IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 30, 2020, accepted January 9, 2021, date of publication January 21, 2021, date of current version February 1, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3053041

Adaptive ε-Constraint Multi-Objective Evolutionary Algorithm Based on Decomposition and Differential Evolution

BING-JIE LIU[1] AND XIAO-JUN BI[2]

[1]College of Information and Communications Engineering, Harbin Engineering University, Harbin 150001, China
[2]Department of Information Engineering, Minzu University of China, Beijing 100081, China

Corresponding author: Xiao-Jun Bi (bixiaojun@hrbeu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51779050.

ABSTRACT To improve distribution and convergence of the obtained solution set in constrained multi-objective optimization problems, this paper presents an adaptive ε-constraint multi-objective evolutionary algorithm based on decomposition and differential evolution (ε-MOEA/D-DE). First, an adaptive ε-constraint strategy based on both evolution generation and constraint violation is designed to make better use of excellent evolution individuals and improve population diversity. Then, an adaptive differential evolution (DE) mutation strategy with full utilization of infeasible individuals is proposed to increase search efficiency and avoid falling into the local optimum. Finally, a replacement mechanism is suggested to take advantage of the infeasible individuals in the population with better objective function values and constraint violation degree, and thus both diversity and convergence are well coordinated. A comparative experiment with four other excellent constrained multi-objective algorithms was implemented on standard constrained multi-objective optimization problems (CF series), and the results showed that the diversity and convergence of our algorithm were both improved.

INDEX TERMS Constrained many-objective optimization, ε-constrain handling techniques, differential evolution algorithm, MOEA/D.

I. INTRODUCTION

In practical engineering applications, there are many constrained multi-objective optimization problems (CMOPs) [1], [2] in which multiple objectives and constraints need to be optimized. Without a loss of generality, CMOPs can be formulated as formula (1):

$$\begin{aligned} &\text{minimize } F(\boldsymbol{X}) = (f_1(\boldsymbol{X}), f_2(\boldsymbol{X}), \ldots, f_m(\boldsymbol{X})), \\ &\text{subject to} \begin{cases} g_i(\boldsymbol{X}) \le 0, & j = 1, 2, \ldots, p \\ h_i(\boldsymbol{X}) = 0, & j = p+1,\ p+2, \ldots, p+q \end{cases} \\ &\text{where } \boldsymbol{X} = \left(x^1, x^2, \ldots, x^n\right) \in \boldsymbol{R}^n, x^i \in \left[L^i, U^i\right] \end{aligned} \tag{1}$$

where $\boldsymbol{X}$ is the decision vector; x^i is the i-th decision variable; L^i and U^i are the lower and upper bounds of x^i, respectively; n is the number of decision variables; $\boldsymbol{R}^n$ is the decision space; $f_i(\boldsymbol{X})$ is the objective function; m is the number of objective functions; $g_i(\boldsymbol{X})$ is the j-th inequality constraint;$h_j(\boldsymbol{X})$ is the $(j-p)$-th equality constraint; p is the number of inequality constraints, and $(q-p)$ is the number of equality constraints. The constraint violation of individual $\boldsymbol{X}$ on the j-th constraint is calculated as formula (2):

$$C_j(X) = \begin{cases} \max\left(0,\ g_j(X)\right), & j = 1, \ldots, p \\ \max\left(0,\ \left|h_j(X) - \delta\right|\right), & j = p+1, \ldots, p+q \end{cases} \tag{2}$$

The feasible region $\boldsymbol{\Omega}$ of a CMOP is a subspace of the decision space $\boldsymbol{R}^n$, and it can be defined as formula (3), where $\boldsymbol{R}^m$ is called the objective space. The attainable objective set is defined as the set $\boldsymbol{\Theta} = \{F(\boldsymbol{X}) | \boldsymbol{X} \in \boldsymbol{\Omega}\}$ [2].

$$\boldsymbol{\Omega} = \left\{\boldsymbol{X} \in \boldsymbol{R}^n \left| C_j(\boldsymbol{X}) = 0, \quad j = 1, \ldots, q \right.\right\} \tag{3}$$

As a promising method to solve the CMOPs, constrained multi-objective evolutionary algorithms (CMOEAs)

The associate editor coordinating the review of this manuscript and approving it for publication was Pasquale De Meo.

consist of two aspects, multi-objective evolutionary algorithms (MOEAs) and constraint handling techniques [3]. In recent decades, a variety of MOEAs have been developed, such as a decomposition-based archiving approach for multi-objective evolutionary optimization [4] and multi-objective particle swarm optimization algorithm [5], [6], and MOEA based on decomposition (MOEA/D) [4], [7], [8] is one of the most attractive algorithms. Many studies have shown its advantages with high search ability and high compatibility on unconstrained multi-objective optimization problems. Zhang *et al.* proposed an efficient decomposition-based archiving approach (DAA) [4] inspired from the decomposition strategy for dealing with multi-objective optimization. At each generation, only one non-dominated solution lying in a subspace is chosen to be used for updating the external archive in consideration of its diversity. When solving CMOPs, many infeasible solutions are generated during the optimization process because of the constraints, and thus constraint handling techniques are needed to measure the advantages and disadvantages of feasible solutions and infeasible solutions. The current constraint handling techniques mainly involve penalty function method, stochastic ranking, feasibility rule, multi-objective optimization, ε-constraint, and dual population storage [9].

Although MOEA/D has shown excellent performance on problems without constraints, few constraint handling techniques have been combined with MOEA/D to solve CMOPs. Jan and Zhang introduced a penalty function into MOEA/D using differential evolution (DE) (CMOEA/D-DE-ATP) [10] for penalizing infeasible solutions, and a dynamic threshold strategy is used to control the penalty coefficient, and thus more infeasible regions near the feasible solutions are explored to improve the population diversity, but it requires setting six other parameters. Jain *et al.* proposed C-MOEA based on dominance (C-MOEA/D) [11], in which the comparison between the offspring and the corresponding parent is made according to the following rules: 1) The feasible solution is always better than the infeasible solution. 2) The individual with a better objective function value wins when the compared individuals are feasible solutions. 3) The smaller constraint violation is taken seriously when two individuals are both infeasible solutions. However, C-MOEA/D's neglect of the valid information carried by the infeasible solutions reduces the search space and causes it to fall into premature convergence on highly constrained problems. Li *et al.* extended MOEA/DD, denoted as C-MOEA based on dominance and decomposition (C-MOEA/DD) [12], to solve constrained optimization problems. By introducing an infeasible solution with the largest constraint violation into the population update procedure when it is associated with an isolated subregion, C-MOEA/DD also makes use of the infeasible solutions, which is important for population diversity. Fan *et al.* proposed an angle-based constrained dominance method, named MOEA/D-ACDP [13], which adopts the angle information of the objective functions to enhance the population diversity in the infeasible region. In MOEA/D-ACDP, for two infeasible solutions, if the angle of the solutions is greater than a given threshold, then they are considered to be non-dominated by each other. For a feasible solution and an infeasible solution, if the angle of the solutions is less than the given threshold, then the feasible solution is better; otherwise, they are non-dominated. These studies made use of the infeasible solutions in the comparison and selection procedure, but none of them introduced an excellent infeasible solution into the mutation and crossover operators.

The ε constrained differential evolution (εDE) [14] proposed by Takahama and Sakai is the first try to combine the ε constrained method and DE. And, the εDE with an archive and gradient-based mutation (εDEag) [15] proposed by Takahama and Sakai is a representative constraint handling method. It has three contributions: (a) In order to improve the stability, the parent can generate another child when the previous one is not better than the parent, and, in order to keep the diversity and efficiency, an archive is utilized which has many individuals initially and is updated using defeated individuals in survivor selection of DE. (b) In order to improve the usability, the parameter value is automatically set based on the state of an initial archive. (c) In order to improve the efficiency, the ε level is increased in order to enlarge searching region and find individuals with better objective values. Furthermore, Takahama and Sakai presented εDEkr [16] by combining the ε-constraint with the estimated comparison. Bu *et al.* suggested SRS-εDEag [17], which selects the infeasible individuals for gradient-based repair. An improved version of εDEag was proposed named εDE-PCGA [18], which combines with pre-estimated comparison gradient based approximation. Saul *et al.* proposed a memetic algorithm [19], in which DE is used as the global search algorithm, and local search is implemented by a mathematical programming method. Asafuddoula *et al.* combined the ε-constraint with MOEA/D, which adaptively decides the ε level based on the number of violated constraints and the number of feasible solutions [20].

To improve the distribution and convergence of the obtained solution set in CMOPs, a DE mutation operator with infeasible solutions in MOEA/D is proposed in this paper. The main contributions of this paper can be summarized as follows:

① We propose an adaptive ε-level strategy based on both evolution generation and constraint violation, which can make better use of the information of excellent infeasible individuals and improve population diversity.

② We present an adaptive DE mutation strategy to increase search efficiency and avoid falling into the local optimum.

③ We suggest a replacement mechanism to take full advantage of the infeasible individuals with a better objective function for coordinating both diversity and convergence of population.

The rest of the paper is structured as follows: Section II provides some background knowledge. Section III introduces the proposed method. Section IV presents the experiment descriptions, results, and discussion. Section V concludes this paper and identifies some directions for future studies.

