IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 23, 2020, accepted July 3, 2020, date of publication July 15, 2020, date of current version July 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3009429

On the Cooperation Between Evolutionary Algorithms and Constraint Handling Techniques: A Further Empirical Study

HUI CAI[1], BIN LIU[2], AND SHU PAN[3]
[1]Business School, University of Shanghai for Science and Technology, Shanghai 200093, China
[2]School of Finance and Economics, Jiangsu University, Zhenjiang 212013, China
[3]School of Computer Science, Jiangsu University of Science and Technology, Zhenjiang 212013, China
Corresponding author: Bin Liu (082015163@fudan.edu.cn)

ABSTRACT Cooperation between Evolutionary Algorithms (EAs) and Constraint Handling Techniques (CHTs) plays an important role in Constrained Optimization Evolutionary Algorithms (COEAs). A constrained composite differential evolution (C^2oDE) sets a good example on the cooperation. This paper tries to further study the inner mechanism, i.e., the effect of different methods on generating and selecting solutions. In the solution generating part, a new method, which adopts stochastic ranking in selecting the best individual in the DE mutation operators, is proposed. In the solution selecting part, a new constraint handling technique combination is added. The experiments on benchmark functions from IEEE CEC2006 verify the effect. During the experiment, it is found that a new cooperation manner performs better than C^2oDE, which reflects the importance of cooperation.

INDEX TERMS Constrained optimization, constraint handling techniques, differential evolution, cooperation manner.

I. INTRODUCTION

Constrained Optimization Problems (COPs) exist widely in the real-world applications [1], [2]. Generally, the COPs can be formed as:

$$\begin{aligned} &\text{Minimize } f(\vec{x}) \\ &\text{Subject to: } g_j(\vec{x}) \le 0, \quad j = 1, \cdots, l \\ &\qquad\qquad h_j(\vec{x}) = 0, \quad j = l+1, \cdots, m. \end{aligned} \tag{1}$$

Here, $\vec{x} = (x_1, \cdots, x_n)$ is the decision variable. The decision variable is bounded by the decision space S, and S is defined by the constraints:

$$L_i \le x_i \le U_i, \quad 1 \le i \le n. \tag{2}$$

Here, l is the number of inequality constraints and m-l is the number of equality constraints.

The Evolutionary Algorithms (EAs) are essentially unconstraint search techniques for generating solutions. To solve the COPs, constraint handling techniques (CHTs) are needed to select the solutions, which form together with EAs as constrained optimization evolutionary algorithms (COEAs) [3].

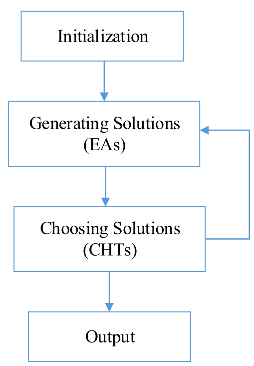

FIGURE 1. Flowchart of COEAs.

The flowchart of COEAs is illustrated as in Fig. 1. After initialization, the EAs generate solutions, and CHTs choose the better solutions as the parents in the EAs for the next generation.

The three most frequently used basic CHTs in COEAs are penalty functions, biasing feasible over infeasible solutions and multi-objective optimization.

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

Based on the basic CHTs, many concepts like cooperative coevolution [4], [5] and ensemble [6], [7] have been proposed. And some researchers tried to solve the problems from some other aspects, e.g., problem characteristics [8], [9].

Many comparisons of different EAs and CHTs are also proposed. Mezura-Montes *et al.* [10] proposed a simple combination of two DE variants (i.e., *DE/rand/1/bin* and *DE/best/1/bin*) based on the empirical analysis of four DE variants. Li *et al.* [11] suggested more experimental comparisons on different constraint-handling techniques are needed. They compared three representative constraint-handling techniques (i.e., Constrained-domination Principle, Self-adaptive Penalty, and Adaptive Tradeoff Model), and the search algorithm is nondominated sorting genetic algorithm II (NSGA II). Three properties of the problems are also summarized: the shape of Pareto front, the dimension of decision vector, and the size of feasible region. Kukkonen and Mezura-Montes [12] compared two existing constraint handling approaches with different ways to select the infeasible solutions. DE served as the searching algorithm. The paper concluded that neither of the constraint handling approaches can be judged to be better than the other. Bin *et al.* [5] proposed a cooperative ranking-based mutation strategy (CRM) for DE when solving COPs. Specially, two different ranking criteria (objective function value-based and constraint violation-based) are adopted in a cooperative way.

Recently, a constrained composite differential evolution (called C^2oDE) [13] was proposed based on CoDE [14] to solve COPs. Three different trial vectors of DE and two CHTs are adopted to get a balance between diversity and convergence, constraints and objective function.

C^2oDE sets a good example for EA and CHT cooperation, i.e., solution generating and solution choosing. As Yang mentioned [15], though researchers know the basic mechanisms of how the algorithms can work in practice, it is not quite clear why they work and under exact what conditions. So is there any inner mechanism behind this method, or which characteristics make the method work so well, is what we will study in this paper. To do this, more empirical studies will be carried out.

In the paper [16], to verify how much can be improved through good evolutionary algorithms, or whether a good enough EA can make up the shortcoming of a simple CHT, four different EAs and Deb's feasibility-based rule are taken as an example. Results show that better performance in EAs is not necessarily the reason for the improved performance of COEAs, and the key point is to find the shortcoming of the CHT and improve the shortcoming in the corresponding revision of EA.

It should be pointed out that not all cooperation is successful and effective. The inner mechanism is similar as that of the social division of labor. The key point lies in the complementarity, not simple over lapping. So it is very important to know the inner mechanism. As to the algorithm design for solving COPs, it is the characteristic of solution generating and selecting.

