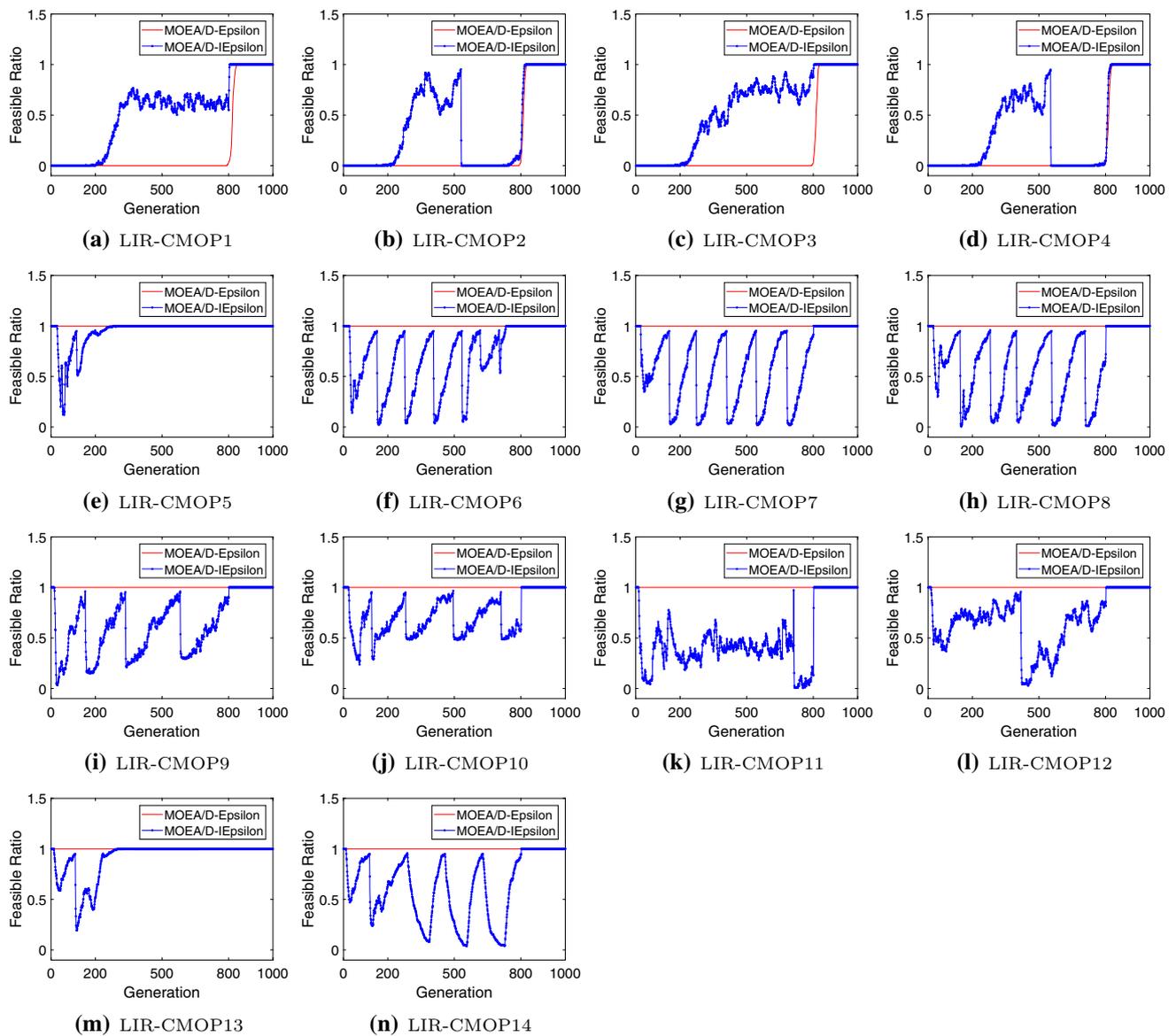
The robot gripper optimization problem considered in this paper is defined as follows:

$$\begin{cases}
 \text{minimize} & f_1(\mathbf{x}) = \frac{P}{\min_z F_k(\mathbf{x}, z)} \\
 \text{minimize} & f_2(x) = a + b + c + e + l \\
 \text{subject to} & c_1(\mathbf{x}) = Y_{\min} - y(\mathbf{x}, Z_{\max}) \geq 0 \\
 & c_2(\mathbf{x}) = y(\mathbf{x}, Z_{\max}) \geq 0 \\
 & c_3(\mathbf{x}) = y(\mathbf{x}, 0) - Y_{\max} \geq 0 \\
 & c_4(\mathbf{x}) = Y_G - y(\mathbf{x}, 0) \geq 0 \\
 & c_5(\mathbf{x}) = (a + b)^2 - l^2 - e^2 \geq 0 \\
 & c_6(\mathbf{x}) = (l - Z_{\max})^2 + (a - e)^2 \geq b^2 \\
 & c_7(\mathbf{x}) = l - Z_{\max} \geq 0 \\
 & c_8(\mathbf{x}) = \min F_k(\mathbf{x}, z) - F_G \geq 0
 \end{cases} \quad (11)$$

where  $\mathbf{x} = [a, b, c, e, l, f, \delta]^T$  has seven decision variables, and each variable is shown in Table 5. The range of each decision variable is as follows:  $10 \text{ mm} \leq a \leq 150 \text{ mm}$ ,  $10 \text{ mm} \leq b \leq 150 \text{ mm}$ ,  $100 \text{ mm} \leq c \leq 200 \text{ mm}$ ,  $0 \text{ mm} \leq e \leq 50 \text{ mm}$ ,  $10 \text{ mm} \leq f \leq 150 \text{ mm}$ ,  $100 \text{ mm} \leq l \leq 300 \text{ mm}$  and



