An Improved Epsilon Constraint Handling Method Embedded in MOEA/D for Constrained Multi-objective Optimization Problems

Zhun Fan
Guangdong Provincial Key Laboratory of Digital Signal and Image Processing,
Department of Electronic Engineering
Shantou University
Guangdong, Shantou 515063

Wenji Li
Department of Electronic Engineering
Shantou University
Guangdong, Shantou 515063

Xinye Cai
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
Jiangsu, Nanjing 210016

Hui Li
School of Mathematics and Statistics
Xi'an Jiaotong University
Xi'an, Shaanxi 710049

Han Huang
School of Software Engineering
South China University of Technology, Guangdong, Guangzhou 510006

Zhaoquan Cai
Department of Computer Science
Huizhou University
Guangdong, Huizhou 516007

Caimin Wei
Department of Mathematics
Shantou University
Guangdong, Shantou 515063

***Abstract*—This paper proposes an improved epsilon constraint handling method embedded in the multi-objective evolutionary algorithm based on decomposition (MOEA/D) to solve constrained multi-objective optimization problems (CMOPs). More specifically, it dynamically adjusts the epsilon level, which is a critical parameter in the epsilon constraint method, according to the feasible ratio of solutions in the current population. In order to verify the effect of the improved epsilon constraint handling method, three algorithms - MOEA/D-CDP, MOEA/D-Epsilon, and MOEA/D-IEpsilon (MOEA/D with the improved epsilon constraint handling mechanism) are tested on nine CMOPs (CMOP1-CMOP9). The comprehensive experimental results indicate that the proposed epsilon constraint handling method is very effective on the performance of both convergence and diversity.**

I. Introduction

In the real world, most engineering optimization problems can be formulated as CMOPs [1] which involve more than one conflicting objective to be optimized and a various of constraints to be met simultaneously. Without loss of generality, a CMOP is defined as follows [2]:

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

where $F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \in R^m$ is a m-dimensional objective vector, $g_i(x) \geq 0$ defines the i-th of q inequality constraints, and $h_j(x) = 0$ defines the j-th of p equality constraints. $x \in R^n$ is a n-dimensional decision vector. In order to evaluate the constraints violation of a solution in a CMOP, an overall constraint violation is adopted which can be defined as follows:

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

If $\phi(x)$ equals to zero, the solution x is feasible, otherwise it is infeasible. For two feasible solutions x^1 and x^2, x^1 is said to dominate x^2 if $f_i(x^1) \leq f_i(x^2)$ for each $i \in \{1, ..., m\}$ and for $f_j(x^1) \leq f_j(x^2)$ at least one $j \in \{1, ..., m\}$, denoted as $x^1 \prec x^2$. For a feasible solution x^*, if there is no other feasible solution dominating x^*, then x^* is called a Pareto optimal solution. The set of all Pareto optimal solutions is called a Pareto Set (PS). Mapping the PS into the objective space, a set of non-dominated objective vectors is obtained, and this set is called a Pareto Front (PF).

Evolutionary algorithms (EAs) are promising methods to solve CMOPs due to the population-based property. They have the ability to achieve a PF in a single running. In this paper, constrained multi-objective evolutionary algorithms (CMOEAs) are adopted to solve CMOPs. In general, CMOEAs consist of two parts - optimizing the multiple objectives and handling the constraints.

In terms of optimizing the multiple objectives, the existing MOEAs can be broadly classified into three categories. They are dominance-based, indicator-based and decomposition-based MOEAs. In the dominance-based methods, typical approaches include NSGA-II [3], SPEA-II [4], and PAES-II [5]. In the indicator-based category, representative methods consist of IBEA [6], R2-IBEA [7] and HypE [8]. In the decomposition-based methods, a multi-objective problem is decomposed into many single objective subproblems, and they are optimized simultaneously in a collaborative way. Representative methods of this type include IMMOGLS [9], MOEA/D [10], [11]. Currently, MOEA/D is a popular algorithm and wins the first place in the CEC2009 MOEA competition. In this paper, MOEA/D is adopted as a MOEA framework to integrate the constraint handling methods to solve CMOPs.

In terms of handling constraints, there are four different types of constraint handling mechanisms in the heuristic algorithms [12]. They are penalty, repair, separatist and hybrid approaches. In the penalty methods, a constrained optimization problem is transformed into an unconstrained one by adding its constraints to the objectives with predefined or adaptive weights which indicate a preference between the constraints and the objectives. Typical methods of this type include static [13], [14], dynamic [15], adaptive [16], [17], self-adaptive [18], annealing-based [19], co-evolutionary-based [20], [21] and death penalty functions [22]. In the repair approaches, a infeasible solution is converted to a feasible one by using a repair operator [23]. In the separatist approaches, objectives and constraints are handled separately. Representative methods of this type consist of multi-objective-based [24], [25], [26], co-evolutionary-based [27], constrained-domination principle (CDP) [28], stochastic ranking (SR) [29], infeasible driven evolutionary algorithm (IDEA) [30] and epsilon constraint methods [31]. In the hybrid approaches, representative methods include Lagrangian multipliers [32], [33], constrained optimization by random evolution [34], fuzzy logic [35], immune system [36], cultural algorithms [37] and ant colony optimization [38] etc.

Currently, the epsilon constraint handling method is very popular. Takahama [39] combines it with the differential evolution (DE) and wins the CEC 2010 competition on constrained single objective optimization problems (CSOPs). The epsilon constraint handling technique was originally proposed to solve CSOPs, and had been successfully used in solving CSOPs. However, for CMOPs, this approach needs to be further studied. In this paper, an improved epsilon constraint handling method is proposed to solve CMOPs.

