Memetic Comp. (2017) 9:69–88
DOI 10.1007/s12293-016-0221-2

REGULAR RESEARCH PAPER

A combined constraint handling framework: an empirical study

Chengyong Si[1] · Junjie Hu[2] · Tian Lan[3] · Lei Wang[4] · Qidi Wu[4]

Received: 19 December 2015 / Accepted: 30 December 2016 / Published online: 9 January 2017

Abstract This paper presents a new combined constraint handling framework (CCHF) for solving constrained optimization problems (COPs). The framework combines promising aspects of different constraint handling techniques (CHTs) in different situations with consideration of problem characteristics. In order to realize the framework, the features of two popular used CHTs (i.e., Deb's feasibility-based rule and multi-objective optimization technique) are firstly studied based on their relationship with penalty function method. And then, a general relationship between problem characteristics and CHTs in different situations (i.e., infeasible situation, semi-feasible situation, and feasible situation) is empirically obtained. Finally, CCHF is proposed based on the corresponding relationship. Also, for the first time, this paper demonstrates that multi-objective optimization technique essentially can be expressed in the form of penalty function method. As CCHF combines promising aspects of different CHTs, it shows good performance on the 22 well-known benchmark test functions. In general, it is comparable to the other four differential evolution-based approaches and five dynamic or ensemble state-of-the-art approaches for constrained optimization.

Keywords Constrained optimization · Constraint handling techniques · Combined constraint handling framework (CCHF) · Differential evolution · Ranking methods

This work is partially done when Chengyong Si was with the Institute for Electronics Engineering, University of Erlangen-Nuernberg in Germany as a joint doctor.

✉ Chengyong Si
sichengyong@163.com

Junjie Hu
junhu@elektro.dtu.dk

Tian Lan
tianlan@mailbox.tu-berlin.de

Lei Wang
wanglei@tongji.edu.cn

Qidi Wu
qidi@mail.tongji.edu.cn

1 Shanghai-Hamburg College, University of Shanghai for Science and Technology, Shanghai 200093, China

2 Department of Electrical Engineering, Center for Electric Power and Energy, Technical University of Denmark, Kgs. Lyngby 2800, Denmark

3 Institute of Sustainable Electric Networks and Sources of Energy, Technical University of Berlin, 10587 Berlin, Germany

4 College of Electronics and Information Engineering, Tongji University, Shanghai 201804, China

1 Introduction

In the real-world applications, constrained optimization problems (COPs) are very common and important. The COPs can be generally expressed by the following formulations:

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

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

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

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 [1] and can be mainly used to generate solutions. Equivalently, choosing the better solutions among the parent and offspring populations, especially for the COPs is another important research area in optimization, which leads to the development of various constrained optimization evolutionary algorithms (COEAs) [2–4] The three most frequently used constraint handling techniques (CHTs) in COEAs are based on the concept of penalty functions, biasing feasible over infeasible solutions and multiobjective optimization [3,5–7].

Penalty function method is generic and applicable to any type of constraints. It constructs a fitness evaluation by adding an amount of constraint violation to an objective function. The fine tuning of penalty parameters, which helps to balance the objective and constraint violation, is the key point.

Methods which compare separately the objective functions and constraint violations were also developed. For example, Deb [5] proposed a feasibility-based rule to pairwise 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.
(3) Among two infeasible solutions, the one having smaller constraint violation is preferred.

Meanwhile, multiobjective optimization technique which considers the objective function and constraint violation at the same time has been employed to handle constraints [3,7].

Besides these basic CHTs, some other concepts like cooperative coevolution [8,9] and ensemble [10,11] have also been proposed. These methods employed different subpopulations and evolve parallel. Normally, the population size of these methods changes with the evolution process. Thus, it can be seen as a dynamic adjustment process.

Taking ensemble of constraint handling techniques (ECHT) [11] as an example, it utilizes multiple subpopulations. Each subpopulation corresponds to one CHT with its own offspring. And the parent population of one CHT will compete with all the offspring populations so that every function call can be utilized effectively.

The evolution process will inevitably experiences three different situations in solving COPs [12], and consequently, some dynamic approaches were developed.

Zhang et al. [13] proposed a dynamic stochastic selection (DSS) within the framework of multimember DE (DSS_MDE). Another adaptive penalty formulation was introduced by Tessema and Yen [14].It uses the number of feasible individuals to determine the amount of penalty added to infeasible individuals.

Wang et al. [12] proposed an adaptive tradeoff model (ATM). To satisfy the different requirements in corresponding situations, different tradeoff schemes during different situations of a search process are designed. Based on it, an improved ATM with each constraint violation first normalized was proposed by Wang and Cai [15]. It was combined with $(\mu + \lambda)$-DE to form the framework of $(\mu + \lambda)$-CDE. In this approach, a constraint-handling mechanism is designed in each situation based on the characteristics of the current population. To overcome the drawbacks of $(\mu + \lambda)$-CDE, an improved version of $(\mu + \lambda)$-CDE, named ICDE, was presented by Jia et al. [16]. ICDE consists of an improved $(\mu + \lambda)$-differential evolution (IDE) and a novel archiving-based adaptive tradeoff model (ArATM). Especially, the hierarchical non-dominated individual selection scheme is utilized and an individual archiving technique is proposed to maintain the diversity of the population in the infeasible situation. In the semi-feasible situation, the feasibility proportion of the population is used to convert the objective function of each individual.