**Fig. 3** Non-dominated solutions achieved by each algorithm with the minimized IGD in the 30 independent runs for LIR-CMOP1, LIR-CMOP4, LIR-CMOP9 and LIR-CMOP11



**Fig. 4** The ratios of feasible to total solutions of each working population of MOEA/D-IEpsilon and MOEA/D-Epsilon on LIR-CMOP1-14

**Table 5** Parameters of the gripper optimization problem

Symbol	Meaning of parameter	Value
$Y_{min}$	Minimal dimension of object to be gripped	50 mm
$Y_G$	Maximal range of the gripper ends displacement	150 mm
$Y_{max}$	Maximal dimension of object to be gripped	100 mm
$z_{max}$	Maximal displacement of the gripper actuator	100 mm
$P$	Actuating force of the gripper	100 N
$F_G$	The lower bound of gripping force	50 N

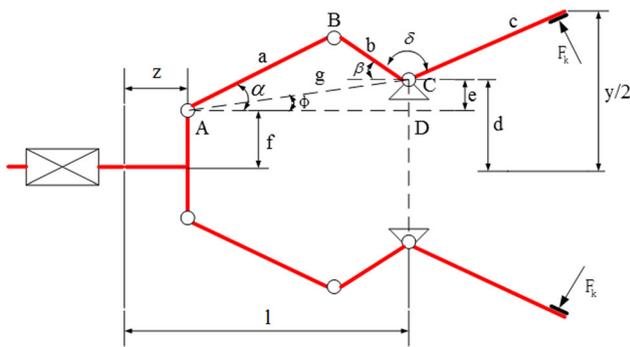


Fig. 5 Geometrical structure of robot gripper mechanism

$1.0 \leq \delta \leq 3.14$ . Two rules are applied to fix the value of  $f$ , and they are defined as follows:

Rule1: if  $(a < 4b \text{ and } c < a + b)$  then  $f = 2e + 10$

Rule2: if  $(a < 4b \text{ and } c > a + b)$  then  $f = e + 50$

According to the geometric analysis, the gripping force  $F_k$  in Fig. 5 can be defined as follows:

$$F_k = \frac{Pb \sin(\alpha + \beta)}{2c \cos \alpha} \tag{12}$$

The displacement of the gripper end is defined as follows:

$$y(\mathbf{x}, z) = 2[e + f + c + \sin(\beta + \delta)]. \tag{13}$$

where  $g = \sqrt{(l - z)^2 + e^2} + \phi$ ,  $\alpha = \arccos(\frac{a^2 + g^2 - b^2}{2ag})$ ,  $\beta = \arccos(\frac{b^2 + g^2 - a^2}{2bg}) - \phi$ ,  $\phi = \arctan \frac{e}{l - z}$  and  $z$  denotes a dynamic displacement of the gripper actuator in the range of 0 to 100 mm.

The first objective  $f_1(x)$  represents a force transmission ratio between the actuating force  $P$  and the minimum gripping force  $\min F_k(\mathbf{x}, z)$ . We prefer to transform more actuating force into the gripper force. Thus, this objective should be minimized. The second objective  $f_2(x)$  is the sum of all elements of the robot gripper. It is relevant to the weight of the robot gripper, and minimizing  $f_2(x)$  can lead to a lightweight design.

To study the distribution of solutions in the objective space for the robot gripper optimization problem, 3,000,000 solutions are generated, where 1,500,000 solutions are generated randomly, and the other 1,500,000 solutions are generated by MOEA/D-IEpsilon. In Fig. 6, we can observe that the robot gripper optimization problem has large infeasible regions ( $RFS = 0.1396$ ), which can be solved well by the proposed method MOEA/D-IEpsilon according to our previous analysis. To verify this hypothesis, MOEA/D-IEpsilon and the other four decomposition-based CMOEAs are tested on the robot gripper optimization problems.