II. RELATED WORK

A. MOEA/D

A MOP is decomposed into many single-objective optimization problems by MOEA/D. Single-objective optimization problems are formulated by scalarizing functions using uniformly distributed weight vectors $\omega(i = 1, 2, \ldots, N)$, where N is the population size. Popular scalarizing approaches include weighted sum, weighted Tchebycheff, and boundary intersection approaches. In this paper, we use the penalty-based boundary intersection (*PBI*) approach [10]. The scalar of the *PBI* function can be formulated by formulas (4) and (5) as follows:

$$\begin{aligned} &\text{minimize } g^{PBI}\left(\boldsymbol{X}|\boldsymbol{\omega}, Z^*\right) = d_1 + \theta d_2, \\ &\text{subject to } \boldsymbol{X} \in \boldsymbol{\Omega} \end{aligned} \tag{4}$$

$$\begin{aligned} d_1 &= \frac{\left\|\left(\boldsymbol{F}(\boldsymbol{X}) - \boldsymbol{Z}^*\right)^{\mathrm{T}} \boldsymbol{\omega}\right\|}{\|\boldsymbol{\omega}\|}, \\ d_2 &= \left\|\boldsymbol{F}(\boldsymbol{X}) - \left(\boldsymbol{Z}^* + d_1 \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|}\right)\right\| \end{aligned} \tag{5}$$

where the reference point $\boldsymbol{Z}_i^* = \min\left\{g_i^{PBI}(X)\,|\boldsymbol{X} \in \boldsymbol{\Omega}\right\}$, $i = 1, 2, \ldots, m$ is the best objective vector in the current population, θ is the parameter of the *PBI*, and its range is $\theta \geq 0$.

B. THE DE PROCESS

Differential evolution [21] has been widely utilized in many fields due to its numerous advantages, including simplicity, efficiency, and ease of implementation. It includes three main evolutionary operators: mutation, crossover, and selection. The mutation operator produces a mutation vector $\boldsymbol{V}_i = \left(v_i^1, v_i^2, \ldots, v_i^n\right)$ corresponding to target vector $\boldsymbol{X}_i$ through the base vector $\boldsymbol{X}_{r1}$ and the difference vector $(\boldsymbol{X}_{r2} - \boldsymbol{X}_{r3})$. Besides, $\boldsymbol{V}_i$ is shown as formula (6).

$$\boldsymbol{V}_i = \boldsymbol{X}_{r1} + F \times (\boldsymbol{X}_{r2} - \boldsymbol{X}_{r3}) \tag{6}$$

where F is the scaling factor, and indices $r1, r2, r3$ are diverse integers uniformly chosen from the set $\{1, 2, \ldots, N\}$.

The crossover operator is applied to the target vector $\boldsymbol{X}_i$ and its mutant vector $\boldsymbol{V}_i$, and the generated trial vector $\boldsymbol{U}_i = \left(u_i^1, u_i^2, \ldots, u_i^n\right)$ is implemented as formula (7).

$$u_i^j = \begin{cases} v_i^j, & \text{rand}(j) \leq CR \\ x_i^j, & \text{otherwise} \end{cases} \tag{7}$$

where u_i^j is the j-th dimension of $\boldsymbol{U}_i$; and the crossover probability CR is a single parameter within the interval [0, 1], which controls the fraction of vector components inherited from the mutation vector. The target vector $\boldsymbol{X}_i$ is compared with its trial vector $\boldsymbol{U}_i$ according to the selection strategy.

C. ε-CONSTRAINT

The ε-constraint [14] compares a pair of individuals with relaxed constraints. If the constraint violation values formulated as formula (8) of the two individuals are equal or both less than a pre-set number ε ($\varepsilon \geq 0$), which is controlled as formula (9) [15], then the individual with a better objective value wins; otherwise, the one with the smaller constraint violation wins. In this equation, parameter t is the iteration of the population; T_c is the control generation, set as $0.2T_{\max}$; cp controls the reducing speed of ε, which is set as 5; and initial ε level $\varepsilon(0)$ is the constraint violation of the top θ-th individual $\boldsymbol{X}_\theta$ in the initial search points.

$$\begin{aligned} G(X) = &\sum\nolimits_{i=1}^{p} \max\left(0, g_i(\boldsymbol{X})\right) \\ &+ \sum\nolimits_{j=1}^{q} \max\left(0, |h_j(\boldsymbol{X})|\right) \end{aligned} \tag{8}$$

$$\begin{aligned} \varepsilon(t) &= \begin{cases} \varepsilon(0) \times (1 - t/T_c)^{cp}, & 0 < t < T_c \\ 0, & t > T_c \end{cases} \\ \varepsilon(0) &= G(\boldsymbol{X}_\theta) \end{aligned} \tag{9}$$

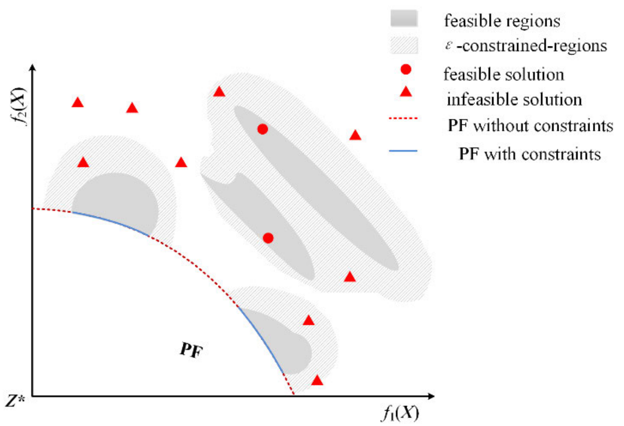
III. INFEASIBLE SOLUTIONS-BASED MOEA/D

In this section, a constrained handling technique, called ε-MOEA/D-DE, is proposed to better balance diversity and convergence based on making use of infeasible solutions. First, we propose an adaptive ε-level based on both the generation and the constraint violation value. Then, a replacement mechanism proposed takes full advantage of the infeasible individuals in the population with a better objective function and avoids the other individuals by a one-to-one comparison.

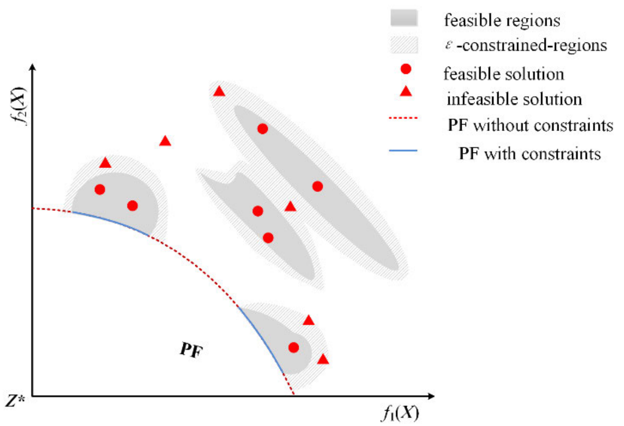
A. ADAPTIVE ε-CONSTRAIN HANDLING TECHNIQUES

Comparing feasible and infeasible solutions is a fundamental issue in constrained evolutionary optimization. Diversity and convergence can both be enhanced by allowing infeasible solutions to take part in the evolution [11]. By adjusting parameter ε, the ε-constraint coordinates feasible solutions and infeasible solutions. In prior work, the value of ε either simply changed because of the iteration increase or for constraints. None of the previous studies investigated the ε value being influenced by both the iteration and the type of constraints. In this part, an adaptive ε-level is proposed considering both the generation and the constraint violation value.

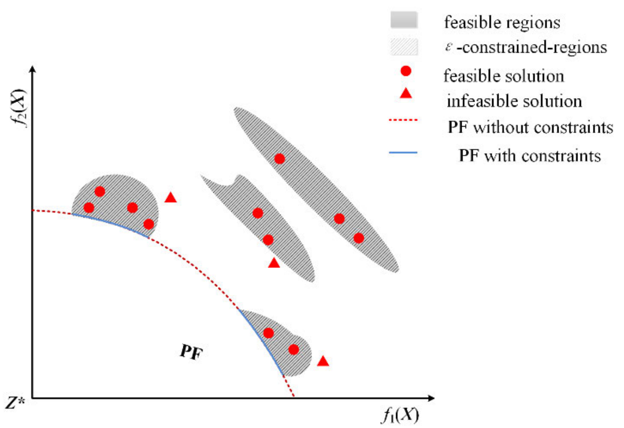
In Figure 1, the triangles denote infeasible solutions, the dots denote the feasible solutions, the red dotted lines display the Pareto front (PF) without constraints, the blue curve is PF with constraints, the gray areas represent the feasible regions, and the slashed areas are the ε-constrained-regions. As shown in Figure 1 (a), in the early stage, the number of infeasible solutions may be much larger than the number of feasible solutions. In order to quickly explore the feasible regions, more infeasible solutions should be participated in the evolution, so the ε-constrained-regions should be large. The greater the proportion of infeasible solutions, the larger the level of ε should be. As shown in Figure 1 (b), as the iteration progresses, the proportion of infeasible solutions



(a) In the early stage, the number of infeasible solutions is much larger than the number of feasible solutions. The ε-constrained-regions should be large, so that more infeasible solutions can guide the population expanding the exploration edges and jumping out of the isolated feasible regions.



(b) In the middle stage, the number of infeasible solutions gradually increases. The ε-constrained-regions need reducing, so that the population can approach the feasible PF while exploring new areas.



(c) In the later stage, the number of infeasible solutions is much larger than the number of feasible solutions. The ε level should be large, so that more infeasible solutions can guide the population expanding the exploration edges and jumping out of the isolated feasible regions.

FIGURE 1. Adaptive ε-constrained level.

in the population gradually increases. The population needs to approach the feasible PF while exploring new areas, the ε-constrained-regions need to be gradually reduced. In the later stage, there are increasingly more feasible individuals to ensure the convergence of the population, and it is worth noting that the final task of evolution is to achieve feasible Pareto solutions; thus, the population needs to be promptly guided by the constrained PFs. Therefore, in this case, there needs a large proportion of feasible solutions, and then the ε value should gradually tend towards zero as the evolutionary generation increases. Based on the above discussion, the improved ε level is calculated by formula (10), where it is adaptive based on both the evolutionary generation and the constraint value.

