

Received November 10, 2020, accepted November 21, 2020, date of publication November 26, 2020, date of current version December 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3040647

Constrained Optimization Based on Ensemble Differential Evolution and Two-Level-Based Epsilon Method

BIN XU¹ AND **ZONGHAO ZHANG**

School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Corresponding author: Bin Xu (bxu@sues.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61703268.

ABSTRACT Constrained optimization problems (COPs) are common in many fields, and the search algorithm and constraint handling technique play important roles in the constrained evolution algorithms. In this article, we propose a new optimization algorithm named CETDE based on ensemble differential evolution (DE) and a two-level epsilon-constrained method. In the ensemble DE variant, some promising parameters and mutation strategies constitute the candidate pool, and each element in the pool coexists throughout the search process and competes to generate new solutions. The two-level epsilon method is proposed by incorporating a generation and a population comparison level to retain more promising solutions without degrading the solution quality. Moreover, a diversity promotion scheme is developed to improve the population distribution when the search becomes trapped in a small region. The superior performance of CETDE is validated by comparison with some state-of-the-art COEAs over two sets of artificial benchmarks and five real-world problems. The competitive results show that CETDE is an effective method for solving COPs.

INDEX TERMS Evolutionary algorithms, differential evolution, constrained optimization, ensemble, two-level-based comparison.

I. INTRODUCTION

Constrained optimization problems (COPs) are frequently encountered in many real-world design domains. Without loss of generality, a COP can be written as [1], [2]:

$$\begin{aligned} \min & f(\vec{x}), \vec{x} = [x_1, \dots, x_n]^T \\ \text{s.t.} & g_j(\vec{x}) \leq 0, (j = 1, 2, \dots, p) \\ & h_j(\vec{x}) = 0, (j = p + 1, \dots, q) \\ & a_i \leq x_i \leq b_i, (i = 1, 2, \dots, n) \end{aligned} \quad (1)$$

where n is the number of the dimensions of the decision variable x , $\vec{a} = [a_1, \dots, a_n]^T$ and $\vec{b} = [b_1, \dots, b_n]^T$ are the lower and upper bounds, respectively, $f(\vec{x})$ is the objective function to be optimized, and $g(\vec{x})$ and $h(\vec{x})$ are the p inequality and $q - p$ equality constraints, respectively. For COPs,

The associate editor coordinating the review of this manuscript and approving it for publication was Juan Liu¹.

the constraint violation of solution \vec{x} is defined as:

$$\phi(\vec{x}) = \sum_{j=1}^p \max\{0, g_j(\vec{x})\} + \sum_{j=p+1}^q \max\{0, |h_j(\vec{x})| - \delta\} \quad (2)$$

where δ is a positive tolerance value for transforming the equality constraints into inequality constraints. We say that \vec{x} is feasible if $\phi(\vec{x}) = 0$, otherwise, \vec{x} is infeasible.

Currently, evolutionary algorithms (EAs) are frequently adopted to solve optimization problems in various domains because of their advantages over traditional mathematical methods [3]–[8]. However, we should be aware that EAs were originally proposed for unconstrained global numerical optimization problems, and the main goal of constrained optimization algorithms is finding the global optimal solution in the feasible domain. Thus, to solve COPs effectively, the powerful EAs should be combined with suitable constraint handling techniques (CHTs) in a proper way [9]–[11].

To date, many CHTs have been designed for nature-inspired search algorithms, and each has its own

merits [12], [13]. Based on the methods used to treat the constraints, we can group the existing CHTs into three different categories, i.e., penalty function (PF)-based methods, superiority of feasibility-based methods and multiobjective-based methods.

The main idea of the PF-based methods is to first transform the original problem into an unconstrained problem by adding a penalized constraint violation to the objective function so that any existing search algorithm can be adopted to solve the unconstrained problems. These methods are easy to use but have also been criticized because the overall performance depends strongly on the associated penalty factor. To avoid this limitation, advanced PF-based methods have been extensively studied. For example, Runarsson and Yao proposed stochastic ranking method which does not require a priori information about COPs and also does not use any additional penalty factors [14]. Gan et al adopted a dynamic exponential function-like penalty factor setting to convert COPs into unconstrained ones and then the simplex crossover-based EA was used as the search algorithm [15]. Wang and Cai proposed an adaptive trade-off model, which the feasible rate of the main population is utilized [9]. Tessema and Yen proposed an adaptive PF method to exploit infeasible solutions with low objective value and low constraint violation [16].

The main principle of the superiority of feasibility-based methods is to prefer the feasible solutions to the infeasible solution. These methods usually do not require any additional parameters, making them quite simple and easy to use [17]. However, these methods often display heavy selection pressure, which is likely to lead to premature convergence, particularly for COPs with complex constraints. To avoid this problem, some improvements have been proposed to alleviate the selection pressure. For example, Takahama and Sakai introduced new CHT named ε -constrained methods by using a relax comparison level in feasibility rule. This method treats some solution with low constrained violation as feasible ones [18]. Fan et al proposed a feasible ratio based epsilon level control method. This method is then incorporated into decomposition-based multiobjective EA to solve constrained multiobjective problems [19]. Zhang et al proposed a adaptive ε -method based on the knowledge of constraint violation in the main population. Meanwhile, a more simpler heuristic rule is designed control the comparison level instead of traditional exponential function-based methods [20]. To solve COPs effectively, Wang et al first adopted the common feasibility rule during the selection procedure and then applied an individual replacement technique to preserve some infeasible solutions with low objective values [21]. Xu et al proposed a cluster-replacement-based feasibility rule to alleviate the selection pressure behind the ordinary feasibility rule [22]. Wang et al proposed an individual-dependent feasibility rule to alleviate the preference from both the objective function and constrained violation [11].

The multiobjective-based methods usually introduce some additional objective function to form a new problem and

then employ the concept in the multiobjective optimization community to solve it [23]–[25]. However, the application of these methods still faces some challenges because solving multiobjective optimization problems is still quite challenging in and of itself. For this kind of method, the key issue is how to use the multiobjective concept to balance the search between objective functions and constraints. For example, Cai and Wang tried to combine the original objective function and constrained violation together to form a bi-objective problem, i.e. $[f(\vec{x}), G(\vec{x})]$ and then the domination concept in multiobjective optimization community is applied to tackle the new problem [23], [24]. Deb and Datta combined the elitist non-dominated sorting genetic algorithm with PF method and proposed a hybrid EA-cum-CHT in a manner complementary to each other [26]. In addition to using static bi-objective structures, some dynamic constrained multiobjective framework is also proposed. Zeng et al converted COP into an equivalent dynamic constrained multiobjective optimization problem with 3 objectives [27], i.e. original objective, constrained violation and niche-count objective. During the search process, the constraint boundary and niche size are dynamically reduced. The new framework is incorporated into Pareto ranking-, decomposition- and hypervolume indicator-based multiobjective evolutionary algorithms, showing some advantages. Later, Jiao et al proposed an improved dynamic constrained multiobjective framework by the Pareto dominance and a new feasible-ratio control mechanism [28]. One remarkable superiority of this new framework is allowing the search from both feasible and infeasible domains. Zeng et al proposed Pareto front transformation model to retain some promising dominated feasible vectors and nondominated infeasible ones with worse objective. Thus the search within both the feasible and infeasible region is allowed [25].

In addition to designing effective CHTs, study of powerful search algorithms is another important issue in the development of competitive COEAs. Many search algorithm are available to the EA community, such as the genetic algorithm (GA), particle swarm optimization (PSO), artificial bee colony algorithm (ABC), and teaching–learning-based optimization (TLBO). Among these types of algorithms, differential evolution (DE) is a simple yet effective search algorithm over a continuous space, and a large number of DE-based methods have been proposed to handle COPs [29], [30]. The simplest approach for the use of DE for COPs is to directly combine basic DE with various CHTs. However, the performance of DE strongly depends on the mutation and crossover strategies and the associated control parameters, i.e., the population size N , scale factor F and crossover rate CR . Thus, these strategies and parameters must be determined carefully and be appropriate for a specific COP. As mentioned above, DE was originally proposed for unconstrained problems. To enhance the search ability of DE when solving complex COPs, various kinds of approaches were proposed, such as the design of new mutation operator [10], [31]–[33], study of advanced parameter control strategies [10], [22],

[34], [35], and introduction of additional diversity schemes [11], [21], [36].

In fact, there has been increasing interest in enhancing the search ability of DE by using multiple strategies and parameters. For these multi-strategy-based variants, each has its own features. For example, Wang et al proposed a composite DE variant by using three different offspring generation strategies and three control parameter settings according to some other researchers' studies [37]. One distinguishing feature of this method is three trial vectors are generated by three different trial vector generation strategies for each target vector in the main population and the best among the three will be used to update the corresponding parent vector during the selection process. Inspired by this framework, Wang et al proposed a constrained composite DE variant to solve COPs [38]. Xu et al proposed a cooperative ranking-based mutation strategy in DE for COPs [33]. Although the commonly used DE/rand/1 mutation strategy is adopted, it uses two different selection technique to determine parent vectors at two different situations. Sun et al proposed new DE variant by adopting two mutation strategies with different features [39]. One strategy is focusing on global exploration while the other on local exploitation. Among all these multi-strategy-based DE variants, ensemble of parameters and mutation strategies is a simple, yet powerful DE variant for global numerical optimization [40]. This approach attempts to provide a systematic framework that combines multiple mutation strategies with some parameter values. In this DE variant, some candidate pools are provided beforehand, and learning information is extracted for further processing. This scheme has been proved to be very effective when solving unconstrained global numerical optimization. How to extend it for COPs needs to be further studied.

The ε -constrained method proposed by Takahama and Sakai [18] seeks to relax the comparison in the feasibility rule and introduces a new comparison level, ε . If the constrained violation values of solutions are less than the ε level, they are considered to be feasible, and the comparison is based solely on the objective function values. Otherwise, the solutions are treated as infeasible, and the solution with the small constrained violation is deemed to be better. By introducing this relaxed ε level, the search algorithms will have more opportunities to explore the infeasible region around the feasible domain, which will be very helpful in the case where the feasible optimal solution lies at the boundary of constraints. However, the original ε -constrained method shares the same comparison level for all solutions in the main population. If the ε level is too large, the search will focus on minimizing the objective functions, degrading the exploitation around the feasible solution. By contrast, if the ε level is too small, more attention will be focused on decreasing the constraint violation.

Motivated by the above considerations, in this article, we propose a new constrained DE algorithm named CETDE by combining an ensemble DE variant with a new two-level-based ε method to solve COPs. In CETDE, some

well-studied mutation strategies and parameter values based on previous studies performed by other researchers are collected to form some candidate pools. If a combination of strategy and parameter can generate promising solutions in a previous search, it will have a greater likelihood to be used in the subsequent evolution process stage. Moreover, a diversity promotion scheme is used to improve the population diversity when the search is trapped into a narrow domain.

The remainder of this article is organized as follows. Section II briefly introduces the basic DE algorithm and the ε -constrained method. Section III presents our proposed CETDE algorithm in detail. Section IV presents the experimental results and performance comparisons. Finally, Section VI draws conclusions

II. DIFFERENTIAL EVOLUTION AND EPSILON-CONSTRAINED METHOD

A. DIFFERENTIAL EVOLUTION

The traditional DE begins with randomly initializing a population with N n -dimensional vectors, i.e., $\mathbf{X}^g = \{\vec{x}_1^g, \dots, \vec{x}_N^g\}$, where $\vec{x}_i^g = [\vec{x}_{i,1}^g, \dots, \vec{x}_{i,n}^g]^g$ is the i th target vector and g is the generation number. Then, DE employs mutation, crossover and selection operators sequentially and iteratively until a termination condition is met [29], [41].

The mutation operator generates a mutant vector for every target vector. The following three operators are used in this article:

- (1) DE/rand/2

$$\vec{v}_i^g = \vec{x}_i^g + F(\vec{x}_{r_2}^g - \vec{x}_{r_3}^g) + F(\vec{x}_{r_4}^g - \vec{x}_{r_5}^g) \quad (3)$$

- (2) DE/current-to-rand/1

$$\vec{v}_i^g = \vec{x}_i^g + K(\vec{x}_{r_1}^g - \vec{x}_i^g) + F(\vec{x}_{r_2}^g - \vec{x}_{r_3}^g) \quad (4)$$

- (3) DE/current-to-best/1

$$\vec{v}_i^g = \vec{x}_i^g + F(\vec{x}_{best}^g - \vec{x}_i^g) + F(\vec{x}_{r_1}^g - \vec{x}_{r_2}^g) \quad (5)$$

where indices r_1, r_2, r_3, r_4 and r_5 are mutually exclusive integers randomly generated within $[1, N]$ that are also different from index i , \vec{x}_{best} is the current found best solution, F is the scale factor lying between 0 and 1, and K is a random number in $[0, 1]$.

The crossover operator generates a trial vector \vec{u}_i^g using the target vector \vec{x}_i^g and mutant vector \vec{v}_i^g . The most frequently used binomial crossover strategy is described by:

$$\vec{u}_{i,j}^g = \begin{cases} \vec{v}_{i,j}^g, & \text{if } (\text{rand} \leq CR) \text{ or } (j = sn) \\ \vec{x}_{i,j}^g, & \text{otherwise.} \end{cases} \quad (6)$$

where $sn = \text{rndint}(1, n)$ is an arbitrary number in $\{1, 2, \dots, n\}$ and $\vec{u}_{i,j}^g$ is the j th element of the i th trial vector. Parameter CR is called the crossover rate and is a real number between 0 and 1.