Among all of these aforementioned methods, the problem characteristics are rarely considered. But as Michalewicz summarized [17], it seems that Evolutionary Algorithms, in the broad sense of this term, provide just a general framework on how to approach complex problems. All their components, from the initialization, through variation operators and selection methods, to constraint-handling methods, might be problem-specific. From this, we believe it is essential to design a general framework from the aspect of problem characteristics.

Besides, there are already some computational time complexity analyses of EAs [18,19] that emphasize the relationship between algorithmic features and problem characteristics. As Yao presented [20], analyzing the relationship between problem characteristics and algorithmic features will shed light on the essential question of when to use which algorithm in solving a difficult problem instance class. And he introduced EA-hard and EA-easy problem instance classes. These classes are based on the functional relationship between the mean number of generations (i.e., the mean first hitting time) and the problem size (in terms of dimensionality). But as he also pointed out, it is still unclear what the relationship is between the optimization time and the problem size for different EAs on different problems. Though a lot of theoretical analysis was obtained, Yao did not give the specific relationship between algorithms and problems.

Additionally, some researchers have emphasized the importance of the relationship between problem characteristics and algorithms, and have tried to realize some simple combination of algorithm variants, although the results are not satisfactory. For example, Tsang and Kwan [21] pointed out the need to map constraint satisfaction problems to algorithms and heuristics. But they did not give an exact relationship between them. Mezura-Montes et al. [22] 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. Gibbs et al. [23] identified the relationship between the optimal number of GA generations and the problem characteristics, through quantifying different problem characteristics of unconstrained problems.

Recently, some researchers have made some beneficial attempt on the use of information during the evolutionary process, and got some good results. It is noted that, in the course of the information use, the relationship between diversity and convergence, exploration and exploitation should be well handled.

Wang et al. [24] proposed the strategy of incorporating objective function information into the Deb's feasibility-based rule, and achieved an effective balance between constraints and objective function in constrained optimization. The paper also presented some new replacement mechanisms and mutation strategy to better exploit the information of individuals with good objective function values.

Qiu et al. [25] developed some adaptive cross-generation differential evolution operators for multi-objective optimization. This mechanism utilized the swarm information between neighbor generations from the aspect of objective spaces into two mutation operators, so as to achieve the good balance between convergence and diversity. This paper also presented a new parameter adaptation mechanism to self-adapt the individuals' associate parameters.

Elsayed and Sarker [26] presented a general differential evolution framework for big data optimization. Three sub-swarms were employed to correspond to a variant respectively. During the evolutionary process, the performance information of each variant was recorded, and an exponential curve was fitted to predict the future performance of each variant.

Feng et al. [27] proposed an evolutionary memetic search, which can learn and evolve knowledge meme across different but related problem domains. It was realized on two combination optimization problems, capacitated vehicle routing problem (CVRP) and capacitated arc routing problem (CARP).

Other methods concerning the problem characteristics were also reported [28]. As presented in [28], a method to construct the relationship between problems and algorithms as well as constraint handling techniques from the qualitative and quantitative point of view was proposed. In the paper, the problem characteristics were also summarized systematically.

Unlike the aforementioned methods, in this work, we try to study the features of different CHTs (i.e., penalty function method, multiobjecitve optimization technique and Deb's feasibility-based rule) and get the corresponding relationship between problem characteristics and CHTs in different situations. A combined constraint handling framework (CCHF) is proposed based on the corresponding relationship.

The rest of this paper is organized as follows. Section 2 briefly introduces DE. Section 3 gives the detail analysis of the relationship among three CHTs (i.e., multi-objective optimization technique and penalty function method, Deb's feasibility-bases rule and penalty function method). The comparison of different CHTs in different situations is analyzed in Sect. 4. Based on this, Sect. 5 presents a detailed description of the proposed CCHF. The experimental results and the comparison with some state-of-the-art methods are given in Sect. 6. Finally, Sect. 7 concludes this paper and provides some possible paths for future research.

2 Differential evolution (DE)

DE, which was proposed by Storn and Price, is a simple and efficient EA. The mutation, crossover and selection operations are introduced in DE. The first two operations are used to generate a trial vector to compete with the target vector while the third one is used to choose the better one for the next generation. Several variants of DE have been proposed [29]. *DE/rand/1/bin* was adopted in this paper as the search algorithm.

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

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

The three operations are defined as follows.

2.1 Mutation operation

Taking into account each individual $\vec{x}_i$ (named a target vector), a mutant vector $\vec{v}_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,n}\}$ is defined as

$$\vec{v}_i = \vec{x}_{r1} + F \cdot (\vec{x}_{r2} - \vec{x}_{r3}) \tag{3}$$

where $r1$, $r2$ and $r3$ are randomly selected from $[1, NP]$ and satisfying: $r1 \neq r2 \neq r3 \neq i$ and F is the scaling factor.