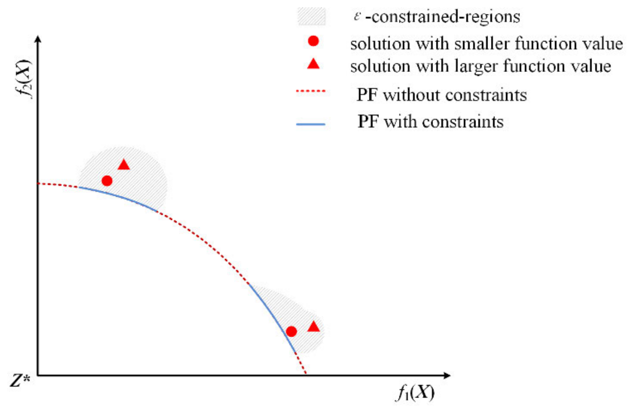
$$\varepsilon(t) = \begin{cases} \varepsilon(0) \times (1 - t/T_c)^{cp}, & 0 < t < T_c \\ 0, & t \geq T_c \end{cases}$$
$$\varepsilon(0) = \frac{N_{in}}{N} \sum\nolimits_{i=1}^{N} G(\boldsymbol{X}_i) \tag{10}$$

where cp controls the reducing speed of ε set which is as 5, referred to [15], T_c is the control generation, which is calculated as formula (11), and N_{in} is the number of infeasible solutions.

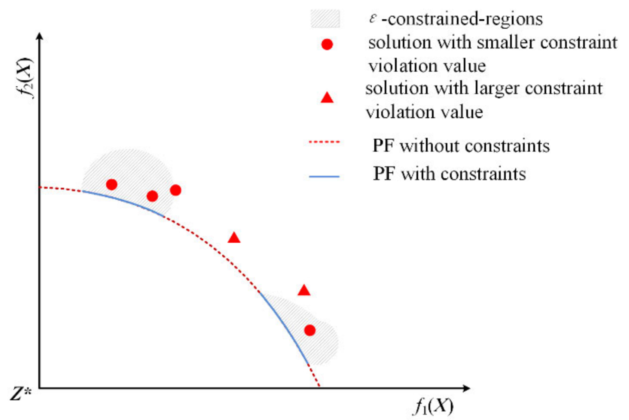
$$T_c = t, \frac{1}{N} \sum\nolimits_{i=1}^{N} G(\boldsymbol{X}_i(t)) \leq 0.01 \tag{11}$$

B. REPLACEMENT MECHANISM

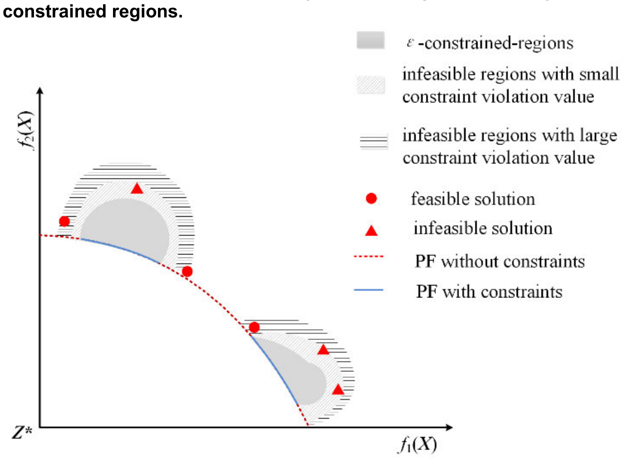
The ε-constraint utilizes a one-to-one replacement mechanism that may result in the loss of a part of the Pareto solutions. The one-to-one replacement may miss an individual who has a constraint violation value slightly larger than ε but has the best objective function value. The replacement mechanism proposed in this part aims at taking full advantage of the infeasible individuals in the population with a better objective function and avoiding individuals by a one-to-one comparison. For example, in Figure 2 (a), the triangles denote the ε-constrained-feasible solutions with larger objective function values, and the dots denote the ε-constrained-feasible solutions with smaller objective function values. This distribution indicates that the dots are better for the evolution. very likely on the boundary of the feasible region. In Figure 2 (b), the triangles denote the solutions with larger constraint violation values, and the dots denote the solutions with smaller constraint violation values. As shown in the figure, by avoiding such solutions, many feasible and infeasible solutions may be generated near the ε-constrained regions, which provides an advantage in searching for the optimal infeasible individuals, many feasible and infeasible individuals may be generated near the feasible region. Thus, an archive $\boldsymbol{A}$ is introduced to store the effective infeasible individuals. During each generation t, for a target vector $\boldsymbol{X}_{i,t}$ in population $\boldsymbol{P}_t$, a trial vector $\boldsymbol{U}_{i,t}$ is generated by DE mutation and crossover operators. Here, $f(\boldsymbol{U}_{i,t})$ ($f(\boldsymbol{X}_{i,t})$) and $G(\boldsymbol{U}_{i,t})$ ($G(\boldsymbol{X}_{i,t})$) are the objective function value and the constraint violation, respectively, at a point $\boldsymbol{U}_{i,t}$ ($\boldsymbol{X}_{i,t}$). To ensure infeasible solutions with better objective functions are included to approach the true constrained PFs, $\boldsymbol{X}_{i,t}$ is compared with $\boldsymbol{U}_{i,t}$ based on the proposed replacement rule as



(a) For the ε-constrained-feasible solutions, those with smaller objective function values are better for the evolution.



(b) For the solutions with smaller constraint violation values can help search for the optimal solutions by surrounding the boundary of the ε-constrained regions.



(c) For the solutions with a slightly larger constraint violation value than ε, store those with smaller function values.

FIGURE 2. The effects of the effective infeasible solutions.

follows: If the constraint violation values of $\boldsymbol{U}_{i,t}$ and $\boldsymbol{X}_{i,t}$ are equal or both less than $\varepsilon(t)$ (formulated as equation (10)), and if $f(\boldsymbol{U}_{i,t}) < f(\boldsymbol{X}_{i,t})$ or if $G(\boldsymbol{U}_{i,t}) < G(\boldsymbol{X}_{i,t})$, then $\boldsymbol{U}_{i,t}$ will replace the compared target vector $\boldsymbol{X}_{i,t}$, this can be described as formula (12); otherwise, if the constraint violation values of $\boldsymbol{U}_{i,t}$ and $\boldsymbol{X}_{i,t}$ are more than $\varepsilon(t)$, $G(\boldsymbol{U}_{i,t}) > G(\boldsymbol{X}_{i,t})$ and $f(\boldsymbol{U}_{i,t}) < f(\boldsymbol{X}_{i,t})$, then $\boldsymbol{X}_{i,t}$ cannot be replaced and $\boldsymbol{U}_{i,t}$ should be stored into the predefined archive $\boldsymbol{A}$. This can be described by formula (13).

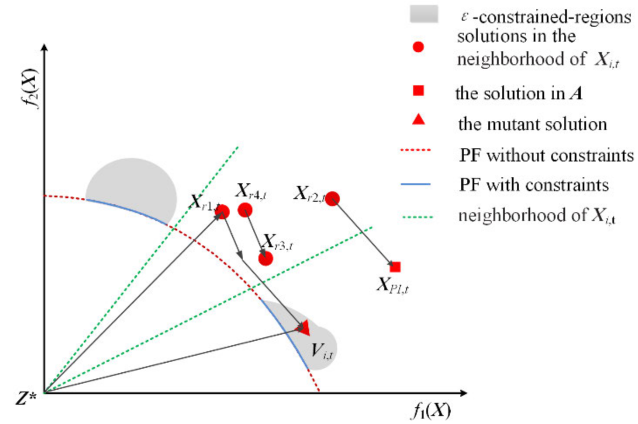


FIGURE 3. The DE mutation operator named DE/rand-infeasible-neighborhood.

$$U_{i,t} \text{ replace } X_{i,t} \;\Leftrightarrow \begin{cases} f(\boldsymbol{U}_{i,t}) < f(\boldsymbol{X}_{i,t}),\, G(\boldsymbol{U}_{i,t}), & G(\boldsymbol{X}_{i,t}) \le \varepsilon(t) \\ f(\boldsymbol{U}_{i,t}) < f(\boldsymbol{X}_{i,t}), & G(\boldsymbol{U}_{i,t}) = G(\boldsymbol{X}_{i,t}) \\ G(\boldsymbol{U}_{i,t}) < G(\boldsymbol{X}_{i,t}). & \end{cases} \tag{12}$$

$$\boldsymbol{U}_{i,t} \text{ stored in } \boldsymbol{A} \Leftrightarrow f(\boldsymbol{U}_{i,t}) < f(\boldsymbol{X}_{i,t}), \\ G(\boldsymbol{U}_{i,t}), G(\boldsymbol{X}_{i,t}) > \varepsilon(t) \wedge G(\boldsymbol{U}_{i,t}) > G(\boldsymbol{X}_{i,t}) \tag{13}$$

C. DE MUTATION OPERATOR WITH INFEASIBLE SOLUTIONS

In this section, a DE mutation operator, named DE/rand-infeasible-neighborhood, is proposed to introduce effective infeasible solutions into producing the mutant vector $\boldsymbol{V}_{i,t}$. As shown in Figure 3, to utilize the information carried by infeasible solutions to guide the population exploring the constraint boundaries and jumping out of an isolated feasible region, an infeasible solution (denoted as $\boldsymbol{X}_{PI,t}$) is introduced into the mutation operator to form the difference vector. The proposed DE mutation operator is calculated as formula (14).

$$\boldsymbol{V}_{i,t} = \boldsymbol{X}_{r1,t} + F_{1,t} \times (\boldsymbol{X}_{PI,t} - \boldsymbol{X}_{r2,t}) \\ + F_{2,t} \times (\boldsymbol{X}_{r3,t} - \boldsymbol{X}_{r4,t}) \tag{14}$$

where four different indexes $r1$, $r2$, $r3$, and $r4$ are randomly selected from the neighbor of $\boldsymbol{X}_{i,t}$, and, to learn information from the infeasible solutions, index PI is selected from the achieve $\boldsymbol{A}$, scaling factors $F_{1,t}$, and $F_{2,t}$ is used to control the two difference vectors in different evolution periods.