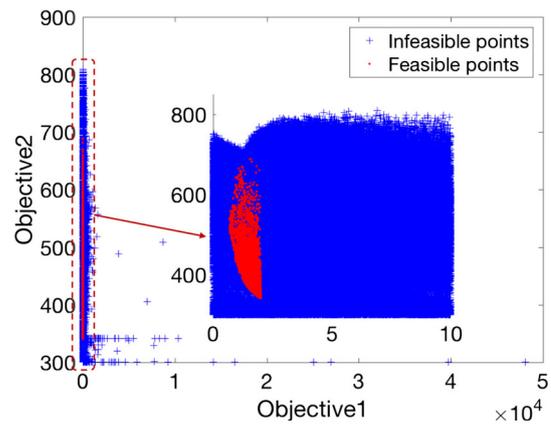


Fig. 6 Distribution of solutions of the robot gripper optimization problem in the objective space,  $RFS = 0.1396$

## 6.2 Experimental study

### 6.2.1 Experimental settings

To solve the robot gripper optimization problem and evaluate the performance of the proposed MOEA/D-IEpsilon, five decomposition-based CMOEAs, including MOEA/D-IEpsilon, MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP and CMOEA/D with the differential evolution (DE) crossover, are tested on the robot gripper optimization problem. The detailed parameters of these five CMOEAs are the same as listed in Sect. 5.1 except for the number of function evaluations. In the case of the robot gripper optimization problem, each CMOEA stops when 600,000 function evaluations are reached. As the ideal PF of the gripper optimization problem is not known in advance, we use only the hypervolume metric (Zitzler and Thiele 1999) to measure the performance of the five tested CMOEAs. In the robot gripper optimization case, the reference point  $z^r = [5, 800]^T$ .

### 6.2.2 Analysis of experiments

Table 6 shows the statistical results of  $HV$  values of MOEA/D-IEpsilon and the other four CMOEAs on the robot gripper optimization problem. It is clear that MOEA/D-IEpsilon is significantly better than the other four CMOEAs. To further demonstrate the superiority of the proposed method MOEA/D-IEpsilon, the non-dominated solutions achieved by each CMOEA during the 30 independent runs are plotted in Fig. 7a–e. The box plot of  $HV$  values of the five CMOEAs is shown in Fig. 7f. From Fig. 7, we see that MOEA/D-IEpsilon has better performance than the other four CMOEAs.

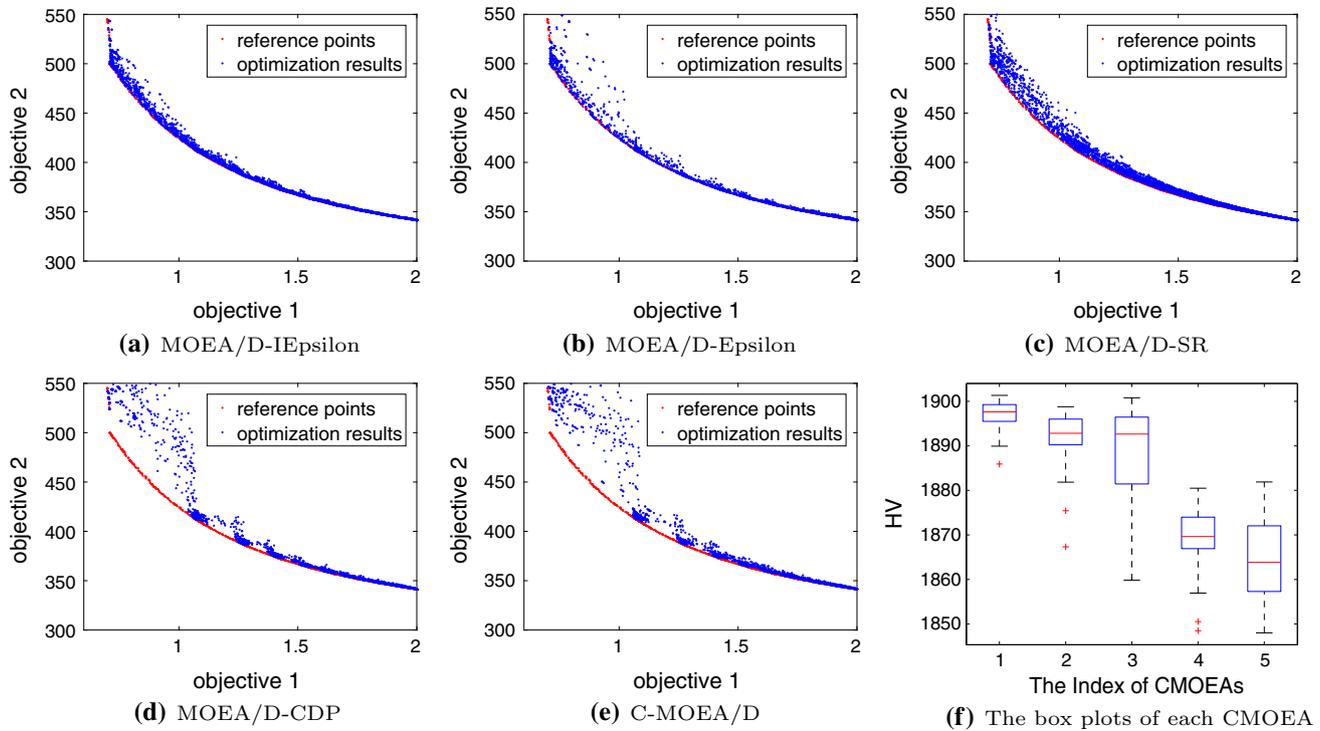
In order to verify the correctness of the optimization results of the robot gripper optimization problem, three representative individuals (A, B and C) are selected from the