In this paper, if $v_{i,j}$ violates the boundary constraint, it will be reset as follows [9]:

$$v_{i,j} = \begin{cases} \min\left\{U_j, 2L_j - v_{i,j}\right\}, & if\ v_{i,j} < L_j \\ \max\left\{L_j, 2U_j - v_{i,j}\right\}, & if\ v_{i,j} > U_j \end{cases} \tag{4}$$

2.2 Crossover operation

A trial vector $\vec{u}_i$ is generated through the crossover operation on the target vector $\vec{x}_i$ and the mutant vector $\vec{v}_i$

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand_j \leq C_r \quad \text{or } j = j_{rand} \\ x_{i,j} & \text{otherwise} \end{cases} \tag{5}$$

where $i = 1, 2, \ldots, NP$, $j = 1, 2, \ldots, n$, 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. The introduction of $j = j_{rand}$ can guarantee the trial vector $\vec{u}_i$ is different from its target vector $\vec{x}_i$.

2.3 Selection operation

Selection operation is realized by comparing the trial vector $\vec{u}_i$ against the target vector $\vec{x}_i$ and the better one will be preserved for the next generation.

$$\vec{x}_i = \begin{cases} \vec{u}_i & \text{if } f(\vec{u}_i) \leq f(\vec{x}_i) \\ \vec{x}_i & \text{otherwise} \end{cases} \tag{6}$$

3 Systematic analysis of CHTs

3.1 Definitions

Unlike the single optimization solution in single-objective optimization problem, there are often a set of the optimization solutions in multiobjective optimization problem. Thereby it is necessary to introduce some essential definitions regarding the multiobjective optimization [30]. These definitions are mostly from the aspect of variables.

Definition 1 (*Pareto dominance*) A multiobjective minimization problem with m decision variables (parameters) and n objectives can be formulated as follows:

$$\begin{aligned} \text{Minimize} \quad & \vec{y} = \vec{f}(\vec{x}) = (f_1(\vec{x}), \ldots, f_n(\vec{x})) \\ \text{where} \quad & \vec{x} = (x_1, \ldots, x_m) \in X \\ & \vec{y} = (y_1, \ldots, y_n) \in Y \end{aligned} \tag{7}$$

and where $\vec{x}$ is decision vector, X is parameter space, $\vec{y}$ is objective vector, and Y is objective space. A decision vector $\vec{a} \in X$ is said to dominate a decision vector $\vec{b} \in X$, denoted as $\vec{a} \prec \vec{b}$, if and only if

$$\begin{aligned} & \forall i \in \{1, \ldots, n\}, f_i(\vec{a}) \leq f_i\left(\vec{b}\right) \quad \text{and} \\ & \exists j \in \{1, \ldots, n\}, f_j(\vec{a}) < f_j\left(\vec{b}\right) \end{aligned} \tag{8}$$

Definition 2 (*Pareto optimality*) The decision vector $\vec{a}$ is said to be nondominated regarding a set $X' \subseteq X$ if and only if there is no vector in X' which dominates $\vec{a}$, as

$$\neg\exists \vec{a}' \in X', \vec{a}' \prec \vec{a} \tag{9}$$

Besides, the decision vector $\vec{a}$ is Pareto-optimal if and only if $\vec{a}$ is nondominated regarding X.

Definition 3 (*Pareto optimal set*) The set X' is called a global Pareto-optimal set if and only if $\forall \vec{a}' \in X', \neg\exists \vec{a} \in X, \vec{a} \prec \vec{a}'$. We can define it as

$$\rho^* = \left\{\vec{a}' \in X' \,\middle|\, \neg\exists \vec{a} \in X, \vec{a} \prec \vec{a}'\right\} \tag{10}$$

The Pareto optimal set is a set of parameters and the corresponding set of objective vectors is denoted as "Pareto-optimal front".

3.2 Systematic analysis of penalty function method and multiobjective optimization technique

As the aforementioned definitions in multiobjective optimization technique can be described in the form of penalty function method, the relationship between them can be analyzed as follows.

For the given $\lambda, \delta > 0$, let the evaluation function L in the penalty function method as

$$L(\vec{x}_i, \lambda, \delta) = f(\vec{x}_i) + \lambda G(\vec{x}_i, \delta) \quad i = 1, 2, \ldots, NP \tag{11}$$

where $\vec{x}_i$ is the NP n-dimensional real-valued vectors of the population as defined in (2), λ is the penalty parameter, δ is the tolerance value for the equality constraints, f is the objective function and G is the real-valued penalty function.