In the early stage of evolution, the population may contain only infeasible solutions, and, in this scenario, the population should quickly approach feasible regions to find feasible solutions. Infeasible solutions with a better objective value can be exploited to expand the exploration range, as discussed in Sections III-A and III-B, the scaling factor $F_{1,t}$ should be large while the $F_{2,t}$ should be small. The parameter $F_{1,t}$, $F_{2,t}$

is controlled as in formula (15).

$$F_{1,t} = F_{\min} + (F_{\max} - F_{\min}) \times \left(2 - \exp\left(\frac{t}{G_{\max}} \times \ln 2\right)\right)$$
$$F_{2,t} = F_{\min} + (F_{\max} - F_{\min}) \times \left(\exp\left(\frac{t}{G_{\max}} \times \ln 2\right) - 1\right) \tag{15}$$

where t is the iteration, $G_{\max}$ is the maximum number of t, and $F_{\min}, F_{\max}$ are the ranges of $F_{1,t}$ and $F_{2,t}$. As the iteration t increases from 0 to $G_{\max}$, $F_{1,t}$ changes from $F_{\max}$ to $F_{\min}$, while $F_{2,t}$ varies from $F_{\min}$ to $F_{\max}$.

D. PROCESS OF THE PROPOSED ALGORITHM

For ease of understanding, in this section, firstly, the proposed algorithm, an adaptive ε-constraint multi-objective evolutionary algorithm based on decomposition and differential evolution (ε-MOEA/D-DE), is briefly described.

For the **step 2:** 4) **Update of Solutions**, firstly, the indexes in set $\boldsymbol{B}_i$ are put into a temporary set S, and a temporary index c is set to 0. Afterwards, the follow-up procedure is done, until $c = nr$ or S is empty, to decide whether the solutions in the neighbor associated with $\boldsymbol{X}_i$ can be replaced by the offspring $\boldsymbol{U}_i$:

If the constraint violation value G of $\boldsymbol{U}_i$ equals that of $\boldsymbol{X}_j$, $j \in S$, and the *PBI* function value g^{PBI} of $\boldsymbol{U}_i$ is smaller than that of $\boldsymbol{X}_j$, then replace $\boldsymbol{X}_j$ by $\boldsymbol{U}_i$, and c plus one.

If both $G(\boldsymbol{U}_i)$ and $G(\boldsymbol{X}_j)$ are smaller than $\varepsilon(t)$, and $g^{PBI}(\boldsymbol{U}_i)$ is smaller than $g^{PBI}(\boldsymbol{X}_j)$, then replace $\boldsymbol{X}_j$ by $\boldsymbol{U}_i$, and c plus one.

If $G(\boldsymbol{U}_i)$ is smaller than $G(\boldsymbol{X}_j)$, then replace $\boldsymbol{X}_j$ by $\boldsymbol{U}_i$, and c plus one.

If any of the above cases does not happened, then retain $\boldsymbol{X}_j$, and, if $g^{PBI}(\boldsymbol{U}_i)$ is smaller than $g^{PBI}(\boldsymbol{X}_j)$ simultaneously, then store $\boldsymbol{U}_i$ into the achieve $\boldsymbol{A}$.

It is worth noting that the index j is selected from set $\boldsymbol{S}$ randomly, and is delete from set $\boldsymbol{S}$ after the above comparison.

IV. EXPERIMENTAL SETUP

This section describes four experimental designs that were developed for investigating the performance of the proposed algorithm, the experiment I is to analyze the effects of the scaling factors, $F_{\min}$ and $F_{\max}$; the experiment II is to verify the adaptive ε-level, the experiment III is to verify the replacement mechanism and the DE mutation operator, and the experiment IV is to verify the proposed algorithm ε-MOEA/D-DE. First, we provide the experimental environment. Then, we describe the benchmark problems used in empirical studies. Afterwards, we introduce the performance metrics used for evaluating the performance of an evolution multi-objective optimization (EMO) algorithm. Finally, we briefly describe the parameters used in the four experiments, and we provide the experimental results and an explanation.

Algorithm 1 ε-MOEA/D-DE

Input:
1) a CMOP;
2) $G_{\max}$: the maximum number of iterations
3) N: the number of the subproblems;
4) a set of N weight vectors: $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_N$;
5) T: the size of the neighborhood of each weight vector;
6) $F_{\min}, F_{\max}$: the ranges of the scaling factors;
7) *CR*: the crossover probability;
8) θ: the parameter of *PBI* function;
9) *nr*: the maximal number of solutions replaced by an offspring.

Output: A set of constrained non-dominated feasible solutions.

Step 1: Initialization:
1) Set the iteration $t = 1$;
2) Decompose the CMOP into N sub-problems associated with $\boldsymbol{\omega}_{1,t}, \boldsymbol{\omega}_{2,t}, \ldots, \boldsymbol{\omega}_{N,t}$;
3) Generate an initial population $\boldsymbol{P}_t = \{\boldsymbol{X}_{1,t}, \boldsymbol{X}_{2,t}, \ldots, \boldsymbol{X}_{N,t}\}$, where $\boldsymbol{X}_{i,t}$ is the current solution to the i-th subproblem. The j-th dimension $x_{i,t}^j$ of $X_{i,t}$ is produced by $x_{i,t}^j = L^i + \mathrm{rand}(t) \times U^i$, $j = 1, 2, \ldots, n$;
4) Compute the Euclidean distance between any two weight vectors and obtain T closest weight vectors to each weight vector. For each $i = 1, 2, \ldots, N$, set $\boldsymbol{B}_i = \{i_1, i_2, \ldots, i_T\}$, where $\omega_{i1,t}, \omega_{i2,t}, \ldots, \omega_{iT,t}$ are the T closest weight vectors to $\omega_{i,t}$;
5) Initialize $\boldsymbol{Z}^* = (z_1, z_2, \ldots, z_m)$;
6) Initialize the achieve $\boldsymbol{A} = \emptyset$.

Step 2: Population update
For $i = 1, 2, \ldots, N$, do:
1) **Production:** Generate the mutant solution $\boldsymbol{V}_{i,t}$ by the DE/rand-infeasible-neighborhood mutation operator, then perform a crossover operator with probability *CR* to produce a new solution $\boldsymbol{U}_{i,t}$;
2) **Repair:** If any element of $\boldsymbol{U}_{i,t}$ is out of the boundary, its value is reset to be randomly selected value inside the boundary;
3) **Update of** Z^***:** For each $j = 1, 2, \ldots, m$, if $z_j > g_j^{PBI}(\boldsymbol{U}_{i,t})$, then set $z_j = g_j^{PBI}(\boldsymbol{U}_{i,t})$;
4) **Update of Solutions**: Set $c = 0$, $\boldsymbol{S} = \boldsymbol{B}_i$, and then do the following:
4.1) If $c = nr$ or $\boldsymbol{S}$ is empty, go to **Step3**. Otherwise, select an index j from $\boldsymbol{S}$ randomly;
4.2) Calculate the constraint violation values and the *PBI* values of $\boldsymbol{X}_{j,t}$ and $\boldsymbol{U}_{i,t}$:
4.2.1) if $G(\boldsymbol{U}_{i,t}) = G(\boldsymbol{X}_{j,t})$ and $g^{PBI}(\boldsymbol{U}_{i,t}|\boldsymbol{\omega}_j, \boldsymbol{Z}^*) \leq g^{PBI}(\boldsymbol{X}_{j,t}|\boldsymbol{\omega}_j, \boldsymbol{Z}^*)$, then set $\boldsymbol{X}_{j,t} = \boldsymbol{U}_{i,t}$ and $c = c + 1$;
4.2.2) if $G(\boldsymbol{U}_{i,t}), G(\boldsymbol{X}_{j,t}) \leq \varepsilon(t)$ and $g^{PBI}(\boldsymbol{U}_{i,t}|\boldsymbol{\omega}_j, \boldsymbol{Z}^*) \leq g^{PBI}(\boldsymbol{X}_{j,t}|\boldsymbol{\omega}_j, \boldsymbol{Z}^*)$, then set $\boldsymbol{X}_{j,t} = \boldsymbol{U}_{i,t}$ and $c = c + 1$;
4.2.3) otherwise, if $G(\boldsymbol{U}_{i,t}) < G(\boldsymbol{X}_{j,t})$, then set $\boldsymbol{X}_{j,t} = \boldsymbol{U}_{i,t}$ and $c = c + 1$;
4.2.4) if not any of the above cases and $f(\boldsymbol{U}_{i,t}) < f(\boldsymbol{X}_{j,t})$, then store $\boldsymbol{U}_{i,t}$ into the achieve A;
4.3) Remove j from S and go to 4.1).

Step 3: Termination: If $t \geq G_{\max}$, output $\boldsymbol{P}_t = \{\boldsymbol{X}_{1,t}, \boldsymbol{X}_{2,t}, \ldots, \boldsymbol{X}_{N,t}\}$. Otherwise, go to **Step 2**.

Step 4:$t = t + 1$. Set $\boldsymbol{A} = \emptyset$.

A. BENCHMARK PROBLEMS

The PC configuration of the system in the experiments was as follows: Windows 10; RAM: 2 G; CPU: G620; CPU 2.60 GHz; computer language: SCILAB 6.1. As a basis for the comparisons, some well-known test problems, as CF1-CF7 [22], were involved in the experiments.