This paper mainly tries to find out the differences of various methods on generating and selecting the solutions. In other words, this paper focus on not only how many problems the algorithms perform better or worse on, but also which problem and the corresponding problem characteristics, which may give some inspiration for future problem solving or algorithm designing.

This paper will further verify this relationship from three aspects: 1) how much will be influenced if we change some parameters in EAs, e.g., the way to choose the best solution, which will guide generating the solutions;2) how much will be influenced if we change some CHTs;3) if we apply the same method in EA and CHT, which part will be improved more?

The contributions of this paper are summarized as follows:

- Effect of different mechanisms in EAs to generate solutions and CHT combinations to select solutions are studied.
- A new method which adopts stochastic ranking in selecting the best individual is proposed.
- A new CHT combination is added and the result is even better than the original C^2oDE.
- Systematic experiments have demonstrated the importance of cooperation between EAs and CHTs, especially with the problem characteristics considered.

The rest of this paper is organized as follows. Section II introduces the basic EAs and CHTs in COEAs. Section III illustrates the proposed method. Section IV presented the experimental results and analysis. Finally, Section V concludes this paper and provides some possible paths for future research.

II. DIFFERENTIAL EVOLUTION AND CONSTRAINT HANDLING TECHNIQUES

A. DIFFERENTIAL EVOLUTION (DE)

DE, as a simple and efficient EA, was proposed by Storn and Price [17]. It can be seen as equation-based algorithms [15]. It mainly uses mutation and crossover operations to generate a trial vector to compete with the target vector and the better one will be preserved for next generation. Many variants of DE have been proposed.

The population of DE consists of NP n-dimensional real-valued vectors

$$\vec{x}_i = \left\{x_{i,1}, x_{i,2}, \ldots, x_{i,n}\right\}, \quad i = 1, 2, \ldots, NP \tag{3}$$

The mutation, crossover and selection operations are as follows.

1) MUTATION OPERATION

In the mutation stage, a mutant vector for each target vector $\vec{x}_i$ is generated.

There are many popular mutation operators as follows [18], [19]:

- DE/rand/1

$$\vec{v}_i^t = \vec{x}_{r_1}^t + F \cdot (\vec{x}_{r_2}^t - \vec{x}_{r_3}^t) \tag{4}$$

- DE/rand/2

$$\vec{v}_i^t = \vec{x}_{r_1}^t + F \cdot (\vec{x}_{r_2}^t - \vec{x}_{r_3}^t) + F \cdot (\vec{x}_{r_4}^t - \vec{x}_{r_5}^t) \quad (5)$$

- DE/rand-to-best/1

$$\vec{v}_i^t = \vec{x}_{r_1}^t + F \cdot (\vec{x}_{best}^t - \vec{x}_{r_1}^t) + F \cdot (\vec{x}_{r_2}^t - \vec{x}_{r_3}^t) \quad (6)$$

- DE/current-to-best/1

$$\vec{v}_i^t = \vec{x}_i^t + F \cdot (\vec{x}_{best}^t - \vec{x}_i^t) + F \cdot (\vec{x}_{r_1}^t - \vec{x}_{r_2}^t) \quad (7)$$

- DE/current-to-rand/1

$$\vec{v}_i^t = \vec{x}_i^t + rand \cdot (\vec{x}_{r_1}^t - \vec{x}_i^t) + F \cdot (\vec{x}_{r_2}^t - \vec{x}_{r_3}^t) \quad (8)$$

Here, r_1, r_2, r_3, r_4, and r_5 are mutually exclusive integers randomly selected from $[1, NP]$, and $rand$ is a uniformly distributed random number between 0 and 1. $\vec{x}_{best}^t$ is the best target vector in the current population.

Different mutation operators have different characteristics, and the operators can be classified according to whether the best individual involved.

2) CROSSOVER OPERATION

In the crossover stage, a trial vector $\vec{u}_i$ will be generated through the binomial crossover operation on the target vector $\vec{x}_i$ and the mutant vector $\vec{v}_i$

$$u_{i,j} = \begin{cases} v_{i,j} & if\ rand_j \le C_r\ or\ j = j_{rand} \\ x_{i,j} & otherwise \end{cases} \quad (9)$$

Here, $i = 1, 2, \ldots, NP$, $j = 1, 2, \ldots, n$, j_{rand} is a randomly chosen integer within the range$[1, n]$, $rand_j$ is the jth evaluation of a uniform random number generator within $[0, 1]$, and C_r is the crossover control parameter. To keep the trial vector $\vec{u}_i$ different from its target vector $\vec{x}_i$, the condition $j = j_{rand}$ is added.

3) SELECTION OPERATION

The selection operation is mainly used to choose the better solution for the next generation between the trial vector $\vec{u}_i$ and the target vector $\vec{x}_i$

$$\vec{x}_i = \begin{cases} \vec{u}_i\ if\ f(\vec{u}_i) \le f(\vec{x}_i) \\ \vec{x}_i\ otherwise \end{cases} \quad (10)$$

B. CONSTRAINT HANDLING TECHNIQUES ADOPTED IN C²oDE

The CHTs adopted in C^{2}oDE is Deb's feasibility-based rule and ε-constrained method. Both of them pair-wise compare the solutions.

1) DEB'S FEASIBILITY-BASED RULE

To compare separately the objective functions and constraint violations, Deb [20] proposed a feasibility-based rule to pair-wise compare individuals:

1) Any feasible solution is preferred to any infeasible solution.

2) Among two feasible solutions, the one having better objective function value is preferred.

