Utilizing the Correlation between Constraints and Objective Function for Constrained Evolutionary Optimization

Yong Wang, *Senior Member, IEEE*, Jia-Peng Li, Xihui Xue, and Bing-Chuan Wang

***Abstract*—When solving constrained optimization problems by evolutionary algorithms, the core issue is to balance constraints and objective function. This paper is the first attempt to utilize the correlation between constraints and objective function to keep this balance. First of all, the correlation between constraints and objective function is mined and represented by a correlation index. Afterward, a weighted sum updating approach and an archiving and replacement mechanism are proposed to make use of this correlation index to guide the evolution. By the above process, a novel constrained optimization evolutionary algorithm is presented. Experiments on a broad range of benchmark test functions indicate that the proposed method shows better or at least competitive performance against other state-of-the-art methods. Moreover, the proposed method is applied to the gait optimization of humanoid robots.**

***Index Terms*—Constrained optimization, evolutionary algorithms, constraints, objective function, correlation, humanoid robots**

I. Introduction

MANY scientific and engineering problems can be formulated as constrained optimization problems (COPs). Without loss of generality, a COP can be formulated as follows:

$$\begin{aligned} \text{minimize}: \quad & f(\vec{x}),\ \vec{x}=(x_1,\ldots,x_D)\in S,\ L_i\le x_i\le U_i \\ \text{subject to}: \quad & g_j(\vec{x})\le 0,\ j=1,\ldots,l \\ & h_j(\vec{x})=0,\ j=l+1,\ldots,m \end{aligned}$$

where $f(\vec{x})$ is the objective function, $\vec{x}=(x_1,\ldots,x_D)$ is the decision vector, L_i and U_i are the lower and upper bounds of x_i, respectively, $S=\prod_{i=1}^{D}[L_i,U_i]$ is the decision space, $g_j(\vec{x})$ is the jth inequality constraint, $h_j(\vec{x})$ is the jth equality constraint, and l and $(m-l)$ are the number of inequality and equality constraints, respectively.

In constrained optimization, the degree of constraint violation of $\vec{x}$ on the jth constraint is computed as

$$G_j(\vec{x})=\begin{cases}\max(0,g_j(\vec{x})), 1\le j\le l\\ \max(0,|h_j(\vec{x})|-\delta), l+1\le j\le m\end{cases} \quad (1)$$

This work was supported in part by the Innovation-Driven Plan in Central South University under Grant 2018CX010, in part by the National Natural Science Foundation of China under Grant 61673397, in part by the Hunan Provincial Natural Science Fund for Distinguished Young Scholars (Grant No. 2016JJ1018), and in part by the Beijing Advanced Innovation Center for Intelligent Robots and Systems under Grant 2018IRS06. (*Corresponding author: Bing-Chuan Wang*).

Y. Wang, J.-P. Li, and X. Xue are with the School of Automation, Central South University, Changsha 410083, China. (e-mail: ywang@csu.edu.cn; ljpcsu@csu.edu.cn; 0909110918@csu.edu.cn).

B.-C. Wang is with the Department of Systems Engineering and Engineering Management, City University of Hong Kong, Hong Kong. (e-mail: bingcwang3-c@my.cityu.edu.hk).

As shown in (1), an equality constraint is relaxed by a positive tolerance value δ. Subsequently, the degree of constraint violation of $\vec{x}$ on all the constraints can be expressed as

$$G(\vec{x})=\sum_{j=1}^{m}G_j(\vec{x}) \quad (2)$$

A solution which satisfies $G(\vec{x})=0$ is called a feasible solution; otherwise, it is called an infeasible solution. The feasible solution with the smallest objective function value is the feasible optimum of a COP. The target of solving a COP is to locate the feasible optimum.

Because of their powerful search ability, evolutionary algorithms (EAs) have attracted increasing attention for dealing with COPs. When applying an EA for constrained optimization, a constraint-handling technique should be integrated. As a result, a variety of constraint-handling techniques has been designed during the last two decades [1], [2], [3]. The current popular constraint-handling techniques can be classified into four categories: 1) methods based on penalty function [4], [5], [6], 2) methods based on treating constraints and objective function separately [7], [8], [9], 3) methods based on multiobjective optimization [10], [11], [12], and 4) hybrid methods [13], [14], [15].

Methods based on penalty function employ a penalty factor to control the use of $f(\vec{x})$ and $G(\vec{x})$. According to the manner of setting the penalty factor, these methods can be divided into three types: 1) static penalty methods, 2) dynamic penalty methods, and 3) adaptive penalty methods. In the static penalty methods, the penalty factor is set beforehand and kept the same throughout the evolution. However, it is not a trivial task to set a proper penalty factor in advance since it is usually problem-dependent. In the dynamic penalty methods, the penalty factor is varied with generation according to a predefined trend function [6]. Though the dynamic penalty methods achieve excellent performance on some COPs, the need of predefining a trend function limits their ability to a certain extent. In the adaptive penalty methods, the feedback information from the evolving process is adopted to adjust the penalty factor [16], [17]. Due to the utilization of feedback information, this type of methods has a significant advantage over the other two types of methods and most of the recent work on penalty function falls into this scope [18], [19], [20].

Methods based on treating constraints and objective function separately make use of $f(\vec{x})$ or $G(\vec{x})$ to compare solutions. Among these methods, the feasibility rule [7] is the most popular one in that it is free of the penalty factor and easy to implement. When two solutions are compared based on the feasibility rule, if both of them are infeasible, the one with smaller $G(\vec{x})$ is preferred; if only one of them is feasible, the feasible one is preferred; and if both of them are feasible, the one with smaller $f(\vec{x})$ is preferred. However, the main weakness of the feasibility rule is its preference to constraints. Stochastic ranking is another representative [8]. When two solutions are compared, the probability of comparing them based on $f(\vec{x})$ or $G(\vec{x})$ is pf or $(1-pf)$, respectively. Moreover, in order to increase the robustness, the bubble-sort-like procedure is implemented. Due to the fact that it can introduce the information of objective function, stochastic ranking can remedy the weakness of the feasibility rule to some degree. Similar to stochastic ranking, the ε constrained method takes advantage of a parameter ε to control the use of $f(\vec{x})$ [21], [22]. The ε constrained method [23] is also integrated with the local search enhanced differential evolution (DE) for constrained optimization. In addition, a dual-population mechanism is proposed in [24] to treat constraints and objective function separately.

In methods based on multiobjective optimization, a COP is transformed into a biobjective optimization problem $(f(\vec{x}), G(\vec{x}))$, or a multiobjecitve optimization problem with $(m+1)$ objectives $(f(\vec{x}), G_1(\vec{x}), \ldots, G_m(\vec{x}))$. Afterward, multiobjective optimization techniques are adopted to tackle the transformed problem [25]. In 2002, Mezura and Coello [26] conducted comprehensive experiments on four methods based on multiobjective optimization. The experimental results reveal that methods based on Pareto dominance outperform other methods [25], [27]. Hence, various Pareto dominance-based methods have been proposed subsequently [10], [11], [12], [28], [29]. Moreover, it is extensively recognized that the biobjective transformation is more reasonable than the $(m+1)$-objective transformation in this kind of methods.

According to the "no free lunch" theorem [30], there does not exist a single constraint-handling technique that can achieve the best performance on all kinds of COPs. Consequently, numerous methods which hybridize different constraint-handling techniques have been presented. For instance, Peng *et al.* [13] proposed a novel dynamic weight-based selection strategy. Moreover, this strategy is combined with the feasibility rule. Datta *et al.* combined multiobjective optimization with penalty function to solve COPs [31], [32], [33]. An adaptive tradeoff model (ATM) is presented by Wang *et al.* [15], [34]. In this model, the whole evolving process is divided into three situations, i.e., infeasible situation, semi-feasible situation, and feasible situation. In addition, different constraint-handling techniques are designed in different situations. A ranking method, inspired by the idea of ATM, is designed to improve the ability of DE to solve COPs [35]. It is noteworthy that the ensemble strategy is also proposed to make use of the advantages of different constraint-handling techniques [36].

It is well known that how to balance constraints and objective function has a significant impact on the performance of a constrained optimization EA (COEA). Unlike the previous work, this paper proposes a new concept, i.e., the correlation between constraints and objective function. Moreover, we mine this correlation via a learning stage. Afterward, we utilize this correlation via a weighted sum updating approach and an archiving and replacement mechanism, and demonstrate that this correlation can be effectively used to strike the balance between constraints and objective function. By mining and utilizing the *cor*relation between *c*onstraints and *o*bjective function, we propose an alternative COEA, named CORCO. The main contributions of this paper are highlighted as follows:

- This paper presents the first attempt to investigate the correlation between constraints and objective function in constrained evolutionary optimization.
- A learning stage is designed to mine the correlation between constraints and objective function.
- A weighted sum updating approach and an archiving and replacement mechanism are proposed to make use of the correlation for constrained optimization.
- Systematic experiments have demonstrated that CORCO provides state-of-the-art performance on three widely used benchmark test suites and the gait optimization of humanoid robots.

The rest of this paper is organized as follows. Section II introduces the proposed method, including motivation, framework, learning stage, evolving stage, and search algorithm (i.e., base optimizer or core optimization algorithm). The experiments and discussions are presented in Section III. Section IV applies the proposed method to the gait optimization of humanoid robots. Finally, Section V concludes this paper.

II. Proposed Approach

A. Motivation

When solving COPs by EAs, an important issue is how to balance constraints (i.e., degree of constraint violation) and objective function (i.e., improvement of objective function value). Note, however, that it is hard to maintain such a balance due to the fact that such a balance is not only dependent on optimization problems, but also the phases of evolution. To address this issue, a number of constraint-handling techniques have been proposed as introduced in Section I. Among them, methods based on penalty function try to balance constraints and objective function by tuning the penalty factor, where the fuzzy adaptive method is a popular one [20]. Methods based on treating constraints and objective function separately put different degree of emphasis on constraints and objective function. Methods based on multiobjective optimization utilize multiobjective optimization techniques to achieve the tradeoff between constraints and objective function. Additionally, hybrid methods intend to combine the advantages of distinct constraint-handling techniques to strike the balance between constraints and objective function. However, all of these methods have not yet mined and exploited the correlation between constraints and objective function explicitly.

It is interesting to notice that there are two kinds of relationships between constraints and objective function:

- The degree of constraint violation $G(\vec{x})$ decreases as the objective function $f(\vec{x})$ decreases.

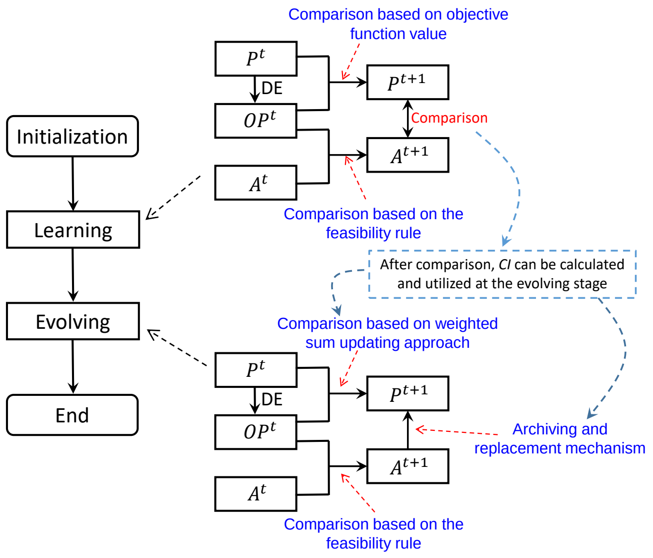

Comparison based on objective function value
Initialization
Learning
Evolving
End
P^t
DE
OP^t
A^t
P^{t+1}
A^{t+1}
Comparison
Comparison based on the feasibility rule
After comparison, CI can be calculated and utilized at the evolving stage
Comparison based on weighted sum updating approach
Archiving and replacement mechanism
Comparison based on the feasibility rule

Fig. 1. Framework of CORCO.

- The degree of constraint violation $G(\vec{x})$ does not decrease as the objective function $f(\vec{x})$ decreases.

In terms of the first kind of relationship, it is clear that $G(\vec{x})$ and $f(\vec{x})$ have a similar variation tendency. As a result, constraints are correlated to objective function. Under this condition, the information of objective function may be very helpful for the population to enter the feasible region, especially for COPs with extremely complex constraints. Note that for COPs with extremely complex constraints, the landscape of the infeasible region determined by constraints would be very complicated. Under this condition, searching only guided by constraints may make the population stagnate in the infeasible region easily. Therefore, if the overall variation tendency of objective function is similar to that of constraints and if the landscape of objective function is simpler, the information of objective function could help the population jump out of the infeasible region.

In addition, with respect to the second kind of relationship, it is obvious that $G(\vec{x})$ and $f(\vec{x})$ have different variation tendencies; thus, constraints are not correlated to objective function. Under this condition, too much information of objective function may prevent the population from entering the feasible region. Hence, we need to suppress the information of objective function and strengthen the information of constraints.

To sum up, the correlation between constraints and objective function can provide an important guidance for the balance of constraints and objective function, which is beneficial to guide the population to find the feasible region. However, this correlation is not discovered and utilized by existing COEAs. So a new COEA, named CORCO, is designed in this paper to address this issue.

B. CORCO

The framework of CORCO is given in Fig. 1. As shown in Fig. 1, CORCO includes two main stages: the learning stage and the evolving stage. The first stage is called the learning stage because it is used to learn the correlation between constraints and objective function. Subsequently, the evolving stage utilizes the correlation to guide the evolution. Indeed, the aim of the correlation is to decide how much information of objective function is utilized to avoid a local optimum caused by complex constraints. It can facilitate the population to enter the feasible region. In this paper, we are more concerned about the overall (global) variation tendencies of constraints and objective function. Thus, we learn the correlation at the early stage due to the fact that the population is distributed widely and the overall trend can be caught. After learning, the rest of computational resources are used for evolving.

At the learning stage, in order to mine the correlation, the concept of correlation index (CI) is proposed. The bigger the value of CI, the stronger the correlation between constraints and objective function. Meanwhile, a method to calculate CI is introduced.