In the CF1-CF7 series, the number of decision variables is ten, and the number of objectives is two. The corresponding PF lies in $f_i \in [0, 1]$. The test problems relate to the PF that has discrete points (CF1), convex (CF2), concave (CF3), continuous (CF4, CF5), disjointed (CF1, CF2, CF3), and differently scaled (CF2, CF3, CF4, CF5).

B. PERFORMANCE METRICS

In this empirical work, we consider the following two widely used performance metrics. Both can simultaneously measure the convergence and diversity of obtained solutions. Convergence describes the closeness of the obtained Pareto front to the true PFs. On the other hand, Diversity depicts how the solutions in the obtained Pareto are distributed.

1) Inverted generation distance (IGD) metric [23]: Let $\boldsymbol{P}^*$ be the ideal Pareto front set, and C is an approximate Pareto front set achieved by the evolutionary multi-objective algorithm. The IGD metric denotes the distance between $\boldsymbol{P}^*$ and C. It is defined by formula (16). The lower is the IGD value, the better is the quality of S for approximating the whole Pareto front set.

$$IGD = \frac{\sum_{y^* \in \boldsymbol{P}^*} d\left(\boldsymbol{y}^*, \boldsymbol{C}\right)}{\left\|\boldsymbol{P}^*\right\|}$$
$$d\left(\boldsymbol{y}^*, \boldsymbol{C}\right) = \min_{y \in C} \left\{ \sqrt{\sum_{i=1}^{m} \left(y_i^* - y_i\right)^2} \right\} \tag{16}$$

2) Hypervolume (HV) Metric [24]: The HV is the m dimensional volume of the region enclosed by the obtained Pareto front set and a dominated point $\boldsymbol{r}$ in the objective space. A high HV indicates a good Pareto front set in both the convergence toward Pareto front and the diversity to approximate a wide range of Pareto front. The reference point for computing the HV is set to $\boldsymbol{r} = (1.2, 1.2, \ldots 1, 2)$ and the HV value is computed by formula (17).

$$HV\left(\boldsymbol{P}\right) = \mathrm{vol}\left(\bigcup_{X \in \boldsymbol{P}} \left[f_1\left(\boldsymbol{X}\right), r_1\right] \times \ldots \times \left[f_m\left(\boldsymbol{X}\right), r_m\right] \right) \tag{17}$$

where vol $(\cdot)$ indicates the Lebesgue measure.

C. EMO ALGORITHMS FOR COMPARISONS

To verify the adaptive ε-level proposed in Section III-A, three state-of-the-art EMO algorithms were considered, including εDE [14], εDEag [15], and SRS-εDEag [17], for comparison. Since εDE, εDEag and SRS-εDEag are proposed for constrained single problems, for a fair comparison, all these algorithms were carried out under the framework of MOEA/D and only those strategies of ε-constraint handling techniques could be used, and the DE operator was carried out by equations (6) and (7).

To verify the replacement mechanism proposed in Section III-B and the DE mutation operator proposed in Section III-C, four most representative algorithms were used for comparison: CMOEA/D-DE-ATP [10], C-MOEA/D [11], C-MOEA/DD [12] and MOEA/D-ACDP [13].

To verify the proposed algorithm ε-MOEA/D-DE, three algorithms were considered, including C-MOEA/DD, MOEA/D-ACDP, C-FRORI [7], and C-NSGA-III [11], for comparison. It should be noted that C-FRORI was incorporated into NSGA-II.

D. PARAMETER SETTINGS

For a fair comparison, parameters of different algorithms are set to be the same, and the parameters used in all the four experiments are listed in Table 1.

TABLE 1. The parameter setting of four experiments.

Parameter	Value
The independently run times	50
The maximum number of iterations: G_{max}	300,000
The population size: N	300
The crossover probability: CR	0.9
The scaling factor: F	0.5
The mutation probability: p_m	$1/n$
The mutation distribution index: η	20
The neighborhood size: T	$0.1N$
The neighborhood selection probability: δ	0.9
The parameter of PBI function: θ	5
The maximal number of solutions replaced by a child: nr	$0.01N$

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. PARAMETER ANALYSIS

The parameters F_1 and F_2 have a direct impact on the evolution. Therefore, in this section, how $F_{\max}$ and $F_{\min}$ affect the results is discussed on test problems CF1-CF7. In experiment I, which is an orthogonal experimental design, $F_{\max}$ ranges from 0.5 to 1 with the step length set as 0.1, and, $F_{\min}$ ranges from 0.5 to 1 with the step length set as 0.1. Two settings are selected to show the effects, one is the extremist setting and the other one got the best IGD values. Table 2 summarizes the IGD metric statistics based on 50 independent runs for each test instance with these two settings. In setting 1, $F_{\min} = 0$, $F_{\max} = 1$, and, in setting 2, $F_{\min} = 0.3$, $F_{\max} = 0.9$.

As shown in Table 2, although F_1 and F_2 control the search scope, they modulate the exploration ability of feasible and infeasible solutions. The variation of these parameters conducts little influence on the evolution. For example, $F_{1,t}$ decreased during the execution of the algorithm, thus leading to a transition from exploration to exploitation of the infeasible solutions. The larger scaling factor value introduces better diversity, but it hampers the convergence. In contrast, although the smaller scaling factor value speeds up the

TABLE 2. **The comparison of *IGD* for three algorithms to verify the proposed adaptive ε-level.**

Problem	parameters	Min	Max	Mean	Std
CF1	set1	0.000682	0.001147	0.000859	**0.000110**
	set2	**0.000304**	**0.001092**	**0.000528**	0.000189
CF2	set1	0.002733	0.003135	0.003420	0.002635
	set2	**0.001079**	**0.003036**	**0.003100**	**0.000435**
CF3	set1	**0.009844**	0.084188	0.052905	0.042127
	set2	0.017862	**0.070731**	**0.043601**	**0.005969**
CF4	set1	0.004964	**0.009239**	0.005232	0.003293
	set2	**0.003222**	0.009886	**0.004995**	**0.001007**
CF5	set1	**0.005881**	0.112996	0. 091730	**0.030676**
	set2	0.006872	**0.076393**	**0.075337**	0.006362
CF6	set1	0.007778	0.052996	0. 019730	0.030676
	set2	**0.005872**	**0.039396**	**0.015773**	**0.006662**
CF7	set1	0.070778	**0.092996**	0.090730	**0.003066**
	set2	**0.048872**	0.096686	**0.085773**	0.000626

convergence, it weakens the diversity and the algorithm may fall into the local optima.

B. COMPARATIVE EXPERIMENT ON ADAPTIVE ε-LEVEL

This section, first, discusses the stability of the adaptive ε-level method, then, comparative experiments were carried out between the proposed adaptive ε-level method and the three best-performing ε-constrained algorithms on the CF1-CF7 series test problems.

To verify the stability of the adaptive ε-level method, 3 independent runs were randomly selected out of 50 independent runs, and the IGD convergence curve for the representative CF1, CF3, CF5, and CF7 problems were given in Figure 4–7.

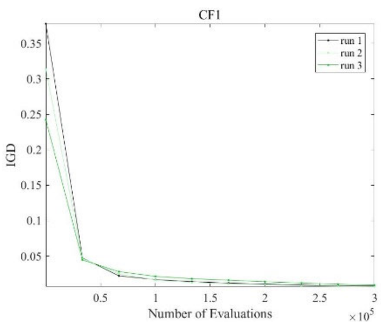


FIGURE 4. **IGD convergence curve for CF1.**

We can conclude from Figs. 4–7 that the proposed adaptive ε-level method does not affect the stability of the algorithm although the ε-level constantly changed, since the IGD values decreased steadily with the iteration for all the test problems. The adaptive ε-level method investigated the ε value being influenced by both the iteration and the type of constraints, which considered both the generation and the constraint violation value, and, lead the algorithm more stable.

Tables 3 compares the *IGD* statistics (all 50 independent runs). To detect the statistical differences systematically, a multiple-problem a Wilcoxon's test at a 0.05 significance level [25] was implemented between the proposed adaptive

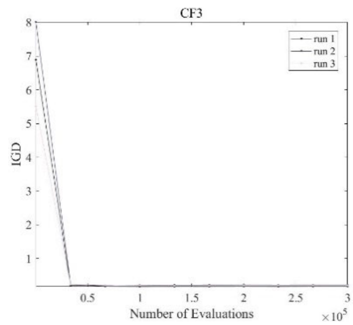


FIGURE 5. **IGD convergence curve for CF3.**

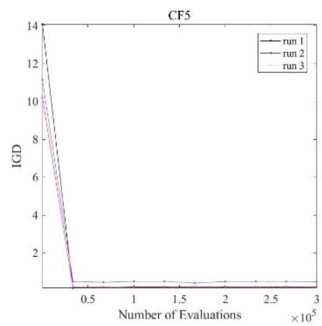


FIGURE 6. **IGD convergence curve for CF5.**

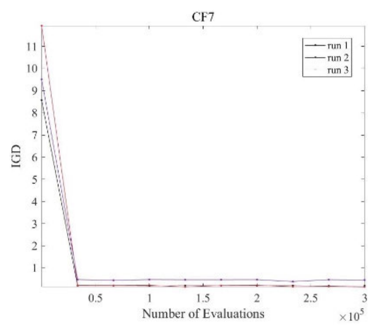


FIGURE 7. **IGD convergence curve for CF7.**

ε-level method, εDE, εDEag and SRS-εDEag. It should be noted that '+' indicates that the proposed adaptive ε method was significantly better than the comparative algorithms,

TABLE 3. **The comparison of *IGD* for three algorithms to verify the proposed adaptive ε-level.**