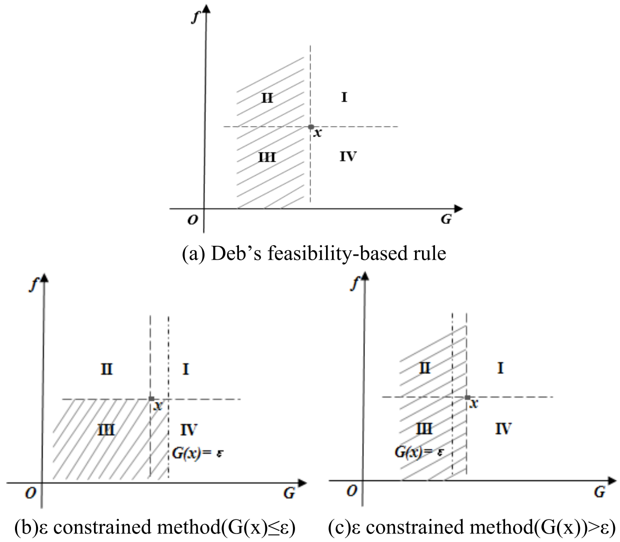

FIGURE 2. Comparison of different CHTs on the *f*-*G* space.

3) Among two infeasible solutions, the one having smaller constraint violation is preferred.

As can be seen from Fig. 2, Deb's feasibility-based rule considers that individuals in Regions II and III are unconditionally superior to individuals x, which indicates the guidelines are highly greedy. It is a very popular CHT for the simplicity and easy to realize.

2) ε CONSTRAINED METHOD

ε constrained method was proposed by Takahama and Sakai [21], [22]. It first compares the constraint violation of two solutions, and if the constraint violation is the same or less than a threshold, then the solution with less objective function will be better; otherwise, the solution with less constraint violation will be preferred.

Compare with Deb's feasibility-based rule, ε constrained method reduces the greedy to a certain extend.

$$(\phi_1, f_1) < (\phi_2, f_2) \Leftrightarrow \begin{cases} f_2 < f_1 & if\ \phi_1, \phi_2 < \varepsilon_k \\ f_2 < f_1 & if\ \phi_1 = \phi_2 \\ \phi_2 < \phi_1 & otherwise \end{cases} \quad (11)$$

where

$$\varepsilon_k = \begin{cases} \varepsilon(0)(1 - \dfrac{k}{T_c})^{cp} & if\ \frac{k}{T_c} \le p \\ 0 & otherwise \end{cases} \quad (12)$$

$$cp = -\frac{\log \varepsilon(0) + \lambda}{\log(1 - p)} \quad (13)$$

Here, $\varepsilon(0)$ is the initial threshold set to be maximum degree of constraint violation of the initial population. T_c is the maximum generation number, and k is the current generation number. λ is set to 6 in this paper, and p controls the degree that the information of objective function is exploited.

III. BASIC IDEA

A. DIFFERENT COMBINATION OF BEST INDIVIDUAL SELECTION

As mentioned in C^2oDE, the individual with the least degree of constraint violation is chosen as the "best" individual in the modified *DE/rand-to-best/1/bin* while the individual with the best objective function value is selected as the "best" individual in *DE/current-to-best/1/bin*. Some experiments are also carried out to verify the effect of this method.

Needless to say, best individual selection plays an important role in guiding the search process. Since pair-wise comparison is used in the two CHTs adopted in C^2oDE, how to select the best individual is really a key task, which may directly relate with the solution in the next generation.

So it is very necessary to further verify which factor influences the selection, and which manner will be the best with CHTs adopted.

As two mutation operators (modified *DE/rand-to-best/1/bin* and *DE/current-to-best/1/bin*) and two selection criteria (through the objective function value f or through the constraint violation g) are adopted, there are four combinations altogether, i.e., f-f, f-g, g-f, g-g, as shown in Table 1. For example, f-g in the table means that the manners of selecting the "best" individual in the modified *DE/rand-to-best/1/bin* and *DE/current-to-best/ 1/bin* are in terms of the objective function value and constraint violation respectively.

TABLE 1. Best individual selection combination.

DE/rand-to-best/1/bin \ DE/current-to-best/1/bin	f	g
f	f-f	f-g
g	g-f	g-g

B. STOCHASTIC RANKING BASED BEST INDIVIDUAL SELECTION

Besides the above four fixed combinations, a new stochastic ranking based best individual selection is proposed as Fig.3.

Algorithm 1: Stochastic Ranking Based Best Individual Selection

Input: selection pool: through f, through g
  Pf: the parameter for selection
  N: generation number
Output: selection method
Begin
  For i=1 to *N*
    sample $u \in$ U(0, 1)
    If $u < Pf$ **then**
      choose f as the ranking criteria
    Else
      choose g as the ranking criteria
    EndIf
  EndFor
End

FIGURE 3. Framework of SRBIS.

C. GENERAL MODEL FOR COMPARISON

In the solution selecting part, two CHTs (i.e., Deb's feasibility-based rule and ε constrained method) are adopted in the two phases. Normally, there should be four combinations altogether, i.e., Deb's feasibility-based rule and ε constrained method in the first and second phase, named D-E, D-D, E-E, E-D respectively. In [11], the first three combinations are compared, and this paper will further study the fourth combination (C^2oDE-ED), i.e., ε constrained method in the first phase and Deb's feasibility-based rule in the second phase.

The search algorithm and the framework of general model for comparison (GMC) are shown in Fig.4 and Fig.5 respectively.