The rest of this paper is organized as follows. Section II introduces the improved epsilon constraint handling approach. Section III introduces the test problems and the framework of MOEA/D with the improved epsilon constraint handling mechanism. Section IV gives the comprehensive experimental results of MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-IEpsilon, and Section V concludes the paper.

II. Improved Epsilon Constraint Handling Approach

In this section, the epsilon constraint handling method and the improved epsilon constraint handling method are described.

A. Epsilon Level Comparison

In the epsilon constraint handling approach [40], the relaxation of the constraints is controlled by the epsilon level ϵ. The epsilon level comparison for CMOPs is defined as an order relation on the set of $(F(x), \phi(x))$. For two solutions x^1 and x^2, their objective values and constraint violation are $F(x^1), F(x^2)$ and $\phi(x^1), \phi(x^2)$ respectively. Then, for any ϵ satisfying $\epsilon \geq 0$, the epsilon level comparison $\prec_\epsilon$ is defined as follows:

$$(F(x^1), \phi(x^1)) \prec_\epsilon (F(x^2), \phi(x^2))$$
$$\Updownarrow$$
$$\begin{cases} F(x^1) \prec F(x^2), \text{if } \phi(x^1), \phi(x^2) \leq \epsilon \\ F(x^1) \prec F(x^2), \text{if } \phi(x^1) = \phi(x^2) \\ \phi(x^1) < \phi(x^2), \text{otherwise} \end{cases} \tag{3}$$

In Equation (3), the epsilon level comparison is equivalent to the CDP constraint handling method [28] when ϵ equals to zero. In the case of $\epsilon = \infty$, the epsilon level comparison is same to the non-dominated ranking for the objectives.

According to Equation (2), a CMOP defined in Equation (1) can be transformed as follows:

$$\begin{aligned} \text{minimize} \quad & F(x) = (f_1(x), \ldots, f_m(x))^T \\ \text{subject to} \quad & \phi(x) = 0 \end{aligned} \tag{4}$$

The epsilon level comparison is equivalent to transforming the CMOP in Equation (4) into the following problem:

$$\begin{aligned} \text{minimize} \quad & F(x) = (f_1(x), \ldots, f_m(x))^T \\ \text{subject to} \quad & \phi(x) \leq \epsilon \end{aligned} \tag{5}$$

It is worth noting that the PS and PF of the CMOP in Equation (4) can be obtained by converging ϵ to zero in Equation (5).

B. Epsilon Level Setting

In the epsilon constraint handling method, the setting of ϵ is quite critical. In [40], a control of the ϵ parameter is suggested as follows:

$$\epsilon(0) = \phi(x^\theta) \tag{6}$$

$$\epsilon(G) = \begin{cases} \epsilon(0)(1 - \frac{G}{T_c})^{cp}, 0 < G < T_c \\ 0, G \geq T_c \end{cases} \tag{7}$$

where x^θ is the top θ-th individual of the current population which is sorted by the overall constraint violation in a descending order, and cp is a parameter to control the speed of reducing relaxation of constraints. The ϵ level is updated until the generation counter G reaches the control generation T_c. When the generation counter G exceeds T_c, the ϵ level is set to zero.

C. Improved Epsilon Level Setting

According to Equation (6), the $\epsilon(0)$ is set to the constraint violation of the top θ-th individual in the initial population. However, if $\epsilon(0)$ equals to zero, then the epsilon level $\epsilon(G)$ identically equals to zero according to Equation (7). This hinders a CMOEA to explore the infeasible regions in the search space. In order to avoid this issue, a new mechanism to set the initial value of $\epsilon(0)$ is suggested as follows:

$$\epsilon(0) = \begin{cases} \phi(x^\theta), \text{if } \phi(x^\theta) > 0 \text{ and } G = 0 \\ \infty, \text{if } \phi(x^\theta) = 0 \text{ and } G = 0 \end{cases} \tag{8}$$

$$\epsilon(0) = \phi^k_{max}, \text{ if } \epsilon(0) = \infty \quad \text{where} \quad k = \arg\min_G G, \text{ subject to } r^G_f < 1 \tag{9}$$

where r^G_f is the feasible ratio (the number of feasible solutions divided by the population size) of solutions in the G-th generation. ϕ^k_{max} is the maximum constraint violation in the k-th generation. In Equation (8), if $\phi(x^\theta)$ is greater than zero in the initial population ($G = 0$), the $\epsilon(0)$ is set to $\phi(x^\theta)$ as done in Equation (6), otherwise the $\epsilon(0)$ is set to ∞ in order to increase the probability of searching in the infeasible regions. When the feasible ratio of solutions in the working population is less than one, which means there are some infeasible solutions in the current population, the $\epsilon(0)$ is updated to ϕ^k_{max} according to Equation (9). This can help to better control the speed of reducing the $\epsilon(G)$.

In terms of setting the $\epsilon(G)$, it decreases gradually along with the generation counter G as mentioned in Equation (7). In the case of $G \geq T_c$, the $\epsilon(G)$ equals to zero. However, this epsilon setting method may not suitable for solving the CMOP which has large infeasible regions near its PF as shown in Figure 1.