At the evolving stage, two methods are designed. One is called the weighted sum updating approach, where the fitness value of an individual is the weighted sum of normalized $G(\vec{x})$ and normalized $f(\vec{x})$, and CI is used to calculate the weight coefficient. The other is called the archiving and replacement mechanism, in which some individuals of the main population are replaced with some individuals in a predefined archive, and CI is used to control the condition of replacement. Overall, the weighted sum updating approach and the archiving and replacement mechanism aim at balancing constraints and objective function. Moreover, CI plays an important role in both methods. Consequently, if CI is calculated and utilized properly, the balance between constraints and objective

Algorithm 1: Procedure of CORCO

```
t = 1;
Initialize $P^t = \{\vec{x}_1^t, \ldots, \vec{x}_{NP}^t\}$;
Initialize $A^t = \{\vec{a}_1^t, \ldots, \vec{a}_{NP}^t\} = P^t$;
Set $Ctr_1 = 0$ and $Ctr_2 = 0$;
Calculate $DI^1$ according to (3);
$P^{t+1} = \emptyset$ and $OP^{t+1} = \emptyset$;
/*The learning stage*/
while $t \leq LearnGen$ do
    Implement the search algorithm in Algorithm 4 on $P^t$ to generate an offspring population $OP^t = \{\vec{u}_1^t, \ldots, \vec{u}_{NP}^t\}$;
    for $i = 1 : NP$ do
        Compare $\vec{u}_i^t$ and $\vec{x}_i^t$ based on objective function value and the better one denoted as $\vec{x}_i^{t+1}$ is stored into $P^{t+1}$;
        Compare $\vec{u}_i^t$ and $\vec{a}_i^t$ based on the feasibility rule and the better one denoted as $\vec{a}_i^{t+1}$ is stored into $A^{t+1}$;
    Update $Ctr_1$ and $Ctr_2$;
    $t = t + 1$;
Calculate $DI^{LearnGen}$ according to (4);
Calculate $DV$ according to (5);
Calculate $CI$ according to (6);
/*The evolving stage*/
while $t > LearnGen$ && $t \leq MaxGen$ do
    Implement the search algorithm in Algorithm 4 on $P^t$ to generate an offspring population $OP^t = \{\vec{u}_1^t, \ldots, \vec{u}_{NP}^t\}$;
    for $i = 1 : NP$ do
        Compare $\vec{u}_i^t$ and $\vec{x}_i^t$ based on the weighted sum updating approach in Algorithm 2 and the better one denoted as $\vec{x}_i^{t+1}$ is stored into $P^{t+1}$;
        Compare $\vec{u}_i^t$ and $\vec{a}_i^t$ based on the feasibility rule and the better one denoted as $\vec{a}_i^{t+1}$ is stored into $A^{t+1}$;
    $t = t + 1$;
    Implement the archiving and replacement mechanism in Algorithm 3;
Output: The best individual in $P^t$.
```

function can be achieved.

Algorithm 1 gives the procedure of CORCO. At the beginning of CORCO, two populations will be initialized, namely, the main population $P^t = \{\vec{x}_1^t, \ldots, \vec{x}_{NP}^t\}$ and the archive $A^t = \{\vec{a}_1^t, \ldots, \vec{a}_{NP}^t\}$, where $t \in \{1, \ldots, MaxGen\}$ is the current generation number, $MaxGen$ is the maximum generation number, NP is the population size, and the initial archive is the same with the initial main population, that is, $A^1 = P^1$. After initialization, the diversity index of the initial population is calculated as

$$DI^1 = \frac{1}{D}\sum_{i=1}^{D} \frac{\text{std}(x_{1,i}^1, \ldots, x_{NP,i}^1)}{U_i - L_i} \tag{3}$$

where "std" is a function to calculate the standard deviation of the members in a vector. Subsequently, the learning stage of CORCO begins. At the end of the learning stage, the diversity index of P^t is calculated again as

$$DI^{LearnGen} = \frac{1}{D}\sum_{i=1}^{D} \frac{\text{std}(x_{1,i}^{LearnGen}, \ldots, x_{NP,i}^{LearnGen})}{U_i - L_i} \tag{4}$$

where $LearnGen$ is the maximum generation number of the learning stage. Afterward, two values will be calculated with the information acquired from the learning stage. One is CI, and the other is the diversity variation (DV) of P^t

$$DV = DI^1 - DI^{LearnGen} \tag{5}$$

Then, the evolving stage of CORCO begins. During the evolving stage, the weighted sum updating approach is used to update P^t, while the feasibility rule [7] is used to update A^t. In addition, to accelerate the convergence, the archiving and replacement mechanism is implemented. When $t = MaxGen$, the iteration ends.

C. Learning Stage

The main purpose of the learning stage is to calculate CI. At the beginning of the learning stage, two counters are initialized, namely $Ctr1 = 0$ and $Ctr2 = 0$. In each generation of the learning stage, P^t generates an offspring population OP^t by the search algorithm, which will be introduced in Section II-E. Afterward, each individual in OP^t is compared with the corresponding individual in P^t based on the objective function value, and the one with a smaller objective function value will survive into P^{t+1}. Meanwhile, each individual in OP^t is compared with the corresponding individual in A^t based on the feasibility rule, and the one with smaller degree of constraint violation will be stored into A^{t+1}. After the above processes, we verify two conditions:

- **Condition 1**: If $\min\limits_{i=1,\ldots,NP} f(\vec{a}_i^{t+1}) < \max\limits_{i=1,\ldots,NP} f(\vec{x}_i^{t+1})$, then $Ctr1 = Ctr1 + 1$.
- **Condition 2**: If $\min\limits_{i=1,\ldots,NP} G(\vec{x}_i^{t+1}) \leq \max\limits_{i=1,\ldots,NP} G(\vec{a}_i^{t+1})$, then $Ctr2 = Ctr2 + 1$.

At the end of the learning stage (i.e., after $LearnGen$ generations), CI is calculated as follows

$$CI = \frac{\min(Ctr1, Ctr2)}{LearnGen} \tag{6}$$

Next, we explain why CI calculated in this way can reflect the correlation between constraints and objective function. According to the comparison manners between OP^t and P^t and between OP^t and A^t, in general, P^{t+1} is composed of solutions with relatively smaller objective function values, while A^{t+1} is composed of solutions with relatively smaller degree of constraint violation. Once condition 1 is satisfied, it means that A^{t+1}, which is guided by the degree of constraint violation, can generate a solution with a smaller objective function value, then the value of $Ctr1$ is increased by one. Similarly, once condition 2 is satisfied, it means that P^{t+1}, which is guided by the objective function value, can generate a solution with smaller degree of constraint violation, then the value of $Ctr2$ is increased by one. When the learning stage ends, a bigger value of CI means that the objective function value and the degree of constraint violation have a more similar variation tendency in many generations. Therefore, we believe that the correlation between constraints and objective function is stronger, and vice versa.

Remark 1: As we know, most of the standard correlation analyses are derived under certain assumptions [37]. For example, Pearson's correlation coefficient is effective on time series obeying normal distribution. However, the objective function value series and the degree of constraint violation series of a COP may not satisfy these assumptions. Additionally, when calculating CI, most of the standard correlation analyses should store all values of a series. Compared with the standard correlation analyses, the proposed method does not impose any assumptions on two series, which would be more effective on different problems. Moreover, the proposed method calculates CI without storing all values, which can save much storage space. The advantage of the proposed method over Pearson's correlation coefficient has been verified in the experimental study.

D. Evolving Stage

The evolving stage aims at utilizing CI and DV obtained from the learning stage to guide the evolution. Two methods are designed, i.e., the weighted sum updating approach and the archiving and replacement mechanism.

1) Weighted Sum Updating Approach: This approach defines the following fitness value for each individual $\vec{x}_i^t$ in P^t:

$$F_{\omega_i^t}(\vec{x}_i^t) = \omega_i^t * \widetilde{f}(\vec{x}_i^t) + (1 - \omega_i^t) * \widetilde{G}(\vec{x}_i^t) \tag{7}$$

$$\widetilde{f}(\vec{x}_i^t) = \frac{f(\vec{x}_i^t) - f_{min}^t}{f_{max}^t - f_{min}^t} \tag{8}$$

$$\widetilde{G}(\vec{x}_i^t) = \frac{G(\vec{x}_i^t) - G_{min}^t}{G_{max}^t - G_{min}^t} \tag{9}$$

where $\widetilde{f}(\vec{x}_i^t)$ and $\widetilde{G}(\vec{x}_i^t)$ are the normalized objective function value and the normalized degree of constraint violation, respectively, f_{min}^t and f_{max}^t are the minimum and maximum objective function values of $P^t \cup OP^t$, respectively, and G_{min}^t and G_{max}^t are the minimum and maximum degree of

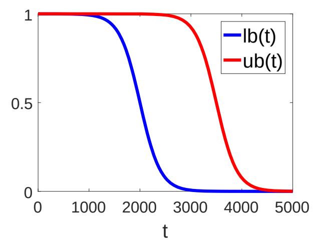
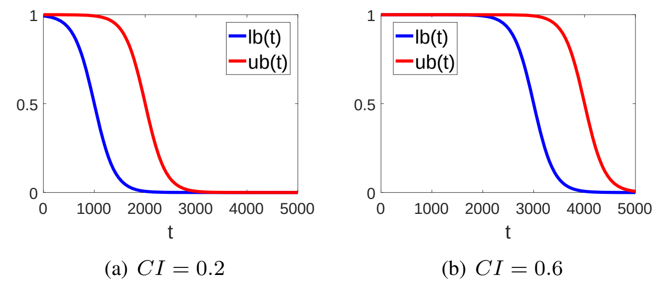
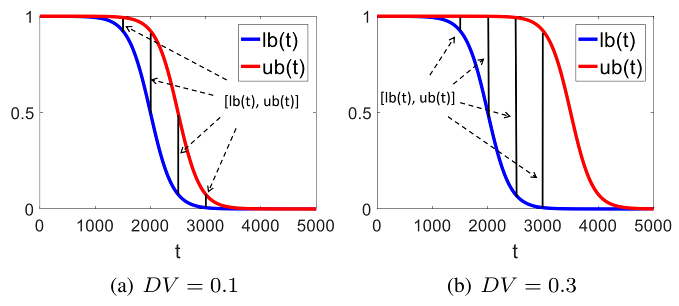


Fig. 2. Variation of $[lb(t), ub(t)]$ with respect to the generation number.



(a) $CI = 0.2$ (b) $CI = 0.6$

Fig. 3. Principle of CI ($MaxGen = 5000$ and $DV = 0.2$).



(a) $DV = 0.1$ (b) $DV = 0.3$

Fig. 4. Principle of DV ($MaxGen = 5000$ and $CI = 0.4$).

constraint violation of $P^t \cup OP^t$, respectively. ω_i^t, calculated by (10), is the weight coefficient of $\widetilde{f}(\vec{x}_i^t)$.

$$\omega_i^t = lb(t) + \frac{i}{NP}(ub(t) - lb(t)), i = 1, \dots, NP \tag{10}$$

Accordingly, $(1 - \omega_i^t)$ is the weight coefficient of $\widetilde{G}(\vec{x}_i^t)$. It is necessary to note that when computing the fitness value of each individual $\vec{u}_i^t$ in OP^t, the objective function value and the degree of constraint violation of $\vec{u}_i^t$ should also be normalized based on (8) and (9). Moreover, $\vec{x}_i^t$ and $\vec{u}_i^t$ have the same weight ω_i^t.

At generation t, $\vec{x}_i^t$ and $\vec{u}_i^t$ have their own ω_i^t, so we have NP weights: $\{\omega_1^t, \dots, \omega_{NP}^t\}$. These NP weights are uniformly distributed in an interval defined by $[lb(t), ub(t)]$, and $lb(t)$ and $ub(t)$ are controlled by (11) and (12), respectively.

$$lb(t) = \frac{1}{1 + e^{\alpha(t/MaxGen - CI)}} \tag{11}$$

$$ub(t) = \frac{1}{1 + e^{\alpha(t/MaxGen - CI - DV)}} \tag{12}$$

where α is a paramter to control the shapes of $lb(t)$ and $ub(t)$, and is set to 25. The reasons for selecting the sigmoid function in (11) and (12) are threefold. First, the sigmoid function is widely utilized in the community of evolutionary computation [38]. Additionally, the decreasing trend of a sigmoid function can be controlled by a value between 0 and 1. In this manner, CI and DV can be incorporated into it naturally. More importantly, when lb and ub are set according to (11) and (12), we can guarantee that ub is bigger than lb. For the sake of illustration, let $MaxGen = 5000$, CI=0.4, and DV=0.2, the variation of $[lb(t), ub(t)]$ with respect to the generation number is presented in Fig. 2.

From (10)-(12), we can give the following comments:

- **Feature 1:** $lb(t)$ and $ub(t)$ are sigmoid curves and decrease as t increases. Under this condition, the information of objective function can be utilized to some extent, which is beneficial to tackle COPs with disjoint feasible regions or equality constraints [39].
- **Feature 2:** CI controls the generation at which the weights begin to decrease. Fig. 3 gives an example, in which $MaxGen = 5000$ and $DV = 0.2$. As shown in Fig. 3, the bigger the value of CI, the later the decrease of weights.
- **Feature 3:** DV controls the decrease speed of $ub(t)$, and thus controls the decrease speed of the range of $[lb(t), ub(t)]$. Fig. 4 gives an example, where $MaxGen = 5000$ and $CI = 0.4$. The four black line segments in Fig. 4 depict the ranges defined by $[lb(t), ub(t)]$ when $t = 1500$, 2000, 2500, and 3000, respectively. As can be seen from Fig. 4, the bigger the value of DV, the slower the decrease of the range of $[lb(t), ub(t)]$.

The rationale of CI and DV is discussed in the following.

- *The rationale of CI*: The bigger the value of CI, the stronger the correlation between constraints and objective function. Under this condition, more information of objective function should be utilized especially for some COPs with extremely complex constraints as pointed out in Section II-A. Therefore, the weight coefficient of objective function should decrease later, which is consistent with the conclusion in **Feature 2** and Fig. 3.
- *The rationale of DV*: The bigger the value of DV, the more the diversity of P^t has been lost at the learning stage. Hence, to increase the diversity, the weight of each individual should be distributed in a wider range. As a result, the decrease of the weight range $[lb(t), ub(t)]$ should be slower, which is consistent with the conclusion in **Feature 3** and Fig. 4. One may be interested in why a wider range of weights is beneficial to the diversity of population. The reason should be attributed into the search algorithm in this paper, where the offspring are generated by DE [40]. One of the DE operators in this paper is DE/rand-to-best/1/bin, where the best individual is selected based on the weighted sum updating approach. A wider distribution of weights implies that DE/rand-to-best/1/bin will search in a bigger area, so the diversity can be maintained.

Based on the above discussion, CI and DV have the capability to adapt the weights. **Algorithm 2** describes the weighted sum updating approach.

Algorithm 2: Weighted Sum Updating Approach

```
1 Calculate $F(\vec{x}_i^t)$ and $F(\vec{u}_i^t)$ via (7)-(12);
2 if $F(\vec{u}_i^t) < F(\vec{x}_i^t)$ then
3     $\vec{x}_i^{t+1} = \vec{u}_i^t$;
4 else
5     $\vec{x}_i^{t+1} = \vec{x}_i^t$;
```

Algorithm 3: Archiving and Replacement Mechanism

```
1 if $rand < (1 - CI)$ then
2     $r = \underset{i=1,\ldots,NP}{argmax}\ (G(\vec{x}_i^t))$;
3     $\vec{x}_r^t = \vec{a}_r^t$;
4     Calculate $feasRate$ of $P^t$;
5     if $feasRate == 0\ \&\&\ rand < \mu$ then
6         $P^t = A^t$;
```

2) Archiving and Replacement Mechanism: At the learning stage, P^t is updated based on objective function value. In addition, at the early evolving stage, the values of weights are relatively larger according to the characteristics of (10)-(12), and thus the updating of P^t depends largely on objective function. Therefore, a great deal of objective function information is exploited in P^t during the learning stage and early evolving stage. It is necessary to note that if the correlation between constraints and objective function is weak, too much information of objective function will have a negative effect on the balance between constraints and objective function in P^t. Moreover, under this condition, too much information of objective function may result in the premature convergence of P^t in the infeasible region. To remedy this issue, an archiving and replacement mechanism is proposed.

The main idea is to use some promising individuals in A^t, the updating of which is based mainly on constraints, to replace some low-quality individuals in P^t. As shown in **Algorithm 3**, the replacement contains two parts and is controlled by two factors, i.e., the percentage of feasible solutions in P^t ($feasRate$) and CI.

- Part I: The individual with the maximum degree of constraint violation in P^t is replaced with the individual with the same index in A^t according to the probability $(1 - CI)$.
- Part II: Under the condition of part I, if the feasible rate of P^t is 0, then P^t is replaced with A^t according to a small probability, that is, μ.