Algorithm 2: Search Algorithm

/*DE/current-to-rand/1*/

Select $\vec{x}_{r_1}^t$, $\vec{x}_{r_2}^t$, and $\vec{x}_{r_3}^t$ from the population;
Randomly choose a F value from F_{pool};
$\vec{v}_{i1}^t = \vec{x}_i^t + rand \cdot (\vec{x}_{r_1}^t - \vec{x}_i^t) + F \cdot (\vec{x}_{r_2}^t - \vec{x}_{r_3}^t)$;
$\vec{u}_{i1}^t = \vec{v}_{i1}^t$;

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

Select $\vec{x}_{best}^t$ as **Algorithm 1**, $\vec{x}_{r_1}^t$, $\vec{x}_{r_2}^t$, $\vec{x}_{r_3}^t$ and $\vec{x}_{r_4}^t$ from the population;
Randomly choose a F value from F_{pool} and a CR value from CR_{pool};
$\vec{v}_{i2}^t = \vec{x}_{r_1}^t + F \cdot (\vec{x}_{best}^t - \vec{x}_{r_2}^t) + F \cdot (\vec{x}_{r_3}^t - \vec{x}_{r_4}^t)$
Generate $\vec{u}_{i2}^t$ by applying the binomial crossover on $\vec{v}_{i2}^t$ and $\vec{x}_i^t$;

/* DE/current-to-best/1/bin*/

Select $\vec{x}_{best}^t$ as **Algorithm 1**, $\vec{x}_{r_1}^t$, $\vec{x}_{r_2}^t$ from the population;
Randomly choose a F value from F_{pool} and a CR value from CR_{pool};
$\vec{v}_{i3}^t = \vec{x}_i^t + F \cdot (\vec{x}_{best}^t - \vec{x}_i^t) + F \cdot (\vec{x}_{r_1}^t - \vec{x}_{r_2}^t)$
Generate $\vec{u}_{i3}^t$ by applying the binomial crossover on $\vec{v}_{i3}^t$ and $\vec{x}_i^t$;

FIGURE 4. Search algorithm adopted in this paper.

Input: *NP*: the size of population at each generation
  Max_FES: maximum number of function evaluations
Output: $\vec{x}_{best}$: the best solution in the final population
Step 1 Initialization
  Step 1.1 t=0;
  Step 1.2 Randomly generate an initial population $P_0 = \{\vec{x}_{1,0}, \cdots, \vec{x}_{NP,0}\}$.
  Step 1.3 Evaluate the objective function values $f(\vec{x}_{i,0})$, the degree of constraint violations $G(\vec{x}_{i,0})$.
  Step 1.4 *FES*=*NP*.
Step 2 Evolutional model
  Step 2.1 Choose the way of best individual selection **(Algorithm 1)**
  Step 2.2 Update P_t using **Algorithm 2** to create offspring. These *NP* offspring form the offspring population Q_t .
  Step 2.3 Evaluate $f(\vec{x}_{i,t})$, $G(\vec{x}_{i,t})$ ($i = 1, \cdots, NP$).
  Step 2.4 Choose the corresponding CHT.
  Step 2.5 Rank the population and select the best *NP* individuals to constitute the next population P_{t+1} .
  Step 2.7 *FES*=*FES*+*NP*.
Step 3 Set t=t+1.
Step 4 Stopping Criterion: If *FES*⩾*Max_FES*, stop and output the best solution $\vec{x}_{best}$, otherwise go to Step2

FIGURE 5. Framework of General Model for Comparison (GMC).

TABLE 2. Details of the benchmark test functions.

Prob.	n	Type of objective function	ρ	*LI*	*NI*	*LE*	*NE*	*a*	$f(\vec{x}^*)$
g01	13	quadratic	0.0111%	9	0	0	0	6	-15.0000000000
g02	20	nonlinear	99.9971%	0	2	0	0	1	-0.8036191042
g03	10	polynomial	0.0000%	0	0	0	1	1	-1.0005001000
g04	5	quadratic	52.1230%	0	6	0	0	2	-30665.5386717834
g05	4	cubic	0.0000%	2	0	0	3	3	5126.4967140071
g06	2	cubic	0.0066%	0	2	0	0	2	-6961.8138755802
g07	10	quadratic	0.0003%	3	5	0	0	6	24.3062090681
g08	2	nonlinear	0.8560%	0	2	0	0	0	-0.0958250415
g09	7	polynomial	0.5121%	0	4	0	0	2	680.6300573745
g10	8	linear	0.0010%	3	3	0	0	6	7049.2480205286
g11	2	quadratic	0.0000%	0	0	0	1	1	-0.7499000000
g12	3	quadratic	4.7713%	0	1	0	0	0	-1.0000000000
g13	5	nonlinear	0.0000%	0	0	0	3	3	0.0539415140
g14	10	nonlinear	0.0000%	0	0	3	0	3	-47.7648884595
g15	3	quadratic	0.0000%	0	0	1	1	2	961.7150222899
g16	5	nonlinear	0.0204%	4	34	0	0	4	-1.9051552586
g17	6	nonlinear	0.0000%	0	0	0	4	4	8853.5396748065
g18	9	quadratic	0.0000%	0	13	0	0	6	-0.8660254038
g19	15	nonlinear	33.4761%	0	5	0	0	0	32.6555929502
g21	7	linear	0.0000%	0	1	0	5	6	193.7245100700
g23	9	linear	0.0000%	0	2	3	1	6	-400.0551000000
g24	2	linear	79.6556%	0	2	0	0	2	-5.5080132716

TABLE 3. Classification of bechmark functions.