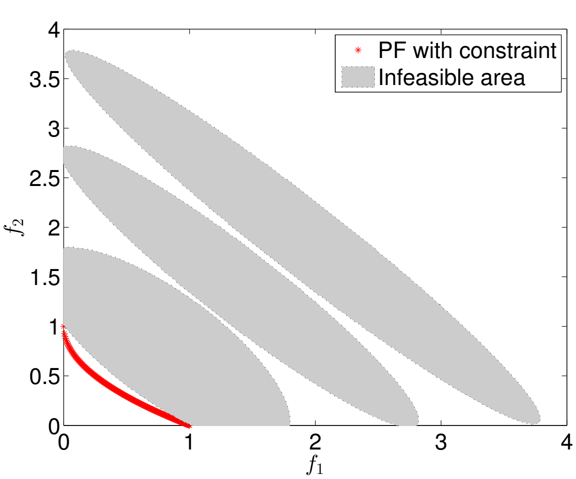

Fig. 1. Illustration of a CMOP with large infeasible regions near its PF

To overcome the shortage of the $\epsilon(G)$ setting in Equation (7), an improved approach of setting $\epsilon(G)$ is suggested as follows:

$$\epsilon(G) = \begin{cases} \epsilon(0)(1 - \frac{G}{T_c})^{cp}, \text{if } r_f < \alpha \text{ and } G < T_c \\ (1+\tau)\phi_{max}, \text{if } r_f \geq \alpha \text{ and } G < T_c \\ 0, G \geq T_c \end{cases} \tag{10}$$

where r_f is the feasible ratio in the current population, α and τ are two factors and belong to $[0, 1]$. ϕ_{max} is the maximum constraint violation found so far. The parameter α is introduced to control the searching preference between the feasible and infeasible regions for a CMOEA. If $r_f < \alpha$, the setting of $\epsilon(G)$ is same to that in Equation (7). In this circumstance, a CMOEA mainly focuses on searching in the feasible regions. If $r_f \geq \alpha$, the $\epsilon(G)$ is set to $(1+\tau)\phi_{max}$, which strengths the searching preference in the infeasible regions. This epsilon level setting method has the ability to increase the $\epsilon(G)$, which can solve the CMOPs with large infeasible regions near their PFs.

III. Test Problems and the Proposed Algorithm

In this section, the CMOPs used to evaluate the CMOEAs are first listed. Then, the improved epsilon constraint handling approach embedded in the framework of MOEA/D is described.

A. Test Instances

In [41], a set of difficulty controllable and scalable CMOPs (CMOP1-CMOP8) are constructed. In this paper, the eight CMOPs are adopted to evaluate the performance of the improved epsilon constraint approach. In Section II, a CMOP with large infeasible regions near its PF is proposed. It is named CMOP9 and added to the test set. The detailed definition of CMOP9 is stated as follows:

$$\begin{aligned} \text{minimize} \quad & f_1(x) = x_1 + g_1(x) \\ \text{minimize} \quad & f_2(x) = 1 - \sqrt{x_1} + g_2(x) \\ \text{subject to} \quad & c_k(x) = ((f_1 - p_k)cos\theta - (f_2 - q_k)sin\theta)^2/a_k^2 \\ & +((f_1 - p_k)sin\theta + (f_2 - q_k)cos\theta)^2/b_k^2 \geq 0.1 \\ \text{where} \quad & g_1(x) = \sum_{j \in J_1} (x_j - sin(0.5\pi x_1))^2 \\ & g_2(x) = \sum_{j \in J_2} (x_j - cos(0.5\pi x_1))^2 \\ & J_1 = \{j | j \text{ is odd and } 2 \leq j \leq n\} \\ & J_2 = \{j | j \text{ is even and } 2 \leq j \leq n\} \\ & p = [0.8, 1.4, 1.9], q = [0.8, 1.4, 1.9] \\ & a = [1.5, 1.0, 1.0], b = [4.0, 6.0, 8.0] \\ & \theta = -0.25\pi, n = 30, x_j \in [0, 1], k = 1, 2, 3. \end{aligned} \tag{11}$$

B. The Improved CMOEA

In order to facilitate the description, the improved epsilon constraint handling method embedded in MOEA/D is called MOEA/D-IEpsilon for short. The pseudo-codes of MOEA/D-IEpsilon are listed in Algorithm I.

Algorithm I: MOEA/D-IEpsilon

Input: A CMOP;
Output: A set of non-dominated feasible solutions NS;
Step 1: Initialization:
a) Generate an initial population $P = \{x^1, \ldots, x^N\}$.
b) Initialize $\epsilon(0)$ according to Equation (8), set $G = 0$.
c) Calculate neighbors of x^i, denoted as S^i, set n_r.
Step 2: Population update
For $i = 1, \ldots, N$, do
  a) Generate a new individual y^i using DE.
  b) Perform a polynomial mutation on y^i.
  c) Update the ideal point z^*.
  d) Update of Solutions: Set $c = 0$, $\epsilon = \epsilon(G)$, and then do the following:
    1) If $c = n_r$ or S^i is empty, continue. Otherwise, select an index j from S randomly.
    2) If $(g(y^i|\lambda^j, z^*), \phi(y^i)) \prec_\epsilon (g(x^j|\lambda^j, z^*), \phi(x^j))$, then set $x^j = y^i$ and $c = c + 1$.
    3) Remove j from S^i and go to 1).
End
Step 3:Update NS and epsilon level
a) Set $U = NS \cup P$, select at most N feasible solutions from U to construct NP according the non-dominated ranking.
b) Update $\epsilon(0)$ according to Equation (9), if $\epsilon(0) = \infty$.
c) Set $\epsilon(G)$ according to Equation (10).
Step 4: Termination If stopping criteria are satisfied, output NS. Otherwise, go to **Step 2** and set G += 1.