Problem	Algorithm	Min	Max	Mean	Std	Statistical significance
CF1	εDE	0.000682	0.001147	0.000859	0.000110	-
	εDEag	0.013855	0.023597	0.019187	0.002568	≈
	SRS-εDEag	0.000504	0.003366	0.001537	0.000680	-
	adaptive ε-level	**0.000404**	**0.001492**	**0.000828**	**0.000589**	
CF2	εDE	0.002733	0.013135	0.004203	0.002635	-
	εDEag	0.004142	0.051816	0.026779	0.014715	-
	SRS-εDEag	0.005912	0.006875	0.006374	**0.000273**	≈
	adaptive ε-level	**0.003979**	**0.005036**	**0.004100**	0.000635	
CF3	εDE	0.090844	0.251884	0.182905	0.042127	-
	εDEag	0.075301	0.142828	0.104460	0.015595	-
	SRS-εDEag	0.070131	0.178339	0.156801	0.020042	-
	adaptive ε-level	**0.047862**	**0.100731**	**0.093601**	**0.005699**	
CF4	εDE	0.008964	0.023999	0.014232	0.003293	≈
	εDEag	0.008937	0.014276	0.011096	0.001369	-
	SRS-εDEag	0.034653	0.041964	0.038480	0.001214	-
	adaptive ε-level	**0.003291**	**0.029886**	**0.026995**	**0.000107**	
CF5	εDE	0.058818	0.192996	0.109730	0.306706	-
	εDEag	0.017565	0.027832	0.020780	0.024022	≈
	SRS-εDEag	0.052164	0.124174	0.089620	0.014444	-
	adaptive ε-level	**0.036872**	**0.096393**	**0.085337**	**0.012862**	
CF6	εDE	**0.007778**	0.092116	0.052973	0.002667	-
	εDEag	0.017565	0.083221	0.042078	0.006422	-
	SRS-εDEag	0.052164	0.074174	0.044620	0.001444	-
	adaptive ε-level	0.017872	**0.070396**	**0.041773**	**0.006662**	
CF7	εDE	0.081238	0.192996	0.109730	0.006730	-
	εDEag	0.076620	0.127832	0.020780	0.007422	-
	SRS-εDEag	**0.064452**	0.124174	0.089620	0.005944	≈
	adaptive ε-level	0.078872	**0.106686**	**0.095573**	**0.005015**	

'≈' means the difference was not statistically significant, and '−' signifies that the proposed method was inferred to the comparative ones.

Table 3 shows the minimum, maximum, mean, and standard deviation values of the *IGD* calculated by εDE, εDEag and SRS-εDEag, and the proposed adaptive ε-level. The results in boldface indicate the best results. Table 3 shows that the adaptive ε-level statistically found a substantially the best min of the *IGD* on the test problems CF1, CF2, CF3, CF4 and CF5, and obtained the best results in terms of max and mean of *IGD* values on all the test problems, and found the best standard deviation of the *IGD* on all the test problems apart from CF2. εDEag and SRS-εDEag are essentially two advanced algorithms based on εDE. The εDEag proposed a scheme of automatically setting the control parameter *cp* of the ε level simply based on the generation. SRS-εDEag adaptively decided the ε level based on the number of violated constraints and the number of feasible solutions only. Although the ε-constraint coordinates feasible solutions and infeasible solutions, the generation and the constraint violation value are considered separately. However, the proposed adaptive ε-level method investigated the ε value being influenced by both the iteration and the type of constraints. As a result, the adaptive ε-level can get better mean and standard deviation than the comparison mechanism.

C. COMPARATIVE EXPERIMENT ON A REPLACEMENT MECHANISM AND DE MUTATION OPERATOR

This section discusses the experiments carried out between the replacement mechanism and the mutation operator proposed in Section III-B, C with the four most representative replacements and mutation operators on the CF1-CF7 series test problems. Note that, the ε value is set as the original model (controlled as formula (9)). Table 4 compares the *IGD* statistics (all 50 independent runs). Moreover, to test the statistical differences, a Wilcoxon's test at a 0.05 significance level was conducted between the replacement mechanism and the DE mutation and CMOEA/D-DE-ATP, C-MOEA/D, C-MOEA/DD, and MOEA/D-ACDP.

TABLE 4. The comparison of *IGD* for four algorithms to verify the replacement mechanism and DE mutation.

Problem	Algorithm	Min	Max	Mean	Std	Statistical significance
CF1	CMOEA/D-DE-ATP	0.015973	0.023854	0.017918	0.002685	-
	C-MOEA/D	0.000633	**0.001147**	0.000895	0.002101	≈
	C-MOEA/DD	0.007016	0.019362	0.013111	0.002587	-
	MOEA/D-ACDP	0.000314	0.001210	**0.000550**	0.001088	-
	Proposed	**0.000309**	0.001233	0.000665	**0.000899**	
CF2	CMOEA/D-DE-ATP	0.004142	0.051816	0.026779	0.014715	-
	C-MOEA/D	0.002733	0.013135	0.004203	0.002635	≈
	C-MOEA/DD	0.001579	**0.003063**	**0.002100**	0.000453	≈
	MOEA/D-ACDP	0.002568	0.089054	0.009402	0.021539	-
	Proposed	**0.001129**	0.004836	0.003596	**0.000399**	
CF3	CMOEA/D-DE-ATP	0.075300	0.142828	0. 104460	0.015595	-
	C-MOEA/D	0.090844	0.251884	0. 182905	0.042127	-
	C-MOEA/DD	0.038062	0.090089	0.070072	0.007573	≈
	MOEA/D-ACDP	0.052966	0.387998	0.185618	0.079894	-
	Proposed	**0.020186**	**0.070731**	**0.056305**	**0.005969**	
CF4	CMOEA/D-DE-ATP	0.008937	0.014276	0.011096	0.001369	-
	C-MOEA/D	0.008964	0.023999	0.014232	0.003293	-
	C-MOEA/DD	0.005523	**0.011552**	0.006995	0.001457	≈
	MOEA/D-ACDP	0.005301	0.128202	0.015356	0.023468	-
	Proposed	**0.004523**	0.011844	**0.005874**	**0.001023**	
CF5	CMOEA/D-DE-ATP	0.175650	0.278326	0.200780	**0.002422**	-
	C-MOEA/D	0.058818	0.192996	0.109730	0.030676	-
	C-MOEA/DD	0.070872	0. 230999	0. 157033	0.006662	-
	MOEA/D-ACDP	0.060908	0.474480	0.227288	0.016536	-
	Proposed	**0.050114**	**0.081741**	**0.069986**	0.009866	
CF6	CMOEA/D-DE-ATP	0.009568	0.038112	0.016168	0.005985	-
	C-MOEA/D	0.009522	**0.019939**	**0.013948**	0.002586	≈
	C-MOEA/DD	0.006186	0.031120	0.015020	0.006462	-
	MOEA/D-ACDP	0.007778	0.091374	0.090450	0.002871	≈
	Proposed	**0.006166**	0.026654	0.014201	**0.001724**	
CF7	CMOEA/D-DE-ATP	**0.041024**	0. 403714	0.109730	0.030676	-
	C-MOEA/D	0.058691	0.203867	0.120780	0.004222	-
	C-MOEA/DD	0.053510	0.083321	0.080962	0.003444	≈
	MOEA/D-ACDP	0.048766	0.525922	0.052041	0.015305	-
	Proposed	0.047461	**0.097001**	**0.050866**	**0.000971**	

Here, '+' indicates that the proposed adaptive ε method was significantly better than the comparative algorithms, '≈' means that the difference was not statistically significant, and '−' signifies that the proposed method was inferred to the comparative ones.

Table 4 shows the minimum, maximum, mean, and standard deviation values of the *IGD* calculated by MOEA/D-DE, C-MOEA/D, C-MOEA/DD, MOEA/D-ACDP, and the proposed method. The results in boldface indicate the best results. For the min values, the proposed method got the best results on all the test problems apart from CF7; for the max values, the proposed method obtained the best results on CF3, CF5, and CF7; for the max values, the proposed method obtained the best results on CF3, CF4, CF5, and CF7; and for the standard deviation values, the proposed method got the best results on all the test problems apart from CF5. CMOEA/D-DE-ATP approach is based on a penalty function concept that requires two penalty parameters, and the suitable parameters are different for different test problems. In C-MOEA/D, the feasible solution is favored over the infeasible solution, so the infeasible solution with good diversity is not fully utilized. In C-MOEA/DD, infeasible solutions are preserved when they are associated with the isolated region, which contribute to escape from the locally feasible regions. However, in the mating selection, the feasible solution is still favored over the infeasible solution and the infeasible solution with good diversity is not fully utilized. MOEA/D-ACDP adopts the angle information of the objective functions to enhance the population diversity in the infeasible region, but the feasible solution is still favored over the infeasible solution and the infeasible solution with good diversity is not fully utilized. In the proposed replacement mechanism and the mutation operator, the infeasible solutions are fully applied in both stages. As discussed above, utilizing the information carried by infeasible solutions can guide the population exploring the constraint boundaries and jumping out of an isolated feasible region, as a result, the proposed replacement mechanism and the mutation operator have good convergence and distribution, and can achieve better convergence accuracy.

D. COMPARATIVE EXPERIMENT ON THE WHOLE ALGORITHM

This section discusses experiments that were carried out between the proposed ε-MOEA/D-DE and the four best-performing algorithms on the CF1-CF7 series test

TABLE 5. The comparison of *IGD* for four algorithms to the proposed ε-MOEA/D-DE.