After generating \vec{u}_i^g , DE uses a greedy selection strategy to form the target vector \vec{x}_i^{g+1} at generation $g + 1$:

$$\vec{x}_i^{g+1} = \begin{cases} \vec{u}_i^g, & \text{if } f(\vec{u}_i^g) \leq f(\vec{x}_i^g), \\ \vec{x}_i^g, & \text{otherwise.} \end{cases} \quad (7)$$

B. THE EPSILON-CONSTRAINED METHOD

The ε constrained method is a commonly used CHT. The main idea of this method is to treat some infeasible solutions as feasible solutions during the early stage based on a predefined comparison level ε controlled by Eq. (8). Meanwhile, the value of ε is gradually decreased until the generation number reaches T_c .

$$\begin{cases} \varepsilon^0 &= \phi(\vec{x}_\theta) \\ \varepsilon^g &= \begin{cases} \varepsilon^0(1 - g/T_c)^{cp}, & 0 < g < T_c \\ 0, & g \geq T_c \end{cases} \end{cases} \quad (8)$$

where ε^0 is the initial comparison level, \vec{x}_θ is the top θ vector and $\theta = 0.2N$, and $cp \in [2, 10]$ is used to control the decreasing speed.

Let $f_1(f_2)$ and $\phi_1(\phi_2)$ be the objective value and constraint violation at solution $\vec{x}_1(\vec{x}_2)$. Then, the ε -level comparison, denoted as \leq_ε , between \vec{x}_1 and \vec{x}_2 is given by:

$$(f_1, \phi_1) \leq_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 \leq f_2, & \text{if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 \leq f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 \leq \phi_2, & \text{otherwise} \end{cases} \quad (9)$$

It can be observed from Eq. (9) that for $\varepsilon = 0, \leq_0$ is equivalent to the feasibility rule [17]; by contrast, for $\varepsilon = \infty, \leq_\infty$ becomes the pure comparisons between the objective values.

III. PROPOSED CETDE

A. BASIC IDEA OF CETDE

Many researchers have pointed out that the search algorithm and constraint handling technique (CHT) are highly important for solving COPs [9]–[11], [33], [42]. Thus, improvement in the search ability and development of effective CHTs are commonly considered when solving COPs. In our CETDE, DE with the ensemble of mutation strategies and parameters is adopted as the main search engine, and the proposed two-level ε -constrained method is used as the main CHT. Moreover, a diversity promotion scheme is used to improve the population diversity. The framework of CETDE is given in Algorithm 1. Next, we will describe these three key components of CETDE in detail.

B. ENSEMBLE DIFFERENTIAL EVOLUTION FOR COPs

The effectiveness of DE in solving a given problem strongly depends on the mutation strategy and its associated parameter values. While many mutation strategies are available, none of them can achieve high overall performance for all problems with different characteristics. Thus, DE with multiple mutation strategies and parameters is popular. To exploit the concept of multiple strategies, ensemble DE [40] is applied to design the search algorithm when solving the COPs. The main idea of ensemble DE is that three candidate pools, i.e., the mutation strategy pool (Mpool), scale factor pool (Fpool) and crossover rate pool (CRpool), are introduced. Generally, we expect that the elements presented in the each pool should have distinct characteristics and

Algorithm 1 Pseudocode of CETDE Algorithm

- 1: Set the parameters: population size N , mutation strategy pool Mpool, scale factor pool Fpool, crossover rate pool CRpool, the maximize fitness evaluation number MaxFES.
- 2: Set the generation number $g = 1$ and randomly generate a population with N vectors uniformly in the search space, i.e. $\mathbf{X}^t = \{\vec{x}_1^g, \dots, \vec{x}_N^g\}$, with $\vec{x}_i^g = [x_{i,1}^g, \dots, x_{i,n}^g]^T, i = 1, \dots, N$.
- 3: Evaluate the objective function $f(\vec{x}_i^g)$ and constraint violation $\phi(\vec{x}_i^g), i = 1, 2, \dots, N$; Let $FES = N$.
- 4: Calculate the initial ε level: $\varepsilon^0 = \max_{k=1}^N (\phi(\vec{x}_k))$.
- 5: Randomly select a mutation strategy M_i^g , a scale factor F_i^g and a crossover rate CR_i^g from the corresponding candidate pool for $\vec{x}_i^g, i = 1, \dots, N$.
- 6: **while** $FES < \text{MaxFES}$ **do**
- 7: **for** $i = 1 : N$ **do**
- 8: Generate the trial vector \vec{u}_i^g for \vec{x}_i^g using the combination $[M_i^g, F_i^g, CR_i^g]$ associated with \vec{x}_i^g .
- 9: Evaluate the objective function $f(\vec{u}_i^g)$ and constraint violation $\phi(\vec{u}_i^g)$.
- 10: $FES = FES + 1$.
- 11: **end for**
- 12: **for** $i = 1 : N$ **do**
- 13: Get the ranking value $r(i)$ for i th vector.
- 14: Calculate the ε_i^g comparison level for i th vector at g th generation using Algorithm 2.
- 15: **if** $((f(\vec{u}_i^g), \phi(\vec{u}_i^g)) <_{\varepsilon_i^g} (f(\vec{x}_i^g), \phi(\vec{x}_i^g)))$ **then**
- 16: Replace \vec{x}_i^g by \vec{u}_i^g .
- 17: **end if**
- 18: **if** $f(\vec{u}_i^g) < f(\vec{x}_i^g)$ **then**
- 19: Store the combination $[M_i^g, F_i^g, CR_i^g]$ into the successful combination pool.
- 20: **end if**
- 21: Randomly select M_i^g, F_i^g and CR_i^g from the corresponding candidate pools or a combination from the stored successful combination pool.
- 22: **end for**
- 23: Perform the diversity promotion scheme using Algorithm 3.
- 24: $g = g + 1$.
- 25: **end while**
- 26: Out put the best vector as the final result.

therefore can be effectively applied to solve different kinds of problems.

In view of the above analysis and results obtained in previous studies, three mutation strategies named “DE/rand/2”, “DE/current-to-rand/1” and “DE/current-to-best/1” in Section II-A are used in Mpool. For the control parameter pools, $F_{\text{pool}} = [0.6 \ 0.8 \ 1.0]$ and $CR_{\text{pool}} = [0.1 \ 0.4 \ 1.0]$ are selected to balance the exploration and exploitation abilities of the search algorithm while solving particular problems with different characteristics.

During the evolution stage, little learning information is available at the first stage, and much information can be used at the later stage. Thus, we randomly select the mutation strategy and parameters from the candidate pools for every vector at $g = 0$. Suppose that the mutation strategy, scale factor and crossover rate associated with vector \vec{x}_i^g are M_i^g , F_i^g and CR_i^g , respectively. Based on the comparison results during the selection process, we keep $[M_i^g, F_i^g, CR_i^g]$ unchanged at generation $g + 1$ and store the combination of $[M_i^g, F_i^g, CR_i^g]$ into a successful combination pool when \vec{x}_i^g has a better objective value than that of the trial vector \vec{u}_i^g . By contrast, when \vec{u}_i^g is worse than \vec{x}_i^g , the current M_i^g , F_i^g and CR_i^g may be not suitable, and they can be re-selected. In this case, we randomly select the mutation strategy and parameters from the corresponding candidate pools or from the successful combination pool with equal probability.

C. TWO-LEVEL EPSILON-CONSTRAINED METHOD

For the ordinary ε -constrained method, all of the vectors in the main population share the same comparison level at generation g . This configuration will introduce some more infeasible vectors to promote the diversity of main population, but it will be highly likely to reduce the convergence speed and degenerate the current feasible optimal solution. It will also most likely cause the search to become trapped in the infeasible region so that no feasible solutions will be obtained at the end of the search.

To overcome these drawbacks to some extent, we proposed a new two-level ε method. The main idea of this method is that two comparison levels, named the generation level and population level here, are introduced. The generation level functions as the ordinary ε method. For the population level, we hope that different vectors have different comparison levels. To achieve this goal, we first sort all the vector from the best to worst according to the following criteria:

- (1) Feasible vectors are better than infeasible ones;
- (2) Ranking feasible vectors according to their function values;
- (3) Ranking infeasible solutions based on their constraint violations.

Based on the sorting values, we can obtain the ranking value for each vector \vec{x}_i^g as $r(i) = N - i$, $i = 1, 2, \dots, N$. And then the combined level can be calculated using a calculation model [33]. Many models are available, and the simplest linear model is described by:

$$\varepsilon_i^g = \varepsilon^g \frac{r(i) - 1}{N - 1} \quad (10)$$

where ε^g is the generation level calculated by Eq. (8); ε_i^g is the combined level for i th vector at generation g ; $r(i)$ is the ranking value for the i th vector \vec{x}_i^g in the main population. It can be observed from Eq. (10) that each vector has its own comparison level value and the superior ones have rigorous levels while the inferior ones own some relatively relaxed level. The main advantage of this idea is that the algorithm

can retain much better solutions with respect to both objective function and constrained violation around the superior solutions. It will also make the search to focus on minimizing objective function values for the inferior vectors. Thus the combined two-level approach can improve the search algorithm's exploration and exploitation abilities.

To clearly illustrate the difference between the ordinary and two-level ε method, Fig. 1 provides some additional information for these two methods with $N = 10$ and $T_c = 20$. From the perspective of generation level, both ordinary and two-level epsilon methods share the same parameter settings, as shown in Eq. (8) and Fig. 1(a). However, from the perspective of population level, these two methods are different. Supposing that the i th vector at generation g is ε_i^g , for the ordinary epsilon method, all vectors have equal ε values, i.e. $\varepsilon_1^g = \varepsilon_2^g = \dots = \varepsilon_N^g$, at every generation, as illustrated in Fig. 1(b). By contrary, the two-level epsilon method allow each vector to have different ε values, i.e. $\varepsilon_1^g \neq \varepsilon_2^g \neq \dots \neq \varepsilon_N^g$. More specifically, the best vector has lowest ε value and the worst has the largest value, as illustrated in Fig. 1(d). When both the population and generation levels are considered, we get the combined level for ordinary and two-level ε method, as illustrated in Fig. 1(c) and Fig. 1(e), respectively. The main procedure of the two-level ε method is given in Algorithm 2.

Algorithm 2 The Two-Level ε Method

Require: rank of the vector: $r(i)$, the current generation number g , initial epsilon value ε^0 .

Ensure: The epsilon value for vector \vec{x}_i^g , i.e. ε_i^g .

- 1: Calculate the epsilon generation level value ε^g at g th generation using Eq. (8).

$$\begin{cases} \varepsilon^0 = \max_{k=1}^N (\phi(\vec{x}_k)) \\ \varepsilon^g = \begin{cases} \varepsilon^0(1 - g/T_c)^{cp}, & 0 < g < T_c \\ 0, & g \geq T_c \end{cases} \end{cases}$$
 - 2: **for** $i = 1 : N$ **do**
 - 3: Calculate the epsilon population level ε_i^g for i th vector \vec{x}_i^g using Eq. (10).

$$\varepsilon_i^g = \varepsilon^g \frac{r(i) - 1}{N - 1}$$
 - 4: **end for**
-

D. DIVERSITY PROMOTION SCHEME

For some COPs with extremely complicated objective function or constraints, the feasible region becomes highly non-linear and always exhibits a multimodal property. Thus, the search is highly likely to become trapped in the region of a local minimum. Introduction of new solutions into the population to enhance the population diversity is a common approach to alleviate this problem. Many mechanisms can be applied to achieve this goal, such as the mutation operator [9], [21], restart scheme [38], and solution replacement [22] techniques. To introduce new solutions, three issues must be considered, i.e., when to introduce new solutions, which solution should be introduced and how to introduce the solutions.