Problem characteristics		Problems
Number of variables	10-20 (High)	g01, g02, g03, g07, g14, g19
	5-9 (Medium)	g04, g09, g10, g13, g16, g17, g18, g21, g23
	2-4 (Low)	g05, g06, g08, g11, g12, g15, g24
Type of objectives	Polynomial	g01, g03, g04, g05, g06, g07, g09, g11, g12, g15, g18
	Nonlinear	g02, g08, g13, g14, g16, g17, g19
	Linear	g10, g21, g23, g24
Type of constraints	Only inequalities	g01, g02, g04, g06, g07, g08, g09, g10, g12, g16, g18, g19, g24
	Only equalities	g03, g11, g13, g14, g15, g17
	Both inequalities and equalities	g05, g21, g23

IV. EXPERIMENTAL STUDY

A. EXPERIMENTAL SETTINGS

As feasible solutions are very difficult to be found in g20 and g22 for most of the algorithms, 22 benchmark functions [23] were used in our experiment. The details of these benchmark functions are reported in Table 2. Here, n is the number of decision variables, $\rho = |F| \big/ |S|$ is the estimated ratio between the feasible region and the search space, *LI, NI, LE, NE* is the number of linear inequality constraints, nonlinear inequality constraints, linear equality constraints and nonlinear equality constraints respectively, a is the number of active constraints at the optimal solution and $f(\vec{x}^*)$ is the objective function value of the best known solution. We also classify these benchmark functions into different groups [9] as shown in Table 3.

The following parameters are the same as in [13]. The population size (NP) is set to 60; the scaling factor pool $Fpool = [0.6, 0.8, 1.0]$, the crossover control parameter pool

TABLE 4. Best individual selection with C^{2}oDE.

Prob. \ Algo.		*f-g* Mean	*f-g* Std	*f-f* Mean	*f-f* Std	*g-g* Mean	*g-g* Std	*g-f* Mean	*g-f* Std
g01	-15	-15	0	-15	0	-15	0	-15	0
g02	-0.803619	**-0.803619**	**2.1672E-08**	**<u>-0.803619</u>**	**<u>2.2022E-03</u>**	**<u>-0.803179</u>**	**<u>2.2022E-03</u>**	**-0.803619**	**4.2612E-09**
g03	-1.0005	-1.0005	0	-1.0005	0	-1.0005	0	-1.0005	1.0E-15
g04	-30665.5387	-30665.5387	0	-30665.5387	0	-30665.5387	0	-30665.5387	0
g05	5126.4967	5126.4967	3.0E-12	5126.4967	3.0E-12	5126.4967	3.0E-12	5126.4967	3.0E-12
g06	-6961.8139	-6961.8139	0	-6961.8139	0	-6961.8139	0	-6961.8139	0
g07	24.3062	24.3062	6.0E-15	24.3062	7.0E-15	24.3062	3.4E-14	24.3062	1.2E-14
g08	-0.09582504	-0.09582504	0	**<u>-0.09308255</u>**	**<u>1.3712E-02</u>**	-0.09582504	0	-0.09582504	0
g09	680.630057	680.630057	3.0E-13	680.630057	3.0E-13	680.630057	3.0E-13	680.630057	3.0E-13
g10	7049.2480	7049.2480	3.0E-12	7049.2480	4.0E-12	7049.2480	1.912E-09	7049.2480	1.48E-11
g11	0.7499	0.7499	0	0.7499	0	0.7499	0	0.7499	0
g12	-1.0000	-1.0000	0	-1.0000	0	-1.0000	0	-1.0000	0
g13	0.05394151	0.05394151	0	0.05394151	0	0.05394151	0	0.05394151	0
g14	-47.764888	-47.764888	2.9E-14	-47.764888	2.9E-14	-47.764888	2.7E-14	-47.764888	2.7E-14
g15	961.715022	961.715022	6.0E-13	961.715022	6.0E-13	961.715022	6.0E-13	961.715022	6.0E-13
g16	-1.905155	-1.905155	0	-1.905155	0	-1.905155	0	-1.905155	0
g17	8853.533875	**<u>8854.733385</u>**	**<u>4.7997E+00</u>**	**8853.533875**	**1.0E-15**	**<u>8901.049928</u>**	**<u>2.88E+01</u>**	**<u>8884.080093</u>**	**<u>2.0945E+01</u>**
g18	-0.86602540	**-0.86602540**	**7.4470E-11**	**-0.86602540**	**1.5657E-11**	-0.86602358	3.1462E-06	-0.86602404	1.9091E-06
g19	32.655593	32.655593	2.0208E-10	32.655593	2.0781E-10	32.655593	1.0966E-07	32.655593	3.2346E-08
g21	193.724510	**193.724510**	**3.42E-11**	**193.724510**	**3.89E-11**	**<u>219.920176</u>**	**<u>5.3472E+01</u>**	**<u>214.681043</u>**	**<u>4.9007E+01</u>**
g23	-400.0551	**-400.0551**	**6.0E-13**	**-400.0551**	**4E-15**	-400.0551	4.8632E-09	**<u>-340.0534</u>**	**<u>1.2248E+02</u>**
g24	-5.5080	-5.5080	1.0E-15	-5.5080	1.0E-15	-5.5080	1.0E-15	-5.5080	1.0E-15

TABLE 5. Best individual selection with C^{2}oDE-ED.