The reason for the above replacement is not difficult to understand. First of all, a smaller CI reflects the weaker correlation between constraints and objective function. As pointed out in Section II-A, under this condition, more information of constraints should be used to guide the population toward the feasible region. Hence, if $rand < (1 - CI)$ where $rand$ is a uniformly distributed random number between 0 and 1, part I is implemented to introduce the information of constraints by replacing the worst individual in terms of the degree of constraint violation in P^t. Moreover, the smaller the value of CI, the higher the probability of replacement. In addition, if the feasible rate of P^t is equal to zero, which means that more information of constraints is required, then all the individuals in P^t will be replaced with those in A^t with the probability μ, which is set to 0.01 in this paper.

In fact, in the weighted sum updating approach, more information of objective function will be utilized at the early stage. Subsequently, through the archiving and replacement mechanism, the information of constraints can be utilized to some extent. Hence, combining the weighted sum updating approach with the archiving and replacement mechanism can be considered as an adaptive relaxation method which is controlled by CI. In this manner, the balance between constraints and objective function can be achieved.

E. Search Algorithm

In addition to constraint-handling technique, search algorithm is another vital element of a COEA. In general, a good search algorithm is expected to achieve both the tradeoff between diversity and convergence and the tradeoff between constraints and objective function. In CORCO, two DE operators, which have been employed to solve COPs successfully [39], [41], are combined to satisfy these two tradeoffs. Their formulations are given as follows. For more details of DE, please refer to two survey papers [42], [43].

- DE/current-to-rand/1

$$\vec{u}_i^t = \vec{x}_i^t + rand \cdot (\vec{x}_{r_1}^t - \vec{x}_i^t) + F \cdot (\vec{x}_{r_2}^t - \vec{x}_{r_3}^t) \quad (13)$$

- DE/rand-to-best/1/bin

$$\vec{v}_i^t = \vec{x}_{r_1}^t + rand \cdot (\vec{x}_{best}^t - \vec{x}_{r_1}^t) + F \cdot (\vec{x}_{r_2}^t - \vec{x}_{r_3}^t) \quad (14)$$

$$u_{i,j} = \begin{cases} v_{i,j}, \text{if } rand_j < CR \text{ or } j = j_{rand} \\ x_{i,j}, \text{otherwise} \end{cases}, j = 1, \ldots, D \quad (15)$$

where $i = 1, \ldots, NP$, $\vec{v}_i^t$ and $\vec{u}_i^t$ are the ith mutant vector and the ith trial vector, respectively, $x_{i,j}^t$, $v_{i,j}^t$ and $u_{i,j}^t$ are the jth dimension of $\vec{x}_i^t$, $\vec{v}_i^t$, and $\vec{u}_i^t$, respectively, $\vec{x}_{r_1}^t$, $\vec{x}_{r_2}^t$, and $\vec{x}_{r_3}^t$ are mutually different individuals randomly selected from P^t, $\vec{x}_{best}^t$ is the best individual in P^t for $\vec{x}_i^t$, F is the scaling factor, CR is the crossover control parameter, $rand_j$ is a uniformly distributed random number between 0 and 1, and j_{rand} is a random integer selected from $\{1, \ldots, D\}$.

In (13), $\vec{x}_i$ learns the information of a randomly selected individual $\vec{x}_{r_1}$, thus enhancing the diversity. In contrast, in (14), the information of the best individual is employed, hence promoting the convergence. Although many adaptive methods [44] based on success history have been presented, in this paper these two operators are executed in a random manner. Specifically, each of them is selected with the same probability, i.e., $ps = 0.5$. The reasons are the following. First, this manner is simple and its effectiveness has been validated in [39]. Additionally, designing an adaptive strategy is nontrivial.

In order to achieve the tradeoff between constraints and objective function, the best individual in (14) is chosen based on the weighted sum updating approach. To be specific, for the ith individual in the population, the best individual is the individual with the smallest fitness value in (7) with ω_i^t. It

Algorithm 4: Search Algorithm

```
1  $OP^t = \emptyset$;
2  if $t \le LearnGen$ then
3      $best = \underset{j=1,\ldots,NP}{argmin}\ f(\vec{x}_j^t)$;
4  for $i = 1 : NP$ do
5      Randomly select a $F$ value from the pool $\{0.6, 0.8, 1.0\}$;
6      Randomly select a $CR$ value from the pool $\{0.1, 0.2, 1.0\}$;
7      if $rand < ps$ then
8          Select $\vec{x}_{r_1}^t$, $\vec{x}_{r_2}^t$, and $\vec{x}_{r_3}^t$ from $P^t$;
9          Generate $\vec{u}_i^t$ according to (13);
10     else
11         if $t > LearnGen$ then
12             $best = \underset{j=1,\ldots,NP}{argmin}\ F_{\omega_i^t}(\vec{x}_j^t)$;
13         Select $\vec{x}_{best}^t$, $\vec{x}_{r_1}^t$, $\vec{x}_{r_2}^t$, and $\vec{x}_{r_3}^t$ from $P^t$;
14         Generate $\vec{u}_i^t$ according to (14) and (15);
15     $OP^t = OP^t \cup \vec{u}_i^t$;
```

should be noted that, at the learning stage, the best solution is selected according to objective function value to mine CI which is critical to calculate the weights.

Overall, the proposed search algorithm, described in **Algorithm 4**, can balance not only diversity and convergence but also constraints and objective function. It is necessary to point out that the search algorithm in this paper is similar to that of FROFI [39]. The main difference between them is the selection of the best individual. In FROFI, the best individual is selected according to objective function value. However, in CORCO, the best individual is selected based on the weighted sum updating approach.

F. Computational Time Complexity

At the learning stage, the computational time complexity of the search algorithm is $O(NP)$, and the computational time complexity of updating the archive and the population is $O(NP)$. The calculation of $DI^{LearnGen}$, DV, and CI has a computational time complexity of $O(NP)$. In addition, the computational time complexity of the evolving stage is $O(NP^2)$, which is dominated by the search algorithm. In summary, the overall computational time complexity of CORCO is $O(NP^2)$.

III. Experimental Study

A. Proof-of-Principle Results

1) Significance of the Correlation Index: First of all, two test functions (C09 with 10D from IEEE CEC2010 and g11 from IEEE CEC2006) were used to investigate the effectiveness of the correlation between constraints and objective function proposed in this paper.

C09 minimize : $f(\vec{x}) = \sum_{i=1}^{D-1}(100(z_i^2 - z_{i+1})^2 + (z_i - 1)^2)$

subject to : $h(\vec{x}) = \sum_{i=1}^{D}(y_i \sin(\sqrt{|y_i|})) = 0$

where $\vec{z} = \vec{x} + 1 - \vec{o}$, $\vec{y} = \vec{x} - \vec{o}$, $\vec{x} \in [-500, 500]^D$, and $\vec{o}$ is a shifted vector.

g11 minimize : $f(\vec{x}) = x_1^2 + (x_2 - 1)^2$

subject to : $h(\vec{x}) = x_2 - x_1^2 = 0$

where $\vec{x} \in [-1, 1]^2$.

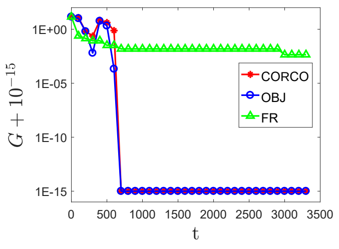


Fig. 5. Evolution of the minimum degree of constraint violation on C09 with 10D from IEEE CEC2010.

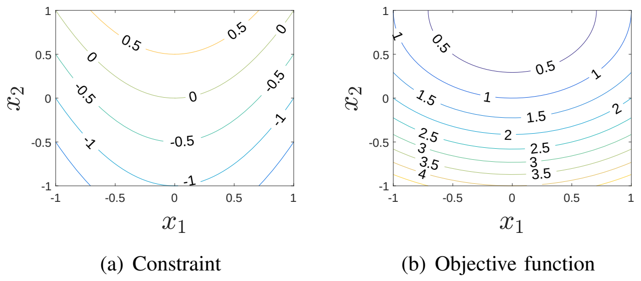


Fig. 6. Contours of g11 from IEEE CEC2006.

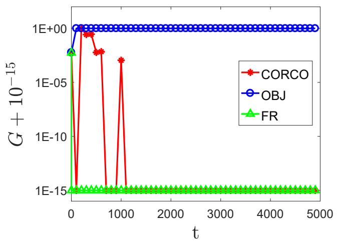


Fig. 7. Evolution of the minimum degree of constraint violation on g11 from IEEE CEC2006.

C09 includes a complex nonlinear equality constraint; therefore, it is difficult to find feasible solutions. We run CORCO on this test function. After the learning stage of CORCO, we found that $CI = 1$, which indicates that the correlation between constraints and objective function is very strong. According to the previous analysis in Section II-A, under this condition, if the comparison among individuals is based only on constraints, an algorithm might not find any feasible solutions. On the contrary, the information of objective function may be beneficial to find feasible solutions. To verify this, we designed two additional algorithms, called FR and OBJ. In FR, we combined the search algorithm in Section II-E with the feasibility rule [7]. In addition, OBJ also involves the search algorithm in Section II-E, while the comparison among individuals is based on objective function value. Note that in FR and OBJ, the best individual is chosen based on the feasibility rule and objective function value, respectively. The evolution of the minimum degree of constraint violation of FR, OBJ, and CORCO is described in Fig. 5. For the sake of visualization, we added a very small real number (i.e., 10^{-15}) into the minimum degree of constraint violation. As shown in Fig. 5, FR fails to find any feasible solution in the end.

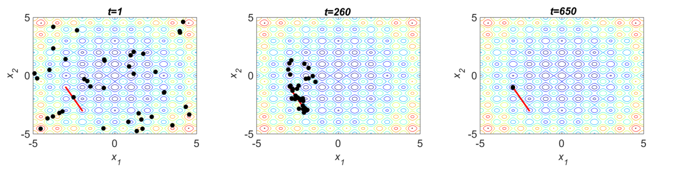


Fig. 8. Evolution of CORCO over a typical run on ATF1.

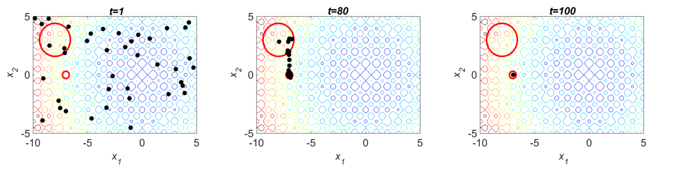


Fig. 9. Evolution of CORCO over a typical run on ATF2.

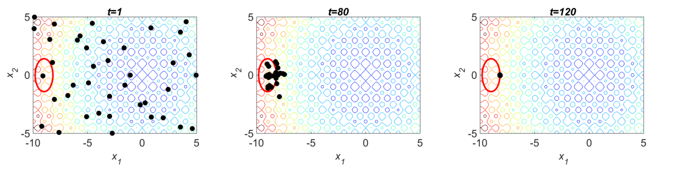


Fig. 10. Evolution of CORCO over a typical run on ATF3.

It is interesting to see that after about 700 generations, the minimum degree of constraint violation in OBJ is equal to zero, which confirms our analysis in Section II-A. Due to the fact that CORCO also exploits the information of objective function, similar to OBJ, CORCO is able to attain feasible solutions at the early stage of evolution. Moreover, the best solution achieved by CORCO is $\vec{o}$, which is a Karush-Kuhn-Tucker (KKT) point.

g11 has a relatively simple constraint. As a result, it is easy to find feasible solutions for g11. Similarly, we run CORCO on this test function and found that $CI = 0$, which signifies that the correlation between constraints and objective function is weak. In addition, Fig. 6 gives the contours of constraint and objective function of g11. As shown in Fig. 6, with the decrease of x_2, the degree of constraint violation decreases while the objective function value increases, which again means a weak correlation between constraint and objective function of g11. According to the previous analysis in Section II-A, under this condition, too much information of objective function may prevent the population from entering the feasible region. However, the information of constraints is essential for the population to enter the feasible region. Similarly, the evolution of the minimum degree of constraint violation of FR, OBJ, and CORCO on g11 is described in Fig. 7. From Fig. 7, OBJ cannot find any feasible solution. In contrast, feasible solutions can be easily found by FR, which is consistent with our analysis. Besides, because the weight coefficient in the weighted sum updating approach is large for the degree of constraint violation, CORCO is capable of finding feasible solutions after about 1200 generations. The experimental results are in line with the above analysis.

TABLE I
MAXIMUM NUMBER OF FUNCTION EVALUATIONS $MaxFEs$ AND POPULATION SIZE NP

Test Functions	$MaxFEs$	NP
24 test functions from IEEE CEC2006	5.0E+05	100
18 test functions with 10D from IEEE CEC2010	2.0E+05	60
18 test functions with 30D from IEEE CEC2010	6.0E+05	100
28 test functions with 50D from IEEE CEC2017	1.0E+06	100
28 test functions with 100D from IEEE CEC2017	2.0E+06	100

Moreover, the best solution provided by CORCO is (0.707, 0.500), which is a KKT point indeed.

The above experimental results reveal that the information derived from constraints and objective function has different effects on the balance of constraints and objective function when solving different COPs. Meanwhile, CI calculated by the learning stage of CORCO can reflect the correlation between constraints and objective function of a COP; thus, it can help to achieve the balance between constraints and objective function.

2) Capability to Solve Challenging COPs: As we know, COPs with disjoint feasible regions, equality constraints, or the optimum located on the boundary between feasible and infeasible regions are challenging. To ascertain the performance of CORCO on these three kinds of COPs, three artificial test functions (ATFs) designed in [39] were adopted. To be specific, ATF1 involves an equality constraint, ATF2 has two disjoint feasible regions, and the optimum of ATF3 is located on the boundary between feasible and infeasible regions. When these three test functions are solved, the population size NP and the maximum number of function evaluations $MaxFEs$ were set to 40 and 40000, respectively. Additionally, other parameter settings were kept unchanged which will be specified in Section III-B.

As shown in Fig. 8, CORCO has the capability to approach the feasible region from both sides. As shown in Fig. 9, CORCO can locate two disjoint feasible regions and find the optimum finally. As shown in Fig. 10, CORCO is able to probe the area around the boundary between feasible and infeasible regions. Hence, it can locate the optimum on the boundary. Overall, CORCO is capable of solving the above-mentioned three kinds of challenging COPs.

B. Benchmark Test Functions and Parameter Settings

In order to systematically validate the effectiveness of CORCO, three sets of benchmark test functions were employed to compare CORCO with other state-of-the-art COEAs. These test functions include 24 test functions from IEEE CEC2006 [45], 18 test functions with 10D and 30D from IEEE CEC2010 [46], and 28 test functions with 50D and 100D from IEEE CEC2017 [47]. They cover various tough properties such as strong nonlinearity, multi-modality, extremely small feasible region, and rotated landscape.