**Table 6** HV results of MOEA/D-Epsilon and the other four CMOEAs on the gripper optimization problem

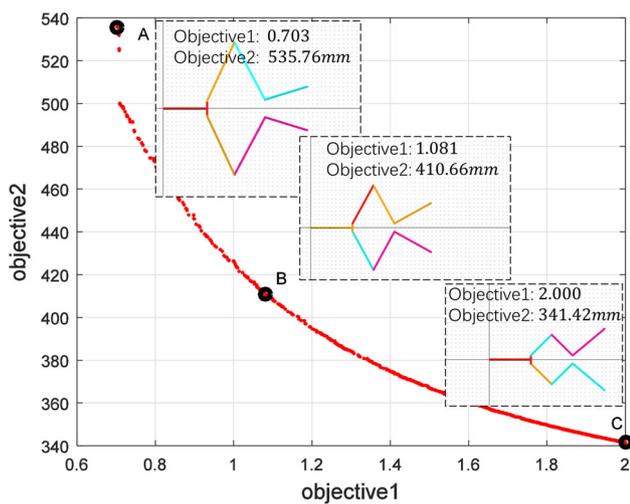
Test Instances	MOEA/D-Epsilon	MOEA/D-Epsilon	MOEA/D-SR	MOEA/D-CDP	cMOEA/D
Mean	<b>1.897E+03</b>	1.891E+03 <sup>†</sup>	1.889E+03 <sup>†</sup>	1.869E+03 <sup>†</sup>	1.865E+03 <sup>†</sup>
SD	3.510E+00	7.151E+00	9.839E+00	8.124E+00	9.048E+00

Wilcoxon rank sum test at a 0.05 significance level is performed between MOEA/D-Epsilon and each of the other four CMOEAs  
 The best mean is highlighted in boldface

<sup>†,‡</sup>Performance of the corresponding algorithm is significantly worse than or better than that of MOEA/D-Epsilon, respectively



**Fig. 7** Non-dominated solutions achieved by each algorithm during the 30 independent runs are plotted in (a)–(e). In (f), the box plots of each CMOEA are plotted



**Fig. 8** Non-dominated solutions achieved by MOEA/D-Epsilon

non-dominated solutions achieved by MOEA/D-Epsilon as shown in Fig. 8. The configurations of the robot gripper mechanism at each point are also plotted in Fig. 8.

To measure the minimum gripping force  $\min_z F_k(\mathbf{x}, z)$ , a spring with a large stiffness coefficient is set vertically at the end of the robot gripper during the simulation process. The spring force is regarded as the gripping force when the robot gripper is balanced by the spring. The simulation tool is ADAMS 2013, and the stiffness coefficient of the spring is  $10^{13}$  N/m.

Table 7 shows the simulation results of the minimum gripping force  $\min_z F_k(\mathbf{x}, z)$  with three different configurations of the robot gripper. The relative errors between the theoretical gripping forces and the simulated gripping forces are less than 0.1%. Thus, we can conclude that the optimization results of the robot gripper optimization problem achieved by MOEA/D-Epsilon are correct.

**Table 7** Simulated results of the minimum gripping force  $\min_z F_k(\mathbf{x}, z)$  with three different robot gripper configurations

Sampled point	The theoretical gripping force (N)	The simulated result (N)	Relative error (%)
A	50.0000	50.0002	0.0004
B	142.3168	142.4582	0.0994
C	92.5285	92.5877	0.0639

## 7 Conclusion

This paper proposes an improved epsilon constraint-handling method embedded in the framework of MOEA/D. A new CMOEA named MOEA/D-IEpsilon has been proposed. The comprehensive experimental results indicate that MOEA/D-IEpsilon has the ability to cross the large infeasible regions. Compared with the other four decomposition-based CMOEAs including MOEA/D-Epsilon, MOEA/D-SR, MOEA/D-CDP and CMOEA/D, MOEA/D-IEpsilon has the following advantages:

- MOEA/D-IEpsilon has the ability to explore the feasible and infeasible regions simultaneously during the evolutionary process.
- MOEA/D-IEpsilon utilizes the ratio of feasible to total solutions of the current population to dynamically balance the exploration between the feasible regions and infeasible regions. It keeps a good balance of the searching between infeasible and feasible regions.
- MOEA/D-IEpsilon is suitable for solving CMOPs with large infeasible regions.

In terms of CMOPs, a new set of CMOPs named LIR-CMOP1-14 was designed and presented in this paper. A common feature of these test instances is that they have large infeasible regions. The experimental results show that MOEA/D-IEpsilon is significantly better than the other four CMOEAs on this test suite. Thus, we hypothesize that MOEA/D-IEpsilon is better than the other four CMOEAs in solving CMOPs with large infeasible regions, in general. To demonstrate the capacity of MOEA/D-IEpsilon to solve real engineering problems, a robot gripper optimization problem with two conflicting objectives and eight constraints was used as a test problem. The experimental results also demonstrated that MOEA/D-IEpsilon outperformed the other four CMOEAs.