Problem	Algorithm	Min	Max	Mean	Std	Statistical significance
CF1	C-MOEA/DD	0.007016	0.019362	0.013111	0.002587	-
	MOEA/D-ACDP	0.000314	0.001210	0.000550	0.001088	-
	C-FROFI	0.000397	0.001015	0.000614	0.000989	≈
	NSGA-III	0.000794	0.001054	0.000868	0.000759	-
	ε-MOEA/D-DE	**0.000304**	**0.001092**	**0.000528**	**0.000189**	
CF2	C-MOEA/DD	0.001579	0.003063	**0.002100**	0.000453	≈
	MOEA/D-ACDP	0.002568	0.089054	0.009402	0.021539	-
	C-FROFI	0.001291	0.004386	0.003756	0.000531	≈
	C-NSGA-III	0.002369	0.009732	0.006551	0.001514	-
	ε-MOEA/D-DE	**0.001079**	**0.003036**	0.003100	**0.000435**	
CF3	C-MOEA/DD	0.038062	0.090089	0.070072	0.007573	≈
	MOEA/D-ACDP	0.052966	0.387998	0.185618	0.079894	-
	C-FROFI	0.020770	0.070731	0.065305	0.007573	-
	C-NSGA-III	0.090021	0.474518	0.148796	0.004489	-
	ε-MOEA/D-DE	**0.017862**	**0.060273**	**0.043601**	**0.005969**	
CF4	C-MOEA/DD	0.005523	0.011552	0.006995	0.001457	≈
	MOEA/D-ACDP	0.005301	0.128202	0.015356	0.023468	-
	C-FROFI	0.004489	0.011776	0.007147	0.001681	≈
	C-NSGA-III	0.003435	0.019802	0.007762	0.002544	-
	ε-MOEA/D-DE	**0.003222**	**0.009886**	**0.004995**	**0.001007**	
CF5	C-MOEA/DD	0.070872	0.230999	0.157033	0.006662	-
	MOEA/D-ACDP	0.060908	0.474480	0.227288	0.016536	-
	C-FROFI	0.050004	**0.074811**	0.069900	0.008965	≈
	C-NSGA-III	0.070001	0.093442	0.083360	**0.001061**	-
	ε-MOEA/D-DE	**0.004872**	0.076393	**0.065337**	0.006362	
CF6	C-MOEA/DD	0.006186	0.031120	0.015020	0.006462	-
	MOEA/D-ACDP	0.007778	0.091374	0.090450	0.002871	≈
	C-FROFI	0.006166	0.056654	0.034201	**0.001724**	≈
	C-NSGA-III	0.006534	0.057823	0.009945	0.010461	-
	ε-MOEA/D-DE	**0.005872**	**0.039396**	**0.013577**	0.006662	
CF7	C-MOEA/DD	0.053510	0.083321	0.080962	0.003444	-
	MOEA/D-ACDP	0.048766	0.525922	0.152041	0.015305	-
	C-FROFI	0.063461	**0.077001**	**0.070866**	0.000971	-
	C-NSGA-III	0.060606	0.086190	0.072041	0.015389	≈
	ε-MOEA/D-DE	**0.042872**	0.096686	0.085773	**0.000626**	

problems. Tables 5 and 6 compare the *IGD* and *HV* statistics (all 50 independent runs).

Moreover, to test the statistical differences, a Wilcoxon's test at a 0.05 significance level was conducted between the proposed method and C-MOEA/DD, MOEA/D-ACDP, C-FRORI, and NSGA-III. Here, '+' indicates that the proposed adaptive ε method was significantly better than the comparative algorithms, '≈' means that the difference was not statistically significant, and '−' signifies that the proposed method was inferred to the comparative ones.

Tables 5 and 6 show the minimum, maximum, mean, and standard deviation values of *IGD* and *HV* calculated by C-MOEA/DD, MOEA/D-ACDP, C-FRORI, NSGA-III, and the proposed ε-MOEA/D-DE. The results in boldface indicate the best results.

Table 5 shows that the ε-MOEA/D-DE statistically found all the best min values of the *IGD* on the test problems; obtained the best results in terms of max values on all the test problems expect CF5 and CF7; found the best mean values of the *IGD* on all the test problems apart from CF2 and CF7; got the best standard deviation values of the *IGD* on all the test problems apart from CF5 and CF6. A higher *HV* indicates a good Pareto front set in both the convergence toward Pareto front and the diversity to approximate a wide range of Pareto front. As shown in table 6, the ε-MOEA/D-DE statistically found all the best min values of the *HV* on the test problems apart from CF4; obtained the best results in terms of max values on all the test problems expect CF5; found the best mean values of the *HV* on all the test problems apart from CF2 and CF5. In general, ε-MOEA/D-DE performed better on convergence and distribution than the comparison algorithm. C-NSGA-III utilizes the Deb constraint dominance principle, and preferentially selects feasible solutions to generate offspring populations in matching selection, although it inherits NSGA-III's high convergence and distribution performance in unconstrained MOPs, but in the whole evolution process, it emphasizes that feasible solutions are dominant, while ignoring the useful information carried by infeasible solutions, which affects the convergence accuracy and convergence speed of the algorithm. For C-FROFI, an archive is introduced to store the effective infeasible individuals, which lead to a primary rale for the constraint violation in environment selection, in the mating selection, the feasible solution is still favored over

TABLE 6. The comparison of *HV* for four algorithms to the proposed ε-MOEA/D-DE.

Problem	Algorithm	Min	Max	Mean	Std	Statistical significance
CF1	C-MOEA/DD	0.820006	1.901147	1.720859	**0.001100**	-
	MOEA/D-ACDP	1.013855	2.023597	1.787191	0.025686	-
	C-FROFI	1.390504	2.060336	2.001537	0.068000	≈
	C-NSGA-III	1.256119	1.892335	1.458686	0.007599	-
	ε-MOEA/D-DE	**2.000304**	**3.001092**	**2.528347**	0.008937	
CF2	C-MOEA/DD	1.344733	2.013135	1.904203	**0.002635**	-
	MOEA/D-ACDP	2.000042	3.051816	**3.026779**	0.014715	-
	C-FROFI	2.005912	2.916875	2.637045	0.002793	≈
	NSGA-III	2.159679	3.000321	2.606551	0.051514	-
	ε-MOEA/D-DE	**2.311579**	**3.003036**	2.594100	0.004358	
CF3	C-MOEA/DD	0.908544	1.251884	0.982905	0.042127	-
	MOEA/D-ACDP	0.753051	1.142828	0.804460	0.015595	-
	C-FROFI	0.701341	1.178339	0.956801	0.020042	-
	C-NSGA-III	0.999871	1.005473	1.048796	**0.001489**	-
	ε-MOEA/D-DE	**1.537862**	**1.970731**	**1.675053**	0.007573	
CF4	C-MOEA/DD	**1.938964**	2.023999	2.014232	**0.000323**	≈
	MOEA/D-ACDP	1.504937	2.347196	2.011096	0.001369	-
	C-FROFI	1.349746	2.041964	1.938480	0.001214	≈
	C-NSGA-III	1.853574	2.359568	2.221772	0.002544	≈
	ε-MOEA/D-DE	1.765523	**2.911552**	**2.785695**	0.051457	
CF5	C-MOEA/DD	0.584818	0.992996	0.809730	0.030676	-
	MOEA/D-ACDP	0.917565	1.027832	0.920780	**0.001422**	-
	C-FROFI	1.012164	**2.124174**	**1.689620**	0.044874	≈
	C-NSGA-III	1.034653	1.914276	1.533360	0.010461	≈
	ε-MOEA/D-DE	**1.007872**	1.839396	1.115773	0.006662	
CF6	C-MOEA/DD	1.058818	2.192996	1.909730	0.030676	-
	MOEA/D-ACDP	1.017565	1.927832	1.562079	0.208822	≈
	C-FROFI	1.205164	1.794174	1.689620	0.021444	≈
	C-NSGA-III	1.534782	2.112394	1.833360	0.100461	-
	ε-MOEA/D-DE	**2.007872**	**2.439396**	**2.015773**	**0.006762**	
CF7	C-MOEA/DD	0.558818	0.992996	0.809730	0.036067	-
	MOEA/D-ACDP	0.517565	0.827832	0.670780	0.002242	-
	C-FROFI	0.752164	0.924174	0.896200	**0.001444**	-
	C-NSGA-III	0.746895	0.894568	0.733360	0.001461	≈
	ε-MOEA/D-DE	**0.917872**	**1.211019**	**1.015773**	0.006662	

the infeasible solution and the infeasible solution with good diversity is not fully utilized. As a result of the fully application for infeasible solutions in both stages. the proposed ε-MOEA/D-DE got good convergence and distribution, and can achieve better convergence accuracy.

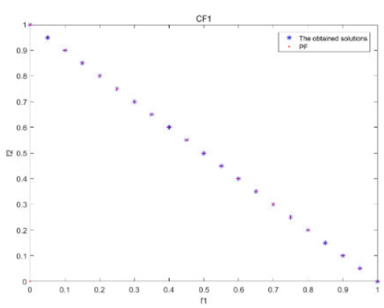


FIGURE 8. Obtained solutions by ε-MOEA/D-DE for CF1.

In order to see the convergence and distribution more intuitively, Figures 8-14 present the non-dominated solutions obtained by ε-MOEA/D-DE on the test problems CF1-CF7.

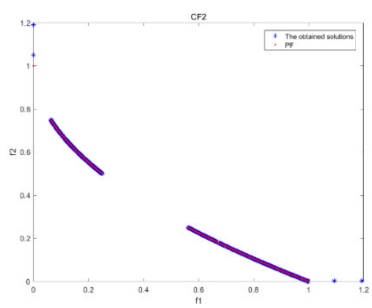


FIGURE 9. Obtained solutions by ε-MOEA/D-DE for CF2.

These solutions were selected from the final population of the run with the best *IGD* and *HV* among the 50 independent runs.