Prob. \ Algo.		***f-g*** **Mean**	***f-g*** **Std**	*f-f* Mean	*f-f* Std	*g-g* Mean	*g-g* Std	*g-f* Mean	*g-f* Std
g02	-0.803619	**<u>-0.803402</u>**	1.0874E-03	-0.803619	7.2262E-09	-0.803619	5.5374E-09	-0.803619	7.7281E-09
g03	-1.0005	-1.0005	0	-1.0005	0	**<u>-0.99028</u>**	3.6738E-02	-1.0005	1.0E-15
g11	0.7499	0.7499	0	0.7499	0	0.7499	0	**<u>0.7514</u>**	7.5537E-03
g13	0.05394151	**<u>0.06992907</u>**	7.6896E-02	**0.05394151**	0	**<u>0.15257549</u>**	1.6787E-01	**<u>0.24177465</u>**	1.9472E-01
g17	8853.533875	8853.533875	1.0E-12	8853.533875	4.0294E-08	8853.533875	6.16E-10	8853.533875	1.0E-12
g18	-0.86602540	-0.86602540	1.402E-12	-0.86602540	1.9770E-12	**<u>-0.86602512</u>**	3.2039E-07	**<u>-0.86602523</u>**	1.7112E-07
g19	32.655593	32.655593	7.05E-13	32.655593	3.36E-13	32.655593	1.7896E-09	32.655593	1.2887E-09
g21	193.724510	193.724510	8.16E-11	193.724510	2.48E-11	**<u>209.441910</u>**	4.3441E+01	**<u>209.4419</u>**	4.3441E+01
g23	-400.0551	-400.0551	5.1964E-09	-400.0551	3.40E-11	**<u>-388.0548</u>**	6.0E+01	-400.0551	5.6782E-09

$CRpool = [0.1, 0.2, 1.0]$. p in the ε constrained method was set to 0.5, and μ in the restart scheme was set to 10^{-8}.

B. EXPERIMENTAL RESULTS

Twenty-five independent runs were performed for each test function using 5×10^5 FES at maximum, as suggested by Liang *et al.* [23]. Additionally, the tolerance value δ for the equality constraints was set to 0.0001.

1) COMPARISON OF BEST INDIVIDUAL SELECTION

In this part, the comparison will be made based on C^{2}oDE and C^{2}oDE-ED.

a: COMPARISON BASED ON C^{2}oDE

Table 4 lists the result of different best individual selection combinations on TR2006. In all the tables in this paper, bold numbers mean that the obtained results are optimal and much better than other methods, and bold underlined numbers mean that the results are worse than other methods. If the numbers are all optimal or similar, these numbers will not be in bold or underlined.

From Table 4, it can be seen that the four combinations shows similar performance on 16 benchmark functions out of 22.

TABLE 6. Different combinations of CHT with C^2oDE.

Prob. \ Algo.		C^2oDE		E-D		E-E		D-D	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
g02	-0.803619	**-0.803619**	2.1672E-08	**-0.803402**	1.0874E-03	**-0.803179**	2.2022E-03	**-0.803619**	7.5244E-09
g03	-1.0005	-1.0005	0	-1.0005	0	-1.0005	0	**-0.83119**	1.9959E-01
g11	0.7499	0.7499	0	0.7499	0	0.7499	0	**0.9134**	1.1899E-01
g13	0.05394151	**0.05394151**	0	0.06992907	7.6896E-02	**0.05394151**	**0**	**0.15712899**	1.7543E-01
g17	8853.533875	**8854.733385**	4.7997E+00	**8853.533875**	1.0E-12	**8853.533875**	**1.0E-12**	**8871.368857**	3.2393E+01
g18	-0.86602540	-0.86602540	7.4470E-11	-0.86602540	1.402E-12	-0.86602540	6.4917E-11	-0.86602540	0
g19	32.655593	32.655593	2.0208E-10	32.655593	7.05E-13	32.655593	3.5030E-09	32.655593	1.6E-14
g21	193.724510	**193.724510**	3.42E-11	**193.724510**	8.16E-11	**198.963643**	2.6196E+01	**324.70284**	1.718E-10

TABLE 7. Different SR with C^2oDE.

Prob. \ Algo.		C^2oDE		SR(Pc=0.475 with 1 phase)		SRBIS(Pc=0.475)	
		Mean	Std	Mean	Std	Mean	Std
g02	-0.803619	**-0.803619**	2.1672E-08	-0.725357	5.5594E-02	**-0.803619**	6.9843E-09
g10	7049.2480	7049.2480	3.0E-12	7049.2480	8.0570E-05	7049.2480	3.0E-12
g11	0.7499	**0.7499**	0	0.9199	1.1907E-01	**0.7499**	0
g13	0.05394151	**0.05394151**	0	0.08473040	1.0656E-01	**0.05394151**	0
g17	8853.533875	8854.733385	4.7997E+00	**8853.548483**	**7.3044E-02**	8857.355762	1.4935E+01
g18	-0.86602540	**-0.86602540**	7.4470E-11	-0.85138438	7.3205E-02	**-0.86602540**	1.8742E-08
g19	32.655593	**32.655593**	2.0208E-10	32.655598	8.0716E-06	**32.655593**	2.9059E-09
g21	193.724510	198.963643	2.6196E+01	303.746308	4.9008E+01	**193.724510**	3.14E-11
g23	-400.0551	**-400.0551**	6.0E-13	-323.4952	1.2864E+02	**-400.0551**	2.75E-11

Among the other 6 benchmark functions, *f-g* shows a relatively worse performance on g17; *f-f* shows worse performance on g02 and g08; *g-g* can not get a good performance on g02, g17, g18 and g21; *g-f* can not get a good performance on g17, g21 and g23.

Considering the problem characteristics, the selection criteria through *g* in *DE/rand-to-best* and *DE/current-to-best* can better solve the problem with nonlinear objectives, and only inequalities; the selection criteria through *f* in *DE/current-to-best* can solve the problems with only inequality constraints.

b: COMPARISON BASED ON C^2oDE-ED

To verify whether the conclusion can be used with different CHTs, we run another four tests, as in Table 5. From the result, we can see the situation is similar but more different benchmark functions are added.