Proposed further work includes studying new constraint-handling mechanisms to solve CMOPs with different types of difficulty. One possible way is to collect more information about the working population and utilize such information to guide a CMOEA to select appropriate constraint-handling methods in different evolutionary stages.

**Acknowledgements** This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grants 61300159, 75661473241 and 61332002, by the Natural Science Foundation of Jiangsu Province of China under Grant SBK2018022017, by the Project of International as well as Hong Kong, Macao & Taiwan Science and Technology Cooperation Innovation Platform in Universities in Guangdong Province under Grant 2015KGGJH2014, by China Postdoctoral Science Foundation under Grant 2015M571751, by the Science and Technology Planning Project of Guangdong Province of China under Grant 2013B011304002, by Educational Commission of Guangdong Province of China under Grant 2015KGGJHZ014, by the Fundamental Research Funds for the Central Universities of China under Grant NZ2013306, by the Guangdong High-Level University Project “Green Technologies” for Marine Industries, by the Scientific Startup Research Foundation of Shantou University under Grant NTF12024 and by the State Key Lab of Digital Manufacturing Equipment & Technology under grant DMETKF2019020.

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## Appendix

In this section, the detailed definitions of LIR-CMOP1-14 are listed in Table 8. The source codes of MOEA/D-IEpsilon and LIR-CMOPs can be found in the Web site.<sup>2</sup>

<sup>2</sup> [http://imagelab.stu.edu.cn/Content.aspx?type=content&Content\\_ID=238](http://imagelab.stu.edu.cn/Content.aspx?type=content&Content_ID=238).

**Table 8** Objectives and constraints of LIR-CMOP1-14

Problem	Objectives	Constraints
LIR-CMOP1	$\begin{cases} f_1(x) = x_1 + g_1(x) \\ f_2(x) = 1 - x_1^2 + g_2(x) \\ g_1(x) = \sum_{j \in J_1} (x_j - \sin(0.5\pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(0.5\pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_1(x) = (a - g_1(x)) * (g_1(x) - b) \geq 0 \\ c_2(x) = (a - g_2(x)) * (g_2(x) - b) \geq 0 \\ a = 0.51, b = 0.5 \\ x \in [0, 1]^{30} \end{cases}$
LIR-CMOP2	$\begin{cases} f_1(x) = x_1 + g_1(x) \\ f_2(x) = 1 - \sqrt{x_1} + g_2(x) \\ g_1(x) = \sum_{j \in J_1} (x_j - \sin(0.5\pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(0.5\pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_1(x) = (a - g_1(x)) * (g_1(x) - b) \geq 0 \\ c_2(x) = (a - g_2(x)) * (g_2(x) - b) \geq 0 \\ a = 0.51, b = 0.5 \\ x \in [0, 1]^{30} \end{cases}$
LIR-CMOP3	$\begin{cases} f_1(x) = x_1 + g_1(x) \\ f_2(x) = 1 - x_1^2 + g_2(x) \\ g_1(x) = \sum_{j \in J_1} (x_j - \sin(0.5\pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(0.5\pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_1(x) = (a - g_1(x)) * (g_1(x) - b) \geq 0 \\ c_2(x) = (a - g_2(x)) * (g_2(x) - b) \geq 0 \\ c_3(x) = \sin(c\pi x_1) - 0.5 \geq 0 \\ a = 0.51, b = 0.5, c = 20 \\ x \in [0, 1]^{30} \end{cases}$