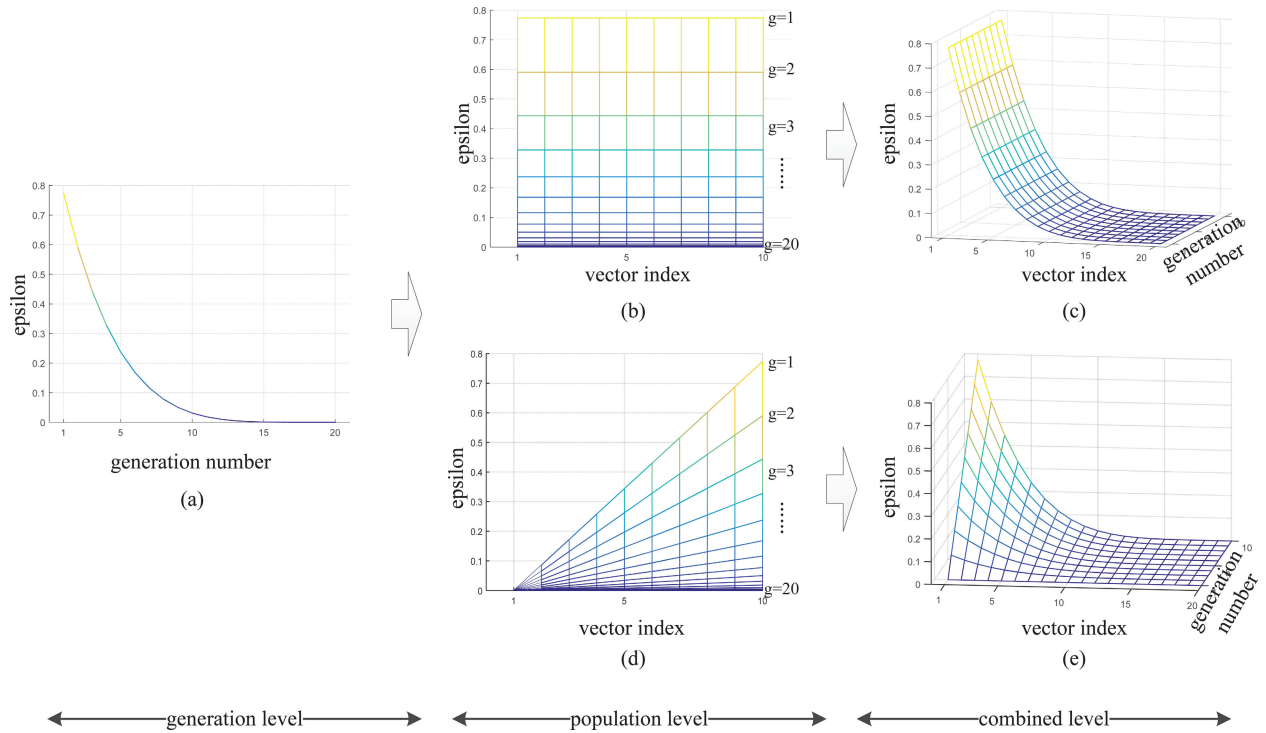


FIGURE 1. Comparison between the ordinary and two-level epsilon methods with $N = 10$ and $T_c = 20$.

Intuitively, if all of the vectors are located in a very small search region, we can say that premature trapping occurs, and we can introduce some new solutions to enhance the population diversity. However, the measurement of the similarity of all of the vectors in the main population is still a tough problem. Here, a simple indicator, i.e., standard deviations of both constraint violations and objective values, is used to evaluate the similarity of a population. Based on the standard deviation values, some infeasible trial vectors with low objective functions are conditionally maintained, and then, a greedy selection strategy is applied to replace some target vectors in the population. The diversity scheme used in this article is given in Algorithm 3.

Algorithm 3 Diversity Promotion Scheme

- 1: **if** $\text{std}(f(\vec{x})) \leq \eta$ & $\text{std}(\phi(\vec{x})) \leq \eta$ **then**
- 2: Set $p_m = p_m + 0.001$.
- 3: **else**
- 4: Set $p_m = 0$.
- 5: **end if**
- 6: **for** $i = 1 : N$ **do**
- 7: **if** $(f(\vec{u}_i^g) < \min_{k=1}^N f(\vec{x}_k^g))$ & $(\text{rand} < p_m)$ **then**
- 8: Replace \vec{x}_i^g by \vec{u}_i^g .
- 9: **end if**
- 10: **end for**

In Algorithm 3, threshold η is applied to decides whether to use the diversity promotion scheme or not. If the η is too big, the diversity promotion scheme will be frequently

applied and infeasible region with low objective value will be located. On the contrary, if η is too small, the diversity promotion scheme will be called rarely, which will lead to the failure of increasing population diversity. In fact, we have made some experiments with different values (i.e. 1×10^{-4} , 1×10^{-5} and 1×10^{-6}), on problems from IEEE CEC2006, and we did not notice any significant difference in results. Therefore, we arbitrarily peak 1×10^{-4} in this article.

IV. EXPERIMENTAL STUDY

A. TEST PROBLEMS AND EXPERIMENTAL SETTINGS

To evaluate the performance of CETDE, two well-known benchmark sets from IEEE CEC2006 and CEC2018 are selected as the test instances. The first set contains 24 problems that involve 5 types of objective functions (linear, nonlinear, polynomial, quadratic, and cubic), with different numbers of design variables (between 2 and 24) and 4 kinds of constraints (linear inequalities, linear equalities, nonlinear inequalities, and nonlinear equalities) with different numbers of constraints (between 1 and 38). The second set contains 28 problems with a wide variety of constraints and dimensions. The mathematical formulas and properties of these functions can be found in [1] and [43], respectively.

For all experiments, the positive tolerance value δ is set to 0.0001 to transform the equality constraints into inequality constraints. For the ϵ -constrained method, T_c is set to $0.2T_{\max}$, and cp is set to 6.

TABLE 1. Error Values Obtained When FES = 5×10^3 , 5×10^4 and 5×10^5 for Instance G01–G06.

		G01	G02	G03	G04	G05	G06
5.00E+03	Best	8.85E+00(0)	3.07E-01(0)	9.92E-01(0)	1.32E+02(0)	6.76E+01(3)	1.02E+01(0)
	Median	7.73E+00(2)	3.85E-01(0)	1.00E+00(1)	2.76E+02(0)	-4.31E+01(3)	3.20E+02(0)
	Worst	2.93E+00(6)	4.39E-01(0)	1.00E+00(1)	3.76E+02(0)	3.24E+02(3)	6.78E+02(0)
	c	1,1,0	0,0,0	0,0,1	0,0,0	3,0,0	0,0,0
	v	2.11E-01	0	3.25E-04	0	9.01E+00	0
	Mean	9.84E+00	3.85E-01	9.61E-01	2.70E+02	1.90E+02	2.86E+02
	Std	2.38E+00	3.10E-02	6.84E-02	6.39E+01	2.83E+02	1.79E+02
5.00E+04	Best	5.24E+00(0)	3.39E-02(0)	1.58E-01(0)	7.21E+00(0)	2.94E+02(3)	1.02E+01(0)
	Median	6.46E+00(0)	8.13E-02(0)	4.21E-01(0)	1.16E+01(0)	3.44E+01(3)	8.92E+01(0)
	Worst	7.45E+00(0)	1.50E-01(0)	9.09E-01(0)	1.92E+01(0)	1.37E+01(3)	2.22E+02(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	1,2,0	0,0,0
	v	0	0	0	0	4.67E-01	0
	Mean	6.34E+00	8.17E-02	4.34E-01	1.25E+01	1.31E+02	9.08E+01
	Std	5.76E-01	3.08E-02	1.65E-01	2.98E+00	1.46E+02	5.05E+01
5.00E+05	Best	0.00E+00(0)	1.39E-09(0)	8.99E-09(0)	-3.64E-12(0)	-1.82E-12(0)	-1.64E-11(0)
	Median	0.00E+00(0)	1.02E-08(0)	3.06E-08(0)	6.77E-09(0)	2.20E-08(0)	-1.64E-11(0)
	Worst	0.00E+00(0)	5.44E-03(0)	4.56E-08(0)	1.62E-07(0)	4.55E-08(0)	3.50E-07(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	v	0	0	0	0	0	0
	Mean	0.00E+00	2.17E-04	3.00E-08	2.42E-08	1.85E-08	4.90E-08
	Std	0.00E+00	1.09E-03	8.17E-09	3.69E-08	1.66E-08	9.49E-08

TABLE 2. Error Values Obtained When FES = 5×10^3 , 5×10^4 and 5×10^5 for Instance G07–G12.

		G07	G08	G09	G10	G11	G12
5.00E+03	Best	3.43E+01(0)	1.16E-05(0)	3.76E+01(0)	1.13E+04(1)	9.07E-04(0)	2.66E-11(0)
	Median	1.02E+02(0)	1.03E-03(0)	1.25E+02(0)	4.44E+03(1)	2.14E-01(0)	3.16E-08(0)
	Worst	4.73E+02(0)	1.17E-02(0)	3.62E+02(0)	-1.22E+03(3)	2.50E-01(1)	2.80E-06(0)
	c	0,0,0	0,0,0	0,0,0	0,1,0	0,0,0	0,0,0
	v	0	0	0	1.32E-01	0	0
	Mean	1.24E+02	2.11E-03	1.29E+02	4.26E+03	8.14E-02	3.82E-07
	Std	9.15E+01	2.57E-03	6.58E+01	4.53E+03	8.69E-02	7.77E-07
5.00E+04	Best	7.34E+00(0)	7.09E-09(0)	6.61E+00(0)	1.64E+04(0)	2.81E-05(0)	0.00E+00(0)
	Median	1.30E+01(0)	1.68E-05(0)	1.65E+01(0)	1.18E+03(1)	5.77E-04(0)	0.00E+00(0)
	Worst	2.27E+01(0)	1.83E-04(0)	4.29E+01(0)	-4.02E+03(1)	5.84E-03(0)	0.00E+00(0)
	c	0,0,0	0,0,0	0,0,0	0,1,0	0,0,0	0,0,0
	v	0	0	0	8.98E-02	0	0
	Mean	1.33E+01	3.69E-05	1.93E+01	1.26E+03	1.35E-03	0.00E+00
	Std	4.83E+00	4.18E-05	1.05E+01	4.44E+03	1.69E-03	0.00E+00
5.00E+05	Best	6.70E-08(0)	2.78E-17(0)	1.33E-09(0)	4.92E-10(0)	0.00E+00(0)	0.00E+00(0)
	Median	1.17E-07(0)	2.78E-17(0)	2.41E-09(0)	5.53E-09(0)	5.16E-10(0)	0.00E+00(0)
	Worst	5.33E-07(0)	2.78E-17(0)	4.20E-09(0)	2.60E-08(0)	2.96E-09(0)	0.00E+00(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	v	0	0	0	0	0	0
	Mean	1.66E-07	2.78E-17	2.73E-09	8.06E-09	8.35E-10	0.00E+00
	Std	1.24E-07	0.00E+00	9.63E-10	6.41E-09	8.83E-10	0.00E+00

B. GENERAL PERFORMANCE OF CETDE ON IEEE CEC2006 PROBLEMS

In this section, the general performance of CETDE is tested over 24 problems collected in CEC2006. For this experiment, the population size N is set to 50 for all 24 test instances. First, we execute CETDE 25 times and record the optimal function error values, i.e., $(f(x) - f(x^*))$, after 5×10^3 , 5×10^4 and 5×10^5 function evaluations, respectively. Then, we calculate some statistical indicators in terms of ‘Best’, ‘Median’, ‘Worst’, c , ‘ \bar{v} ’, ‘Mean’ and standard deviation (‘Std’), where c is a sequence of three numbers denoting the number of violated constraints for the median solution that are greater than 1.0, between 0.01 to 1.0, and between 0.0001 to 0.1, \bar{v} is the mean value of all constraint violations for the median solution, and the numbers in the parenthesis following ‘Best’,

‘Median’ and ‘Worst’ are the numbers of constraints unable to satisfy the feasible condition for the best, median and worst solutions, respectively.

The examination results presented in Tables 1, 2, 3 and 4 shows that the proposed CETDE approach can find feasible solutions by using 5×10^3 FES for 9 instances(i.e. G02, G04, G06, G07, G08, G09, G12, G19 and G24) for all 25 runs. As for other instances, CETDE obtains feasible solutions by using 5×10^3 FES for 4 instances (i.e. G01, G03, G11 and G16) and using 5×10^5 FES for 10 instances (i.e. G05, G10, G12, G13, G14, G15, G17, G18, G21 and G23). CETDE fails to finds any feasible solutions for G20 and G22 because these two instances are extremely constrained and no feasible solutions are reported so far. Furthermore, CETDA finds very close solutions to the known optimal

TABLE 3. Error Values Obtained When FES = 5 × 10³, 5 × 10⁴ and 5 × 10⁵ for Instance G13–G18.

		G13	G14	G15	G16	G17	G18
5.00E+03	Best	9.32E-01(3)	-4.77E+01(3)	9.82E+00(2)	1.22E-01(0)	1.19E+02(4)	9.01E-01(9)
	Median	9.39E-01(3)	-7.39E+01(3)	3.91E+00(2)	3.91E-01(0)	-2.67E+02(4)	3.37E+00(11)
	Worst	7.67E-01(3)	-1.01E+02(3)	7.11E-01(2)	9.03E-01(1)	-1.15E+03(4)	2.73E-01(10)
	c	0,3,0	3,0,0	0,2,0	0,0,0	3,1,0	9,2,0
	v	1.58E-01	2.75E+00	3.55E-01	0	7.97E+00	2.71E+00
	Mean	7.29E-01	-8.48E+01	2.73E+00	4.20E-01	-4.09E+01	9.58E-01
	Std	2.81E-01	1.81E+01	2.86E+00	1.53E-01	4.26E+02	1.89E+00
5.00E+04	Best	9.67E-03(3)	4.13E+00(3)	2.01E-01(2)	7.00E-02(0)	1.08E+02(4)	3.69E-01(0)
	Median	7.69E-03(3)	1.28E+00(3)	1.85E+00(2)	1.99E-01(0)	1.07E+02(4)	4.54E-01(1)
	Worst	1.02E-02(3)	5.40E+00(3)	2.15E-01(2)	2.91E-01(0)	1.32E+02(4)	6.14E-01(2)
	c	0,3,0	0,3,0	0,2,0	0,0,0	0,4,0	0,1,0
	v	2.02E-02	2.02E-02	2.37E-02	0	3.78E-01	2.10E-03
	Mean	3.05E-01	4.84E+00	1.23E+00	1.94E-01	8.47E+01	6.18E-01
	Std	3.44E-01	1.41E+00	1.34E+00	6.08E-02	6.45E+01	1.45E-01
5.00E+05	Best	-1.94E-16(0)	1.15E-09(0)	8.82E-09(0)	4.26E-11(0)	0.00E+00(0)	2.22E-16(0)
	Median	1.14E-11(0)	2.95E-09(0)	4.30E-08(0)	5.56E-10(0)	0.00E+00(0)	1.84E-14(0)
	Worst	2.94E-11(0)	3.65E-08(0)	1.22E-07(0)	1.46E-09(0)	1.82E-12(0)	1.81E-09(0)
	c	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0	0,0,0
	v	0	0	0	0	0	0
	Mean	1.10E-11	6.62E-09	4.40E-08	5.96E-10	1.46E-13	7.44E-11
	Std	7.39E-12	8.55E-09	2.50E-08	3.65E-10	5.04E-13	3.61E-10

TABLE 4. Error Values Obtained When FES = 5 × 10³, 5 × 10⁴ and 5 × 10⁵ for Instance G19–G24.