As δ can be supposed as a constant, the effect of λ is mainly concerned. The formula (11) can be transformed as

$$L(\vec{x}_i, \lambda) = f(\vec{x}_i) + \lambda G(\vec{x}_i) \quad i = 1, 2, \ldots, NP \tag{12}$$

Given two population members, $\vec{x}_s$ and $\vec{x}_t$, where s and t are randomly selected from [1, NP] and satisfying: $s \neq t$, the difference between their evaluation function values is:

$$\begin{aligned} \Delta(\vec{x}_s, \vec{x}_t, \lambda) &= L(\vec{x}_s, \lambda) - L(\vec{x}_t, \lambda) \\ &= \left[f(\vec{x}_s) + \lambda G(\vec{x}_s)\right] - \left[f(\vec{x}_t) + \lambda G(\vec{x}_t)\right] \\ &= \left[f(\vec{x}_s) - f(\vec{x}_t)\right] + \lambda\left[G(\vec{x}_s) - G(\vec{x}_t)\right] \end{aligned} \tag{13}$$

We define $\Delta f_{st} = f(\vec{x}_s) - f(\vec{x}_t)$, $\Delta G_{st} = G(\vec{x}_s) - G(\vec{x}_t)$, then formula (13) can be written as:

$$\Delta\left(\vec{x}_s, \vec{x}_t, \lambda\right) = \Delta f_{st} + \lambda \cdot \Delta G_{st} \quad (14)$$

According to the definition, $\Delta f_{st} \neq \pm\infty$ and $\Delta G_{st} \neq \pm\infty$.

The multiobjective optimization technique converts a COP into a biobjective or multiobjective optimization problem, for the sake of clarity, let $\vec{f}(\vec{x}) = (\vec{f}_1(\vec{x}), \vec{f}_2(\vec{x})) = (f(\vec{x}), G(\vec{x}))$

By employing the aforementioned definitions and the form of penalty function, we obtain the following conclusions:

Theorem 1 $\vec{x}_s \prec \vec{x}_t \Leftrightarrow \forall\lambda > 0, L\left(\vec{x}_s, \lambda\right) < L\left(\vec{x}_t, \lambda\right)$.

Proof As mentioned above, $\forall\lambda > 0, L(\vec{x}_s, \lambda) < L(\vec{x}_t, \lambda)$ is equal to $\forall\lambda > 0, \Delta f_{st} + \lambda \cdot \Delta G_{st} < 0$.

The sufficient condition can be easily proved.

From the definition of dominance, if$\vec{x}_s \prec \vec{x}_t$, then $\Delta f_{st} \leq 0$, $\Delta G_{st} \leq 0$, and Δf_{st} and ΔG_{st} can't be equal to 0 simultaneously.

The conclusion $\forall\lambda > 0, \Delta f_{st} + \lambda \cdot \Delta G_{st} < 0$ is obtained.

Next, we prove the necessary condition.

From $\forall\lambda > 0, \Delta f_{st} + \lambda \cdot \Delta G_{st} < 0$, we can conclude:

$$\Delta G_{st} < -\frac{\Delta f_{st}}{\lambda} \quad (15)$$

Let us define $A = -\frac{\Delta f_{st}}{\lambda}$, then $\Delta G_{st} < A$.

For the different properties of Δf_{st}, there are three different cases.

(1) $\Delta f_{st} > 0$: in this case, $A < 0$, $\Delta G_{st} < A < 0$. When $\lambda \to 0^+$, then $A \to -\infty$, and $\Delta G_{st} = -\infty$, which contradicts the previous assumption.
(2) $\Delta f_{st} = 0$: in this case, A=0, $\Delta G_{st} < 0$.
3) $\Delta f_{st} < 0$: in this case, $A > 0$, $\Delta G_{st} < A$. When $\lambda \to +\infty$, then $A \to 0^+$, and $\Delta G_{st} \leq 0$,

In general, $\Delta f_{st} \leq 0, \Delta G_{st} \leq 0$, and Δf_{st} and ΔG_{st} are not equal to 0 simultaneously. So the conclusion $\vec{x}_s \prec \vec{x}_t$ is obtained.

Likewise, we can get two related theorems as follows.

Theorem 2 $\vec{x}_s$ *nondominates* $\vec{x}_t \Leftrightarrow \exists\lambda > 0, L(\vec{x}_s, \lambda) \geq L(\vec{x}_t, \lambda)$.

Proof As mentioned above, $\exists\lambda > 0, L(\vec{x}_s, \lambda) \geq L(\vec{x}_t, \lambda)$ is equal to $\exists\lambda > 0, \Delta f_{st} + \lambda \cdot \Delta G_{st} \geq 0$

$\vec{x}_s$ nondominates $\vec{x}_t$

$\Leftrightarrow \exists i, f_i(\vec{x}_s) > f_i(\vec{x}_t)$ or $\forall i, f_i(\vec{x}_s) = f_i(\vec{x}_t)$ (where $i = 1, 2$) $\Leftrightarrow f(\vec{x}_s) > f(\vec{x}_t)$ or $G(\vec{x}_s) > G(\vec{x}_t)$, or $f(\vec{x}_s) = f(\vec{x}_t)$ and $G(\vec{x}_s) = G(\vec{x}_t)$.

Let us first prove the sufficient condition.

From the definition of nondominated, if $\vec{x}_s$ nondominates $\vec{x}_t$, then $\Delta f_{st} > 0$, or$\Delta G_{st} > 0$, or $\Delta f_{st} = \Delta G_{st} = 0$.

Then four cases are listed as follows.