In Algorithm I, the input is a CMOP and the output is a set of non-dominated feasible solutions. The algorithm MOEA/D-IEpsilon mainly consists of four steps. In the first step, the working population, the epsilon level $\epsilon(0)$ and the parameters in MOEA/D are initialized. In the second step, a new solution is generated by using a differential evolution operator. A polynomial mutation is performed on the newly generated solution, and a repair operator is adopted to fix the solution. In the replacement stage, the newly generated solution is compared with its neighbors based on the epsilon level comparison. In the third step, a non-dominated ranking operator is executed on the feasible solutions in the union of the external archive and the working population. Solutions in the first rank of the union is selected into the external archive. If the number of solutions in the first rank is great than the population size, the crowding distance is calculated to select the top N feasible solutions into the external archive. Then, the epsilon level $\epsilon(0)$ and $\epsilon(G)$ are updated according to Equation (9) and Equation (10) respectively. In Step 4, if stopping criteria are met, output the non-dominated solutions, otherwise, go to Step 2.

IV. Experimental Study

A. Experimental Settings

To verify the effect of the improved epsilon constraint handling approach, three CMOEAs (i.e., MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-IEpsilon) are adopted in the experiments. The parameters of these three CMOEAs are listed as follows:

1) Setting for reproduction operators: The mutation probability $Pm = 1/n$ (n is the number of decision variables) and its distribution index is set to be 20. For the DE operator, we set $CR = 1.0$ and $f = 0.5$.
2) Population size: $N = 300$.
3) The number of runnings and the stopping condition: Each algorithm runs 30 times independently on each test problem. The algorithm stops until 300 000 function evaluations.
4) Neighborhood size: $T = 20$.
5) Probability used to select in the neighborhood: $\delta = 0.9$.
6) The maximal number of solutions replaced by a child: $n_r = 2$.
7) The parameters in the improved epsilon constraint handling approach are defined as follow: $\alpha = 0.8$, $T_c = 800$, $cp = 2$, $\tau = 0.1$ and $\theta = 0.2NI$. NI is the number of infeasible solutions in the initial population.

B. Performance Metrics

To compare the performance of MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-IEpsilon, two popular metrics - inverted generation distance (IGD)[42] and relative hypervolume indicator (I_H^-) [43] are employed. The definitions of IGD and I_H^- are stated as follows:

- **Inverted Generational Distance** (IGD):

$$\begin{cases} IGD(P^*, A) = \frac{\sum_{y^* \in P^*} d(y^*, A)}{|P^*|} \\ \\ d(y^*, A) = \min_{y \in A} \{\sqrt{\sum_{i=1}^{m} (y_i^* - y_i)^2}\} \end{cases} \tag{12}$$

where P^* is a set of representative points in the PF, A is an approximate PF achieved by algorithms. IGD metric represents the distance between P^* and A, the smaller value of IGD indicates the better performance of both convergence and diversity.

- **Relative Hypervolume Indicator** (I_H^-):

$$\begin{cases} I_H^-(A, P^*, R) = I_H(P^*, R) - I_H(A, R) \\ I_H(P^*, R) = Vol_{v \in P^*}(v) \\ I_H(A, R) = Vol_{v \in A}(v) \end{cases} \tag{13}$$

where $I_H(P^*, R)$ is a volume enclosed by P^* and the reference vector $R = (R_1, \ldots, R_m)$. $I_H(A, R)$ is the volume enclosed by A and R. I_H^- simultaneously considers the distribution of the obtained solutions set - A and its vicinity to the PF. For CMOP1-CMOP2 and CMOP7-CMOP9, the reference point R is set to $(1.2, 1.2)^T$. For CMOP3-CMOP6, R is set to $(1.6, 1.6)^T$. The smaller value of I_H^- represents the better performance of both diversity and convergence. It is worth noting that if a MOEA can not get any feasible solutions, the IGD and I_H are set to one.

C. Experimental Results and Discussions

Figure 2 shows the final non-dominated solutions with the best IGD metric in 30 independent runnings by using MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-IEpsilon. It can be observed that MOEA/D-IEpsilon is much better than MOEA/D-CDP and MOEA/D-Epsilon on CMOP3-CMOP7, and not bad on the rest of CMOPs. This demonstrates the effect of the improved epsilon handling method.

Table I and Table II show the best, median and worst IGD and I_H^- values obtained by MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-IEpsilon, respectively. In these two tables, MOEA/D-IEpsilon is significantly better than MOEA/D-CDP and MOEA/D-Epsilon on all of test problems. Therefore, the proposed method MOEA/D-IEpsilon has superiority over MOEA/D-CDP and MOEA/D-Epsilon on CMOP1-CMOP9.

The box plots of IGD and I_H^- metrics of CMOP1-CMOP9 on MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-IEpsilon are shown in Figure 3. For CMOP3-CMOP5 and CMOP7-CMOP9, the performance of MOEA/D-IEpsilon is very stable and significantly better than MOEA/D-CDP and MOEA/D-Epsilon. For the rest of test problems, MOEA/D-IEpsilon is also the best, which further verify that the improved epsilon constraint handling approach is effective.