		G19	G20	G21	G22	G23	G24
5.00E+03	Best	8.43E+01(0)	6.11E-01(20)	1.54E+02(5)	-2.28E+02(20)	-4.20E+02(5)	1.08E-02(0)
	Median	1.49E+02(0)	1.27E+00(20)	3.87E+01(5)	-1.81E+02(20)	4.36E+02(5)	2.78E-02(0)
	Worst	2.55E+02(0)	1.55E+00(20)	6.87E+01(5)	-2.06E+02(20)	-1.17E+03(5)	6.32E-02(0)
	c	0,0,0	2,17,1	1,4,0	19,1,0	1,4,0	0,0,0
	v	0	6.80E-01	1.69E+00	5.52E+06	1.17E+00	0
	Mean	1.55E+02	1.09E+00	2.13E+02	5.70E+01	-6.16E+02	3.09E-02
	Std	4.70E+01	4.30E-01	1.86E+02	5.33E+02	4.06E+02	1.38E-02
5.00E+04	Best	2.37E+01(0)	-1.28E-01(20)	1.29E+02(5)	-2.36E+02(20)	-1.09E+02(5)	1.19E-04(0)
	Median	3.27E+01(0)	-1.22E-01(20)	5.77E-01(5)	-2.36E+02(20)	2.10E+02(5)	1.32E-03(0)
	Worst	4.97E+01(0)	-1.19E-01(20)	-1.55E+02(4)	-2.36E+02(20)	-8.26E+02(5)	2.69E-03(0)
	c	0,0,0	0,13,7	0,5,0	19,1,0	0,5,0	0,0,0
	v	0	3.84E-02	2.31E-01	9.72E+05	2.65E-01	0
	Mean	3.28E+01	-1.17E-01	-2.52E+01	-2.36E+02	-2.40E+02	1.42E-03
	Std	6.19E+00	9.52E-03	1.30E+02	4.31E-04	3.44E+02	6.87E-04
5.00E+05	Best	2.34E-07(0)	-7.81E-04(9)	-4.02E-10(0)	5.87E+02(14)	-6.82E-13(0)	3.29E-14(0)
	Median	8.05E-07(0)	-3.23E-03(11)	-3.47E-10(0)	-2.36E+02(9)	7.86E-06(0)	3.29E-14(0)
	Worst	7.48E-06(0)	-1.04E-02(10)	5.70E-06(0)	-2.36E+02(8)	2.20E-05(0)	3.38E-14(0)
	c	0,0,0	0,1,3	0,0,0	2,4,0	0,0,0	0,0,0
	v	0	7.76E-03	0	2.13E+01	0	0
	Mean	1.38E-06	-3.24E-03	1.16E-06	9.70E+02	7.87E-06	3.30E-14
	Std	1.60E-06	2.45E-03	2.13E-06	1.79E+03	7.58E-06	2.95E-16

values for most instances by using 5 × 10⁵ FES except G02. As for this problem, The worst function error is 5.44 × 10⁻³. However, the best and median values are 1.39 × 10⁻⁹ and 1.02 × 10⁻⁸, which means they are very close to the known optimal value.

For this experiment, the optimal function error values after 5 × 10⁵ FES over 25 runs are summarized in terms of ‘Best’, Median, Worst, Std, feasible rate, success rate and success performance values in Tables 5 and 6. As shown in Tables 5 and 6, the obtained best values for all 24 instances by CETDE on 25 independent runs are almost equal to the known optimal value $f(\vec{x}^*)$, except G20 and G22. Since G20 and G22 are extremely hard and no feasible solutions have been found so far, we do not report the statistical values in these tables. It is observed that CETDE obtained a 100% feasible rate on all

22 instances. For the success rate, CETDE achieved 100% on 21 out of 22 instances. In particular, CETDE obtained a 96% success rate on G02.

The experimental results of the CETDE on 24 instances from IEEE CEC2006 are shown in Table 6, where the known optimal value and the achieved Best, Median, Worst, Mean and Std values according to 25 independent runs are presented. It is found from the results presented in this table that CETDE can find the optimal solutions in all 25 runs consistently for all of the instances, except G02. Although CETDE cannot consistently achieve the optimal value on G02, the obtained values in 25 runs are quite close to the known optimal value, as verified by the small Std value. Thus, we deduce that the proposed CETDE is an effective and robust method for COPs.

TABLE 5. Number of FES to Achieve the Fixed Accuracy Level, Success Rate, Feasible Rate and Success Performance.

Prob.	Best	Median	Worst	Mean	Std	Feasible Rate	Success Rate	Success Performance
G01	100850	102900	104400	102866	864	100%	100%	102866
G02	87050	100350	500000	120138	80426	100%	96%	125144
G03	87200	89900	91400	89882	1226	100%	100%	89882
G04	96900	97700	98950	97716	551	100%	100%	97716
G05	96600	98850	107450	99014	1893	100%	100%	99014
G06	101850	103700	105150	103672	750	100%	100%	103672
G07	115850	119850	128350	120092	2826	100%	100%	120092
G08	2750	29900	52700	29398	15435	100%	100%	29398
G09	102450	104500	106300	104290	1107	100%	100%	104290
G10	139400	147550	159050	148144	5104	100%	100%	148144
G11	23250	61800	73150	59632	11807	100%	100%	59632
G12	1000	2400	3450	2488	579	100%	100%	2488
G13	78700	95900	274050	112548	48282	100%	100%	112548
G14	107000	109500	115350	110068	2332	100%	100%	110068
G15	83250	91900	93050	91396	1906	100%	100%	91396
G16	99800	101400	102350	101304	699	100%	100%	101304
G17	121400	156650	239900	164178	30360	100%	100%	164178
G18	107950	119600	135900	119598	6013	100%	100%	119598
G19	130400	141050	173500	143312	10619	100%	100%	143312
G20	500000	500000	500000	500000	-	0%	0%	-
G21	114800	123050	189900	126228	14839	100%	100%	126228
G22	500000	500000	500000	500000	-	0%	0%	-
G23	134500	176000	308550	187230	44059	100%	100%	187230
G24	60600	72900	76800	71696	3983	100%	100%	71696

TABLE 6. The Experimental Results of CETDE on 24 Instances From IEEE CEC2006.

Prob.	$f(\vec{x}^*)$	Best	Median	Worst	Mean	Std
G01	-15	-15	-15	-15	-15	0.00E+00
G02	-0.803619104	-0.803619	-0.803619	-0.798182	-0.803402	1.09E-03
G03	-1.0005001	-1.0005	-1.0005	-1.0005	-1.0005	8.17E-09
G04	-30665.53867	-30665.53867	-30665.53867	-30665.53867	-30665.53867	3.69E-08
G05	5126.496714	5126.496714	5126.496714	5126.496714	5126.496714	1.66E-08
G06	-6961.813876	-6961.813876	-6961.813876	-6961.813875	-6961.813876	9.49E-08
G07	24.30620907	24.306209	24.306209	24.30621	24.306209	1.24E-07
G08	-0.095825042	-0.095825	-0.095825	-0.095825	-0.095825	1.42E-17
G09	680.6300574	680.630057	680.630057	680.630057	680.630057	9.63E-10
G10	7049.248021	7049.248021	7049.248021	7049.248021	7049.248021	6.41E-09
G11	0.7499	0.7499	0.7499	0.7499	0.7499	8.83E-10
G12	-1	-1	-1	-1	-1	0.00E+00
G13	0.053941514	0.053942	0.053942	0.053942	0.053942	7.39E-12
G14	-47.76488846	-47.764888	-47.764888	-47.764888	-47.764888	8.55E-09
G15	961.7150223	961.715022	961.715022	961.715022	961.715022	2.50E-08
G16	-1.905155259	-1.905155	-1.905155	-1.905155	-1.905155	3.65E-10
G17	8853.533875	8853.533875	8853.533875	8853.533875	8853.533875	5.25E-13
G18	-0.866025404	-0.866025	-0.866025	-0.866025	-0.866025	3.61E-10
G19	32.65559295	32.655593	32.655594	32.6556	32.655594	1.60E-06
G20	0.2049794	0.1946	0.202326	0.204464	0.201741	2.45E-03
G21	193.7245101	193.72451	193.72451	193.724516	193.724511	2.13E-06
G22	236.4309755	0	20.362973	6379.138866	1206.692442	1.79E+03
G23	-400.0551	-400.0551	-400.055092	-400.055078	-400.055092	7.58E-06
G24	-5.508013272	-5.508013	-5.508013	-5.508013	-5.508013	1.06E-15

To provide additional information about the CETDE’s search ability when solving instances from IEEE CEC2006, we show the convergence graphs of the function error values, i.e., $\lg(f(\vec{x}) - f(\vec{x}^*))$, versus fitness evaluations (FES) at the median run over 25 runs in Figs. 2-5. Since CETDE cannot find any feasible solutions on G20 and G22 over 25 runs, we do not show the convergence graphs of these two instances in the figures. It is observed from these figures that for most instances, the function error values decrease dramatically at the early stage and evolve slowly at the later stage. This is because the obtained values are quite close to the known

optimal value. Based on these figures, we conclude that the proposed CETDE shows relatively fast convergence for most instances.

C. COMPARISON TO OTHER STATE-OF-THE-ART COEAs

Table 7 gives the mean and standard deviation of objective values with respect to CETDE and nine competing algorithms. Among all nine competitors, the first four are DE-based methods (i.e., DPDE [31], rank-iMDDE [36], eDE [44], and FRC-CEA [28]), the next four are non-DE-based methods (i.e., AIS [45], CMPSOWV [46], I-ABC [47] and

TABLE 7. Experimental Results of CETDE and 4 Other Algorithms on G01-G24 over 25 Independent Runs.