(1) $\Delta f_{st} > 0, \Delta G_{st} > 0$: in this case, $\Delta f_{st} + \lambda \cdot \Delta G_{st} \geq 0$ holds for $\forall\lambda > 0$.
(2) $\Delta f_{st} > 0, \Delta G_{st} \leq 0$: if $\Delta G_{st} = 0$, then the inequality $\Delta f_{st} + \lambda \cdot \Delta G_{st} \geq 0$ holds; else suppose$\frac{\Delta f_{st}}{-\Delta G_{st}} = \eta \geq \lambda > 0$, then the inequality holds.
(3) $\Delta f_{st} \leq 0, \Delta G_{st} > 0$: suppose $\frac{-\Delta f_{st}}{\Delta G_{st}} = \eta \geq \lambda > 0$, then the inequality holds.
(4) $\Delta f_{st} = \Delta G_{st} = 0$: the conclusion $\Delta f_{st} + \lambda \cdot \Delta G_{st} \geq 0$is obtained.

Next, we prove the necessary condition.

The main aim is to find out the relationship of Δf_{st} and ΔG_{st} under the conditions.

For the different properties of ΔG_{st}, there are three different cases.

(1) $\Delta G_{st} > 0$: in this case, $\lambda \geq -\frac{\Delta f_{st}}{\Delta G_{st}} = \eta$ holds for any Δf_{st}, and in this situation, $\vec{x}_s$ nondominates $\vec{x}_t$ (as cases 1 and 3 in the previous part);
(2) $\Delta G_{st} = 0$: in this case,$\Delta f_{st} \geq 0$, then $\vec{x}_s$ nondominates $\vec{x}_t$ (as cases 2 and 4 in the previous part);
(3) $\Delta G_{st} < 0$: in this case, $\lambda \leq -\frac{\Delta f_{st}}{\Delta G_{st}} = \eta, \Delta f_{st} > 0$, then $\vec{x}_s$ nondominates $\vec{x}_t$ (as case 2 in the previous part).

In general, $\Delta G_{st} > 0$, or $\Delta G_{st} = 0$ and $\Delta f_{st} \geq 0$, or $\Delta G_{st} < 0$ and $\Delta f_{st} > 0$. So the conclusion $\vec{x}_s$ nondominates $\vec{x}_t$ is obtained.

To expand the individuals to a set, then Theorem 3 is obtained.

Theorem 3

$$\begin{aligned}\rho^* &= \left\{\vec{x}_s \in X' | \neg\exists\vec{x}_t \in X, \vec{x}_t \prec \vec{x}_s\right\} \\ &\Leftrightarrow \left\{\vec{x}_s \in X' | \forall\vec{x}_t \in X, \exists\lambda > 0, L\left(\vec{x}_t, \lambda\right) \geq L\left(\vec{x}_s, \lambda\right)\right\}\end{aligned}$$

3.3 Systematic analysis of penalty function method and Deb's feasibility-based rule

As analyzed in [31], Deb's feasibility-based rule corresponds to one special case of penalty function method when penalty parameter is large enough (i.e., larger than $\lambda_{\max}$) for the following reason:

(1) For the feasible situation, both methods have the same effect on ranking due to the fact that only objective function values are used for ranking.
(2) For the infeasible and semi-feasible situations, when $\lambda > \lambda_{\max}$, the two methods have the same effect on ranking the whole population. While, when $\lambda < \lambda_{\max}$, these two methods present different effect on ranking. Here, $\lambda_{\max}$ is determined by the current solutions.

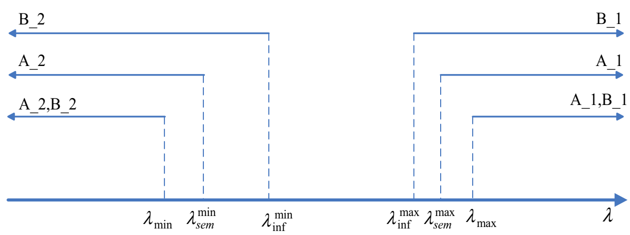

Fig. 1 The corresponding rule for penalty parameter λ

The general results can be illustrated in Fig. 1, where A_1, A_2, B_1, B_2 are four rules for comparing feasible and infeasible solutions. A_1, B_1 stands for Deb's feasibility-based rule.

4 Comparison of different CHTs in different situations

To fully compare the effect of different CHTs on different situations, two experiments, will be carried out. One is under the infeasible situation while the other one is under the semi-feasible situation. All our experiments are based on the benchmark functions in [32]. The details of these benchmark functions and the classifications (which takes some idea from [22]) are presented in Tables 1 and 2 respectively.

To make fair comparison, all CHTs will be compared under the same circumstance (i.e., with the same initial solutions and the same setting of DE).

The average method (which divides the range of each independent variable equally) is adopted to generate the initial solutions. 15 out of 22 benchmark functions are in the infeasible situation. As the rest seven benchmark functions are not enough to analyze the characteristics of CHT in the semi-feasible situation, we adopt the other 15 benchmark functions with the semi-feasible situation. Deb's feasibility-based rule is applied to the 15 benchmark functions to get at least a feasible solution (i.e., semi-feasible situation).

Therefore, 15 and 22 benchmark functions are used in experiment 1 and experiment 2 respectively for comparison. The experimental results are listed in Tables 3 and 4.

Table 1 Details of the benchmark 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	−30,665.5386717834
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	8853.5396748064
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

n is the number of decision variables, $\rho = |F|/|S|$ is the estimated ratio between the feasible region and the search space, *LI*, *NI*, *LE*, *NE* stand for 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.