For CMOP3-CMOP6, which have large portion of infeasible regions in the search space, MOEA/D-CDP has only found a part of the PFs. The reason is that the CDP constraint handling method always prefers the feasible regions to the infeasible regions. When MOEA/D-CDP gets some feasible solutions, they quickly replaces the infeasible solutions, and the diversity of the working population is bad. Because the feasible region is very narrowed, it is difficult for MOEA/D-CDP to expand its search regions. However, the epsilon constraint handling method allows a CMOEA to search infeasible regions, which helps to improve the diversity of the working population. This is the reason that MOEA/D-Epsilon is better than MOEA/D-CDP on CMOP3-CMOP6. Comparing with MOEA/D-Epsilon, MOEA/D-IEpsion increases the epsilon level when the feasible ratio is greater than a given threshold, which further strengths the searching preference to the infeasible regions. Therefore, MOEA/D-IEpsilon is better than MOEA/D-CDP and MOEA/D-Epsilon on the performance of diversity.

For CMOP7 and CMOP9, they have large infeasible regions near their PFs. As the epsilon level of MOEA/D-Epsilon is decreased gradually, it is difficult for MOEA/D-Epsilon to expand the infeasible regions on CMOP7 and CMOP9. MOEA/D-CDP is equivalent to MOEA/D-Epsilon when the epsilon level equals to zero. Therefore, it is difficult for MOEA/D-CDP and MOEA/D-Epsilon to solve CMOPs with large infeasible regions near their PFs. However, the epsilon level of MOEA/D-IEpsilon is increased when the ratio of feasible solutions is greater than a threshold, which helps MOEA/D-IEpsilon to cross the large infeasible regions. That is why MOEA/D-IEpsilon is significantly better than MOEA/D-CDP and MOEA/D-Epsilon on the performance of convergence on CMOP7 and CMOP9.

V. Conclusion

This paper proposed an improved epsilon constraint handling approach embedded in MOEA/D to solve CMOPs. The proposed method MOEA/D-IEpsilon along with

TABLE I

Best, median and worst IGD values obtained by MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-IEpsilon on CMOP1-CMOP9. Best performance is highlighted in bold font

Problem	MOEA/D-CDP	MOEA/D-Epsilon	MOEA/D-IEpsilon
CMOP1	2.51E-03	2.31E-03	**1.48E-03**
	3.88E-03	3.77E-03	**2.91E-03**
	4.89E-03	5.28E-03	**4.13E-03**
CMOP2	1.79E-03	1.67E-03	**1.25E-03**
	1.97E-03	2.04E-03	**1.34E-03**
	2.42E-03	2.38E-03	**1.50E-03**
CMOP3	3.69E-02	4.72E-03	**2.40E-03**
	2.32E-01	3.68E-02	**2.84E-03**
	3.05E-01	1.05E-01	**4.28E-03**
CMOP4	1.15E-01	6.71E-03	**2.29E-03**
	2.15E-01	1.86E-02	**2.52E-03**
	2.67E-01	1.36E-01	**3.45E-03**
CMOP5	1.73E-01	2.76E-02	**2.35E-03**
	2.73E-01	7.01E-02	**3.03E-03**
	3.32E-01	1.24E-01	**2.08E-02**
CMOP6	1.63E-01	2.71E-02	**1.87E-03**
	2.64E-01	4.31E-02	**3.66E-03**
	3.13E-01	8.96E-02	**4.24E-02**
CMOP7	6.59E-02	6.62E-02	**9.01E-04**
	7.03E-02	6.92E-02	**1.02E-03**
	2.31E-01	2.32E-01	**1.15E-03**
CMOP8	1.68E-03	1.73E-03	**1.21E-03**
	2.06E-03	1.91E-03	**1.36E-03**
	2.46E-03	2.38E-03	**1.48E-03**
CMOP9	2.22E-03	2.17E-03	**2.07E-03**
	9.78E-01	9.80E-01	**2.27E-03**
	9.80E-01	9.80E-01	**2.75E-03**

TABLE II

Best, median and worst I_H^- values obtained by MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-IEpsilon on CMOP1-CMOP9. Best performance is highlighted in bold font

Problem	MOEA/D-CDP	MOEA/D-Epsilon	MOEA/D-IEpsilon
CMOP1	5.39E-04	4.55E-04	**1.16E-04**
	1.20E-03	9.88E-04	**3.57E-04**
	1.92E-03	2.01E-03	**7.02E-04**
CMOP2	2.09E-04	1.49E-04	**-5.19E-05**
	4.49E-04	4.31E-04	**8.53E-05**
	7.76E-04	7.32E-04	**3.84E-04**
CMOP3	5.90E-02	3.09E-03	**1.74E-03**
	2.36E-01	3.15E-02	**2.22E-03**
	2.91E-01	1.06E-01	**3.21E-03**
CMOP4	1.53E-01	7.41E-03	**1.43E-03**
	2.89E-01	2.24E-02	**1.76E-03**
	3.45E-01	1.73E-01	**3.32E-03**
CMOP5	1.84E-01	2.47E-02	**4.00E-03**
	2.51E-01	6.29E-02	**5.91E-03**
	2.81E-01	1.26E-01	**1.32E-02**
CMOP6	1.97E-01	3.10E-02	**3.13E-03**
	3.05E-01	5.02E-02	**7.13E-03**
	3.66E-01	9.09E-02	**3.74E-02**
CMOP7	1.91E-01	1.91E-01	**-1.60E-04**
	1.93E-01	1.92E-01	**2.11E-04**
	2.90E-01	2.90E-01	**5.97E-04**
CMOP8	1.32E-04	1.42E-04	**-2.76E-05**
	5.79E-04	3.88E-04	**1.34E-04**
	1.45E-03	6.04E-04	**4.10E-04**
CMOP9	1.05E-03	7.92E-04	**4.26E-04**
	1.10E+00	1.10E+00	**7.04E-04**
	1.10E+00	1.10E+00	**2.34E-03**