LIR-CMOP4	$\begin{cases} f_1(x) = x_1 + g_1(x) \\ f_2(x) = 1 - \sqrt{x_1} + g_2(x) \\ g_1(x) = \sum_{j \in J_1} (x_j - \sin(0.5\pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(0.5\pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_1(x) = (a - g_1(x)) * (g_1(x) - b) \geq 0 \\ c_2(x) = (a - g_2(x)) * (g_2(x) - b) \geq 0 \\ c_3(x) = \sin(c\pi x_1) - 0.5 \geq 0 \\ a = 0.51, b = 0.5, c = 20 \\ x \in [0, 1]^{30} \end{cases}$
LIR-CMOP5	$\begin{cases} f_1(x) = x_1 + 10 * g_1(x) + 0.7057 \\ f_2(x) = 1 - \sqrt{x_1} + 10 * g_2(x) + 0.7057 \\ g_1(x) = \sum_{i \in J_1} (x_i - \sin(\frac{0.5i}{30} \pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(\frac{0.5j}{30} \pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_k(x) = ((f_1 - p_k) \cos \theta_k - (f_2 - q_k) \sin \theta_k)^2 / a_k^2 \\ + ((f_1 - p_k) \sin \theta_k + (f_2 - q_k) \cos \theta_k)^2 / b_k^2 \geq r \\ p_k = [1.6, 2.5], q_k = [1.6, 2.5] \\ a_k = [2, 2], b_k = [4, 8] \\ r = 0.1, \theta_k = -0.25\pi \\ x \in [0, 1]^{30}, k = 1, 2 \end{cases}$
LIR-CMOP6	$\begin{cases} f_1(x) = x_1 + 10 * g_1(x) + 0.7057 \\ f_2(x) = 1 - x_1^2 + 10 * g_2(x) + 0.7057 \\ g_1(x) = \sum_{i \in J_1} (x_i - \sin(\frac{0.5i}{30} \pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(\frac{0.5j}{30} \pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_k(x) = ((f_1 - p_k) \cos \theta_k - (f_2 - q_k) \sin \theta_k)^2 / a_k^2 \\ + ((f_1 - p_k) \sin \theta_k + (f_2 - q_k) \cos \theta_k)^2 / b_k^2 \geq r \\ p_k = [1.8, 2.8], q_k = [1.8, 2.8] \\ a_k = [2, 2], b_k = [8, 8] \\ r = 0.1, \theta_k = -0.25\pi \\ x \in [0, 1]^{30}, k = 1, 2 \end{cases}$
LIR-CMOP7	$\begin{cases} f_1(x) = x_1 + 10 * g_1(x) + 0.7057 \\ f_2(x) = 1 - \sqrt{x_1} + 10 * g_2(x) + 0.7057 \\ g_1(x) = \sum_{i \in J_1} (x_i - \sin(\frac{0.5i}{30} \pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(\frac{0.5j}{30} \pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_k(x) = ((f_1 - p_k) \cos \theta_k - (f_2 - q_k) \sin \theta_k)^2 / a_k^2 \\ + ((f_1 - p_k) \sin \theta_k + (f_2 - q_k) \cos \theta_k)^2 / b_k^2 \geq r \\ p_k = [1.2, 2.25, 3.5], q_k = [1.2, 2.25, 3.5] \\ a_k = [2, 2.5, 2.5], b_k = [6, 12, 10] \\ r = 0.1, \theta_k = -0.25\pi \\ x \in [0, 1]^{30}, k = 1, 2, 3 \end{cases}$
LIR-CMOP8	$\begin{cases} f_1(x) = x_1 + 10 * g_1(x) + 0.7057 \\ f_2(x) = 1 - x_1^2 + 10 * g_2(x) + 0.7057 \\ g_1(x) = \sum_{i \in J_1} (x_i - \sin(\frac{0.5i}{30} \pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(\frac{0.5j}{30} \pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_k(x) = ((f_1 - p_k) \cos \theta_k - (f_2 - q_k) \sin \theta_k)^2 / a_k^2 \\ + ((f_1 - p_k) \sin \theta_k + (f_2 - q_k) \cos \theta_k)^2 / b_k^2 \geq r \\ p_k = [1.2, 2.25, 3.5], q_k = [1.2, 2.25, 3.5] \\ a_k = [2, 2.5, 2.5], b_k = [6, 12, 10] \\ r = 0.1, \theta_k = -0.25\pi \\ x \in [0, 1]^{30}, k = 1, 2, 3 \end{cases}$