Table 2 Classification of benchmark functions based on the number of decision variables and the type of objectives and constraints

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

The Deb's feasibility-based rule [5] and multi-objective optimization technique without any variants [12] are used here.

4.1 Comparison under infeasible situation

In this situation, both Deb's feasibility-based rule and multi-objective optimization technique can always find the feasible solutions. This is because that these two methods take the constraint violation as a metric for evaluation (i.e., comparing the constraint violation directly).

Multi-objective optimization technique shows a better performance in g03, g05, g15, g16, g17, g21 and g23 while Deb's feasibility-based rule performs better in g06, g07 and g18. They show similar performance in other functions.

Considering the problem characteristics, it indicates that if equality constraints are involved, multiobjective optimization technique is preferred; otherwise, Deb's feasibility-based rule is preferred.

4.2 Comparison under semi-feasible situation

In this experiment, multiobjective optimization technique performs better than Deb's feasibility-based rule in most test functions, especially in g03, g05, g06, g10 and g14. However, it performs worse than Deb's feasibility-based rule in g08.

All these two CHTs have the same or similar performance in g04, g07, g08, g09, g12, g16, g18, g19 and g24 with the known optimal value reached. It is worthy noting that multiobjective optimization technique needs less fitness evaluations (FES) comparing with Deb's feasibility-based rule.

It also should be pointed out that these two CHTs can not find the optimal solutions in g13, g21 and g23.

Considering the problem characteristics, Deb's feasibility-based rule performs better in solving problems with inequality constraints and the nonlinear objective function's type; multiobjective optimization technique performs better in the other types of problems.

4.3 General conclusion

We can conclude that different CHTs can solve different problems effectively in corresponding situations. These conclusions can be generalized as follows.

(1) For the infeasible situation, Deb's feasibility-based rule and multiobjective optimization technique can effectively solve problems with only inequality constraints and the others respectively.
(2) For the semi-feasible situation, Deb's feasibility-based rule and multiobjective optimization technique can effectively solve problems with inequality constraints with nonlinear objective function and the other types of problems respectively.
(3) For the feasible situation, these CHTs have the same performance as there is no constraint violation considered.

This conclusion forms a good basis for combining promising aspects of different CHTs on different problems into a new approach, as demonstrated in next section.

5 Combined constraint handling framework (CCHF)

As mentioned in Sect. 4, different CHTs have different effects on solving different problems in different situations. Based on this, a generalized CCHF is proposed.

Table 3 Comparison of different CHTs in infeasible situation

Func. and optimal value	FES_D	FES_M	FIT_D	FIT_M
G01 −15.0000				
Best	600	300	−1.1218	−1.7463
Median	1200	1100	0.4490	0.5844
Mean	1104	1060	0.5357	0.3665
Worst	1400	1400	2.3860	2.1513
SD	1.9891E+02	2.3274E+02	8.4033E−01	9.3501E−01
FR	1	1	1	1
G03 −1.0005				
Best	500	700	−0.7240	−0.9728
Median	1800	1400	−0.1414	−0.8385
Mean	2604	1424	−0.1841	−0.7915
Worst	7400	2600	−5.9421E−06	−0.3758
SD	1.9747E+03	4.1960E+02	2.0049E−01	1.5278E−01
FR	1	1	1	1
G05 5126.4967				
Best	12,100	10,000	5127.4556	5126.4978
Median	14,600	11,700	5253.0267	5126.5072
Mean	14,472	11,664	5258.2013	5126.5245
Worst	16,300	13,300	5418.3266	5126.6695
SD	1.0964E+03	8.8265E+02	9.7182E+01	4.1436E−02
FR	1	1	1	1
G06 −6961.8139				
Best	100	100	−6117.3390	−6179.5483
Median	400	500	−6045.4509	−6067.7217
Mean	344	400	−5692.7229	−5729.7016
Worst	600	800	−3271.9841	−2321.7015
SD	1.6093E+02	1.8708E+02	8.7138E+02	8.8309E+02
FR	1	1	1	1
G07 24.3062				
Best	400	700	119.5695	257.6533
Median	1100	1200	553.1970	453.9109
Mean	1120	1212	639.0553	572.2661
Worst	1700	1600	1571.2309	2457.7759
SD	3.1623E+02	2.0478E+02	3.7773E+02	4.3877E+02
FR	1	1	1	1
G08 −0.09582504				
Best	100	100	−0.07046294	−0.08718057
Median	300	300	5.6641E−04	−5.3796E−04
Mean	304	316	−1.8560E−04	−0.00298423
Worst	600	700	0.05896248	0.08705198
SD	1.3064E+02	1.5727E+02	3.1780E−02	2.9881E−02
FR	1	1	1	1
G10 7049.2480				
Best	1700	1600	10,795.6388	11,542.8820
Median	3000	2300	19,881.4967	16,904.5752
Mean	2852	2332	19,842.7470	16,899.4635
Worst	3900	3200	28,330.7877	28,304.7614
SD	5.4705E+02	3.8914E+02	4.4126E+03	3.6109E+03
FR	1	1	1	1