For page limited, the same or similar results are omitted from this section.

And the result is quite different. As to *f-g*, g17 and g21 shows a good performance, but g02 and g13 shows a worse performance. As to *f-f*, it shows quite a good performance on all functions. **It should be pointed out that this is the best cooperation way (i.e., the *f* criteria in the best individual selection, ε constrained method in the first phase and Deb's feasibility-based rule in the second phase), even better than C^2oDE.**

This also reflects the importance of the cooperation between EAs and CHTs. As mentioned in C^2oDE [13], the feasibility rule in the second phase might discard an individual with promising objective function value selected by the ε constrained method in the first phase, and this would make the population bias toward constraints ultimately. But if we adopt the f criteria in the best individual selection, the shortcoming will be compensated, and the result is quite competitive.

As to *g-g*, besides g18 and g21, it shows a worse performance on g03, g13, and g23; as to *g-f*, besides g18 and g21, g11 and g13 shows up instead of g17 and g23.

2) COMPARISON OF DIFFERENT CHTs WITH C^2oDE

In this part, four different combinations (i.e., D-E, D-D, E-E, and E-D) were tested. The same search algorithm is adopted. The results are shown in Table 6.

From the table, we can see the result shows a little difference with that of the paper [13]. D-E shows a worse performance on g17 and g21, with g02 and g13 in E-D. The benchmark functions with E-E are g02 and g21, while more functions with D-D, as g03, g11, g13, g17 and g21.

TABLE 8. Results of SR with different phases.

Prob. \ Algo.		SR/Pc=0.475-2 phases		SR/Pc=0.475-1 phase	
		Mean	Std	Mean	Std
g02	-0.803619	-0.732981	5.9811E-02	-0.725357	5.5594E-02
g10	7049.2480	7049.2480	1.4788E-04	7049.2480	8.0570E-05
g11	0.7499	0.8999	1.2505E-01	0.9199	1.1907E-01
g13	0.05394151	0.11551928	1.4400E-01	0.08473040	1.0656E-01
g17	8853.533875	8856.509627	1.4809E+01	**8853.548483**	**7.3044E-02**
g18	-0.86602540	-0.85074189	5.2898E-02	-0.85138438	7.3205E-02
g19	32.655593	32.655601	2.8976E-05	32.655598	8.0716E-06
g21	193.724510	319.463708	2.6196E+01	303.746308	4.9008E+01
g23	-400.0551	-359.4534	9.8221E+01	-323.4952	1.2864E+02

3) COMPARISON OF STOCHASTIC RANKING (SR) IN EA AND CHT

To compare the effect of the same method in EA and CHT, we choose SR as an example, i.e., SRBIS and SR respectively. The results are shown in Table 7.

From the table, we can see that SR in best individual choosing obtains much better results than that in CHT, which implies that the **solution generating is a bit more important that in solution choosing with the same condition**.

4) INVESTIGATION ON TWO PHASES ON THE SAME CHT

To further verify whether two phases are necessary for the same CHT, we run SR as an example. As there is no ranking difference in Deb's feasibility-based rule and ε constrained method with one phase or two phases, we did not test these two cases. In fact, the discussion of two phases is the same as whether the parent and the offspring are stored in one pool or not.

The results are shown in Table 8.

From the result, it can be seen that there is not much difference between these two manners. SR with one pool shows a relative better performance on g17.

We should also point out that the parameter *pf* in SR is really important. If *pf* is set as 0.5, no feasible solutions will be found in all the functions except g08 and g12, the two functions whose best objective function value can be found with any penalty parameter in penalty function method [24].

V. CONCLUSION

This paper further studies the cooperation between EAs and CHTs based on a well designed algorithm C^{2}oDE. Firstly, different ways of selecting the best individual in DE mutation operators are compared on TR2006. SR is employed to balance the biases of fixed ways, which shows a good performance. In the solution generating phases, effects on different combinations of CHTs are also studied. Specially, C^{2}oDE-ED is added. Through experiments on the effect of different factors, the importance of cooperation between EA and CHTs are emphasized.

In the future, more functions will be used to check the cooperation, and more parameter tuning will be analyzed with some significance tests [25], and some more algorithms can be added [26]. Specially, the problem characteristics will be summarized in a more reasonable way, which will be the base for problem solving and new algorithm design.

ACKNOWLEDGMENT

The authors are very grateful to Prof. Wang Yong, for providing the code of C^{2}oDE.

REFERENCES

[1] A. Kumar, G. Wu, M. Z. Ali, R. Mallipeddi, P. N. Suganthan, and S. Das, "A test-suite of non-convex constrained optimization problems from the real-world and some baseline results," *Swarm Evol. Comput.*, vol. 56, pp. 1–27, 2020, doi: 10.1016/j.swevo.2020.100693.

[2] Z. Wang, Y. Fu, C. Song, P. Zeng, and L. Qiao, "Power system anomaly detection based on OCSVM optimized by improved particle swarm optimization," *IEEE Access*, vol. 7, pp. 181580–181588, 2019, doi: 10.1109/ACCESS.2019.2959699.

[3] E. Mezura-Montes and C. A. Coello Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm Evol. Comput.*, vol. 1, no. 4, pp. 173–194, Dec. 2011.

[4] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 2, pp. 210–224, Apr. 2012.

[5] B. Xu, H. Zhang, M. Zhang, and L. Liu, "Differential evolution using cooperative ranking-based mutation operators for constrained optimization," *Swarm Evol. Comput.*, vol. 49, pp. 206–219, Sep. 2019.