	G01	G02	G03	G04
DPDE [31]	-15.0000 ± 0.00E+00 *	-0.8024 ± 1.18E-03	-1.0005 ± 0.00E+00 *	-30665.5387 ± 0.00E+00 *
rank-iMDDE [36]	-15.0000 ± 7.78E-10 *	-0.8036 ± 7.20E-08 *	-1.0005 ± 6.01E-07 *	-30665.5390 ± 2.56E-08 *
eDE [44]	-15.0000 ± 0.00E+00 *	-0.8020 ± 1.40E-05	-1.0005 ± 2.80E-31 *	-30665.5387 ± 1.38E-23 *
FRC-CEA [28]	-15.0000 ± 2.38E-10 *	-0.8036 ± 7.57E-08 *	-1.0005 ± 6.48E-10 *	-30665.5390 ± 7.80E-12 *
AIS [45]	-15.0000 ± 0.00E+00 *	-0.8022 ± 5.19E-10	-1.0005 ± 1.77E-11 *	-30665.5390 ± 3.69E-13 *
CMPSOVV [46]	-15.0000 ± 0.00E+00 *	-0.8036 ± 2.79E-07 *	-1.0005 ± 2.13E-11 *	-30665.5390 ± 0.00E+00 *
I-ABC [47]	-15.0000 ± 4.59E-16 *	-0.8001 ± 7.05E-03	-1.0000 ± 3.36E-04	-30665.5390 ± 1.02E-05 *
ITLBO [48]	-15.0000 ± 0.00E+00 *	-0.8023 ± 3.26E-03	-1.0005 ± 2.58E-09 *	-30665.5390 ± 3.71E-12 *
DPD [49]	-15.0000 ± 0.00E+00 *	-0.8035 ± 3.54E-03 *	-1.0005 ± 0.00E+00 *	-30665.5386 ± 0.00E+00 *
CETDE	-15.0000 ± 0.00E+00 *	-0.8034 ± 1.09E-03	-1.0005 ± 2.57E-08 *	-30665.5387 ± 1.97E-07 *
	G05	G06	G07	G08
DPDE [31]	5126.4967 ± 0.00E+00 *	-6961.8139 ± 0.00E+00 *	24.3062 ± 6.25E-09 *	-0.0958 ± 0.00E+00 *
rank-iMDDE [36]	5126.4967 ± 3.41E-07 *	-6961.8139 ± 5.59E-08 *	24.3062 ± 2.11E-04 *	-0.0958 ± 0.00E+00 *
eDE [44]	5126.4967 ± 2.75E-12 *	-6961.8139 ± 0.00E+00 *	24.3062 ± 1.87E-23 *	-0.0958 ± 2.00E-34 *
FRC-CEA [28]	5126.4967 ± 7.43E-13 *	-6961.8139 ± 2.73E-12 *	24.3062 ± 3.64E-05 *	-0.0958 ± 0.00E+00 *
AIS [45]	5126.4981 ± 1.70E-02 *	-6961.8139 ± 1.90E-12 *	24.3557 ± 8.20E-03	-0.0958 ± 0.00E+00 *
CMPSOVV [46]	5126.4975 ± 1.32E-01	-6961.8140 ± 0.00E+00 *	24.3061 ± 2.51E-11 *	-0.0958 ± 0.00E+00 *
I-ABC [47]	5131.8610 ± 5.53E+00	-6961.8140 ± 1.82E-12 *	24.3660 ± 6.91E-02	-0.0958 ± 2.78E-17 *
ITLBO [48]	5126.4967 ± 2.78E-12 *	-6961.8139 ± 0.00E+00 *	24.3062 ± 1.51E-05 *	-0.0958 ± 1.42E-17 *
DPD [49]	5126.4967 ± 0.00E+00 *	-6961.8139 ± 0.00E+00 *	24.3062 ± 0.00E+00 *	-0.0958 ± 0.00E+00 *
CETDE	5126.4967 ± 6.10E-08 *	-6961.8139 ± 2.04E-07 *	24.3062 ± 1.63E-07 *	-0.0958 ± 1.42E-17 *
	G09	G10	G11	G12
DPDE [31]	680.6301 ± 3.65E-14 *	7049.2480 ± 8.36E-08 *	0.7499 ± 0.00E+00 *	-1.0000 ± 0.00E+00 *
rank-iMDDE [36]	680.6300 ± 1.56E-06 *	7049.2490 ± 8.24E-04	0.7499 ± 1.61E-07 *	-1.0000 ± 0.00E+00 *
eDE [44]	680.6301 ± 1.10E-25 *	7049.2480 ± 4.89E-19 *	0.7499 ± 1.28E-32 *	-1.0000 ± 0.00E+00 *
FRC-CEA [28]	680.6300 ± 2.49E-13 *	7049.2480 ± 1.40E-04 *	0.7499 ± 0.00E+00 *	-1.0000 ± 0.00E+00 *
AIS [45]	680.6300 ± 0.00E+00 *	7049.5700 ± 4.50E-04	0.7499 ± 1.40E-08 *	-1.0000 ± 0.00E+00 *
CMPSOVV [46]	680.6300 ± 2.76E-12 *	7085.1718 ± 7.13E+02	0.7499 ± 1.03E-12 *	-1.0000 ± 0.00E+00 *
I-ABC [47]	680.6330 ± 1.53E-03	7124.0420 ± 5.94E+01	0.7499 ± 2.44E-06 *	-1.0000 ± 0.00E+00 *
ITLBO [48]	680.6300 ± 3.36E-13 *	7049.2490 ± 4.29E-05	0.7499 ± 1.13E-16 *	-1.0000 ± 0.00E+00 *
DPD [49]	680.6301 ± 0.00E+00 *	7049.2480 ± 0.00E+00 *	0.7499 ± 0.00E+00 *	-1.0000 ± 0.00E+00 *
CETDE	680.6301 ± 1.88E-09 *	7049.2480 ± 6.24E-08 *	0.7499 ± 1.54E-09 *	-1.0000 ± 0.00E+00 *
	G13	G14	G15	G16
DPDE [31]	0.0539 ± 1.16E-17 *	-47.7649 ± 2.56E-09 *	961.7150 ± 0.00E+00 *	-1.9052 ± 0.00E+00 *
rank-iMDDE [36]	0.0539 ± 1.55E-07 *	-47.7649 ± 2.43E-04 *	961.7150 ± 1.71E-07 *	-1.9052 ± 7.04E-06 *
eDE [44]	0.0539 ± 1.27E-10 *	-47.7649 ± 5.47E-23 *	961.7150 ± 2.41E-14 *	-1.9052 ± 2.05E-31 *
FRC-CEA [28]	0.0539 ± 0.00E+00 *	-47.7649 ± 3.05E-04 *	961.7150 ± 4.55E-13 *	-1.9052 ± 0.00E+00 *
AIS [45]	0.0539 ± 7.80E-10 *	-47.7649 ± 1.00E-12 *	961.7150 ± 0.00E+00 *	-1.9052 ± 0.00E+00 *
CMPSOVV [46]	0.4345 ± 1.59E-01	-47.7649 ± 1.46E-07 *	961.7150 ± 4.64E-13 *	-1.9052 ± 3.48E-14 *
I-ABC [47]	0.0541 ± 2.69E-04	-47.2010 ± 3.51E-01	961.7160 ± 1.44E-03	-1.9050 ± 6.86E-16 *
ITLBO [48]	0.0540 ± 3.30E-04 *	-47.7649 ± 3.80E-05 *	961.7200 ± 5.80E-13	-1.9052 ± 4.53E-16 *
DPD [49]	0.0539 ± 0.00E+00 *	-47.7649 ± 0.00E+00 *	961.7150 ± 0.00E+00 *	-1.9052 ± 0.00E+00 *
CETDE	0.0549 ± 4.73E-03	-47.7649 ± 8.35E-09 *	961.7150 ± 3.12E-08 *	-1.9052 ± 7.19E-10 *
	G17	G18	G19	G21
DPDE [31]	8853.5339 ± 2.34E-12 *	-0.8660 ± 1.65E-12 *	32.6556 ± 6.17E-08 *	193.7262 ± 8.36E-04
rank-iMDDE [36]	8895.0034 ± 3.92E+01	-0.8660 ± 1.57E-06 *	32.6572 ± 1.80E-03	193.7245 ± 1.27E-07 *
eDE [44]	8853.8165 ± 1.70E-02	-0.8660 ± 1.14E-12 *	32.6556 ± 3.97E-11 *	193.7275 ± 6.00E-10
FRC-CEA [28]	8872.1108 ± 3.12E+01	-0.8660 ± 4.25E-10 *	32.6614 ± 1.28E-02	193.7245 ± 4.56E-10 *
AIS [45]	8853.5397 ± 1.90E-09	-0.8660 ± 1.30E-15 *	32.6556 ± 0.00E+00 *	196.7245 ± 1.10E+00
CMPSOVV [46]	8874.2046 ± 3.64E-01	-0.8660 ± 3.64E-15 *	32.6556 ± 1.53E-05 *	193.7535 ± 2.52E-03
I-ABC [47]	8909.9940 ± 2.90E+01	-0.8653 ± 1.90E-03	33.3440 ± 4.96E-01	622.6780 ± 1.67E+02
ITLBO [48]	8959.8000 ± 3.77E+01	-0.8660 ± 1.68E-05 *	32.6620 ± 1.06E-02	222.2200 ± 4.84E+01
DPD [49]	8853.5397 ± 0.00E+00	-0.8660 ± 0.00E+00 *	32.6556 ± 6.47E-05 *	193.7245 ± 0.00E+00 *
CETDE	8853.5339 ± 1.98E-05 *	-0.8660 ± 1.58E-09 *	32.6556 ± 3.04E-06 *	193.7245 ± 3.44E-06 *
	G23	G24	Sum	
DPDE [31]	-399.2752 ± 1.52E+00	-5.5080 ± 0.00E+00 *	19	
rank-iMDDE [36]	-393.1349 ± 1.86E+01	-5.5080 ± 3.58E-11 *	18	
eDE [44]	-400.0551 ± 1.00E-06 *	-5.5080 ± 8.20E-31 *	19	
FRC-CEA [28]	-398.2812 ± 6.27E+00	-5.5080 ± 0.00E+00 *	19	
AIS [45]	-399.8743 ± 2.00E+00	-5.5080 ± 0.00E+00 *	15	
CMPSOVV [46]	-360.2836 ± 1.43E+01	-5.5080 ± 0.00E+00 *	16	
I-ABC [47]	169.0210 ± 3.67E+02	-5.5080 ± 1.78E-15 *	7	
ITLBO [48]	-256.4000 ± 1.42E+02	-5.5080 ± 9.06E-16 *	15	
DPD [49]	-398.1809 ± 2.97E+00	-5.5080 ± 0.00E+00 *	20	
CETDE	-399.4736 ± 2.88E+00	-5.5080 ± 5.69E-11 *	19	

ITLBO [48]), and the last one is a PSO- and DE-based hybrid method (i.e., DPD [49]). Since the best feasible optimal values are known, we compare the mean function values with the

best known optima. If the obtained mean function error value is less than a predefined threshold, i.e., $\overline{f(\vec{x})} - f(\vec{x}^*) \leq 10^{-4}$, we denote it using symbol \star in Table 7. Because the optimal

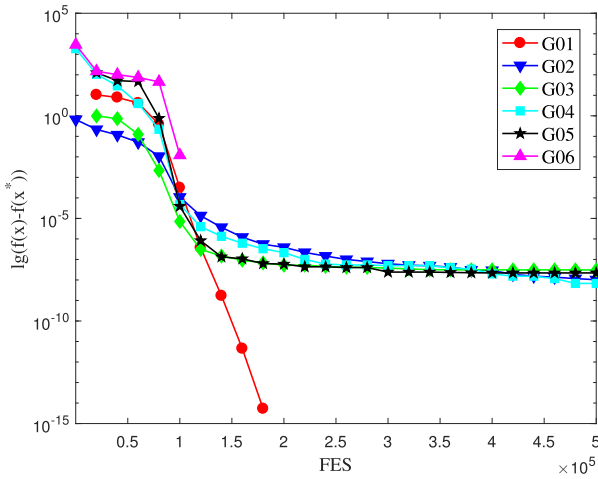


FIGURE 2. Convergence graph for G01-G06.

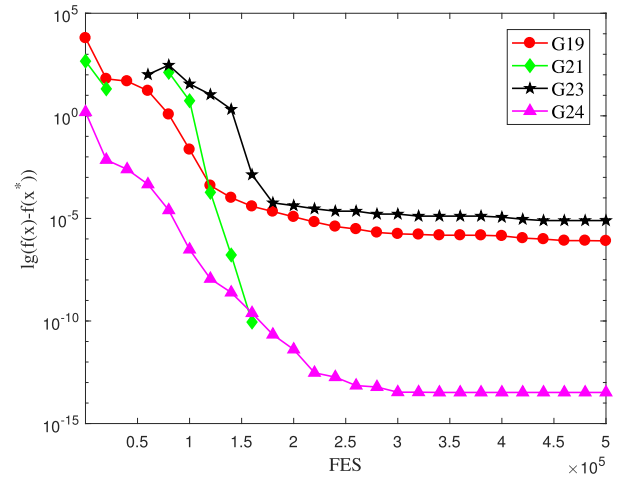


FIGURE 5. Convergence graph for G19-G24.

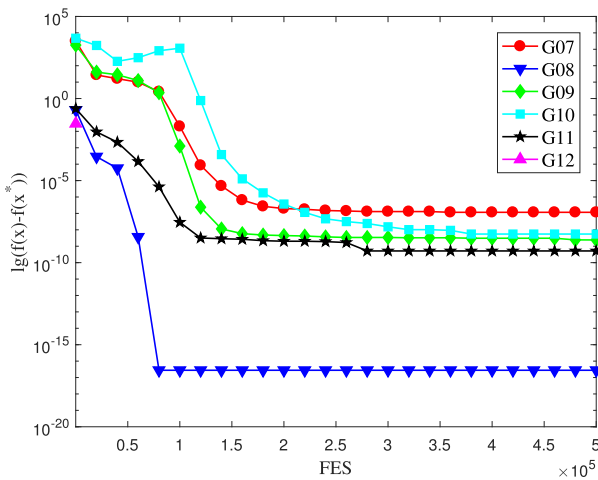


FIGURE 3. Convergence graph for G07-G12.

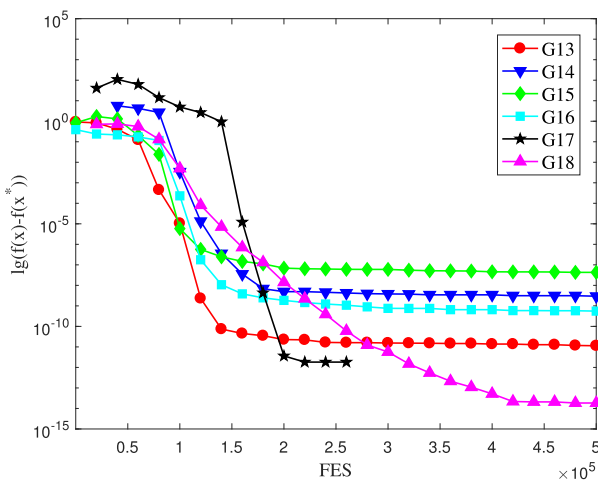


FIGURE 4. Convergence graph for G13-G18.

feasible solutions of G20 and G22 are unknown, Table 7 does not contain the statistical information on these two instances. We note that the maximal function evaluation number for this experiment is set to 2.4×10^5 and that the Mean and Std values

provided by all competing algorithms are taken directly from the corresponding references for fair comparison.

As shown in Table 7, the hybrid DPD method ranks first on all ten algorithms because it obtained 20 successful runs out of 22. The proposed CETDE and other three competing algorithms (i.e., DPDE, eDE and FRC-CEA) rank second because they only obtained 19 successful runs out of 22. The non-DE-based algorithms (i.e., AIS, CMPSOVV, I-ABC and ITLBO) perform slightly worse than the DE-based approaches, obtaining 15, 16, 7 and 15 successful runs out of 22, respectively. Based on the results of this experiment, we conclude that CETDE either outperforms or performs similarly to most competitors on the 22 test instances from IEEE CEC2006.