In the experimental study, the population size NP and the maximum number of function evaluations $MaxFEs$, which includes the function evaluations consumed at the learning and evolving stages, were given in Table I. 25 independent runs were performed on each test function and the tolerance value

TABLE II
EXPERIMENTAL RESULTS OF CORCO AND FOUR OTHER SELECTED METHODS OVER 25 INDEPENDENT RUNS ON THE 22 TEST FUNCTIONS FROM IEEE CEC2006

IEEE CEC2006	NSES Mean OFV±Std Dev	AIS-IRP Mean OFV±Std Dev	FROFI Mean OFV±Std Dev	APF-GA Mean OFV±Std Dev	CORCO Mean OFV±Std Dev
g01	-1.5000E+01±4.20E-30*	-1.5000E+01±0.00E+00*	-1.5000E+01±0.00E+00*	-1.5000E+01±0.00E+00*	-1.5000E+01±0.00E+00*
g02	-8.0362E-01±2.41E-32*	-8.0219E-01±5.19E-10	-8.0362E-01±1.78E-07*	-8.0352E-01±1.00E-04	-8.0362E-01±7.01E-09*
g03	-1.0005E+00±5.44E-19*	-1.0005E+00±1.77E-11*	-1.0005E+00±4.49E-16*	-1.0004E+00±0.00E+00	-1.0005E+00±5.32E-15*
g04	-3.0666E+04±2.22E-24*	-3.066553E+04±3.69E-13*	-3.066553E+04±3.71E-12*	-3.066553E+04±1.00E-04*	-3.066553E+04±3.71E-12*
g05	5.1265E+03±0.00E+00*	5.1264981E+03±1.70E-02	5.1264967E+03±2.78E-12*	5.12754E+03±1.43E+00	5.1265E+03±3.00E-15*
g06	-6.9618E+03±0.00E+00*	-6.961813E+03±1.90E-12*	-6.961813E+03±0.00E+00*	-6.961813E+03±0.00E+00*	-6.961813E+03±0.00E+00*
g07	2.4306E+01±7.37E-09*	2.435572E+01±8.20E-03	2.430621E+01±6.32E-15*	2.430621E+01±0.00E+00*	2.4306E+01±8.00E-15*
g08	-9.5825E+02±2.01E-34*	-9.5825E+02±0.00E+00*	-9.5825E+02±1.42E-17*	-9.5825E+02±0.00E+00*	-9.5825E+02±1.42E-17*
g09	6.8063E+02±1.100E-25*	6.80650308E+02±1.20E-08	6.8063006E+02±3.64E-13*	6.8063006E+02±0.00E+00*	6.8063006E+02±3.00E-13*
g10	7.0492480E+03±2.07E-24*	7.049570318E+03±4.50E-04	7.0492480E+03±3.26E-12*	7.077682E+03±5.12E+01	7.0492480E+03±2.21E-08*
g11	7.499E-01±0.00E+00*	7.499E-01±1.40E-08*	7.499E-01±1.13E-16*	7.499E-01±0.00E+00*	7.499E-01±0.00E+00*
g12	-1.00E+00±0.00E+00*	-1.00E+00±0.00E+00*	-1.00E+00±0.00E+00*	-1.00E+00±0.00E+00*	-1.00E+00±0.00E+00*
g13	5.3942E-02±1.98E-34*	5.3942E-02±7.80E-10*	5.3942E-02±2.41E-17*	5.4042E-02±0.00E+00	5.3942E-02±3.28E-17*
g14	-4.776489E+01±0.00E+00*	-4.776489E+01±1.00E-12*	-4.776489E+01±2.34E-14*	-4.776489E+01±1.00E-04*	-4.776489E+01±2.30E-14*
g15	9.617150E+02±0.00E+00*	9.617150E+02±0.00E+00*	9.617150E+02±5.80E-13*	9.617150E+02±0.00E+00*	9.617150E+02±0.00E+00*
g16	-1.90516E+00±2.62E-30*	-1.90516E+00±0.00E+00*	-1.90516E+00±4.53E-16*	-1.90510E+00±0.00E+00*	-1.90516E+00±5.80E-13*
g17	8.853533E+03±2.51E-23*	8.853533E+03±1.90E-09*	8.853533E+03±0.00E+00*	8.888481E+03±2.90E+01	8.853533E+03±1.24E-12*
g18	-8.66025E-01±4.62E-33*	-8.66025E-01±1.30E-15*	-8.66025E-01±6.94E-16*	-8.65925E-01±0.00E+00	-8.66025E-01±2.00E-15 *
g19	3.265559E+01±1.52E-05*	3.265559E+01±0.00E+00*	3.265559E+01±2.18E-14*	3.265559E+01±0.00E+00*	3.265559E+01±2.10E-14*
g21	1.937245E+02±1.62E-22*	1.9672451E+02±1.10E+00	1.937245E+02±2.95E-11*	1.99516E+02±2.35E+00	1.937245E+02±3.77E-08*
g23	-4.000551E+02±9.08E-26*	-3.9987432E+02±2.00E+00	-4.000551E+02±1.71E-13*	-3.947627E+02±3.87E+00	-4.000551E+02±2.28E-11*
g24	-5.50801E+00±0.00E+00*	-5.50801E+00±0.00E+00*	-5.50801E+00±9.06E-16*	-5.50801E+00±0.00E+00*	-5.50801E+00±0.00E+00*
*	**22**	**15**	**22**	**13**	**22**

TABLE III
EXPERIMENTAL RESULTS OF CORCO AND FIVE OTHER SELECTED METHODS OVER 25 INDEPENDENT RUNS ON THE 18 TEST FUNCTIONS WITH 10D FROM IEEE CEC2010

IEEE CEC2010 with 10D	AIS-IRP Mean OFV±Std Dev	εDEag Mean OFV±Std Dev	DE-AOPS Mean OFV±Std Dev	FROFI Mean OFV±Std Dev	ECHT-DE Mean OFV±Std Dev	CORCO Mean OFV±Std Dev
C01	-7.47E-01±1.30E-03≈	-7.47E-01±1.32E-03≈	-7.47E-01±2.82E-16≈	-7.47E-01±1.35E-03≈	-7.47E-01±1.40E-03≈	-7.47E-01±2.87E-03
C02	-2.27E+00±2.00E-03+	-2.26E+00±2.39E-02+	-1.87E+00±4.90E-01−	-2.02E+00±1.41E-01−	-2.27E+00±6.70E-03+	-2.21E+00±6.88E-02
C03	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C04	-9.97E-06±4.28E-03−	-9.92E-06±1.55E-07−	-1.00E-05±0.00E+00≈	-1.00E-05±0.00E+00≈	-1.00E-05±0.00E+00≈	-1.00E-05±0.00E+00
C05	-4.80E+02±6.30E+00−	-4.84E+02±3.89E-13≈	-4.84E+02±3.48E-13≈	-4.84E+02±8.09E-07≈	-4.11E+02±7.63E+01−	-4.84E+02±3.00E-13
C06	-5.80E+02±7.30E-08+	-5.79E+02±3.63E-03≈	-5.79E+02±1.41E-13≈	-5.79E+02±5.04E-04≈	-5.62E+02±4.51E+01−	-5.79E+02±4.82E-10
C07	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	1.33E-01±7.28E-01−	0.00E+00±0.00E+00
C08	4.09E+00±1.46E+00+	6.73E+00±5.56E+00−	6.59E+00±5.12E+00−	7.11E+00±4.79E+00−	6.16E+00±6.45E+00−	5.27E+00±5.26E+00
C09	2.70E+01±7.50E+01−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	2.50E+01±3.92E+01−	1.47E-01±8.05E-01−	0.00E+00±0.00E+00
C10	1.62E+03±5.00E+02−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	4.17E+01±8.69E-06−	1.71E+00±7.66E+00−	0.00E+00±0.00E+00
C11	-9.20E-04±8.23E-04−	-1.52E-03±6.34E-11≈	-1.52E-03±7.80E-18≈	-1.52E-03±5.63E-14≈	-4.40E-03±1.57E-02∇−	-1.52E-03±4.32E-14
C12	-4.36E+02±6.02E+01+	-3.37E+02±1.78E+02+	-6.73E+01±1.14E+02−	-3.84E+02±2.17E+02+	-1.72E+02±2.21E+02∇−	-1.14E+02±2.54E+02
C13	-6.79E+01±3.11E-01−	-6.84E+01±1.03E-06≈	-6.84E+01±2.63E-14≈	-6.84E+01±2.52E-09≈	-6.51E+01±2.38E+00−	-6.84E+01±2.90E-14
C14	1.22E-04±2.90E-08−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	7.02E+05±3.19E+06−	0.00E+00±0.00E+00
C15	0.00E+00±0.00E+00≈	1.80E-01±8.81E-01−	0.00E+00±0.00E+00≈	3.09E+00±1.37E+00−	2.34E+13±5.30E+13−	0.00E+00±0.00E+00
C16	0.00E+00±0.00E+00≈	3.70E-01±3.71E-01−	0.00E+00±0.00E+00≈	1.19E-02±2.07E-02−	3.93E-02±4.28E-02−	0.00E+00±0.00E+00
C17	2.93E+00±2.29E+00−	1.25E-01±1.94E-01−	0.00E+00±0.00E+00≈	7.83E-02±2.25E-01−	1.12E-01±3.32E-01−	0.00E+00±0.00E+00
C18	1.66E+00±1.27E+00−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
−	**9**	**5**	**3**	**7**	**13**	/
+	**4**	**2**	**0**	**1**	**1**	/
≈	**5**	**11**	**15**	**10**	**4**	/

TABLE IV
STATISTICAL TEST RESULTS OF CORCO AND FIVE OTHER SELECTED METHODS BY THE MULTIPLE-PROBLEM WILCOXON'S SIGNED RANK TEST ON THE 18 TEST FUNCTIONS WITH 10D FROM IEEE CEC2010

CORCO VS	R^+	R^-	p-value	α=0.1	α=0.05
AIS-IRP	120.0	33.0	3.95E-02	Yes	Yes
εDEag	106.5	64.5	≥ 0.2	No	No
DE-APOS	118.5	52.5	1.61E-01	No	No
FROFI	112.0	41.0	9.83E-02	Yes	No
ECHT-DE	151.0	20.0	2.80E-03	Yes	Yes

TABLE V
RANKING OF CORCO AND FIVE OTHER SELECTED METHODS BY THE FRIEDMAN'S TEST ON THE 18 TEST FUNCTIONS WITH 10D FROM IEEE CEC2010

Algorithm	Ranking
CORCO	**2.7778**
DE-APOS	3.1111
εDEag	3.3611
FROFI	3.5
AIS-IRP	3.8333
ECHT-DE	4.4167

δ in (1) was set to 0.0001. Note that the settings of $MaxFEs$, number of runs, and δ were in accordance with the suggestions in [45], [46], and [47], and they were the same for all compared algorithms. In addition, $LearnGen$ in CORCO was set to $\lfloor MaxGen/20 \rfloor$, where $MaxGen = \lfloor MaxFEs/NP \rfloor$.

C. Experiments on the 24 Benchmark Test Functions from IEEE CEC2006

Firstly, the performance of CORCO was compared with that of four excellent COEAs on the 24 test functions from IEEE

TABLE VI
EXPERIMENTAL RESULTS OF CORCO AND FIVE OTHER SELECTED METHODS OVER 25 INDEPENDENT RUNS ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

IEEE CEC2010 with 30D	AIS-IRP Mean OFV±Std Dev	εDEag Mean OFV±Std Dev	DE-AOPS Mean OFV±Std Dev	FROFI Mean OFV±Std Dev	ECHT-DE Mean OFV±Std Dev	CORCO Mean OFV±Std Dev
C01	-8.20E-01±3.25E-04−	-8.21E-01±7.10E-04≈	-8.22E-01±8.64E-04≈	-8.21E-01±2.36E-03≈	-8.00E-01±1.79E-02−	-8.21E-01±2.98E-03
C02	-2.21E+00±2.84E-03+	-2.15E+00±1.20E-02≈	-1.94E+00±2.70E-01−	-2.00E+00±4.35E-02−	-1.99E+00±2.10E-01−	-2.13E+00±1.22E-01
C03	6.68E+01±4.26E+02−	2.88E+01±8.05E-01−	0.00E+00±0.00E+00≈	2.87E+01±6.24E-08−	9.89E+01±6.26E+01−	0.00E+00±0.00E+00
C04	1.98E-03±1.61E-03−	8.16E-03±3.07E-03−	-3.33E-06±7.20E-14≈	-3.33E-06±4.13E-10≈	-1.03E-06±9.01E-03−	-3.33E-06±4.00E-03
C05	-4.36E+02±2.51E+01−	-4.50E+02±2.90E+00−	-4.84E+02±7.07E-12≈	-4.81E+02±2.84E+00−	-1.06E+02±1.67E+02−	-4.84E+02±5.56E-04
C06	-4.54E+02±4.79E+01−	-5.28E+02±4.75E-01−	-5.31E+02±3.23E-11≈	-5.29E+02±5.71E-01≈	-1.38E+02±9.89E+01−	-5.30E+02±8.12E-03
C07	1.07E+00±1.61E+00−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	1.33E-01±7.28E-01−	0.00E+00±0.00E+00
C08	1.65E+00±6.41E-01−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	3.36E+01±1.11E+02−	0.00E+00±0.00E+00
C09	1.57E+00±1.96E+00−	1.07E+01±2.82E+01−	0.00E+00±0.00E+00≈	4.30E+01±3.27E+01−	4.24E+01±1.38E+02−	0.00E+00±0.00E+00
C10	1.78E+01±1.88E+01−	3.33E+01±4.55E-01−	0.00E+00±0.00E+00≈	3.13E+01±8.22E-02−	5.34E+01±8.83E+01−	0.00E+00±0.00E+00
C11	-1.58E-04±4.67E-05−	-2.86E-04±2.71E-05−	-3.92E-04±1.20E-10≈	-3.92E-04±2.64E-06≈	2.60E-03±6.00E-03∇−	-3.92E-04±3.36E-07
C12	4.29E-06±4.52E-04−	3.56E+02±2.89E+02∇−	-1.99E-01±1.01E-08≈	-1.99E-01±1.42E-06≈	-2.51E+01±1.37E+02∇−	-1.99E-01±2.19E-08
C13	-6.62E+01±2.27E-01−	-6.54E+01±5.73E-01−	-6.46E+01±3.03E+00−	-6.83E+01±1.95E-01≈	-6.46E+01±1.67E+00−	-6.80E+01±6.63E-01
C14	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	1.24E+05±6.77E+05−	0.00E+00±0.00E+00
C15	3.41E+01±3.82E+01−	2.16E+01±1.10E-04−	0.00E+00±0.00E+00≈	2.16E+01±8.03E-05−	1.94E+11±4.35E+11−	0.00E+00±0.00E+00
C16	8.21E-02±1.12E-01−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C17	3.61E+00±2.54E+00−	6.33E+00±4.99E+00−	0.00E+00±0.00E+00≈	1.59E-01±3.82E-01−	2.75E-01±3.78E-01−	0.00E+00±0.00E+00
C18	4.02E+01±1.80E+01−	8.75E+01±1.66E+02−	0.00E+00±0.00E+00≈	4.87E-01±1.25E+00−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
−	**16**	**12**	**2**	**8**	**16**	/
+	**1**	**0**	**0**	**0**	**0**	/
≈	**1**	**6**	**16**	**10**	**2**	/

TABLE VII
STATISTICAL TEST RESULTS OF CORCO AND FIVE OTHER SELECTED METHODS BY THE MULTIPLE-PROBLEM WILCOXON'S SIGNED RANK TEST ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

CORCO VS	R^+	R^-	p-value	α=0.1	α=0.05
AIS-IRP	152.0	1.0	3.05E-05	Yes	Yes
εDEag	160.5	10.5	3.74E-04	Yes	Yes
DE-AOPS	100.5	52.5	≥ 0.2	No	No
FROFI	143.5	27.5	9.69E-03	Yes	Yes
ECHT-DE	169.5	1.5	1.91E-05	Yes	Yes

TABLE VIII
RANKING OF CORCO AND FIVE OTHER SELECTED METHODS BY THE FRIEDMAN'S TEST ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

Algorithm	Ranking
CORCO	**2**
DE-AOPS	2.4722
FROFI	2.9167
εDEag	4.1111
AIS-IRP	4.4444
ECHT-DE	5.0556

CEC2006:

- Method based on multiobjective optimization: NSES [12].
- Methods based on treating constraints and objective function separately: AIS-IRP [48] and FROFI [39].
- Method based on penalty function: APF-GA [5].