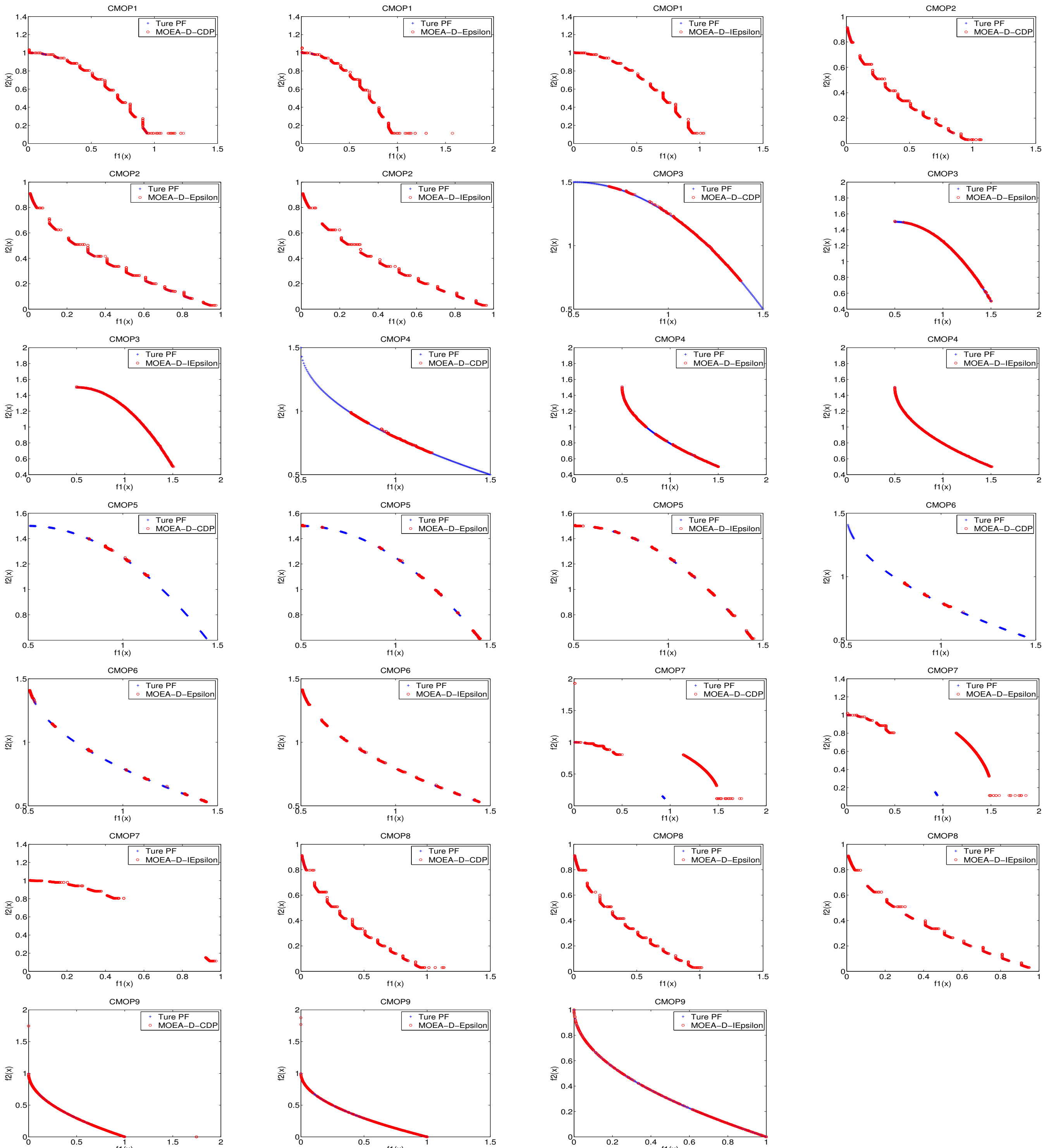

Fig. 2. The final populations with the best IGD metric in 30 independent runnings by using MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-IEpsilon