Based on the Mean indicator summarized in Table 7, the average ranks obtained by applying the Friedman procedure are shown in Fig. 6. We can find that CETDE ranks fourth among all ten algorithms, which means that CETDE performs worse than DPDE, rank-iMDDE and FCR-CEA and better than the rest six COEAs on IEEE CEC2006 problems.

D. COMPARISONS WITH COEAs ON IEEE CEC2018 PROBLEMS

To further test the performance of CETDE when solving more challenging instances, in this section 28 instances selected from IEEE CEC2018 are used as test problems. Four different dimension levels, i.e. 10, 30, 50, 100, are applied in the experiment. Here we set the population size N to 50, 50, 80 and 100 when the dimension level is 10, 30, 50 and 100, respectively. Still 25 independent runs are carried out for every problem at each dimension level. Table 8 summarized the mean objective values, mean constraint violation values and feasibility rate for 28 problems over 25 runs.

Similarly, we choose three algorithms, i.e. MAg-ES [50], IUDE [51] and LSHADE [52], as the competitors because these algorithms achieved very competitive results in the IEEE CEC2018 competition. Because the optimal values of these problems are not provided in [43], we compare the

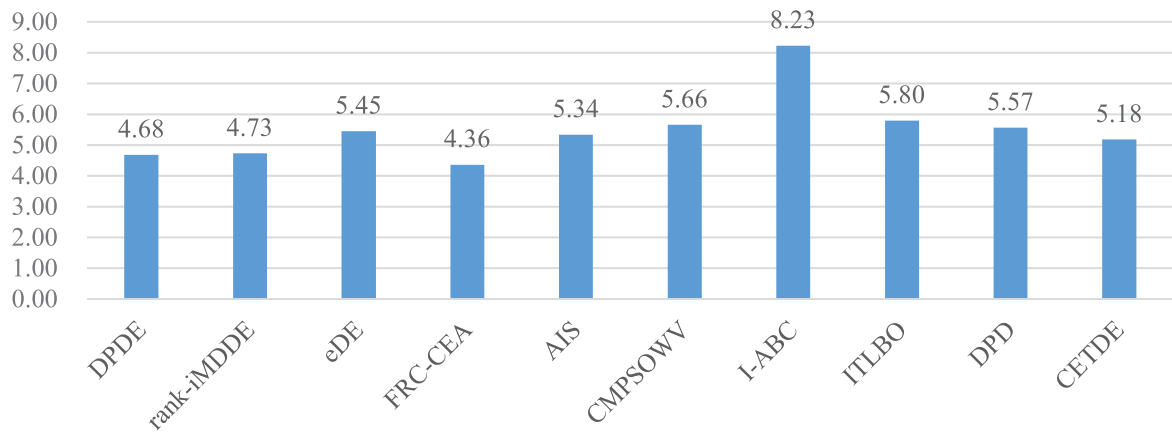


FIGURE 6. Average ranks obtained by applying the Friedman procedure.

TABLE 8. Mean Objective, Constraint Violation and Feasibility Rate Obtained by CETDE Over 25 Runs on 28 Instances With Four Dimension Levels Over IEEE CEC2018 Problems.

Prob.	Mean objective / (Mean constraint violation) / Feasibility Rate			
	D = 10	D = 30	D = 50	D = 100
C01	0.000E+0 / (0.00E+0) / [100%]	7.124E-29 / (0.00E+0) / [100%]	4.614E-27 / (0.00E+0) / [100%]	1.201E-12 / (0.00E+0) / [100%]
C02	0.000E+0 / (0.00E+0) / [100%]	5.597E-29 / (0.00E+0) / [100%]	3.280E-27 / (0.00E+0) / [100%]	7.110E-13 / (0.00E+0) / [100%]
C03	3.684E+4 / (0.00E+0) / [100%]	1.721E+5 / (0.00E+0) / [100%]	5.511E+5 / (0.00E+0) / [100%]	2.229E+6 / (0.00E+0) / [100%]
C04	1.357E+1 / (0.00E+0) / [100%]	1.357E+1 / (0.00E+0) / [100%]	2.208E+1 / (0.00E+0) / [100%]	2.051E+2 / (0.00E+0) / [100%]
C05	0.000E+0 / (0.00E+0) / [100%]	2.625E-28 / (0.00E+0) / [100%]	1.963E-27 / (0.00E+0) / [100%]	1.521E-10 / (0.00E+0) / [100%]
C06	1.538E+2 / (4.13E-2) / [20%]	3.000E+2 / (0.00E+0) / [100%]	6.019E+2 / (0.00E+0) / [100%]	1.949E+3 / (0.00E+0) / [100%]
C07	-2.345E+2 / (0.00E+0) / [100%]	-3.904E+2 / (0.00E+0) / [100%]	-6.602E+2 / (0.00E+0) / [100%]	-1.015E+3 / (0.00E+0) / [100%]
C08	-1.348E-3 / (0.00E+0) / [100%]	-2.840E-4 / (0.00E+0) / [100%]	-1.293E-4 / (0.00E+0) / [100%]	1.511E-3 / (0.00E+0) / [100%]
C09	-4.975E-3 / (0.00E+0) / [100%]	-2.666E-3 / (0.00E+0) / [100%]	-2.037E-3 / (0.00E+0) / [100%]	0.000E+0 / (0.00E+0) / [100%]
C10	-5.096E-4 / (0.00E+0) / [100%]	-1.028E-4 / (0.00E+0) / [100%]	-4.760E-5 / (0.00E+0) / [100%]	5.085E-4 / (0.00E+0) / [100%]
C11	-1.688E-1 / (0.00E+0) / [100%]	-1.101E+1 / (8.77E-14) / [16%]	-2.641E+2 / (3.25E+0) / [0%]	-5.849E+3 / (2.69E+2) / [0%]
C12	3.989E+0 / (0.00E+0) / [100%]	3.983E+0 / (0.00E+0) / [100%]	3.982E+0 / (0.00E+0) / [100%]	3.235E+1 / (0.00E+0) / [100%]
C13	0.000E+0 / (0.00E+0) / [100%]	3.509E-26 / (0.00E+0) / [100%]	1.375E+2 / (0.00E+0) / [100%]	1.817E+4 / (5.35E+1) / [0%]
C14	2.960E+0 / (0.00E+0) / [100%]	1.692E+0 / (0.00E+0) / [100%]	1.291E+0 / (0.00E+0) / [100%]	8.099E-1 / (0.00E+0) / [100%]
C15	8.639E+0 / (0.00E+0) / [100%]	1.492E+1 / (0.00E+0) / [100%]	1.492E+1 / (0.00E+0) / [100%]	1.806E+1 / (0.00E+0) / [100%]
C16	3.770E+1 / (0.00E+0) / [100%]	1.319E+2 / (0.00E+0) / [100%]	2.325E+2 / (0.00E+0) / [100%]	4.964E+2 / (0.00E+0) / [100%]
C17	1.301E-1 / (9.00E+0) / [0%]	9.084E-1 / (3.10E+1) / [0%]	1.021E+0 / (5.10E+1) / [0%]	1.096E+0 / (1.01E+2) / [0%]
C18	3.701E+3 / (3.20E+4) / [0%]	1.916E+3 / (1.84E+7) / [0%]	9.999E+3 / (3.95E+8) / [0%]	6.875E+4 / (5.06E+9) / [0%]
C19	0.000E+0 / (1.33E+4) / [0%]	0.000E+0 / (4.27E+4) / [0%]	0.000E+0 / (7.22E+4) / [0%]	0.000E+0 / (1.46E+5) / [0%]
C20	5.202E-1 / (0.00E+0) / [100%]	1.811E+0 / (0.00E+0) / [100%]	3.341E+0 / (0.00E+0) / [100%]	8.774E+0 / (0.00E+0) / [100%]
C21	3.988E+0 / (0.00E+0) / [100%]	9.775E+0 / (0.00E+0) / [100%]	1.794E+1 / (0.00E+0) / [100%]	5.029E+0 / (0.00E+0) / [100%]
C22	3.325E-27 / (0.00E+0) / [100%]	8.060E+1 / (0.00E+0) / [96%]	3.609E+2 / (0.00E+0) / [100%]	5.299E+3 / (0.00E+0) / [76%]
C23	2.980E+0 / (0.00E+0) / [100%]	1.409E+0 / (0.00E+0) / [100%]	1.100E+0 / (0.00E+0) / [100%]	7.842E-1 / (0.00E+0) / [100%]
C24	8.639E+0 / (0.00E+0) / [100%]	1.178E+1 / (0.00E+0) / [100%]	1.492E+1 / (0.00E+0) / [100%]	1.806E+1 / (0.00E+0) / [100%]
C25	4.398E+1 / (0.00E+0) / [100%]	1.524E+2 / (0.00E+0) / [100%]	2.592E+2 / (0.00E+0) / [100%]	5.168E+2 / (0.00E+0) / [100%]
C26	7.048E-2 / (9.00E+0) / [0%]	9.306E-1 / (3.10E+1) / [0%]	9.920E-1 / (5.10E+1) / [0%]	1.092E+0 / (1.01E+2) / [0%]
C27	1.850E+4 / (9.15E+6) / [0%]	1.236E+4 / (7.69E+8) / [0%]	9.916E+4 / (9.02E+9) / [0%]	2.169E+5 / (9.83E+10) / [0%]
C28	0.000E+0 / (1.33E+4) / [0%]	6.056E+1 / (4.29E+4) / [0%]	1.385E+2 / (7.25E+4) / [0%]	3.295E+2 / (1.47E+5) / [0%]

performance of all algorithms based on the mean objective values over 25 runs. We obtained the experimental results regarding to competitors from their corresponding original references. The multi-problem Friedman’s test is carried out by the KEEL software [53] and the results are shown in Fig. 7. It can be found that CETDE performs worse than IUDE and better than MAG-ES at 4 different dimension levels. When compared with LSHADE, CETDE performs worse at $D = 100$ and performs better at other three dimension levels. Overall, CETDE achieves very competitive results when solving problems from IEEE CEC2018.

E. APPLICATION TO SOME REAL-WORLD PROBLEMS

Experiments on artificial benchmarks have shown the competitive performance of CETDE. To further validate the performance of CETDE when solving real-world problems, here five commonly used mechanical design problems are selected to evaluate CETDE. These five real-world problems are welded beam, tension/compression spring, speed reducer, three-bar truss and pressure vessel design problems. For each problem, we run CETDE 30 times and then compare the ‘Best’, ‘Median’, ‘Worst’, ‘Mean’ and ‘Std’ values with

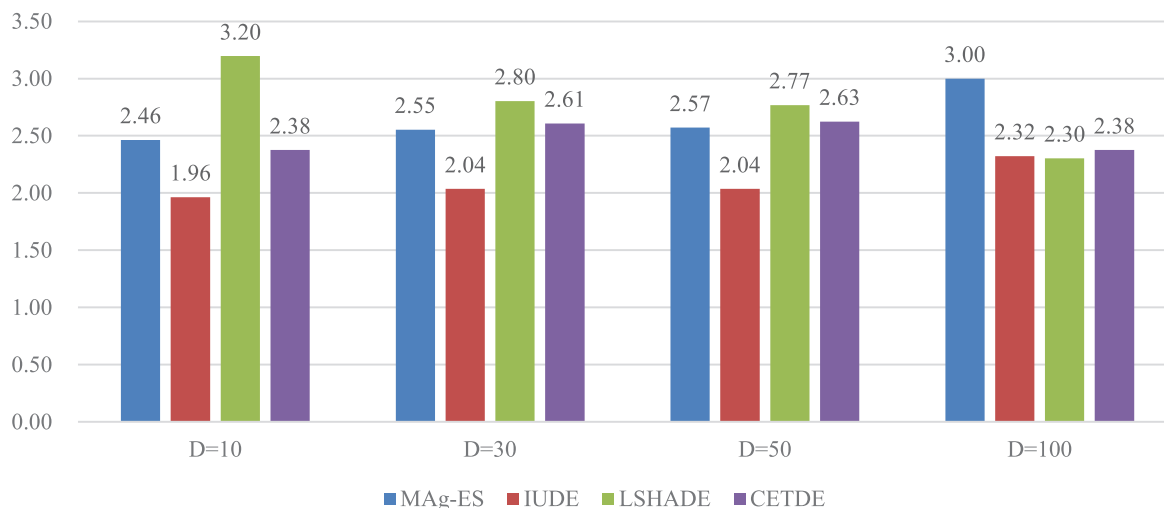


FIGURE 7. Ranking of MAG-ES, IUDE, LSHADE and CETDE by Friedman test for IEEE CEC2018 problems with 10D, 30D, 50D and 100D.

TABLE 9. Comparison of CETDE With Other Seven COEAs on the Five Real-World Problems in Terms of the Best, Worst, Mean, Median and Standard Deviation (SD) Values.