Table II summarizes the mean objective function value and standard deviation (denoted as "Mean OFV" and "Std Dev") derived from the five compared methods over 25 independent runs. Since it is very difficult to find a feasible solution for g20 and g22, Table II does not include the experimental results for them. The true optima of these 24 test functions are known *a priori* [45]. Hence, it is possible to judge whether a run is successful or not. A run is successful if and only if $f(\vec{x}') - f(\vec{x}^*) < 10^{-4}$, where $f(\vec{x}^*)$ is the true optimum of a test function and $f(\vec{x}')$ is the best feasible solution provided by a COEA. In Table II, "*" denotes that a method can yield 25 successful runs on the corresponding test function.

As shown in Table II, NSES, FROFI, and CORCO can consistently solve 22 test functions. However, AIS-IRP and APF-GA fail to achieve 100% successful runs on seven and nine test functions, respectively. The experimental results suggest that CORCO outperforms or performs similarly to the four competitors on the 24 test functions from IEEE CEC2006.

D. Experiments on the 18 Benchmark Test Functions from IEEE CEC2010

Subsequently, we tested the performance of CORCO on the 18 more complicated test functions with 10D and 30D from IEEE CEC2010. Five well-established COEAs were selected as the competitors:

- Methods based on treating constraints and objective function separately: AIS-IRP [48], εDEag [22], DE-AOPS [49], and FROFI [39].
- Hybrid method: ECHT-DE [50].

Different from the 24 test functions from IEEE CEC2006, the true optima of the 18 test functions with 10D and 30D from IEEE CEC2010 are unknown [46]. Hence, we cannot compute the number of successful runs. Based on the mean objective function value and standard deviation (denoted as "Mean OFV" and "Std Dev"), we implemented the statistical test between CORCO and each of the five competitors. Due to the fact that the experimental results of FROFI over 25 runs can be obtained from our previous study, the Wilcoxon's rank sum test at a 0.05 significance level was adopted to compare CORCO with FROFI. However, we can just obtain "Mean OFV" and "Std Dev" of AIS-IRP, εDEag, DE-AOPS, and ECHT-DE from their original papers. Therefore, the t-test at a 0.05 significance level was employed. Furthermore, the multiple-problem Wilcoxon's signed rank test and the Friedman's test with the Bonferroni-Dunn method were carried out via KEEL

TABLE IX
EXPERIMENTAL RESULTS OF CORCO AND LSHADE44 OVER 25 INDEPENDENT RUNS ON THE 22 TEST FUNCTIONS WITH 50D AND 100D FROM IEEE CEC2017

IEEE CEC2017	50D		100D	
	LSHADE44 Mean OFV±Std Dev	CORCO Mean OFV±Std Dev	LSHADE44 Mean OFV±Std Dev	CORCO Mean OFV±Std Dev
C01	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C02	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C03	8.95E+05±7.40E+05−	0.00E+00±0.00E+00	2.73E+06±9.66E+05−	0.00E+00±0.00E+00
C04	1.36E+01±5.44E-15≈	1.36E+01±1.05E+00	1.37E+02±4.62E-01−	6.69E+01±2.07E+01
C05	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00	3.28E-05±9.25E-05+	2.29E+00±4.84E+00
C06	7.51E+03±1.42E+03−	3.38E+02±5.00E+01	1.55E+04±1.59E+03−	6.74E+02±8.31E+01
C07	-1.79E+02±8.97E+01+	-9.45E+01±2.42E+02	-3.02E+02±1.35E+02+	-4.85E+01±2.30E+02
C08	-1.30E-04±2.77E-20−	-1.34E-04±1.17E-06	-4.81E-05±1.33E-07+	1.60E-03±1.65E-04
C09	-2.04E-03±1.33E-18≈	-2.03E-03±5.24E-04	-1.43E-03±2.21E-19+	0.00E+00±0.00E+00
C10	-4.83E-05±0.00E+00+	-4.81E-05±0.00E+00	-1.72E-05±1.29E-08+	4.67E-04±5.75E-05
C11	-1.77E+00±3.33E-01+	∇	∇≈	∇
C12	4.98E+01±1.99E+01−	8.81E+00±6.33E-01	3.25E+01±8.19E-01−	9.74E+00±2.89E+00
C13	2.67E+01±1.36E+01−	7.97E-01±1.68E+00	8.07E+01±7.37E+00−	3.72E+01±9.92E+01
C14	1.40E+00±3.74E-02≈	1.40E+00±2.62E-01	9.72E-01±1.94E-02−	8.78E-01±1.38E-01
C15	1.78E+01±3.00E+00−	2.35E+00±1.82E-03	1.81E+01±1.28E+00−	1.46E+01±1.78E+00
C16	2.53E+02±1.62E+01−	0.00E+00±0.00E+00	5.35E+02±3.08E+01−	0.00E+00±0.00E+00
C20	3.20E+00±1.43E-01≈	3.20E+00±4.35E-01	9.36E+00±3.78E-01−	7.68E+00±5.86E-01
C21	6.29E+01±1.59E+00+	∇	3.16E+01±2.99E-03+	∇
C22	∇−	5.88E+00±3.02E+00	∇−	4.61E+02±5.73E+02
C23	1.33E+00±6.16E-02−	1.23E+00±2.54E-01	9.69E-01±4.26E-02−	8.76E-01±1.32E-01
C24	1.43E+01±1.28E+00−	2.34E+00±3.85E-03	1.71E+01±1.43E+00−	2.35E+00±4.07E-04
C25	2.48E+02±1.58E+01−	4.00E-06±1.26E-05	5.44E+02±2.86E+01−	2.40E+01±2.00E+01
−	**11**	/	**13**	/
+	**4**	/	**6**	/
≈	**7**	/	**3**	/

software [51], which can compare the performance of multiple methods concurrently.

In the case of $D = 10$, the experimental results are collected in Tables III, IV, and V. In Table III, "$-$", "$+$", and "$\approx$" represent that the performance of the corresponding algorithm is worse than, better than, and similar to that of CORCO, respectively, based on the Wilcoxon's rank sum test/t-test. In addition, "∇" represents that the corresponding algorithm cannot consistently find feasible solutions over 25 runs. As described in Table III, CORCO has an edge over AIS-IRP, εDEag, DE-AOPS, FROFI, and ECHT-DE on nine, five, three, seven, and 13 test functions, respectively. However, AIS-IRP, εDEag, DE-AOPS, FROFI, and ECHT-DE perform better than CORCO on four, two, zero, one, and one test function, respectively. According to the multiple-problem Wilcoxon's signed rank test, CORCO provides higher R^+ values than R^- values in all cases. In terms of the Friedman's test, CORCO ranks the first among the six compared methods, followed by DE-AOPS.

In the case of $D = 30$, the experimental results are summarized in Tables VI, VII, and VIII. From Table VI, CORCO is superior to AIS-IRP, εDEag, DE-APOS, FROFI, and ECHT-DE on 16, 12, two, eight, and 16 test functions, respectively. In contrast, εDEag, DE-AOPS, FROFI, and ECHT-DE cannot surpass CORCO on any test function, and AIS-IRP outperforms CORCO on one test function. As far as the multiple-problem Wilcoxon's signed rank test is concerned, CORCO provides higher R^+ values than R^- values in all cases, and performs significantly better than AIS-IRP, εDEag, FROFI, and ECHT-DE as the p-values are less than 0.05. Furthermore, CORCO achieves the first rank regarding the Friedman's test, followed by DE-AOPS.

The above comparison demonstrates that CORCO exhibits better performance than the five competitors on the 18 test functions with 10D and 30D from IEEE CEC2010. Moreover, it seems that the superiority of CORCO against the five competitors except DE-AOPS increases with the increase of the number of decision variables.

E. Experiments on the 28 Benchmark Test Functions from IEEE CEC2017

To further test the performance of CORCO on high-dimensional COPs, it was evaluated on the 28 test functions with 50D and 100D from IEEE CEC2017. The performance of CORCO was compared with that of LSHADE44 [52], which is the champion of the competition at IEEE CEC2017. The experimental results are summarized in Table IX. Note that the experimental results of those test functions, in which both CORCO and LSHADE44 cannot find feasible solutions consistently, were excluded from the comparison. When a method cannot achieve at least a feasible solution on a test function at the end of some runs, "∇" was placed in Table IX. In addition, the t-test at a 0.05 significance level was employed to compare CORCO and LSHADE44 on each test function due to the fact that we can only obtain the mean objective function values and standard deviations of LSHADE44 from its original literature.

As shown in Table IX, when $D = 50$, CORCO performs better than LSHADE44 on 11 test functions while LSHADE44 provides better results on four test functions. In the case of $D = 100$, CORCO outperforms LSHADE44 on 13 test functions while LSHADE44 surpasses CORCO on six test functions. Therefore, the above comparison verifies that, over-

TABLE X
EXPERIMENTAL RESULTS OF THE ORIGINAL CORCO, CORCO WITH FIVE FIXED CI VALUES, AND CORCO WITH PCC OVER 25 INDEPENDENT RUNS ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

IEEE CEC2010 with 30D	$CI = 0.0$ Mean OFV±Std Dev (feasible rate)	$CI = 0.25$ Mean OFV±Std Dev (feasible rate)	$CI = 0.5$ Mean OFV±Std Dev (feasible rate)	$CI = 0.75$ Mean OFV±Std Dev (feasible rate)	$CI = 1.0$ Mean OFV±Std Dev (feasible rate)	$CI = PCC$ Mean OFV±Std Dev (feasible rate)	CORCO Mean OFV±Std Dev (feasible rate)
C01	-8.19E-01±2.77E-03−	-8.19E-01±2.85E-03−	-8.20E-01±2.04E-03≈	-8.21E-01±1.83E-03≈	-1.59E-01±2.35E-02−	-8.19E-01±1.77E-03−	-8.21E-01±2.98E-03
C02	-1.79E+00±2.95E-01−	-1.75E+00±2.21E-01−	-2.02E+00±1.10E-01−	-1.90E+00±1.35E-01−	(0%)−	-1.83E+00±2.51E-01−	-2.13E+00±1.22E-01
C03	2.75E+01±5.73E+00−	2.67E+01±7.40E+00−	1.26E+01±1.45E+01−	0.00E+00±0.00E+00≈	(60%)−	0.00E+00 ±0.00E+00≈	0.00E+00±0.00E+00
C04	-3.33E-06±1.41E-11≈	-3.33E-06±6.81E-11≈	1.19E-05±2.31E-05−	2.66E-03±9.54E-04−	(0%)−	-3.33E-06 ±5.07E-11≈	-3.33E-06±4.00E-03
C05	-4.57E+02±4.69E+01−	-4.52E+02±3.70E+01−	-4.84E+02±1.17E-01≈	-4.83E+02±2.22E-01≈	(0%)−	-4.60E+02±5.13E+01−	-4.84E+02±5.56E-04
C06	-5.22E+02±8.05E+00−	-5.23E+02±5.33E+00−	-5.29E+02±1.26E+00≈	-5.29E+02±1.32E+00≈	(0%)−	-5.26E+02±2.16E+00−	-5.30E+02±8.12E-03
C07	0.00E+00±0.00E+00≈	4.78E-01±1.32E+00−	0.00E+00±0.00E+00≈	1.59E-01±7.97E-01−	1.59E-01±7.97E-01−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C08	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	4.25E+00±2.12E+01−	1.57E+01±3.95E+01−	0.00E+00±0.00E+00≈	0.00E+00 ±0.00E+00≈	0.00E+00±0.00E+00
C09	1.85E+04±2.09E+04−	3.87E+04±8.01E+04−	2.55E+00±8.23E+00−	3.35E+01±1.36E+02−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C10	2.72E+04±7.85E+04−	1.88E+04±3.79E+04−	3.42E+01±3.56E+01−	5.80E+00±1.16E+01−	1.07E+00±3.69E+00−	5.88E+02±1.25E+03−	0.00E+00±0.00E+00
C11	-3.92E-04±4.85E-11≈	-3.92E-04±6.87E-11≈	-3.92E-04±8.99E-11≈	-3.92E-04±4.47E-11≈	-3.92E-04±3.77E-11≈	-3.92E-04 ±1.48E-10≈	-3.92E-04±3.36E-07
C12	-1.99E-01±1.07E-08≈	-1.99E-01±2.35E-09≈	-1.99E-01±3.78E-07≈	-1.99E-01±1.94E-09≈	-1.99E-01±2.28E-09≈	-1.99E-01 ±7.65E-07≈	-1.99E-01±2.19E-08
C13	-6.78E+01±1.11E+00≈	-6.80E+01±5.47E-01≈	-6.78E+01±9.50E-01≈	-6.79E+01±1.09E+00≈	-6.82E+01±5.43E-01≈	-6.79E+01±1.26E+00≈	-6.80E+01±6.63E-01
C14	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	3.19E-01±1.10E+00−	1.59E-01±7.97E-01−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C15	2.07E+01±4.32E+00−	2.16E+01±2.33E-07−	2.98E+00±1.49E+01−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	1.30E+01±1.18E+01−	0.00E+00±0.00E+00
C16	5.26E-04±2.63E-03−	0.00E+00±0.00E+00≈	6.78E-04±3.39E-03−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C17	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	(0%)−	8.37E-02±1.78E-01−	0.00E+00±0.00E+00
C18	2.95E-01±8.06E-01−	1.23E+01±6.13E+01−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	(0%)−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
−	**10**	**10**	**8**	**7**	**11**	**7**	/
+	**0**	**0**	**0**	**0**	**0**	**0**	/
≈	**8**	**8**	**10**	**11**	**7**	**11**	/

all, CORCO is better than LSHADE44 on high-dimensional test functions.

F. Effectiveness of CI Obtained from the Learning Stage

In this paper, the correlation between constraints and objective function is measured by the correlation index CI and CI is obtained from the learning stage. Subsequently, CI is incorporated into three important parts of CORCO, i.e., the weighted sum updating approach, the archiving and replacement mechanism, and the search algorithm. Therefore, CI plays a critical role in CORCO. In order to validate the effectiveness of CI obtained from the learning stage, we tested five fixed values of CI: 0.0, 0.25, 0.5, 0.75, and 1.0. Besides, we also implemented a variant of CORCO, where CI is equal to Pearson's correlation coefficient (PCC). The 18 test functions with 30D from IEEE CEC2010 were selected to produce the experimental results and the Wilcoxon's rank sum test at a 0.05 significance level was used for statistical comparison.

The experimental results are summarized in Table X. Note that the feasible rate, i.e., percentage of runs in which at least one feasible solution can be found, is recorded if an algorithm cannot consistently find feasible solutions over 25 runs. As shown in Table X, CORCO performs better than $CI = 0.0$, $CI = 0.25$, $CI = 0.5$, $CI = 0.75$, and $CI = 1.0$ on 10, 10, eight, seven, and 11 test functions, respectively. However, CORCO with fixed settings of CI cannot surpass the original CORCO on any test function. Moreover, $CI = 1.0$ cannot provide any feasible runs on six test functions: C02, C04-C06, and C17-C18. Additionally, CORCO performs better than CORCO with $CI = PCC$ on seven test functions. In contrast, CORCO with $CI = PCC$ cannot be better than CORCO on any test function.