Table 3 continued

Func. and optimal value	FES_D	FES_M	FIT_D	FIT_M
G13 0.05394151				
Best	13,200	13,000	0.59453698	0.43886839
Median	19,200	15,900	0.99759030	0.44112443
Mean	20,264	16,920	0.95363631	0.55068088
Worst	35,400	27,800	0.99999747	0.99149309
SD	5.1288E+03	3.3608E+03	9.8749E−02	2.1020E−01
FR	1	1	1	1
G14 −47.764888				
Best	9100	14,500	−45.900715	−47.480493
Median	11,400	21,700	−42.870229	−46.051264
Mean	112,80	21,148	−43.017299	−45.907825
Worst	12,800	28,500	−40.525691	−42.805479
SD	9.4207E+02	4.3285E+03	1.3108E+00	1.2910E+00
FR	1	1	1	1
G15 961.715022				
Best	6300	5000	961.723166	961.715075
Median	8500	5700	962.273256	961.715279
Mean	8844	5732	962.950058	961.717110
Worst	12,100	6600	968.367416	961.747885
SD	1.4131E+03	3.8914E+02	1.7101E+00	6.6709E−03
FR	1	1	1	1
G16 −1.905155				
Best	500	700	−1.659527	−1.463994
Median	2200	1900	−1.220891	−1.256614
Mean	3500	2540	−1.227879	−1.243692
Worst	11700	8100	−0.795549	−0.912618
SD	3.3637E+03	1.6055E+03	2.0455E−01	1.3708E−01
FR	1	1	1	1
G17 **8853.533875**				
Best	24,500	22,800	8863.244581	8859.048260
Median	28,000	26,500	8949.788386	8866.532519
Mean	27,860	26,772	8956.506847	8892.594486
Worst	33,000	34,500	9165.634865	8963.392570
SD	2.3585E+03	2.4330E+03	6.9404E+01	3.7512E+01
FR	1	1	1	1
G18 −0.86602540				
Best	1400	2500	−0.48211156	−0.48426308
Median	3300	3500	−0.24068045	−0.16692896
Mean	3212	3516	−0.25222633	−0.19867864
Worst	4300	4700	−0.05572148	−0.06870847
SD	7.1956E+02	4.7142E+02	1.1151E−01	9.4618E−02
FR	1	1	1	1
G21 193.724510				
Best	13,500	14,700	300.889263	195.257693
Median	14,500	19,500	643.552798	218.397609
Mean	15,248	19,752	664.071306	241.883079
Worst	20,500	26,200	995.855832	443.999855
SD	2.0486E+03	2.9066E+03	2.0376E+02	5.8805E+01
FR	1	1	1	1

Table 3 continued

Func. and optimal value	FES_D	FES_M	FIT_D	FIT_M
G23 −400.0551				
Best	16,500	1,7700	−45.5844	−201.2285
Median	21,000	23,100	43.9778	−50.3251
Mean	21,264	23,512	59.4635	−54.1843
Worst	28,800	32,500	235.2444	129.4536
SD	3.2183E+03	3.9431E+03	7.3017E+01	9.4735E+01
FR	1	1	1	1

FES_D and FES_M stand for the FES needed for finding a feasible solution with corresponding CHTs while FIT_D and FIT_M stand for the fitness values of the first feasible solution with corresponding CHTs. FR means the feasible rate. "–" means no feasible solutions were found. Values in boldface mean that the obtained result is much better with respect to the CHTs compared

Table 4 Comparison of different CHTs in semi-feasible situation

Func. and optimal value	FES_D	FES_M	FIT_D	FIT_M
G01 −15.0000				
Best	42,900	44,200	−15.0000	−15.0000
Median	46,750	46,900	−15.0000	−15.0000
Mean	46,469	47,628	−14.1050	−14.3306
Worst	49,300	52,100	−12.4531	−12.4531
SD	1.7617E+03	2.0338E+03	1.2225E+00	1.1036E+00
SR	0.64	0.72	0.64	0.72
G02 −0.803619				
Best	59,100	46,400	−0.803619	−0.803619
Median	64,700	60,750	−0.785626	−0.803619
Mean	65,975	60,663	−0.745453	−0.802776
Worst	79,000	70,100	−0.485595	−0.782551
SD	6.2842E+03	5.8958E+03	9.0213E−02	4.2136E−03
SR	0.32	0.96	0.32	0.96
G03 −1.0005				
Best	233,700	40,300	−1.0005	**−1.0005**
Median	297,200	56,200	−0.6745	**−1.0005**
Mean	337,500	57,244	−0.7636	**−1.0005**
Worst	481,600	83,700	−0.6609	**−1.0005**
SD	1.2877E+05	1.0318E+04	1.2959E−01	**4.2276E−16**
SR	0.12	**1**	0.12	**1**
G04 −30665.5387				
Best	26,700	20,100	−30665.5387	−30,665.5387
Median	30,800	23,600	−30665.5387	−30,665.5387
Mean	30,800	23,676	−30665.5387	−30,665.5387
Worst	36,100	26,400	−30665.5387	−30,665.5387
SD	2.1819E+03	1.6541E+03	3.7130E−12	3.7130E−12
SR	1	1	1	1
G05 5126.4967				
Best	–	10,600	5126.4969	5126.4967
Median	–	17,900	5126.4972	5126.4967
Mean	–	18,656	5126.4972	5126.4967
Worst	–	31,800	5126.4976	5126.4967
SD	–	5.2837E+03	1.6379E−04	2.7847E−12
SR	0	1	0	**1**