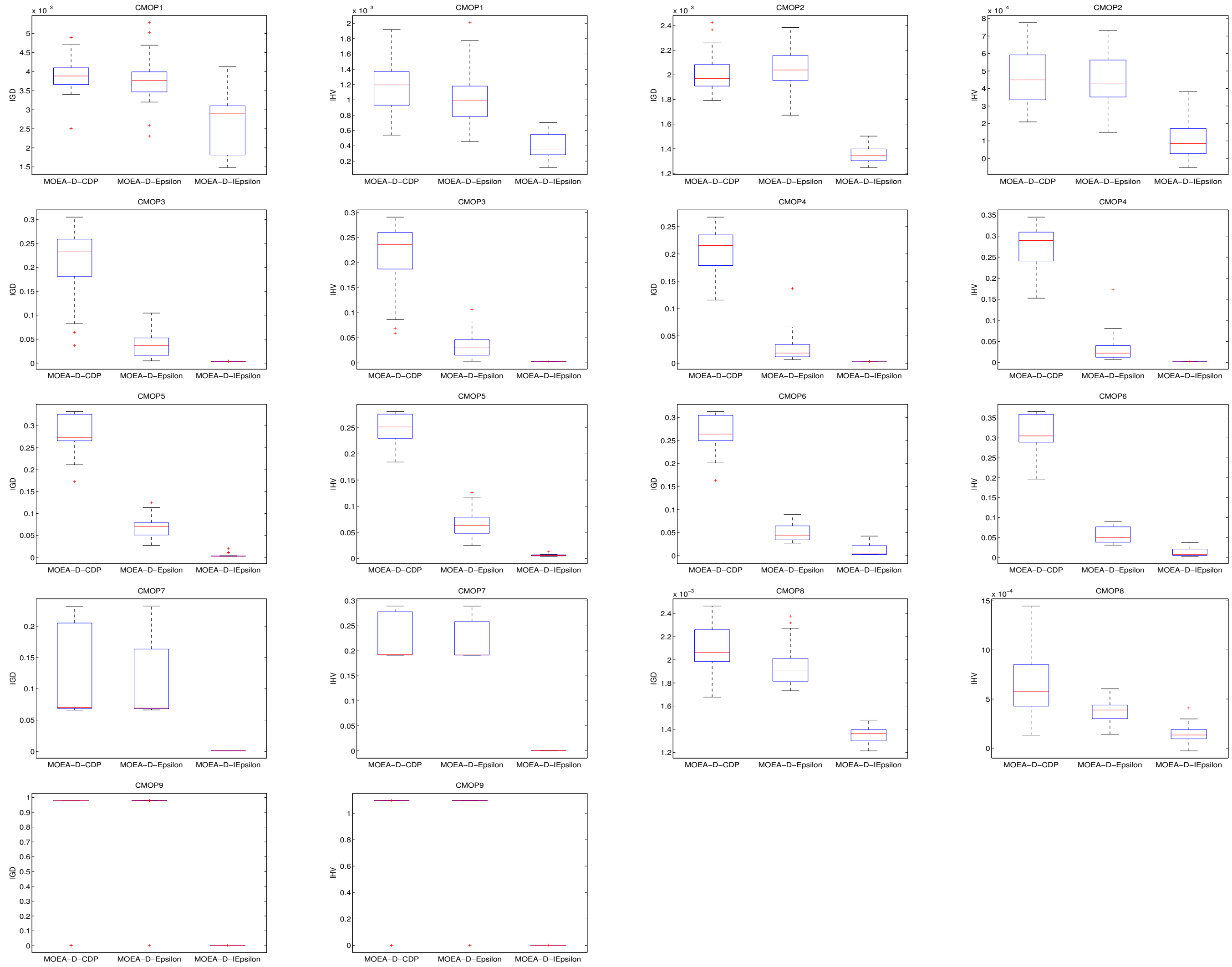

Fig. 3. The box plots of IGD and I_H^- metrics of CMOP1-CMOP9 on MOEA/D-CDP, MOEA/D-Epsilon and MOEA/D-IEpsilon

MOEA/D-CDP and MOEA/D-Epsilon are tested on CMOP1-CMOP9. The comprehensive experimental results verified that MOEA/D-IEpsilon is significantly better than MOEA/D-CDP and MOEA/D-Epsilon on the performance of both convergence and diversity. Unlike the original epsilon constraint method with a decreasing epsilon level, the proposed method increases the epsilon level if the feasible ratio of solutions in the current population is greater than a threshold defined by users. It can be concluded that increasing the searching preference in infeasible regions can help to enhance the performance of a CMOEA. The further work includes testing the MOEA/D-IEpsilon on some real-world optimization problems and comparing it with some other state-of-the-art CMOEAs.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grant (61175073, 61300159, 61332002, 51375287), in part by the Guangdong High-Level University Project Green Technologies for Marine Industries, in part by the Science and Technology Planning Project of Guangdong Province (2013B011304002), in part by the Project of Educational Commission of Guangdong Province, China2015KGJHZ014), and in part by the Jiangsu Natural Science Foundation (BK20130808).

References

[1] Y. Collette and P. Siarry, *Multiobjective optimization: principles and case studies*. Springer Science & Business Media, 2013.

[2] Y. Wang, Z.-X. Cai, Y.-R. Zhou, and C.-X. Xiao, “Constrained optimization evolutionary algorithms,” *Journal of Software*, vol. 20, no. 1, pp. 11–29, 2009.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.

[4] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, and L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm,” 2001.

[5] D. W. Corne, N. R. Jerram, J. D. Knowles, M. J. Oates *et al.*, “PESA-II: Region-based selection in evolutionary multiobjective optimization,” in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001)*. Citeseer, 2001.

[6] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature-PPSN VIII*. Springer, 2004, pp. 832–842.

[7] D. H. Phan and J. Suzuki, "R2-IBEA: R2 indicator based evolutionary algorithm for multiobjective optimization," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*. IEEE, 2013, pp. 1836–1845.

[8] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume-based many-objective optimization," *Evolutionary computation*, vol. 19, no. 1, pp. 45–76, 2011.

[9] H. Ishibuchi and T. Murata, "A multi-objective genetic local search algorithm and its application to flowshop scheduling," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 28, no. 3, pp. 392–403, 1998.

[10] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.

[11] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 2, pp. 284–302, 2009.

[12] A. R. Jordehi, "A review on constraint handling strategies in particle swarm optimisation," *Neural Computing and Applications*, vol. 26, no. 6, pp. 1265–1275, 2015.