The above comparison suggests that no single fixed value of CI can suit different types of COPs and the learning stage of CORCO is able to successfully learn a proper value of CI. Moreover, standard correlation analyses such as Pearson's correlation coefficient may not be able to reveal the correlation between constraints and objective function.

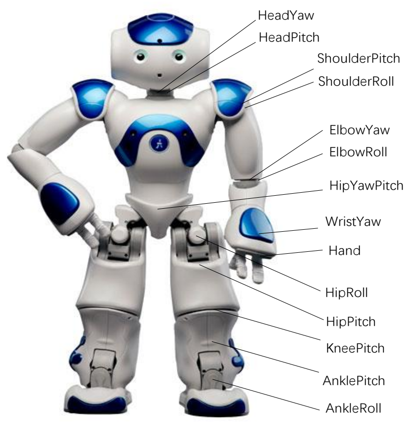


Fig. 11. DOFs in NAO.

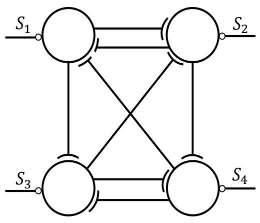
IV. REAL-WORLD APPLICATION

In this section, CORCO is applied to deal with the gait optimization of humanoid robots, with the aim of investigating its effectiveness in solving real-world COPs.

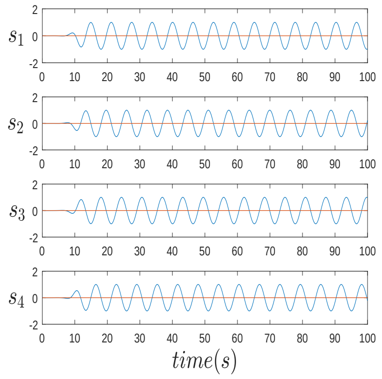
A. Problem Formulation

Gait optimization of humanoid robots is one of the most challenging problems in the control of humanoid robots and can be formulated as a COP. In a humanoid robot, gait optimization is based on the control of degree of freedom (DOF). Taking NAO as an example, there are 25 DOFs, of which 14 are in the upper body and 11 are in the lower body. Fig. 11 depicts the DOFs in NAO.

In NAO, 12 DOFs need to be controlled to ensure stable walking. They are HipRoll, HipPitch, KneePitch, AnklePitch, AnkleRoll, and ShoulderPitch, each of which is in the left body and the right body, respectively. Since some of the DOFs have a similar movement (e.g., HipRoll in the left leg and the right leg), they can be controlled by one base signal with some adjustments. In this case, we utilize four base signals to make NAO walk. In our method, Central Pattern Generator (CPG)



(a) CPG network



(b) Four base signals

Fig. 12. CPG network and four base signals generated by it.

TABLE XI
SIGNAL ASSIGNMENT OF EACH DOF

Name of DOF	Signal
RHipRoll	$k_1 \cdot s_1$
LHipRoll	$k_1 \cdot s_1$
RHipPitch	$k_2 \cdot s_4 - k_3$
LHipPitch	$k_2 \cdot s_2 - k_3$
RKneePitch	$k_4 \cdot max(0, s_1) + k_5$
LKneePitch	$k_4 \cdot max(0, s_1) + k_5$
RAnklePitch	$k_6 \cdot s_2 - k_7$
LAnklePitch	$k_6 \cdot s_4 - k_7$
RAnkleRoll	$k_8 \cdot s_3$
LAnkleRoll	$k_8 \cdot s_3$
RShouderPitch	$-k_9 \cdot s_2 - k_{10}$
LShouderPitch	$k_9 \cdot s_2 + k_{10}$

is used to generate the four base signals. The CPG network is shown in Fig. 12(a). There are four neurons in this network, and they are connected with each other to generate coupling signals. Each of the connections is attached with a weight, the value of which is set to 2. Then we can get the four base signals (i.e., s_1, s_2, s_3, and s_4) as shown in Fig. 12(b).

These four base signals need to be adjusted by some parameters to control the 12 DOFs in NAO. Thus, each DOF is assigned an adjusted signal, which is listed in Table XI. Since kneePicths cannot rotate reversely, when s_1 is used to control RkneePicth and LkneePitch, its value is restricted to be positive. In this paper, 10 parameters (i.e., $k_1, \ldots, k_{10}$) in Table XI, a parameter (k_{11}) for controlling the amplitude of the adjusted signals, and a periodic parameter (k_{12}) are encoded as a 12-dimensional decision vector $\vec{x} = (x_1, \ldots, x_{12})$.

TABLE XII
EXPERIMENTAL RESULTS OF FROFI, DEFR, AND CORCO OVER 25 INDEPENDENT RUNS ON THE GAIT OPTIMIZATION PROBLEM OF NAO

Statistic Value	FROFI (m)	DEFR (m)	CORCO (m)
Best	2.544173	1.657536	**2.949078**
Mean	1.183916	0.809671	**2.296031**
Worst	0.484514	0.084154	**1.809316**
Std	0.823426	0.604805	**0.386405**

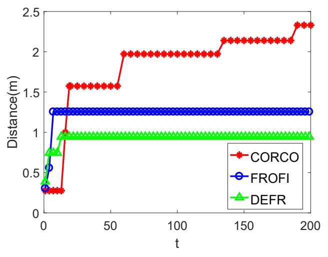


Fig. 13. Convergence curves of the best solution provided by the three compared algorithms in a typical run.

The objective function is formulated as:

$$\text{maximize}: f(\vec{x}) = d(\vec{x}) - p(\vec{x})$$
$$p(\vec{x}) = \begin{cases} 50, \text{if NAO falls down} \\ 0, \text{otherwise} \end{cases} \tag{16}$$

where $d(\vec{x})$ is the distance (m) NAO can walk within 10 seconds, and $p(\vec{x})$ is a penalty if NAO falls down. In order to make NAO walk fast and stably, the objective function needs to be maximized. Since the rotation angle of each DOF is restricted to a certain range, the constraints in (17) should be satisfied.

$$\begin{aligned}
x_{11} * x_1 - 0.44 &\leq 0 \\
x_{11} * x_2 - x_3 - 0.52 &\leq 0 \\
-x_{11} * x_2 + x_3 - 1.57 &\leq 0 \\
x_{11} * x_2 + x_3 - 1.57 &\leq 0 \\
x_{11} * x_4 + x_5 - 2.27 &\leq 0 \\
-x_{11} * x_4 - x_5 &\leq 0 \\
-x_{11} * x_6 - x_7 - 0.79 &\leq 0 \\
-x_{11} * x_6 + x_7 - 1.22 &\leq 0 \\
x_{11} * x_6 + x_7 - 1.22 &\leq 0 \\
x_{11} * x_8 - 0.79 &\leq 0 \\
x_{11} * x_9 + x_{10} - 2.09 &\leq 0
\end{aligned} \tag{17}$$

As a result, the gait optimization problem of NAO is formulated as a COP, and then CORCO is used to solve this COP.

B. Experimental Results

In the experiment, the evaluation of each solution was conducted by a 3D simulating software named webots6.2.4. Two other COEAs were adopted as the compared algorithms. The first one is FROFI [39], and the other is DE/rand1/bin combined with the feasibility rule [7] (named DEFR). 25 independent runs were implemented for each algorithm. The best, mean, and worst distances, and the standard deviation provided by the three compared algorithms are listed in Table XII.

From Table XII, NAO can walk 2.296031m on average when CORCO is applied, while the mean distances NAO can walk are 1.183916m and 0.809671m, when FROFI and DEFR are applied, respectively. Moreover, the standard deviations in Table XII indicate that the solutions generated by CORCO are more stable. Fig. 13 plots the convergence curves of the best solution provided by the three compared algorithms in a typical run, where t is the generation number. It shows that FROFI and DEFR converge to a local optimum quickly. In contrast, the convergence speed of CORCO is slower, because the learning stage of CORCO may delay the optimization at the beginning. However, once the evolving stage begins (after t=10 in this case), CORCO can quickly and continuously improve the quality of solutions. In the end, CORCO converges to a better value than both FROFI and DEFR.

The above experiment shows the effectiveness and stability of CORCO when it is used to solve a real-world COP.

V. Conclusion

In this paper, we found the important significance of the correlation between constraints and objective function in constrained evolutionary optimization. Based on this finding, we proposed an alternative COEA, named CORCO, which includes two stages, namely, the learning stage and the evolving stage. The learning stage was to mine the correlation between constraints and objective function of a COP. At the evolving stage, a weighted sum updating approach and an archiving and replacement mechanism were proposed to utilize the correlation to balance constraints and objective function. Extensive and systematic experiments on three benchmark test suites and a practical case verified that:

1) CORCO shows better or at least competitive performance against other state-of-the-art COEAs.
2) The learning stage can mine the correlation between constraints and objective function of different types of COPs successfully.
3) The weighted sum updating approach and the archiving and replacement mechanism can utilize the information obtained from the learning stage to balance constraints and objective function effectively.

The Matlab source codes of CORCO can be downloaded from: http://www.escience.cn/people/yongwang1/index.html

References

[1] Z. Michalewicz, “A survey of constraint handling techniques in evolutionary computation methods.” *Evolutionary Programming*, vol. 4, pp. 135–155, 1995.

[2] E. Mezura-Montes and C. A. Coello Coello, “Constraint-handling in nature-inspired numerical optimization: past, present and future,” *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 173–194, 2011.

[3] C. A. Coello Coello, “Constraint-handling techniques used with evolutionary algorithms,” in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*. ACM, 2016, pp. 563–587.

[4] Ö. Yeniay, “Penalty function methods for constrained optimization with genetic algorithms,” *Mathematical and Computational Applications*, vol. 10, no. 1, pp. 45–56, 2005.

[5] B. Tessema and G. G. Yen, “An adaptive penalty formulation for constrained evolutionary optimization,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 39, no. 3, pp. 565–578, 2009.

[6] J. Liu, K. L. Teo, X. Wang, and C. Wu, “An exact penalty function-based differential search algorithm for constrained global optimization,” *Soft Computing*, vol. 20, no. 4, pp. 1305–1313, 2016.

[7] K. Deb, “An efficient constraint handling method for genetic algorithms,” *Computer Methods in Applied Mechanics and Engineering*, vol. 186, no. 2, pp. 311–338, 2000.

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

[9] T. Takahama and S. Sakai, “Efficient constrained optimization by the ε constrained adaptive differential evolution,” in *2010 IEEE Congress on Evolutionary Computation*. IEEE, 2010, pp. 1–8.

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

[11] Y. Wang, Z. Cai, G. Guo, and Y. Zhou, “Multiobjective optimization and hybrid evolutionary algorithm to solve constrained optimization problems,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 3, pp. 560–575, 2007.

[12] L. Jiao, L. Li, R. Shang, F. Liu, and R. Stolkin, “A novel selection evolutionary strategy for constrained optimization,” *Information Sciences*, vol. 239, pp. 122–141, 2013.

[13] C. Peng, H.-L. Liu, and F. Gu, “A novel constraint-handling technique based on dynamic weights for constrained optimization problems,” *Soft Computing*, vol. 22, no. 12, pp. 3919–3935, 2018.

[14] A. Mani and C. Patvardhan, “A novel hybrid constraint handling technique for evolutionary optimization,” in *2009 IEEE Congress on Evolutionary Computation (CEC'09)*. IEEE, 2009, pp. 2577–2583.

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

[16] Q. Fan and X. Yan, “Differential evolution algorithm with co-evolution of control parameters and penalty factors for constrained optimization problems,” *Asia-Pacific Journal of Chemical Engineering*, vol. 7, no. 2, pp. 227–235, 2012.

[17] M. Ali and W. Zhu, “A penalty function-based differential evolution algorithm for constrained global optimization,” *Computational Optimization and Applications*, vol. 54, no. 3, pp. 707–739, 2013.

[18] H. J. Barbosa, A. C. Lemonge, and H. S. Bernardino, “A critical review of adaptive penalty techniques in evolutionary computation,” in *Evolutionary Constrained Optimization*. Springer, 2015, pp. 1–27.

[19] V. V. De Melo and G. Iacca, “A modified covariance matrix adaptation evolution strategy with adaptive penalty function and restart for constrained optimization,” *Expert Systems with Applications*, vol. 41, no. 16, pp. 7077–7094, 2014.

[20] C. Saha, S. Das, K. Pal, and S. Mukherjee, “A fuzzy rule-based penalty function approach for constrained evolutionary optimization,” *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2953–2965, 2016.

[21] T. Takahama and S. Sakai, “Efficient constrained optimization by the ε constrained rank-based differential evolution,” in *2012 IEEE Congress on Evolutionary Computation*. IEEE, 2012, pp. 1–8.

[22] ——, “Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation,” in *2010 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2010, pp. 1–9.

[23] S. Domínguez-Isidro and E. Mezura-Montes, “A cost-benefit local search coordination in multimeme differential evolution for constrained numerical optimization problems,” *Swarm and Evolutionary Computation*, vol. 39, pp. 249–266, 2018.

[24] W. Gao, G. G. Yen, and S. Liu, “A dual-population differential evolution with coevolution for constrained optimization.” *IEEE Trans. Cybernetics*, vol. 45, no. 5, pp. 1094–1107, 2015.

[25] C. A. Coello Coello, “Treating constraints as objectives for single-objective evolutionary optimization,” *Engineering Optimization*, vol. 32, no. 3, pp. 275–308, 2000.

[26] E. Mezura-Montes and C. A. Coello Coello, “A numerical comparison of some multiobjective-based techniques to handle constraints in genetic algorithms,” *Departamento de Ingeniera Elctrica, CINVESTAV-IPN, México, Tech. Rep. Technical Report EVOCINV-03-2002*, 2002.

[27] C. M. Fonseca and P. J. Fleming, “Multiobjective optimization and multiple constraint handling with evolutionary algorithms. i. a unified formulation,” *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 28, no. 1, pp. 26–37, 1998.

[28] Y. Wang and Z. Cai, "Combining multiobjective optimization with differential evolution to solve constrained optimization problems," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 1, pp. 117–134, 2012.

[29] ——, "A dynamic hybrid framework for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 1, pp. 203–217, 2012.

[30] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[31] R. Datta, K. Deb, and A. Segev, "A bi-objective hybrid constrained optimization (hycon) method using a multi-objective and penalty function approach," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 317–324.

[32] R. Datta and K. Deb, "An adaptive normalization based constrained handling methodology with hybrid bi-objective and penalty function approach," in *2012 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2012, pp. 1–8.

[33] ——, "Individual penalty based constraint handling using a hybrid bi-objective and penalty function approach," in *2013 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2013, pp. 2720–2727.

[34] Y. Wang and Z. Cai, "Constrained evolutionary optimization by means of (μ+ λ)-differential evolution and improved adaptive trade-off model," *Evolutionary Computation*, vol. 19, no. 2, pp. 249–285, 2011.

[35] W. Gong, Z. Cai, and D. Liang, "Adaptive ranking mutation operator based differential evolution for constrained optimization," *IEEE Transactions on Cybernetics*, vol. 45, no. 4, pp. 716–727, 2015.

[36] R. Mallipeddi, S. Das, and P. N. Suganthan, "Ensemble of constraint handling techniques for single objective constrained optimization," in *Evolutionary Constrained Optimization*. Springer, 2015, pp. 231–248.

[37] M. M. Mukaka, "A guide to appropriate use of correlation coefficient in medical research," *Malawi Medical Journal*, vol. 24, no. 3, pp. 69–71, 2012.