**Table 8** continued

Problem	Objectives	Constraints
LIR-CMOP9	$\begin{cases} f_1(x) = 1.7057x_1(10 * g_1(x) + 1) \\ f_2(x) = 1.7057(1 - x_1^2)(10 * g_2(x) + 1) \\ g_1(x) = \sum_{i \in J_1} (x_i - \sin(\frac{0.5i}{30}\pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(\frac{0.5j}{30}\pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_1(x) = ((f_1 - p_1) \cos \theta_1 - (f_2 - q_1) \sin \theta_1)^2 / a_1^2 \\ + ((f_1 - p_1) \sin \theta_1 + (f_2 - q_1) \cos \theta_1)^2 / b_1^2 \geq r \\ c_2(x) = f_1 \sin \alpha + f_2 \cos \alpha \\ - \sin(4\pi(f_1 \cos \alpha - f_2 \sin \alpha)) - 2 \geq 0 \\ p_1 = 1.4, q_1 = 1.4, a_1 = 1.5, b_1 = 6.0 \\ r = 0.1, \alpha = 0.25\pi, \theta_1 = -0.25\pi \\ x \in [0, 1]^{30} \end{cases}$
LIR-CMOP10	$\begin{cases} f_1(x) = 1.7057x_1(10 * g_1(x) + 1) \\ f_2(x) = 1.7057(1 - \sqrt{x_1})(10 * g_2(x) + 1) \\ g_1(x) = \sum_{i \in J_1} (x_i - \sin(\frac{0.5i}{30}\pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(\frac{0.5j}{30}\pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_1(x) = ((f_1 - p_1) \cos \theta_1 - (f_2 - q_1) \sin \theta_1)^2 / a_1^2 \\ + ((f_1 - p_1) \sin \theta_1 + (f_2 - q_1) \cos \theta_1)^2 / b_1^2 \geq r \\ c_2(x) = f_1 \sin \alpha + f_2 \cos \alpha \\ - \sin(4\pi(f_1 \cos \alpha - f_2 \sin \alpha)) - 1 \geq 0 \\ p_1 = 1.1, q_1 = 1.2, a_1 = 2.0, b_1 = 4.0 \\ r = 0.1, \alpha = 0.25\pi, \theta_1 = -0.25\pi \\ x \in [0, 1]^{30} \end{cases}$
LIR-CMOP11	$\begin{cases} f_1(x) = 1.7057x_1(10 * g_1(x) + 1) \\ f_2(x) = 1.7057(1 - \sqrt{x_1})(10 * g_2(x) + 1) \\ g_1(x) = \sum_{i \in J_1} (x_i - \sin(\frac{0.5i}{30}\pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(\frac{0.5j}{30}\pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_1(x) = ((f_1 - p_1) \cos \theta_1 - (f_2 - q_1) \sin \theta_1)^2 / a_1^2 \\ + ((f_1 - p_1) \sin \theta_1 + (f_2 - q_1) \cos \theta_1)^2 / b_1^2 \geq r \\ c_2(x) = f_1 \sin \alpha + f_2 \cos \alpha \\ - \sin(4\pi(f_1 \cos \alpha - f_2 \sin \alpha)) - 2.1 \geq 0 \\ p_1 = 1.2, q_1 = 1.2, a_1 = 1.5, b_1 = 5.0 \\ r = 0.1, \alpha = 0.25\pi, \theta_1 = -0.25\pi \\ x \in [0, 1]^{30} \end{cases}$
LIR-CMOP12	$\begin{cases} f_1(x) = 1.7057x_1(10 * g_1(x) + 1) \\ f_2(x) = 1.7057(1 - x_1^2)(10 * g_2(x) + 1) \\ g_1(x) = \sum_{i \in J_1} (x_i - \sin(\frac{0.5i}{30}\pi x_1))^2 \\ g_2(x) = \sum_{j \in J_2} (x_j - \cos(\frac{0.5j}{30}\pi x_1))^2 \\ J_1 = \{3, 5, \dots, 29\}, J_2 = \{2, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_1(x) = ((f_1 - p_1) \cos \theta_1 - (f_2 - q_1) \sin \theta_1)^2 / a_1^2 \\ + ((f_1 - p_1) \sin \theta_1 + (f_2 - q_1) \cos \theta_1)^2 / b_1^2 \geq r \\ c_2(x) = f_1 \sin \alpha + f_2 \cos \alpha \\ - \sin(4\pi(f_1 \cos \alpha - f_2 \sin \alpha)) - 2.5 \geq 0 \\ p_1 = 1.6, q_1 = 1.6, a_1 = 1.5, b_1 = 6.0 \\ r = 0.1, \alpha = 0.25\pi, \theta_1 = -0.25\pi \\ x \in [0, 1]^{30} \end{cases}$
LIR-CMOP13	$\begin{cases} f_1(x) = (1.7057 + g_1) \cos(0.5\pi x_1) \cos(0.5\pi x_2) \\ f_2(x) = (1.7057 + g_1) \cos(0.5\pi x_1) \sin(0.5\pi x_2) \\ f_3(x) = (1.7057 + g_1) \sin(0.5\pi x_1) \\ g_1 = \sum_{i \in J} 10(x_i - 0.5)^2 \\ J = \{3, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_1(x) = (g(x) - 9)(g(x) - 4) \\ c_2(x) = (g(x) - 3.61)(g(x) - 3.24) \\ g(x) = f_1^2 + f_2^2 + f_3^2 \\ x \in [0, 1]^{30} \end{cases}$
LIR-CMOP14	$\begin{cases} f_1(x) = (1.7057 + g_1) \cos(0.5\pi x_1) \cos(0.5\pi x_2) \\ f_2(x) = (1.7057 + g_1) \cos(0.5\pi x_1) \sin(0.5\pi x_2) \\ f_3(x) = (1.7057 + g_1) \sin(0.5\pi x_1) \\ g_1 = \sum_{i \in J} 10(x_i - 0.5)^2 \\ J = \{3, 4, \dots, 30\} \end{cases}$	$\begin{cases} c_1(x) = (g(x) - 9)(g(x) - 4) \\ c_2(x) = (g(x) - 3.61)(g(x) - 3.24) \\ c_3(x) = (g(x) - 3.0625)(g(x) - 2.56) \\ g(x) = f_1^2 + f_2^2 + f_3^2 \\ x \in [0, 1]^{30} \end{cases}$

**References**

Asafuddoula M, Ray T, Sarker R, Alam K (2012) An adaptive constraint handling approach embedded MOEA/D. In: 2012 IEEE Congress on Evolutionary Computation. IEEE, pp 1–8

Bader J, Zitzler E (2011) HypE: an algorithm for fast hypervolume-based many-objective optimization. *Evol Comput* 19(1):45–76

Beume N, Naujoks B, Emmerich M (2007) SMS-EMOA: multiobjective selection based on dominated hypervolume. *Eur J Oper Res* 181(3):1653–1669

Bosman PA, Thierens D (2003) The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans Evol Comput* 7(2):174–188

Cai X, Hu Z, Fan Z (2013) A novel memetic algorithm based on invasive weed optimization and differential evolution for constrained optimization. *Soft Comput* 17(10):1893–1910

Cai X, Li Y, Fan Z, Zhang Q (2015) An external archive guided multiobjective evolutionary algorithm based on decomposition for combinatorial optimization. *IEEE Trans Evol Comput* 19(4):508–523