ε-MOEA/D-DE obtained good convergence on the test problems CF1 and CF2 from Figure 8 and 9. In Figure 10, the PF of the test problem CF3 was discontinuous and concave, and thus it could be more difficult than all the other test problems to solve. The three algorithms did not completely

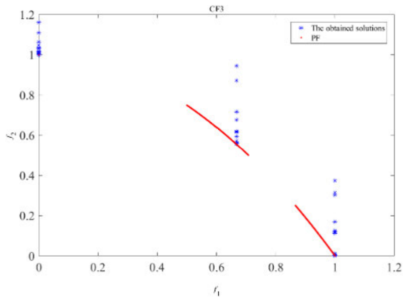


FIGURE 10. Obtained solutions by ε-MOEA/D-DE for CF3.

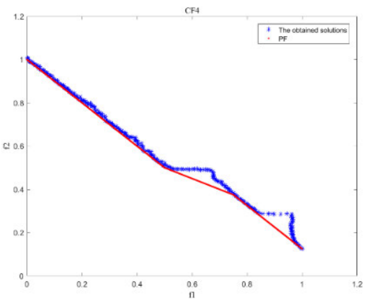


FIGURE 11. Obtained solutions by ε-MOEA/D-DE for CF4.

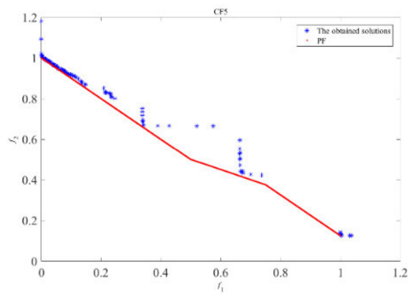


FIGURE 12. Obtained solutions by ε-MOEA/D-DE for CF5.

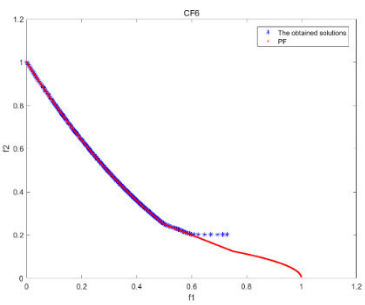


FIGURE 13. Obtained solutions by ε-MOEA/D-DE for CF6.

converge to the PF. Thus, these algorithms needed to further strengthen the exploration and exploitation capability. In Figure 11 and 12, ε-MOEA/D-DE acted relatively poorly on test problems CF4 and CF5 because the non-dominated solutions were not distributed evenly along the entire PF.

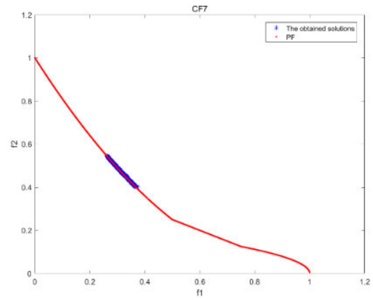


FIGURE 14. Obtained solutions by ε-MOEA/D-DE for CF7.

VI. CONCLUSION

In this paper, we suggested a constrained multi-objective optimization algorithm, called ε-MOEA/D-DE. From the perspective of the complementary advantages of dual population storage and ε-constraint, a novel constraint handling technique, ε-truncation, was presented to coordinate diversity and convergence by adaptively exploiting the feasible Pareto solutions and the infeasible solutions with both a lower constraint violation and a better objective value. Besides, the improved crowding density estimation can evaluate the distribution accurately and cut down the computation load by selecting a part of the Pareto solutions and near solutions to participate in the calculation. Moreover, for achieving a better compromise between global exploration and the local exploitation, exponential variation is introduced following the crossover operation and mutation operation.

To validate the competitiveness, we conducted a comprehensive experimental comparison with six state-of-the-art algorithms that were subject to different kinds of technologies. Many benchmark test problems (CF1-CF7 series) were selected to challenge the different capabilities of the algorithms. The proposed ε-MOEA/D-DE successfully found a well-converged and well-diversified set of solutions over multiple independent runs. However, on some CF-series test problems, ε-MOEA/D-DE encountered the increasingly difficult task of maintaining diversity and converging to the PF. Although different algorithms exhibited their working on different problems, we observed that none of these algorithms were capable of treating all the test problems, which implies that a careful choice of algorithms is still required at present when dealing with a complicated CMOP.

Future work should extend ε-MOEA/D-DE to handle constrained many-objective problems by incorporating advanced EAs. Moreover, we will apply ε-MOEA/D-DE to real-world problems to further confirm its effectiveness.

REFERENCES

[1] L. Zhang, T.-T. Liu, F.-Q. Wen, L. Hu, C. Hei, and K. Wang, "Differential evolution based regional coverage-enhancing algorithm for directional 3D wireless sensor networks," *IEEE Access*, vol. 7, pp. 93690–93700, 2019.

[2] Z. Jiao, L. Zhang, M. Xu, C. Cai, and J. Xiong, "Coverage control algorithm-based adaptive particle swarm optimization and node sleeping in wireless multimedia sensor networks," *IEEE Access*, vol. 7, pp. 170096–170105, 2019.

[3] L. Zhang, X. J. Bi, and Y. J. Wang, "Adaptive truncation technique for constrained multi-objective optimization," *KSII T. Internet. Inf.*, vol. 13, no. 11, pp. 5489–5511, Nov. 2019.

[4] Y. Zhang, D.-W. Gong, J.-Y. Sun, and B.-Y. Qu, "A decomposition-based archiving approach for multi-objective evolutionary optimization," *Inf. Sci.*, vols. 430–431, pp. 397–413, Mar. 2018.

[5] Y. Zhang, D.-W. Gong, and Z. Ding, "A bare-bones multi-objective particle swarm optimization algorithm for environmental/economic dispatch," *Inf. Sci.*, vol. 192, pp. 213–227, Jun. 2012.

[6] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 64–75, Jan. 2017.

[7] Y. Wang, B.-C. Wang, H.-X. Li, and G. G. Yen, "Incorporating objective function information into the feasibility rule for constrained evolutionary optimization," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2938–2952, Dec. 2016.

[8] H. Li and Q. Zhang, "MOEA/D: A multi-objective evolutionary algorithm based on decomposition," *IEEE Trans. Evol. Comput.*, vol. 11, no. 6, pp. 712–731, Dec. 2007.

[9] R. Mallipeddi and P. N. Suganthan, "Ensemble of constraint handling techniques," *IEEE Trans. Evol. Comput.*, vol. 14, no. 4, pp. 561–579, Aug. 2010.

[10] M. A. Jan and Q. Zhang, "MOEA/D for constrained multiobjective optimization: Some preliminary experimental results," in *Proc. UK Workshop Comput. Intell. (UKCI)*, Sep. 2010, pp. 1–6.

[11] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based nondominated sorting approach—Part II: Handling constraints and extending to an adaptive approach," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 602–622, Aug. 2014.

[12] K. Li, K. Deb, Q. Zhang, and S. Kwong, "An evolutionary many-objective optimization algorithm based on dominance and decomposition," *IEEE Trans. Evol. Comput.*, vol. 19, no. 5, pp. 694–716, Oct. 2015.

[13] Z. Fan, W. Li, X. Cai, K. Hu, H. Lin, and H. Li, "Angle-based constrained dominance principle in MOEA/D for constrained multi-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 460–467.

[14] T. Takahama and S. Sakai, "Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–9, doi: 10.1109/CEC.2010.5586484.

[15] T. Takahama and S. Sakai, "Constrained optimization by the μ constrained differential evolution with gradient-based mutation and feasible elites," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 1–8, doi: 10.1109/CEC.2006.1688283.

[16] T. Takahama and S. Sakai, "Efficient constrained optimization by the ε constrained differential evolution with rough approximation using kernel regression," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 1334–1341, doi: 10.1109/CEC.2013.6557719.

[17] C. Bu, W. Luo, and T. Zhu, "Differential evolution with a species-based repair strategy for constrained optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2014, pp. 967–974, doi: 10.1109/CEC.2014.6900526.

[18] W. Yi, X. Li, L. Gao, Y. Zhou, and J. Huang, "ε constrained differential evolution with pre-estimated comparison using gradient-based approximation for constrained optimization problems," *Expert Syst. Appl.*, vol. 44, pp. 37–49, Feb. 2016.

[19] S. Dominguez-Isidro, E. Mezura-Montes, and G. Leguizamon, "Memetic differential evolution for constrained numerical optimization problems," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2013, pp. 2996–3003, doi: 10.1109/CEC.2013.6557934.

[20] M. Asafuddoula, T. Ray, R. Sarker, and K. Alam, "An adaptive constraint handling approach embedded MOEA/D," in *Proc. IEEE Congr. Evol. Comput.*, Jun. 2012, pp. 1–8, doi: 10.1109/CEC.2012.6252868.

[21] R. Storn and K. Price, "Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[22] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the CEC 2009 special session and competition," Univ. Essex, Nanyang Technol. Univ., Clemson Univ., Clemson, SC, USA, Tech. Rep. CES-487, 2009.

[23] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[24] K. Deb, "A fast-elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Feb. 2000.

[25] K. Deb, L. Thiele, M. Laumanns, and Z. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," in *Proc. Evol. Multi-Objective Optim., Theor. Adv. Appl.*, 2005, pp. 105–145.

BING-JIE LIU was born in Hebei, China, in 1989. She received the B.S. degree in information and computing science from the College of Information and Communication Engineering, Harbin Engineering University, Harbin, China, 2012, where she is currently pursuing the Ph.D. degree. Her research interests include intelligent information processing and image processing, evolutionary computation, and multi-objective optimization.

XIAO-JUN BI received the Ph.D. degree from the College of Information and Communication Engineering, Harbin Engineering University, China, in 2006. She is currently a Professor with Minzu University of China. She was a Professor with Harbin Engineering University. Her research interests include evolutionary computation, multi-objective optimization, image/video analysis, and deep learning.