[38] Z. Wang, Q. Zhang, A. Zhou, M. Gong, and L. Jiao, "Adaptive replacement strategies for moea/d," *IEEE Transactions on Cybernetics*, vol. 46, no. 2, pp. 474–486, 2016.

[39] 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 Transactions on Cybernetics*, vol. 46, no. 12, pp. 2938–2952, 2016.

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

[41] B.-C. Wang, H.-X. Li, J.-P. Li, and Y. Wang, "Composite differential evolution for constrained evolutionary optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems, in press, doi: 10.1109/TSMC.2018.2807785*.

[42] S. Das and P. N. Suganthan, "Differential evolution: A survey of the state-of-the-art," *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.

[43] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution–an updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1–30, 2016.

[44] A. Ghosh, S. Das, S. S. Mullick, R. Mallipeddi, and A. K. Das, "A switched parameter differential evolution with optional blending crossover for scalable numerical optimization," *Applied Soft Computing*, vol. 57, pp. 329–352, 2017.

[45] J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. Suganthan, C. A. Coello Coello, and K. Deb, "Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization," *Journal of Applied Mechanics*, vol. 41, no. 8, 2006.

[46] R. Mallipeddi and P. N. Suganthan, "Problem definitions and evaluation criteria for the cec 2010 competition on constrained real-parameter optimization," *Nanyang Technological University, Singapore*, vol. 24, 2010.

[47] G. Wu, R. Mallipeddi, and P. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 competition on constrained real-parameter optimization," *National University of Defense Technology, Changsha, Hunan, PR China and Kyungpook National University, Daegu, South Korea and Nanyang Technological University, Singapore, Technical Report*, 2016.

[48] W. Zhang, G. G. Yen, and Z. He, "Constrained optimization via artificial immune system," *IEEE Transactions on Cybernetics*, vol. 44, no. 2, pp. 185–198, 2014.

[49] S. Elsayed, R. Sarker, C. A. Coello Coello, and T. Ray, "Adaptation of operators and continuous control parameters in differential evolution for constrained optimization," *Soft Computing*, vol. 22, no. 19, pp. 6595–6616, 2018.

[50] R. Mallipeddi and P. N. Suganthan, "Differential evolution with ensemble of constraint handling techniques for solving cec 2010 benchmark problems," in *2010 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2010, pp. 1–8.

[51] J. Alcalá-Fdez, L. Sanchez, S. Garcia, M. J. del Jesus, S. Ventura, J. M. Garrell, J. Otero, C. Romero, J. Bacardit, V. M. Rivas *et al.*, "Keel: a software tool to assess evolutionary algorithms for data mining problems," *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, vol. 13, no. 3, pp. 307–318, 2009.

[52] R. Poláková, "L-SHADE with competing strategies applied to constrained optimization," in *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2017, pp. 1683–1689.

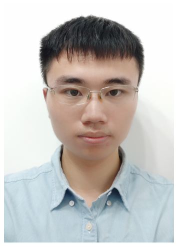
Yong Wang (M'08–SM'17) received the Ph.D. degree in control science and engineering from the Central South University, Changsha, China, in 2011.

He is a Professor with the School of Automation, Central South University, Changsha, China. His current research interests include the theory, algorithm design, and interdisciplinary applications of computational intelligence.

Dr. Wang is an Associate Editor for the *Swarm and Evolutionary Computation*. He was a Web of Science highly cited researcher in Computer Science in 2017 and 2018.

Jia-Peng Li received the B.S. degree in intelligent science and technology and the M.S. degree in control science and engineering both from the Central South University, Changsha, China, in 2015 and 2018, respectively.

He is working as a data analysist in Merchants Unicom Financial Corporation, Shenzhen, China. His current research interests include evolutionary constrained single- and multi-objective optimization, data mining, and machine learning.



Xihui Xue received the B.S. degree in automation and the M.S. degree in control science and engineering both from the Central South University, Changsha, China, in 2015 and 2018, respectively.

He is researching on LIDAR-based perception algorithm in SF Technology, Shenzhen, China. His current research interests include humanoid robots, reinforcement learning, and 3D data processing.

Bing-Chuan Wang received the B.S. degree in automation and the M.S. degree in control science and engineering both from the Central South University, Changsha, China, in 2013 and 2016, respectively.

He is currently pursuing the Ph.D. degree at the City University of Hong Kong, Hong Kong. His current research interests include evolutionary computation, modeling of distributed parameter systems, and battery management system.

Supplementary File for "Utilizing the Correlation between Constraints and Objective Function for Constrained Evolutionary Optimization"

S-1. Effectiveness of Learning CI at the Early Stage

To investigate the effectiveness of learning CI at the early stage, we implemented some variants by cycling the learning stage and the evolving stage. To be specific, these two stages were repeated every 100 generations. Additionally, the first $cycLenGen$ generations were utilized for learning while the remaining $(100 - cycLenGen)$ generations were utilized for evolving. By setting $cycLenGen$ to different values, i.e., $cycLenGen = 10$, $cycLenGen = 20$, $cycLenGen = 30$, $cycLenGen = 40$, and $cycLenGen = 50$, five variants, which were called CORCO-cycle1, CORCO-cycle2, CORCO-cycle3, CORCO-cycle4, and CORCO-cycle5, were taken into consideration. These variants were evaluated on the 18 test functions with 30D from IEEE CEC2010.

The experimental results are recorded in Table S-1. The feasible rate was reported when an algorithm cannot consistently obtain at least one feasible solution over 25 runs and the Wilcoxon's rank sum test at a 0.05 significance level was used to compare CORCO with each competitor. As shown in Table S-1, CORCO performs better than the other five variants on 12, 14, 13, 12, and 14 test functions, respectively. However, these variants cannot provide better results than CORCO on any test function. The experimental results reflect that learning CI at the early stage is better than the manner of cycling the learning stage and the evolving stage.

S-2. Effectiveness of the Archiving and Replacement Mechanism

The archiving and replacement mechanism proposed in Section II-D2 can compensate the weighted sum updating approach to a certain extent. To study its effectiveness, we removed it from CORCO and the resultant algorithm was called CORCO-WoA. Afterward, the performance of CORCO-WoA was compared with that of CORCO on the 18 test functions with 30D from IEEE CEC2010.

As shown in Table S-2, CORCO outperforms CORCO-WoA on seven test functions. However, CORCO-WoA is unable to perform better than CORCO on any test function. More importantly, CORCO-WoA fails to find any feasible solution over 13 and nine runs on C05 and C06, respectively. Obviously, the above experimental results confirm that the performance of CORCO benefits from the archiving and replacement mechanism.

S-3. Effectiveness of the Search Algorithm

To validate the effectiveness of the search algorithm, we implemented two variants of CORCO, i.e., CORCO-BaR and CORCO-Ada. In CORCO-BaR, DE/best/1/bin was utilized to enhance the convergence while DE/rand/1/bin was used to promote the diversity. As the same with CORCO, these two operators were selected in a random manner. In CORCO-Ada, DE/rand-to-best/1/bin and DE/current-to-rand/1 were selected in an adaptive manner as the same with [44]. CORCO, CORCO-BaR, and CORCO-Ada were evaluated on the 18 test functions with 30D from IEEE CEC2010.

As shown in Table S-3, CORCO provides better results than CORCO-BaR and CORCO-Ada on 15 and 10 test functions, respectively. However, both CORCO-BaR and CORCO-Ada cannot perform better than CORCO on any test function. By comparing CORCO with CORCO-BaR, it can be seen that DE/rand-to-best/1/bin and DE/current-to-rand/1 are effective for constrained optimization. By comparing CORCO with CORCO-Ada, we can find that selecting two operators randomly would be a good choice.

S-4. Parameter Sensitivity Analysis

CORCO includes a parameter, i.e., the maximum generation number of the learning stage $LearnGen$. A too small value of $LearnGen$ may not be enough for learning the correlation. However, a too big value of $LearnGen$ may consume great computational resource at the learning stage, thus remarkably reducing the computational resource for the evolving stage. Hence, a reasonable value should be set for this parameter. To this end, we tested five different $LearnGen$, i.e., $LearnGen = MaxGen/30$, $LearnGen = MaxGen/25$, $LearnGen = MaxGen/20$, $LearnGen = MaxGen/10$, and $LearnGen = MaxGen/5$ on the 18 test functions with 30D from IEEE CEC2010. Note that in the original CORCO, $LearnGen = MaxGen/20$. As shown in Table S-4, $LearnGen = MaxGen/20$ performs better than $LearnGen = MaxGen/30$, $LearnGen = MaxGen/25$, and $LearnGen = MaxGen/5$ on five, six, and seven test functions, respectively. However, $LearnGen = MaxGen/30$, $LearnGen = MaxGen/25$, and $LearnGen = MaxGen/5$ cannot provide better

TABLE S-1
EXPERIMENTAL RESULTS OF CORCO AND FIVE VARIANTS (I.E., CORCO-CYCLE1, CORCO-CYCLE2, CORCO-CYCLE3, CORCO-CYCLE4, AND CORCO-CYCLE5) OVER 25 INDEPENDENT RUNS ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

IEEE CEC2010 with 30D	CORCO-cycle1 Mean OFV±Std Dev (feasible rate)	CORCO-cycle2 Mean OFV±Std Dev (feasible rate)	CORCO-cycle3 Mean OFV±Std Dev (feasible rate)	CORCO-cycle4 Mean OFV±Std Dev (feasible rate)	CORCO-cycle5 Mean OFV±Std Dev (feasible rate)	CORCO Mean OFV±Std Dev (feasible rate)
C01	-8.19E-01±2.74E-03−	-8.21E-01±1.98E-03≈	-8.20E-01±1.85E-03≈	-8.21E-01 ±2.19E-03≈	-8.20E-01 ±1.95E-03≈	-8.21E-01±2.98E-03
C02	2.62E+00 ±1.13E+00−	2.05E+00 ±1.20E+00−	1.20E+00±1.20E+00−	1.88E-01±1.22E+00−	3.20E-01±7.69E-01−	-2.13E+00±1.22E-01
C03	9.01E+11±2.46E+12−	4.35E+10±2.13E+11−	2.52E+01±9.51E+00−	2.87E+01±5.71E-08−	2.87E+01±2.97E-07−	0.00E+00±0.00E+00
C04	-3.33E-06 ±7.33E-13≈	5.19E-02±1.80E-01−	-3.33E-06±2.49E-13≈	-3.33E-06±2.05E-11≈	-3.33E-06±8.72E-10≈	-3.33E-06±4.00E-03
C05	5.25E+02 ±4.16E+01−	4.72E+02 ±1.14E+02−	4.80E+02±1.01E+02−	4.37E+02±1.90E+02−	3.57E+02 ±1.84E+02−	-4.84E+02±5.56E-04
C06	5.15E+02 ±1.06E+02−	5.12E+02±9.89E+01−	4.21E+02±2.12E+02−	2.94E+02±2.90E+02−	1.18E+02±4.28E+02−	-5.30E+02±8.12E-03
C07	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	1.59E-01±7.97E-01−	0.00E+00±0.00E+00
C08	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	7.86E+00±2.72E+01−	0.00E+00±0.00E+00
C09	3.90E+13±2.59E+13−	3.37E+13±1.36E+13−	1.96E+13±1.34E+13−	4.00E+12±6.31E+12−	92%−	0.00E+00±0.00E+00
C10	3.82E+13±1.72E+13−	2.61E+13±1.57E+13−	2.17E+13±1.66E+13−	3.38E+12±6.02E+12−	6.09E+11±2.33E+12−	0.00E+00±0.00E+00
C11	-3.92E-04±6.74E-11≈	-3.61E-04 ±4.17E-05−	8%−	0%−	0%−	-3.92E-04±3.36E-07
C12	92%−	88%−	84%−	80%−	44%−	-1.99E-01±2.19E-08
C13	-6.80E+01±1.08E+00≈	-6.81E+01±5.70E-01≈	-6.66E+01±1.20E+00−	-6.18E+01±1.89E+00−	-5.59E+01±.46E+00−	-6.80E+01±6.63E-01
C14	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C15	9.09E+12 ±1.84E+13−	2.64E+11±7.98E+11−	4.70E+09±1.88E+10−	2.18E+01±1.14E+00−	2.16E+01±9.85E-07−	0.00E+00±0.00E+00
C16	7.90E-01±3.69E-01−	92%−	5.65E-02±9.74E-02−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C17	6.62E+02 ±7.40E+02−	92%−	96%−	1.51E+01±1.88E+01−	1.19E+00 ±2.25E+00−	0.00E+00±0.00E+00
C18	1.52E+04 ±8.21E+03−	1.15E+04±6.27E+03−	1.07E+04±7.69E+03−	7.72E+03±7.70E+03−	2.50E+03±5.34E+03−	0.00E+00±0.00E+00
−	**12**	**14**	**13**	**12**	**14**	/
+	**0**	**0**	**0**	**0**	**0**	/
≈	**6**	**4**	**5**	**6**	**4**	/

TABLE S-2
EXPERIMENTAL RESULTS OF CORCO AND CORCO-WOA OVER 25 INDEPENDENT RUNS ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

IEEE CEC2010 with 30D	CORCO-WoA Mean OFV±Std Dev (feasible rate)	CORCO Mean OFV±Std Dev (feasible rate)
C01	-1.59E-01±1.46E-02−	-8.21E-01±2.98E-03
C02	-2.14E+00±5.04E-02≈	-2.13E+00±1.22E-01
C03	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C04	2.53E-04±1.02E-03−	-3.33E-06±4.00E-03
C05	(48%)−	-4.84E+02±5.56E-04
C06	(64%)−	-5.30E+02±8.12E-03
C07	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C08	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C09	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C10	1.31E+00±2.32E+00−	0.00E+00±0.00E+00
C11	-3.92E-04±3.06E-07≈	-3.92E-04±3.36E-07
C12	-1.99E-01±1.97E-07≈	-1.99E-01±2.19E-08
C13	-6.83E+01±4.28E-01≈	-6.80E+01±6.63E-01
C14	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C15	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C16	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C17	4.11E+01±1.83E+00−	0.00E+00±0.00E+00
C18	3.85E+02±1.65E+03−	0.00E+00±0.00E+00
−	**7**	/
+	**0**	/
≈	**11**	/

results than $LearnGen = MaxGen/20$ on any test function. Overall, $LearnGen = MaxGen/20$ performs similarly to $LearnGen = MaxGen/10$. Therefore, a value between $MaxGen/20$ and $MaxGen/10$ was recommended for CORCO.

Besides, the sensitivity of α in (11) and (12) was investigated. As described in (11) and (12), t is a variable between 0 and $maxGen$. Consequently, $\frac{t}{MaxGen}$ is a scale-free variable between 0 and 1, no matter what $MaxGen$ is. Thus, α is not dependent on $MaxGen$. In addition, the other two parameters, i.e., CI and DV, which are related to $lb(t)$ and $ub(t)$, can be learnt at the learning stage. Hence, only α should be set manually. Its sensitivity was studied experimentally. To be specific, five variants with different α, i.e., $\alpha = 5$, $\alpha = 15$, $\alpha = 25$, $\alpha = 35$, and $\alpha = 45$, were implemented. All these five variants were evaluated on the 18 test functions with 30D from IEEE CEC2010, and the experimental results are summarized in Table S-5. From Table S-5, CORCO with $\alpha = 25$ performs better than CORCO with $\alpha = 5$, $\alpha = 15$, $\alpha = 35$, and $\alpha = 45$ on four, one, three, and six test functions, respectively. However, CORCO with $\alpha = 5$, $\alpha = 15$, $\alpha = 35$, and $\alpha = 45$ cannot produce better results than CORCO with $\alpha = 25$ on more than one test function. Thus, $\alpha = 25$ was recommended in this paper. Additionally, CORCO with $\alpha = 15$, $\alpha = 25$, and $\alpha = 35$ perform similarly on most of test functions. Therefore, α is not sensitive and we suggest that an α value between 15 and 35 is a good choice.