Prob.		CETDE	DELIC [54]	COMDE [55]	iMDDE [36]	CCiALF [56]	NDE [57]	MVDE [58]	PSO-DE [59]
Welded beam	Best	1.724852	1.724852	1.724852	1.724852	1.724852	1.724852	1.724853	1.724852
	Median	1.724853	1.724852	1.724852	-	-	1.724852	-	-
	Mean	1.724853	1.724852	1.724852	1.724852	1.724852	1.724852	1.724862	1.724852
	Worst	1.724855	1.724852	1.724852	1.724852	1.724854	1.724852	1.724922	1.724852
	SD	4.60E-07	4.10E-13	1.20E-12	7.71E-11	5.11E-07	3.73E-12	7.88E-06	6.70E-16
	FES	20,000	20,000	20,000	15,000	10,000	8,000	15,000	66,600
Spring design	Best	0.012665	0.012665	0.012665	0.012665	0.012665	0.012665	0.012665	0.012665
	Median	0.012665	0.012665	0.012665	-	-	0.012665	-	-
	Mean	0.012665	0.012665	0.012667	0.012665	0.012665	0.012669	0.012667	0.012665
	Worst	0.012665	0.012666	0.012677	0.012667	0.012665	0.012687	0.012719	0.012665
	SD	9.10E-10	1.30E-07	3.09E-06	8.48E-07	9.87E-08	5.38E-06	2.45E-06	4.90E-12
	FES	24,000	20,000	24,000	10,000	5,000	24,000	10,000	42,100
Speed reducer	Best	2994.471084	2994.471066	2994.471066	2994.471066	2994.471066	2994.471066	2994.471066	2996.348165
	Median	2994.471099	2994.471066	2994.471066	-	-	2994.471066	-	-
	Mean	2994.471108	2994.471066	2994.471066	2994.471066	2994.471066	2994.471066	2994.471066	2996.348165
	Worst	2994.471197	2994.471066	2994.471066	2994.471066	2994.471066	2994.471066	2994.471069	2996.348166
	SD	2.45E-05	1.90E-12	1.54E-12	7.93E-13	2.31E-12	4.17E-12	2.82E-07	1.00E-07
	FES	18,000	30,000	21,000	19,920	10,000	18,000	30,000	70,100
Three bar truss	Best	263.895845	263.895843	263.895843	263.895843	263.895843	263.895843	263.895843	263.895843
	Median	263.895863	263.895843	263.895843	-	-	263.895843	-	-
	Mean	263.895885	263.895843	263.895843	263.895843	263.895843	263.895843	263.895843	263.895843
	Worst	263.896096	263.895843	263.895843	263.895843	263.895843	263.895843	263.895855	263.895843
	SD	5.55E-05	4.34E-14	5.34E-13	0.00E+00	4.23E-14	0.00E+00	2.58E-07	1.20E-10
	FES	5,000	10,000	7,000	4,920	5,000	4,000	7,000	17,600
Pressure vessel	Best	6059.714335	6059.714300	6059.714335	6059.714335	6059.714335	6059.714335	6059.714387	6059.714335
	Median	6059.714339	6059.714300	6059.714335	-	-	6059.714335	-	-
	Mean	6059.714348	6059.714300	6059.714335	6059.714335	6059.714335	6059.714335	6059.997236	6059.714335
	Worst	6059.714557	6059.714300	6059.714335	6059.714335	6059.714335	6059.714335	6090.533528	6059.714335
	SD	3.98E-05	2.10E-11	3.62E-10	7.57E-07	1.01E-11	4.56E-07	2.91E+00	1.00E-10
	FES	20,000	30,000	30,000	15,000	12,000	20,000	15,000	42,100

seven competing methods, i.e. DELIC [54], COMDE [55], iMDDE [36], CCiALF [56], NDE [57], MVDE [58] and PSO-DE [59].

The statistical results for all problems are summarized in Table 9. Please note that the final results of the competing methods are directly taken from the corresponding references and the symbol ‘-’ means the corresponding values are not available. It can be found from Table 9 that for most problems,

CETDE obtained very similar results with other competing methods and CETDE is a potential method to deal with constrained real-world problems.

V. ADDITIONAL DISCUSSION

In our CETDE, three main components contribute to obtaining high performance when solving COPs, i.e., the multistrategy-based DE variant, two-level epsilon method

TABLE 10. Comparison of Ensemble DE Variant With Single Mutation Strategy-Based DE Variants in Terms of Median Function Error Value, Feasible Rate and Success Rate.

Prob.	Mean±Std (feasible rate) [success rate]				
	CDE-1	CDE-2	CDE-3	CETDE	
G02	1.43E-04±2.41E-05 (100%) [0%]	3.28E-03±7.52E-03 (100%) [80%]	1.42E-02±1.41E-02 (100%) [32%]	2.17E-04±1.09E-03 (100%) [96%]	
G03	5.33E-01±1.12E-01 (100%) [0%]	3.24E-10±1.48E-10 (100%) [100%]	9.48E-10±2.03E-09 (100%) [100%]	3.00E-08±8.17E-09 (100%) [100%]	
G07	1.20E-03±4.26E-04 (100%) [0%]	1.80E-09±6.78E-10 (100%) [100%]	2.82E-08±4.74E-08 (100%) [100%]	1.66E-07±1.24E-07 (100%) [100%]	
G10	1.41E-03±8.26E-04 (100%) [0%]	-7.31E-12±3.19E-13 (100%) [100%]	4.52E-05±1.92E-04 (100%) [92%]	8.06E-09±6.41E-09 (100%) [100%]	
G13	8.11E-01±5.16E-01 (0%) [0%]	3.08E-02±1.07E-01 (100%) [92%]	1.61E-01±2.44E-01 (100%) [64%]	1.10E-11±7.39E-12 (100%) [100%]	
G14	8.92E-05±4.70E-05 (100%) [68%]	5.07E-11±1.68E-11 (100%) [100%]	9.36E-10±1.36E-09 (100%) [100%]	6.62E-09±8.55E-09 (100%) [100%]	
G17	1.38E+01±2.71E+01 (100%) [8%]	8.89E+00±2.46E+01 (100%) [88%]	2.52E+01±8.09E+01 (100%) [76%]	1.46E-13±5.04E-13 (100%) [100%]	
G18	7.54E-04±6.58E-04 (100%) [0%]	2.13E-16±3.07E-17 (100%) [100%]	7.64E-03±3.82E-02 (100%) [96%]	7.44E-11±3.61E-10 (100%) [100%]	
G19	5.91E-02±3.93E-02 (100%) [0%]	1.61E-10±5.54E-11 (100%) [100%]	1.66E-07±2.82E-07 (100%) [100%]	1.38E-06±1.60E-06 (100%) [100%]	
G21	3.67E+01±6.00E+01 (100%) [72%]	1.06E-07±6.58E-08 (100%) [100%]	1.05E+01±3.63E+01 (100%) [92%]	1.16E-06±2.13E-06 (100%) [100%]	
G23	3.94E+02±1.63E+02 (72%) [0%]	1.64E-06±7.02E-07 (100%) [100%]	1.20E+01±6.00E+01 (100%) [96%]	7.87E-06±7.58E-06 (100%) [100%]	

TABLE 11. Comparison of Two-Level Epsilon-Based DE Variant With Feasibility Rule and Epsilon Method-Based DE Variants in Terms of Median Function Error Value, Feasible Rate and Success Rate.

Prob.	Mean±Std (feasible rate) [success rate]				
	CDE-4	CDE-5	CDE-6	CDE-7	CETDE
G02	2.85E-03±5.39E-03 (100%) [76%]	2.30E-03±4.80E-03 (100%) [80%]	1.22E-02±1.05E-02 (100%) [24%]	1.48E-02±1.78E-02 (100%) [40%]	2.17E-04±1.09E-03 (100%) [96%]
G03	2.53E-03±8.96E-03 (100%) [92%]	3.36E-10±1.63E-10 (100%) [100%]	-2.74E-15±1.26E-16 (100%) [100%]	-2.74E-15±1.53E-16 (100%) [100%]	3.00E-08±8.17E-09 (100%) [100%]
G10	-7.35E-12±2.52E-13 (100%) [100%]	1.67E+03±4.63E+03 (88%) [88%]	-7.31E-12±1.82E-13 (100%) [100%]	-7.35E-12±3.64E-13 (100%) [100%]	8.06E-09±6.41E-09 (100%) [100%]
G13	6.16E-02±1.44E-01 (100%) [84%]	1.26E-11±6.29E-12 (100%) [100%]	3.08E-02±1.07E-01 (100%) [92%]	-2.15E-16±1.39E-17 (100%) [100%]	1.10E-11±7.39E-12 (100%) [100%]
G17	2.96E+00±1.48E+01 (100%) [96%]	2.96E+01±3.70E+01 (100%) [60%]	3.64E-13±7.43E-13 (100%) [100%]	6.55E-13±8.91E-13 (100%) [100%]	1.46E-13±5.04E-13 (100%) [100%]
G18	4.13E-16±9.10E-16 (100%) [100%]	2.13E-16±3.07E-17 (100%) [100%]	2.22E-16±0.00E+00 (100%) [100%]	7.64E-03±3.82E-02 (100%) [96%]	7.44E-11±3.61E-10 (100%) [100%]
G21	1.21E-07±7.52E-08 (100%) [100%]	-1.94E+02±2.90E-14 (0%) [0%]	3.14E+01±5.71E+01 (100%) [76%]	5.24E+00±2.62E+01 (100%) [96%]	1.16E-06±2.13E-06 (100%) [100%]

and diversity enhancement scheme. To study whether these components are essential and whether they indeed can improve the search ability of DE, in this section, we provide some additional experiments to demonstrate the effectiveness and rationality of these components in CETDE.

A. EFFECTIVENESS OF THE ENSEMBLE DE METHOD

The proposed CETDE contains three different mutation operators when solving COPs, and here, we design three additional methods to test the necessity of using multiple strategies. To facilitate the experiments, we denote these three methods as CDE-1, CDE-2 and CDE-3 as follows:

- (1) CDE-1: DE using strategy “DE/rand/2”;
- (2) CDE-2: DE using strategy “DE/current-to-rand/1”;
- (3) CDE-3: DE using strategy “DE/current-to-best/1”;

We also run each method 25 times and compute the Mean, Std, feasible rate (FR) and success rate (SR) values

after 5×10^5 FES. The experimental results are presented in Table 10. It should be noted that all of the methods obtained similar results on 11 instances (i.e., G01, G04, G05, G06, G08, G09, G11, G12, G15, G16 and G24) and could not find any feasible solutions on two instances (i.e., G20 and G22), and we do not show the results in Table 10 for these instances. An examination of the data presented in Table 10 shows that CETDE achieves better results than the other three methods in terms of most indicators, particularly for FR and SR. Based on this study, we conclude that constrained DE with multiple strategies performs better than DE with a single mutation strategy.

B. EFFECTIVENESS OF THE TWO-LEVEL EPSILON METHOD

In CETDE, the two-level ϵ method is proposed as a CHT to solve COPs. To determine the effectiveness of this new

TABLE 12. Experimental Results of Non-Diversity Promotion Scheme Based Variant and on 22 Instances in Terms of Median Function Error Value, Feasible Rate and Success Rate.

Prob.	Mean±Std (feasible rate) [success rate]	CDE-8	CETDE
		G02	7.49E-03±1.07E-02 (100%) [56%]
G13	4.62E-02±1.28E-01 (100%) [88%]	1.10E-11±7.39E-12 (100%) [100%]	
G17	8.89E+00±2.46E+01 (100%) [88%]	1.46E-13±5.04E-13 (100%) [100%]	

method, four CHTs, i.e., the feasibility rule [17], ordinary ε method [18], an improved ε method [19] and adaptive ε method [20], are combined with the ensemble DE variant to solve COPs. To facilitate the study, we denote these variants as CDE-4, CDE-5, CDE-6 and CDE-7 as follows:

- (1) CDE-4: ensemble DE variant using feasibility rule;
- (2) CDE-5: ensemble DE variant using ordinary ε method
- (3) CDE-6: ensemble DE variant using improved ε method;
- (4) CDE-7: ensemble DE variant using adaptive ε method.

All algorithms are carried out 25 times, and the Mean, Std, FR and SR values after 5×10^5 FES are also summarized in Table 11. Here we do not present the results on 18 instances (i.e., G01, G04, G05, G07, G08, G09, G11, G12, G14, G15, G16, G18, G19, G20, G22, G23 and G24) because all methods achieve similar results. It is found from Table 11 that CETDE obtains slightly better/comparable results than those achieved by other three variants, showing that the two-level epsilon method is an effective CHT for solving COPs.

C. EFFECTIVENESS OF THE DIVERSITY PROMOTION SCHEME

In CETDE, a diversity scheme is proposed to enhance the population diversity. To experimentally validate this scheme's effectiveness, we implemented a new method by removing this scheme. We denote this method as CDE-6 and compare it with CETDE on 24 instances over 25 runs. Table 12 gives the Mean, Std, FR and SR values after 5×10^5 FES. We note that CDE-8 and CETDE obtained different results on 3 out of 24 instances. Thus, only these three instances, i.e., G02, G13 and G17, are summarized. It is found that CETDE performs better than CDE-6 in term of most indicators, particularly for the SR values. Thus, the proposed diversity scheme can improve the search ability of DE.

VI. CONCLUSION

The performance of a COEA strongly depends on the search algorithm and CHT. In this article, we proposed a new COEA, denoted as CETDE, to solve COPs by combining an ensemble DE variant and an improved two-level epsilon method. In CETDE, both the control parameters and mutation strategies with various characteristics are introduced into the DE algorithm. All the parameters and strategies coexist throughout the search process and compete to generate new solutions. We also introduced a generation and a population comparison level into the original epsilon method to retain more

promising solutions without degrading the solution quality. Moreover, a diversity scheme is proposed to increase the population diversity when the search becomes trapped in a local domain.