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

[7] J. Sun, D. Gong, X. Zeng, and N. Geng, "An ensemble framework for assessing solutions of interval programming problems," *Inf. Sci.*, vols. 436–437, pp. 146–161, Apr. 2018.

[8] E. Tsang and A. Kwan, "Mapping constraint satisfaction problems to algorithms and heuristics," Univ. Essex, Colchester, U.K., Tech. Rep. CSM-198, 1993.

[9] C. Si, J. Hu, T. Lan, L. Wang, and Q. Wu, "A combined constraint handling framework: An empirical study," *Memetic Comput.*, vol. 9, no. 1, pp. 69–88, Mar. 2017.

[10] E. Mezura-Montes, M. E. Miranda-Varela, and R. del Carmen Gómez-Ramón, "Differential evolution in constrained numerical optimization: An empirical study," *Inf. Sci.*, vol. 180, no. 22, pp. 4223–4262, Nov. 2010.

[11] J.-P. Li, Y. Wang, S. Yang, and Z. Cai, "A comparative study of constraint-handling techniques in evolutionary constrained multiobjective optimization," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jul. 2016, pp. 4175–4182.

[12] S. Kukkonen and E. Mezura-Montes, "An experimental comparison of two constraint handling approaches used with differential evolution," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Jun. 2017, pp. 2691–2697.

[13] B.-C. Wang, H.-X. Li, J.-P. Li, and Y. Wang, "Composite differential evolution for constrained evolutionary optimization," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 49, no. 7, pp. 1482–1495, Jul. 2019.

[14] Y. Wang, Z. Cai, and Q. Zhang, "Differential evolution with composite trial vector generation strategies and control parameters," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 55–66, Feb. 2011.

[15] X.-S. Yang, "Nature-inspired optimization algorithms: Challenges and open problems," *J. Comput. Sci.*, vol. 2020, Mar. 2020, Art. no. 101104, doi: 10.1016/j.jocs.2020.101104.

[16] C. Si, J. Shen, W. Guo, and L. Wang, "On the cooperation between evolutionary algorithms and constraint handling techniques," in *Advances in Swarm Intelligence* (Lecture Notes in Computer Science), vol. 10941. Cham, Switzerland: Springer, 2018, pp. 42–50.

[17] R. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," UC Berkeley, Berkeley, CA, USA, Tech. Rep. TR-95-012, 1995.

[18] S. Das and P. N. Suganthan, "Differential evolution: A survey of the State-of-the-Art," *IEEE Trans. Evol. Comput.*, vol. 15, no. 1, pp. 4–31, Feb. 2011.

[19] S. Das, S. S. Mullick, and P. N. Suganthan, "Recent advances in differential evolution—An updated survey," *Swarm Evol. Comput.*, vol. 27, pp. 1–30, Apr. 2016.

[20] K. Deb, "An efficient constraint handling method for genetic algorithms," *Comput. Methods Appl. Mech. Eng.*, vol. 186, nos. 2–4, pp. 311–338, Jun. 2000.
[21] T. Takahama and S. Sakai, "Constrained optimization by the μ constrained differential evolution with gradient-based mutation and feasible elites," in *Proc. IEEE Int. Conf. Evol. Comput.*, Mar. 2006, pp. 1–8.
[22] T. Takahama and S. Sakai, "Efficient constrained optimization by the ε Constrained rank-based differential evolution," in *Proc. CEC*, 2012, pp. 1–8.
[23] J. J. Liang, T. P. Runarsson, E. Mezura-Montes, M. Clerc, P. N. Suganthan, C. A. Coello, and K. Deb, "Problem defiitions and evaluation criteria for the CEC 2006," Special Session on Constrained Real-Parameter Optim., Nanyang Technol. Univ., Singapore, Tech. Rep., 2006.
[24] C. Si, J. Shen, X. Zou, and L. Wang, "Comparison of differential evolution algorithms on the mapping between problems and penalty parameters," in *Advances in Swarm and Computational Intelligence* (Lecture Notes in Computer Science), vol. 10385. Cham, Switzerland: Springer, 2017, pp. 420–428.
[25] J. Carrasco, S. García, M. M. Rueda, S. Das, and F. Herrera, "Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review," *Swarm Evol. Comput.*, vol. 54, May 2020, Art. no. 100665.
[26] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Comput. Ind. Eng.*, vol. 137, Nov. 2019, Art. no. 106040.

HUI CAI was born in Zhenjiang, Jiangsu, China, in 1983. She received the bachelor's degree majoring in finance and investment from the Joint Program of the University of Shanghai for Science and Technology, and Queens College, City University of New York, in 2006, and the master's degree majoring in foreign languages and applied linguistics and the Ph.D. degree in management science and engineering from the University of Shanghai for Science and Technology, Shanghai, China, in 2010.

She is currently the Deputy Director of the International Office, University of Shanghai for Science and Technology. She has been in charge of an Elite Youth Teacher Scientific Project from Shanghai Municipal Education Commission. Besides, she has taken part in the Research on Optimization Strategy and Allocation Efficiency of Intellectual Resources in Creative Industry from Ministry of Education of People's Republic of China.

BIN LIU was born in Yichang, Hubei, in 1980. He received the bachelor's degree in material engineering from Central South University, the master's degree in finance from Fudan University, and the Ph.D. degree from the School of Finance and Economics, Jiangsu University. He is currently working at CITIC Group. His research interests include the Internet finance and risk management.

SHU PAN was born in Zhenjiang, Jiangsu, China, in 1981. She received the bachelor's degree in computer science and technology and the master's degree in computer applications from Jiangsu University of Science and Technology, in 2003 and 2008, respectively.

She is currently a Lecturer at the Computer Science School, Jiangsu University of Science and Technology.