[13] A. Homaifar, C. X. Qi, and S. H. Lai, "Constrained optimization via genetic algorithms," *Simulation*, vol. 62, no. 4, pp. 242–253, 1994.

[14] A. K. Morales and C. V. Quezada, "A universal eclectic genetic algorithm for constrained optimization," in *Proceedings of the 6th European congress on intelligent techniques and soft computing*, vol. 1. Citeseer, 1998, pp. 518–522.

[15] J. A. Joines and C. R. Houck, "On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with ga's," in *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*. IEEE, 1994, pp. 579–584.

[16] D. W. Coit, A. E. Smith, and D. M. Tate, "Adaptive penalty methods for genetic optimization of constrained combinatorial problems," *INFORMS Journal on Computing*, vol. 8, no. 2, pp. 173–182, 1996.

[17] S. B. Hamida and M. Schoenauer, "Aschea: new results using adaptive segregational constraint handling," in *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, vol. 1. IEEE, 2002, pp. 884–889.

[18] Y. G. Woldesenbet, G. G. Yen, and B. G. Tessema, "Constraint handling in multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 3, pp. 514–525, 2009.

[19] S. E. Carlson and R. Shonkwiler, "Annealing a genetic algorithm over constraints," in *Systems, Man, and Cybernetics, 1998. 1998 IEEE International Conference on*, vol. 4. IEEE, 1998, pp. 3931–3936.

[20] C. A. C. Coello, "Use of a self-adaptive penalty approach for engineering optimization problems," *Computers in Industry*, vol. 41, no. 2, pp. 113–127, 2000.

[21] F.-z. Huang, L. Wang, and Q. He, "An effective co-evolutionary differential evolution for constrained optimization," *Applied Mathematics and computation*, vol. 186, no. 1, pp. 340–356, 2007.

[22] F. Hoffmeister and J. Sprave, "Problem-independent handling of constraints by use of metric penalty functions," 1996.

[23] J. Xiao, Z. Michalewicz, L. Zhang, and K. Trojanowski, "Adaptive evolutionary planner/navigator for mobile robots," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 18–28, 1997.

[24] P. D. Surry and N. J. Radcliffe, "The comoga method: constrained optimisation by multi-objective genetic algorithms," *Control and Cybernetics*, vol. 26, pp. 391–412, 1997.

[25] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 10, no. 6, pp. 658–675, 2006.

[26] Y. Wang, Z. Cai, Y. Zhou, and W. Zeng, "An adaptive tradeoff model for constrained evolutionary optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 12, no. 1, pp. 80–92, 2008.

[27] J. Paredis, "Co-evolutionary constraint satisfaction," in *International Conference on Parallel Problem Solving from Nature*. Springer, 1994, pp. 46–55.

[28] K. Deb, "An efficient constraint handling method for genetic algorithms," *Computer methods in applied mechanics and engineering*, vol. 186, no. 2, pp. 311–338, 2000.

[29] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 4, no. 3, pp. 284–294, 2000.

[30] T. Ray, H. K. Singh, A. Isaacs, and W. Smith, "Infeasibility driven evolutionary algorithm for constrained optimization," in *Constraint-handling in evolutionary optimization*. Springer, 2009, pp. 145–165.

[31] M. Laumanns, L. Thiele, and E. Zitzler, "An efficient, adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method," *European Journal of Operational Research*, vol. 169, no. 3, pp. 932–942, 2006.

[32] H. Adeli and N.-T. Cheng, "Augmented lagrangian genetic algorithm for structural optimization," *Journal of Aerospace Engineering*, vol. 7, no. 1, pp. 104–118, 1994.

[33] J.-H. Kim and H. Myung, "Evolutionary programming techniques for constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 2, pp. 129–140, 1997.

[34] S. V. Belur, "Core: Constrained optimization by random evolution," in *Late Breaking Papers at the Genetic Programming 1997 Conference*, 1997, pp. 280–286.

[35] T. V. Le, "A fuzzy evolutionary approach to constrained optimization problems," in *Proceedings of the second IEEE conference on evolutionary computation*, 1995, pp. 274–278.

[36] S. Forrest and A. S. Perelson, "Genetic algorithms and the immune system," in *International Conference on Parallel Problem Solving from Nature*. Springer, 1990, pp. 319–325.

[37] R. G. Reynolds, Z. Michalewicz, and M. J. Cavaretta, "Using cultural algorithms for constraint handling in genocop." in *Evolutionary programming*, 1995, pp. 289–305.

[38] G. Bilchev and I. C. Parmee, "The ant colony metaphor for searching continuous design spaces," in *AISB workshop on evolutionary computing*. Springer, 1995, pp. 25–39.

[39] T. Takahama and S. Sakai, "Constrained optimization by the epsilon constrained differential evolution with an archive and gradient-based mutation," in *IEEE Congress on Evolutionary Computation*, July 2010, pp. 1–9.

[40] ——, "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites," in *2006 IEEE International Conference on Evolutionary Computation*. IEEE, 2006, pp. 1–8.

[41] Z. Fan, W. Li, X. Cai, H. Li, K. Hu, and H. Yin, "Difficulty controllable and scalable constrained multi-objective test problems," in *Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), 2015 International Conference on*, Dec 2015, pp. 76–83.

[42] P. A. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *Evolutionary Computation, IEEE Transactions on*, vol. 7, no. 2, pp. 174–188, 2003.

[43] J. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," *Tik report*, vol. 214, pp. 327–332, 2006.