The proposed CETDE method is compared with some state-of-the-art COEAs on two sets of artificial benchmarks and 5 mechanical engineering design problems. Extensive and systematic studies show that the proposed CETDE achieved better or comparable results on these selected problems. The effectiveness and benefits of the ensemble DE variant, two-level epsilon method and diversity scheme are experimentally validated by some additional studies. The experimental results demonstrate the effectiveness of these components for solving COPs.

In the future, we would like to further enhance the search ability of CETDE for more complex COPs. Another interesting research direction would be to extend our DE variant, two-level epsilon method and diversity scheme to constrained multiobjective optimization problems.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable suggestions and comments.

REFERENCES

- [1] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. Coello, and K. Deb, "Problem definitions and evaluation criteria for the CEC 2006 special session on constrained real-parameter optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., 2006.
- [2] W. Gao, L. Huang, Y. Luo, Z. Wei, and S. Liu, "Constrained optimization by artificial bee colony framework," *IEEE Access*, vol. 6, pp. 73829–73845, 2018.
- [3] K. Yu, J. J. Liang, B. Y. Qu, X. Chen, and H. Wang, "Parameters identification of photovoltaic models using an improved JAYA optimization algorithm," *Energy Convers. Manage.*, vol. 150, pp. 742–753, Oct. 2017.
- [4] X. Chen, C. Mei, B. Xu, K. Yu, and X. Huang, "Quadratic interpolation based teaching-learning-based optimization for chemical dynamic system optimization," *Knowl.-Based Syst.*, vol. 145, pp. 250–263, Apr. 2018.
- [5] W. Yi, Y. Zhou, L. Gao, X. Li, and C. Zhang, "Engineering design optimization using an improved local search based epsilon differential evolution algorithm," *J. Intell. Manuf.*, vol. 29, no. 7, pp. 1559–1580, Oct. 2018.
- [6] K. Sallam, S. Elsayed, R. Sarker, and D. Essam, "Landscape-based differential evolution for constrained optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.
- [7] B. Xu, W. Cheng, F. Qian, and X. Huang, "Self-adaptive differential evolution with multiple strategies for dynamic optimization of chemical processes," *Neural Comput. Appl.*, vol. 31, no. 7, pp. 2041–2061, Jul. 2019.
- [8] X. Chen, H. Tianfield, and K. Li, "Self-adaptive differential artificial bee colony algorithm for global optimization problems," *Swarm Evol. Comput.*, vol. 45, pp. 70–91, Mar. 2019.
- [9] Y. Wang and Z. Cai, "Constrained evolutionary optimization by means of $(\mu+\lambda)$ -differential evolution and improved adaptive trade-off model," *Evol. Comput.*, vol. 19, no. 2, pp. 249–285, Jun. 2011.
- [10] F. Qian, B. Xu, R. Qi, and H. Tianfield, "Self-adaptive differential evolution algorithm with α -constrained-domination principle for constrained multi-objective optimization," *Soft Comput.*, vol. 16, no. 8, pp. 1353–1372, Aug. 2012.
- [11] B.-C. Wang, Y. Feng, and H.-X. Li, "Individual-dependent feasibility rule for constrained differential evolution," *Inf. Sci.*, vol. 506, pp. 174–195, Jan. 2020.
- [12] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm Evol. Comput.*, vol. 1, no. 4, pp. 173–194, Dec. 2011.

- [13] H. Cai, B. Liu, and S. Pan, "On the cooperation between evolutionary algorithms and constraint handling techniques: A further empirical study," *IEEE Access*, vol. 8, pp. 130598–130606, 2020.
- [14] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *IEEE Trans. Evol. Comput.*, vol. 4, no. 3, pp. 284–294, 2000.
- [15] M. Gan, H. Peng, X. Peng, X. Chen, and G. Inoussa, "An adaptive decision maker for constrained evolutionary optimization," *Appl. Math. Comput.*, vol. 215, no. 12, pp. 4172–4184, Feb. 2010.
- [16] B. Tessema and G. G. Yen, "An adaptive penalty formulation for constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 39, no. 3, pp. 565–578, May 2009.
- [17] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, Jun. 2000.
- [18] T. Takahama and S. Sakai, "Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 1–8.
- [19] Z. Fan, W. Li, X. Cai, H. Huang, Y. Fang, Y. You, J. Mo, C. Wei, and E. Goodman, "An improved epsilon constraint-handling method in MOEA/D for CMOPs with large infeasible regions," *Soft Comput.*, vol. 23, no. 23, pp. 12491–12510, Dec. 2019.
- [20] C. Zhang, A. K. Qin, W. Shen, L. Gao, K. C. Tan, and X. Li, "Epsilon-constrained differential evolution using an adaptive epsilon-level control method," *IEEE Trans. Syst., Man, Cybern., Syst.*, early access, Aug. 4, 2020, doi: [10.1109/TSMC.2020.3010120](https://doi.org/10.1109/TSMC.2020.3010120).
- [21] 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.
- [22] B. Xu, X. Chen, and L. Tao, "Differential evolution with adaptive trial vector generation strategy and cluster-replacement-based feasibility rule for constrained optimization," *Inf. Sci.*, vol. 435, pp. 240–262, Apr. 2018.
- [23] Z. Cai and Y. Wang, "A multiobjective optimization-based evolutionary algorithm for constrained optimization," *IEEE Trans. Evol. Comput.*, vol. 10, no. 6, pp. 658–675, Dec. 2006.
- [24] Y. Wang and Z. Cai, "Combining multiobjective optimization with differential evolution to solve constrained optimization problems," *IEEE Trans. Evol. Comput.*, vol. 16, no. 1, pp. 117–134, Feb. 2012.
- [25] S. Zeng, R. Zhang, R. Jiao, and Q. Xu, "A Pareto front transformation model for Multi-objective-based constrained optimization," *IEEE Access*, early access, Feb. 24, 2020, doi: [10.1109/ACCESS.2020.2976047](https://doi.org/10.1109/ACCESS.2020.2976047).
- [26] K. Deb and R. Datta, "A bi-objective constrained optimization algorithm using a hybrid evolutionary and penalty function approach," *Eng. Optim.*, vol. 45, no. 5, pp. 503–527, May 2013.
- [27] S. Zeng, R. Jiao, C. Li, X. Li, and J. S. Alkassabeh, "A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization," *IEEE Trans. Cybern.*, vol. 47, no. 9, pp. 2678–2688, Sep. 2017.
- [28] R. Jiao, S. Zeng, and C. Li, "A feasible-ratio control technique for constrained optimization," *Inf. Sci.*, vol. 502, pp. 201–217, Oct. 2019.
- [29] 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.
- [30] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.
- [31] W. F. Gao, G. G. Yen, and S. Y. Liu, "A dual-population differential evolution with coevolution for constrained optimization," *IEEE Trans. Cybern.*, vol. 45, no. 5, pp. 1108–1121, May 2015.
- [32] B. Xu, L. Tao, X. Chen, and W. Cheng, "Adaptive differential evolution with multi-population-based mutation operators for constrained optimization," *Soft Comput.*, vol. 23, no. 10, pp. 3423–3447, May 2019.
- [33] B. Xu, H. Zhang, M. Zhang, and L. Liu, "Differential evolution using cooperative ranking-based mutation operators for constrained optimization," *Swarm Evol. Comput.*, vol. 49, pp. 206–219, Sep. 2019.
- [34] J. Brest, V. Zumer, and M. S. Maucec, "Self-adaptive differential evolution algorithm for constrained real-parameter optimization," in *Proc. IEEE Int. Conf. Evol. Comput.*, Jul. 2006, pp. 215–222.
- [35] S. M. Elsayed, R. A. Sarker, and D. L. Essam, "A self-adaptive combined strategies algorithm for constrained optimization using differential evolution," *Appl. Math. Comput.*, vol. 241, pp. 267–282, Aug. 2014.
- [36] W. Gong, Z. Cai, and D. Liang, "Engineering optimization by means of an improved constrained differential evolution," *Comput. Methods Appl. Mech. Eng.*, vol. 268, pp. 884–904, Jan. 2014.
- [37] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.
- [38] B.-C. Wang, H.-X. Li, J.-P. Li, and Y. Wang, "Composite differential evolution for constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 7, pp. 1482–1495, Jul. 2019.
- [39] G. Sun, B. Yang, Z. Yang, and G. Xu, "An adaptive differential evolution with combined strategy for global numerical optimization," *Soft Comput.*, vol. 24, no. 9, pp. 6277–6296, May 2020.
- [40] R. Mallipeddi, P. N. Suganthan, Q. K. Pan, and M. F. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Appl. Soft Comput.*, vol. 11, no. 2, pp. 1679–1696, Mar. 2011.
- [41] F. Neri and V. Tirronen, "Recent advances in differential evolution: A survey and experimental analysis," *Artif. Intell. Rev.*, vol. 33, nos. 1–2, pp. 61–106, Feb. 2010.
- [42] K. Yu, X. Wang, and Z. Wang, "Constrained optimization based on improved teaching–learning–based optimization algorithm," *Inf. Sci.*, vols. 352–353, pp. 61–78, Jul. 2016.
- [43] G. Wu, R. Mallipeddi, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition and special session on constrained single objective real-parameter optimization," Nat. Univ. Defense Technol., Changsha, China, P. R. China Kyungpook Nat. Univ., Daegu, South Korea, Nanyang Technol. Univ., Singapore, Tech. Rep., 2017.
- [44] X. Yu, Y. Lu, X. Wang, X. Luo, and M. Cai, "An effective improved differential evolution algorithm to solve constrained optimization problems," *Soft Comput.*, vol. 23, no. 7, pp. 2409–2427, Apr. 2019.
- [45] W. Zhang, G. G. Yen, and Z. He, "Constrained optimization via artificial immune system," *IEEE Trans. Cybern.*, vol. 44, no. 2, pp. 185–198, Feb. 2014.
- [46] K. M. Ang, W. H. Lim, N. A. M. Isa, S. S. Tiang, and C. H. Wong, "A constrained multi-swarm particle swarm optimization without velocity for constrained optimization problems," *Expert Syst. Appl.*, vol. 140, pp. 1–23, Feb. 2020.
- [47] Y. Liang, Z. Wan, and D. Fang, "An improved artificial bee colony algorithm for solving constrained optimization problems," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 3, pp. 739–754, Jun. 2017.
- [48] B.-C. Wang, H.-X. Li, and Y. Feng, "An improved teaching–learning–based optimization for constrained evolutionary optimization," *Inf. Sci.*, vol. 456, pp. 131–144, Aug. 2018.
- [49] R. P. Parouha and K. N. Das, "An efficient hybrid technique for numerical optimization and applications," *Comput. Ind. Eng.*, vol. 83, pp. 193–216, May 2015.
- [50] M. Hellwig and H.-G. Beyer, "A matrix adaptation evolution strategy for constrained real-parameter optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.
- [51] A. Trivedi, K. Sanyal, P. Verma, and D. Srinivasan, "A unified differential evolution algorithm for constrained optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 1–18.
- [52] Z. Fan, Y. Fang, W. Li, Y. Yuan, Z. Wang, and X. Bian, "LSHADE44 with an improved constraint-handling method for solving constrained single-objective optimization problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2018, pp. 1–8.
- [53] J. Alcalá-Fdez, L. Sánchez, S. García, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas, J. C. Fernández, and F. Herrera, "KEEL: A software tool to assess evolutionary algorithms for data mining problems," *Soft Comput.*, vol. 13, no. 3, pp. 307–318, Feb. 2009.
- [54] L. Wang and L.-P. Li, "An effective differential evolution with level comparison for constrained engineering design," *Struct. Multidisciplinary Optim.*, vol. 41, no. 6, pp. 947–963, Jun. 2010.
- [55] A. W. Mohamed and H. Z. Sabry, "Constrained optimization based on modified differential evolution algorithm," *Inf. Sci.*, vol. 194, pp. 171–208, Jul. 2012.
- [56] B. Ghasemishabankareh, X. Li, and M. Ozlen, "Cooperative coevolutionary differential evolution with improved augmented lagrangian to solve constrained optimisation problems," *Inf. Sci.*, vol. 369, pp. 441–456, Nov. 2016.
- [57] A. W. Mohamed, "A novel differential evolution algorithm for solving constrained engineering optimization problems," *J. Intell. Manuf.*, vol. 29, no. 3, pp. 659–692, Mar. 2018.

- [58] V. V. de Melo and G. L. C. Carosio, "Investigating multi-view differential evolution for solving constrained engineering design problems," *Expert Syst. Appl.*, vol. 40, no. 9, pp. 3370–3377, Jul. 2013.
- [59] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Appl. Soft Comput.*, vol. 10, no. 2, pp. 629–640, Mar. 2010.



BIN XU received the B.S. degree in information engineering and the Ph.D. degree in control science and engineering from the East China University of Science and Technology (ECUST), in 2007 and 2013, respectively. He is currently an Associate Professor with the School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science (SUES). His current research interests include evolutionary computation, constrained optimization, and multiobjective optimization and their applications in engineering.



ZONGHAO ZHANG received the B.S. degree from Tongling University, in 2018. He is currently pursuing the M.S. degree with the Shanghai University of Engineering Science. His current research interests include intelligent computing algorithms and their application in path planning.

• • •