Cai X, Yang Z, Fan Z, Zhang Q (2017) Decomposition-based-sorting and angle-based-selection for evolutionary multiobjective and

- many-objective optimization. *IEEE Trans Cybern* 47(9):2824–2837
- Coello Coello CA (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Comput Methods Appl Mech Eng* 191(11–12):1245–1287
- 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*. Morgan Kaufmann, pp 283–290
- Črepinšek M, Liu SH, Mernik M (2013) Exploration and exploitation in evolutionary algorithms: a survey. *ACM Comput Surv* 45(3):35
- Datta R, Deb K (2011) Multi-objective design and analysis of robot gripper configurations using an evolutionary-classical approach. In: *Conference on genetic and evolutionary computation*, pp 1843–1850
- Deb K (2001) *Multi-objective optimization using evolutionary algorithms*, vol 16. Wiley, London
- Deb K, Jain H (2014) An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE Trans Evol Comput* 18(4):577–601
- 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
- Derrac J, García S, Molina D, Herrera F (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol Comput* 1(1):3–18
- Dunn OJ (1961) Multiple comparisons among means. *J Am Stat Assoc* 56(293):52–64
- Fan Z, Li W, Cai X, Li H, Hu K, Zhang Q, Deb K, Goodman ED (2016) Difficulty adjustable and scalable constrained multi-objective test problem toolkit. [arXiv preprint arXiv:1612.07603](https://arxiv.org/abs/1612.07603)
- Finner H (1993) On a monotonicity problem in step-down multiple test procedures. *J Am Stat Assoc* 88(423):920–923
- Hochberg Y (1988) A sharper Bonferroni procedure for multiple tests of significance. *Biometrika* 75(4):800–802
- Holland BS, Copenhaver MD (1987) An improved sequentially rejective Bonferroni test procedure. *Biometrics* 43:417–423
- Holm S (1979) A simple sequentially rejective multiple test procedure. *Scand J Stat* 6:65–70
- Hommel G (1988) A stagewise rejective multiple test procedure based on a modified Bonferroni test. *Biometrika* 75(2):383–386
- Hu Z, Cai X, Fan Z (2013) An improved memetic algorithm using ring neighborhood topology for constrained optimization. *Soft Comput* 18(10):2023–2041
- Huband S, Hingston P, Barone L, While L (2006) A review of multi-objective test problems and a scalable test problem toolkit. *IEEE Trans Evol Comput* 10(5):477–506
- Jan MA, Khanum RA (2013) A study of two penalty-parameterless constraint handling techniques in the framework of MOEA/D. *Appl Soft Comput* 13(1):128–148
- Jiang S, Zhang J, Ong YS, Zhang AN, Tan PS (2015) A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. *IEEE Trans Cybern* 45(10):2202–2213
- Li JD (2008) A two-step rejection procedure for testing multiple hypotheses. *J Stat Plann Inference* 138(6):1521–1527
- Li H, Zhang Q (2009) Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Trans Evol Comput* 13(2):284–302
- Liu HL, Gu F, Zhang Q (2014) Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems. *IEEE Trans Evol Comput* 18(3):450–455
- Mezura-Montes E, Coello Coello CA (2011) Constraint-handling in nature-inspired numerical optimization: past, present and future. *Swarm Evol Comput* 1(4):173–194
- Miettinen K (1999) *Nonlinear multiobjective optimization*, vol 12. Springer, Berlin
- Rom DM (1990) A sequentially rejective test procedure based on a modified Bonferroni inequality. *Biometrika* 77(3):663–665
- Runarsson TP, Yao X (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Trans Evol Comput* 4(3):284–294
- Runarsson TP, Yao X (2005) Search biases in constrained evolutionary optimization. *IEEE Trans Syst Man Cybern Part C Appl Rev* 35(2):233–243
- Saravanan R, Ramabalan S, Ebenezer NGR, Dharmaraja C (2009) Evolutionary multi criteria design optimization of robot grippers. *Appl Soft Comput* 9(1):159–172
- Takahama T, Sakai S (2006) Constrained optimization by the  $\epsilon$  constrained differential evolution with gradient-based mutation and feasible elites. In: *2006 IEEE international conference on evolutionary computation*. IEEE, pp 1–8
- Yang Z, Cai X, Fan Z (2014) Epsilon constrained method for constrained multiobjective optimization problems: some preliminary results. In: *Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation*, ACM, pp 1181–1186
- Zhang Q, Li H (2007) MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 11(6):712–731
- Zhang Q, Zhou A, Zhao S, Suganthan PN, Liu W, Tiwari S (2008) Multiobjective optimization test instances for the CEC 2009 special session and competition. University of Essex, Colchester, UK and Nanyang Technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, Technical report 264
- Zitzler E, Künzli S (2004) Indicator-based selection in multiobjective search. In: *International conference on parallel problem solving from nature*. Springer, pp 832–842
- 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
- Zitzler E, Laumanns M, Thiele L (2001) SPEA2: Improving the strength Pareto evolutionary algorithm. *TIK-report* 103