As shown in the archiving and replacement mechanism, μ is the probability of replacing P^t with A^t. Thus, the sensitivity of μ should be investigated. To this end, we implemented five variants of CORCO with five different values: $\mu = 10^{-4}$, $\mu = 10^{-3}$, $\mu = 10^{-2}$, $\mu = 10^{-1}$, and $\mu = 10^{0}$. All these variants were evaluated on the 18 test functions with 30D from IEEE

TABLE S-3
EXPERIMENTAL RESULTS OF CORCO AND TWO VARIANTS WITH DIFFERENT SEARCH ALGORITHMS OVER 25 INDEPENDENT RUNS ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

IEEE CEC2010 with 30D	CORCO-BaR Mean OFV±Std Dev	CORCO-Ada Mean OFV±Std Dev	CORCO Mean OFV±Std Dev
C01	-8.19E-01±2.36E-03−	-8.20E-01±2.47E-03≈	-8.21E-01±2.98E-03
C02	-2.13E+00±8.39E-02≈	-2.11E+00±1.02E-01≈	-2.13E+00±1.22E-01
C03	(96%)−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C04	5.62E-05±1.32E-04−	-3.31E-06 ±3.17E-08−	-3.33E-06±4.00E-03
C05	-4.84E+02±8.29E-02≈	-4.84E+02±7.21E-02≈	-4.84E+02±5.56E-04
C06	(68%)−	(88%)−	-5.30E+02±8.12E-03
C07	1.59E-01±7.97E-01−	1.59E-01±7.97E-01−	0.00E+00±0.00E+00
C08	8.55E+00 ±2.75E+01−	5.46E+00±2.01E+01−	0.00E+00±0.00E+00
C09	(92%)−	(92%) −	0.00E+00±0.00E+00
C10	1.07E+00 ±3.72E+00−	5.24E-01±2.62E+00−	0.00E+00±0.00E+00
C11	(88%)−	(96%)−	-3.92E-04±3.36E-07
C12	(92%)−	-1.99E-01±2.79E-09 ≈	-1.99E-01±2.19E-08
C13	-6.73E+01±1.50E+00≈	-6.81E+01±7.23E-01≈	-6.80E+01±6.63E-01
C14	4.61E+00 ±2.06E+01−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C15	3.44E-01±1.19E+00−	3.43E-01±1.19E+00−	0.00E+00±0.00E+00
C16	1.71E-03±6.03E-03−	0.00E+00 ±0.00E+00≈	0.00E+00±0.00E+00
C17	1.70E+01 ±8.37E+01−	(76%)−	0.00E+00±0.00E+00
C18	2.07E+02±1.02E+03−	3.44E-02±1.21E-01−	0.00E+00±0.00E+00
−	**15**	**10**	/
+	**0**	**0**	/
≈	**3**	**8**	/

TABLE S-4
EXPERIMENTAL RESULTS OF CORCO WITH FIVE DIFFERENT $LearnGen$ VALUES OVER 25 INDEPENDENT RUNS ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

IEEE CEC2010 with 30D	$LearnGen = MaxGen/30$ Mean OFV±Std Dev (feasible rate)	$LearnGen = MaxGen/25$ Mean OFV±Std Dev (feasible rate)	$LearnGen = MaxGen/10$ Mean OFV±Std Dev (feasible rate)	$LearnGen = MaxGen/5$ Mean OFV±Std Dev (feasible rate)	$LearnGen = MaxGen/20$ Mean OFV±Std Dev (feasible rate)
C01	-8.20E-01±1.80E-03≈	-8.18E-01±4.74E-03−	-8.20E-01±3.14E-03≈	-8.20E-01±1.91E-03≈	-8.21E-01±2.98E-03
C02	-1.88E+00±2.40E-01−	-2.01E+00±1.17E-01−	-2.11E+00±1.01E-01≈	-1.97E+00±1.92E-01−	-2.13E+00±1.22E-01
C03	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C04	-1.01E-06±4.47E-06−	1.60E-06±4.81E-05−	-3.33E-06±3.19E-09≈	-3.32E-06±5.43E-09≈	-3.33E-06±4.00E-03
C05	-4.84E+02±6.86E-01≈	-4.84E+02±6.80E-02≈	-4.84E+02±8.17E-02≈	-4.78E+02±1.88E+01−	-4.84E+02±5.56E-04
C06	(8%)−	(40%)−	-5.28E+02±1.62E+00−	-5.26E+02±2.55E+00−	-5.30E+02±8.12E-03
C07	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C08	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C09	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C10	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	1.89E+01±3.39E+01−	0.00E+00±0.00E+00
C11	-3.92E-04±1.56E-07≈	-3.92E-04±3.98E-08≈	-3.92E-04±9.21E-11≈	-3.72E-04±5.83E-05−	-3.92E-04±3.36E-07
C12	-1.99E-01±2.04E-08≈	-1.99E-01±7.99E-08≈	-1.99E-01±4.44E-08≈	-1.99E-01±8.07E-10≈	-1.99E-01±2.19E-08
C13	-6.79E+01±1.10E+00≈	-6.80E+01±4.66E-01≈	-6.76E+01±1.01E+00≈	-6.65E+01±1.61E+00−	-6.80E+01±6.63E-01
C14	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C15	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C16	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C17	(60%)−	(54%)−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C18	6.46E+01±1.35E+02−	2.27E-02±8.82E-02−	0.00E+00±0.00E+00≈	1.13E-02±4.39E-02−	0.00E+00±0.00E+00
−	**5**	**6**	**1**	**7**	/
+	**0**	**0**	**0**	**0**	/
≈	**13**	**12**	**17**	**11**	/

CEC2010. The experimental results are summarized in Table S-6. As shown in Table S-6, CORCO with $\mu = 10^{-2}$ is better than CORCO with $\mu = 10^{-4}$, $\mu = 10^{-3}$, $\mu = 10^{-1}$, and $\mu = 10^{0}$ on four, four, four, and five test functions, respectively. In contrast, CORCO with $\mu = 10^{-4}$, $\mu = 10^{-3}$, $\mu = 10^{-1}$, and $\mu = 10^{0}$ cannot perform better than CORCO with $\mu = 10^{-2}$ on any test function. As a result, $\mu = 10^{-2}$ was adopted in this paper.

In the search algorithm, DE/rand-to-best/1/bin is selected with the probability ps while DE/current-to-rand/1 is selected with the probability $(1 - ps)$. In this manner, a bigger ps would prefer convergence while a smaller one would prefer diversity. To choose a proper ps, we implemented five variants of CORCO with five different values: $ps = 0.1$, $ps = 0.3$, $ps = 0.5$, $ps = 0.7$, and $ps = 0.9$. All these variants were evaluated on the 18 test functions with 30D from IEEE CEC2010. The experimental results are collected in Table S-7. As shown in Table S-7, CORCO with $ps = 0.5$ performs better than CORCO with $ps = 0.1$, $ps = 0.3$, $ps = 0.7$, and $ps = 0.9$ on six, four, two, and eight test functions, respectively. However, CORCO with $ps = 0.1$, $ps = 0.3$, $ps = 0.7$, and $ps = 0.9$ cannot perform better than CORCO with $ps = 0.5$ on more than one test function. Consequently, $ps = 0.5$ was utilized in this paper.

TABLE S-5
EXPERIMENTAL RESULTS OF CORCO WITH FIVE VARYING α OVER 25 INDEPENDENT RUNS ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

IEEE CEC2010 with 30D	$\alpha = 5$	$\alpha = 15$	$\alpha = 35$	$\alpha = 45$	$\alpha = 25$
	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)
C01	-8.20E-01 ±1.84E-03≈	-8.20E-01±2.34E-03≈	-8.20E-01±2.98E-03≈	-8.20E-01±1.86E-03≈	-8.21E-01±2.98E-03
C02	-2.23E+00±4.62E-02+	-2.17E+00±5.67E-02≈	-2.01E+00±1.73E-01−	-2.02E+00±1.29E-01−	-2.13E+00±1.22E-01
C03	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C04	-3.33E-06 ±3.87E-09≈	-3.33E-06±1.25E-07≈	-3.21E-06±5.01E-07−	-3.23E-06±2.28E-07−	-3.33E-06±4.00E-03
C05	-4.84E+02±1.76E-02≈	-4.84E+02±1.92E-02≈	-4.83E+02±1.78E-01≈	(96%)−	-4.84E+02±5.56E-04
C06	-5.30E+02±2.89E-01≈	-5.30E+02±3.47E-01≈	(88%)−	(60%)−	-5.30E+02±8.12E-03
C07	0.00E+00 ±0.00E+00≈	0.00E+00 ±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C08	1.79E+00 ±8.95E+00≈	1.18E+01±3.28E+01≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C09	2.89E+01 ±9.93E+01−	1.77E-01±8.84E-01≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C10	8.60E+00 ±2.80E+01−	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C11	-3.92E-04±3.26E-11≈	-3.92E-04±1.31E-11≈	-3.92E-04±3.24E-11≈	(96%)−	-3.92E-04±3.36E-07
C12	-1.99E-01±2.38E-09≈	-1.99E-01±2.83E-03≈	-1.99E-01±3.33E-03≈	(92%)−	-1.99E-01±2.19E-08
C13	-6.79E+01±9.05E-01≈	-6.77E+01±7.79E-01≈	-6.80E+01±7.91E-01≈	-6.82E+01±4.95E-01≈	-6.80E+01±6.63E-01
C14	0.00E+00 ±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C15	8.60E+01±3.32E+02−	0.00E+00±0.00E+00 ≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C16	0.00E+00±0.00E+00≈	0.00E+00 ±0.00E+00≈	0.00E+00 ±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C17	(64%)−	(76%)−	0.00E+00±0.00E+00≈	0.00E+00 ±0.00E+00≈	0.00E+00±0.00E+00
C18	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
−	**4**	**1**	**3**	**6**	**/**
+	**1**	**0**	**0**	**0**	**/**
≈	**13**	**17**	**15**	**12**	**/**

TABLE S-6
EXPERIMENTAL RESULTS OF CORCO WITH FIVE VARYING μ OVER 25 INDEPENDENT RUNS ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

IEEE CEC2010 with 30D	$\mu = 10^{-4}$	$\mu = 10^{-3}$	$\mu = 10^{-1}$	$\mu = 10^{0}$	$\mu = 10^{-2}$
	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)
C01	-8.20E-01±4.56E-03≈	-8.20E-01 ±2.18E-03≈	-8.20E-01 ±2.27E-03≈	-8.20E-01 ±2.63E-03≈	-8.21E-01±2.98E-03
C02	-2.15E+00±6.68E-02≈	-2.13E+00±7.68E-02≈	-1.41E+00±3.45E-01−	-8.26E-01 ±6.65E-01−	-2.13E+00±1.22E-01
C03	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C04	-2.70E-06±2.00E-06−	-3.19E-06 ±2.98E-07−	-3.33E-06 ±1.20E-09≈	-3.33E-06 ±4.19E-11≈	-3.33E-06±4.00E-03
C05	(76%)−	(84%)−	-9.33E+01±2.32E+02−	3.42E+02±2.32E+02−	-4.84E+02±5.56E-04
C06	(88%)−	-5.27E+02±3.43E-02−	(36%)−	-4.51E+02±1.13E+02−	-5.30E+02±8.12E-03
C07	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C08	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C09	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C10	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C11	-3.92E-04 ±4.38E-11≈	-3.92E-04±6.74E-11≈	-3.92E-04±3.47E-11≈	-3.92E-04±1.73E-11≈	-3.92E-04±3.36E-07
C12	-1.99E-01±6.70E-07≈	-1.99E-01 ±7.20E-07≈	-1.99E-01 ±9.32E-08≈	-1.99E-01 ±1.17E-08≈	-1.99E-01±2.19E-08
C13	-6.81E+01±6.45E-01≈	-6.77E+01±1.26E+00≈	-6.79E+01±9.32E-01≈	-6.79E+01±1.16E+00≈	-6.80E+01±6.63E-01
C14	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00 ±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C15	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00 ±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C16	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00 ±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C17	(72%)−	(84%)−	(88%)−	(96%)−	0.00E+00±0.00E+00
C18	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00 ±0.00E+00≈	1.07E+02±4.40E+02−	0.00E+00±0.00E+00
−	**4**	**4**	**4**	**5**	**/**
+	**0**	**0**	**0**	**0**	**/**
≈	**14**	**14**	**14**	**13**	**/**

TABLE S-7
EXPERIMENTAL RESULTS OF CORCO WITH FIVE VARYING ps OVER 25 INDEPENDENT RUNS ON THE 18 TEST FUNCTIONS WITH 30D FROM IEEE CEC2010

IEEE CEC2010 with 30D	$ps = 0.1$	$ps = 0.3$	$ps = 0.7$	$ps = 0.9$	$ps = 0.5$
	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)	Mean OFV±Std Dev (feasible rate)
C01	-8.21E-01±2.43E-03≈	-8.20E-01±2.97E-03≈	-8.20E-01±2.05E-03≈	-8.20E-01±3.36E-03≈	-8.21E-01±2.98E-03
C02	-2.98E-01±6.35E-01−	-1.76E+00±2.98E-01−	-2.18E+00±7.18E-02≈	-2.20E+00±6.66E-02+	-2.13E+00±1.22E-01
C03	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C04	6.31E-04±7.59E-04−	-3.11E-07 ±5.02E-06−	-3.33E-06 ±7.78E-09≈	-3.33E-06 ±3.61E-09≈	-3.33E-06±4.00E-03
C05	(56%)−	-4.83E+02±5.25E-01≈	-4.84E+02±5.92E-02≈	-4.84E+02±1.82E-01≈	-4.84E+02±5.56E-04
C06	(0%)−	(60%)−	-5.26E+02±4.56E+00 −	-5.26E+02±3.32E+00−	-5.30E+02±8.12E-03
C07	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C08	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	4.07E+01±8.34E+01−	0.00E+00±0.00E+00
C09	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C10	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
C11	-3.92E-04±2.51E-09≈	-3.92E-04 ±7.15E-11≈	-3.92E-04 ±2.32E-11≈	(88%)−	-3.92E-04±3.36E-07
C12	-1.99E-01±1.33E-07≈	-1.99E-01±3.58E-07≈	-1.99E-01±2.58E-08≈	(92%)−	-1.99E-01±2.19E-08
C13	-5.40E+01±1.30E+00−	-6.72E+01±1.25E+00≈	-6.81E+01±7.56E-01≈	-6.81E+01±9.48E-01≈	-6.80E+01±6.63E-01
C14	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	1.60E+01±6.11E+01−	0.00E+00±0.00E+00
C15	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	9.06E-01±2.34E+00−	0.00E+00±0.00E+00
C16	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	5.84E-03±2.66E-02−	0.00E+00±0.00E+00
C17	(64%)−	(64%)−	(84%)−	(84%)−	0.00E+00±0.00E+00
C18	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00≈	0.00E+00±0.00E+00
−	**6**	**4**	**2**	**8**	**/**
+	**0**	**0**	**0**	**1**	**/**
≈	**12**	**14**	**16**	**9**	**/**