Table 4 continued

Func. and optimal value	FES_D	FES_M	FIT_D	FIT_M
G06 −6961.8139				
Best	11,700	5600	−6961.8139	**−6961.8139**
Median	12,400	6100	−6952.4813	**−6961.8139**
Mean	12,400	6120	−6932.2531	**−6961.8139**
Worst	13,100	6600	−6786.8708	**−6961.8139**
SD	9.8995E+02	2.4833E+02	4.6394E+01	**0**
SR	0.08	1	0.08	**1**
G07 24.3062				
Best	72,800	46,100	24.3062	24.3062
Median	87,600	56,100	24.3062	24.3062
Mean	87,912	56,700	24.3062	24.3062
Worst	101,400	63,000	24.3062	24.3062
SD	6.3423E+03	3.4666E+03	1.5619E−08	6.7252E−15
SR	1	1	1	1
G08 −0.09582504				
Best	700	700	−0.09582504	−0.09582504
Median	1200	2000	−0.09582504	−0.09582498
Mean	1184	2096	−0.09582504	−0.09582391
Worst	1500	2900	−0.09582504	−0.09581591
SD	2.0347E+02	5.4657E+02	1.6518E−17	2.2722E−06
SR	1	1	1	1
G09 680.630057				
Best	22,100	15,600	680.630057	680.630057
Median	25,300	18,200	680.630057	680.630057
Mean	25,460	18,124	680.630057	680.630057
Worst	31,500	19,400	680.630057	680.630057
SD	1.9055E+03	8.7144E+02	1.1367E−08	2.3779E−13
SR	1	1	1	1
G10 7049.2480				
Best	700	700	−0.09582504	−0.09582504
Median	1200	2000	−0.09582504	−0.09582498
Mean	1184	2096	−0.09582504	−0.09582391
Worst	1500	2900	−0.09582504	−0.09581591
SD	2.0347E+02	5.4657E+02	1.6518E−17	2.2722E−06
SR	1	1	1	1
G11 0.7499				
Best	94,300	1700	0.7499	0.7499
Median	21,5300	2500	0.9342	0.7499
Mean	215,300	2483	0.9090	0.7522
Worst	33,6300	2900	0.9843	0.8017
SD	1.7112E+05	2.6740E+02	6.7854E−02	1.0370E−02
SR	0.08	0.92	0.08	0.92
G12 −1.0000				
Best	100	100	−1.0000	−1.0000
Median	100	100	−1.0000	−1.0000
Mean	100	100	−1.0000	−1.0000
Worst	100	100	−1.0000	−1.0000
SD	0	0	0	0
SR	1	1	1	1

Table 4 continued

Func. and optimal value	FES_D	FES_M	FIT_D	FIT_M
G13 0.05394151				
Best	–	–	0.61293374	0.48430818
Median	–	–	0.61336015	0.58354355
Mean	–	–	0.61342680	0.56981500
Worst	–	–	0.61464663	0.60034601
SD	–	–	3.8304E−04	3.4001E−02
SR	0	0	0	0
G14 −47.764888				
Best	259,700	43,100	−47.764888	**−47.764888**
Median	409,400	47,300	−47.681166	**−47.764888**
Mean	392,744	48,596	−47.570461	**−47.764888**
Worst	482,100	59,100	−46.956127	**−47.764888**
SD	7.8414E+04	4.7786E+03	2.4485E−01	**2.9008E−14**
SR	0.36	1	0.36	**1**
G15 961.715022				
Best	–	7500	962.168160	961.715022
Median	–	7500	962.172237	961.826461
Mean	–	7500	962.172506	961.833962
Worst	–	7500	962.176638	961.969301
SD	–	0	2.5469E−03	6.6747E−02
SR	0	0.04	0	**0.04**
G16 −1.905155				
Best	18,200	15,000	−1.905155	−1.905155
Median	19,900	17,300	−1.905155	−1.905155
Mean	19,968	17,104	−1.905155	−1.905155
Worst	22,100	18,600	−1.905155	−1.905155
SD	1.0566E+03	8.9045E+02	9.0649E−16	7.4889E−16
SR	1	1	1	1
G17 **8853.533875**				
Best	–	–	8888.511800	8882.428257
Median	–	–	8889.080392	8883.900462
Mean	–	–	8889.051374	8883.924324
Worst	–	–	8889.674809	8884.783828
SD	–	–	3.0127E−01	5.4886E−01
SR	0	0	0	0
G18 −0.86602540				
Best	43,300	27,300	−0.86602540	−0.86602540
Median	56,000	31,600	−0.86602540	−0.86602540
Mean	55,664	31,412	−0.86602540	−0.86602540
Worst	67,100	35,600	−0.86602540	−0.86602540
SD	4.5889E+03	2.0614E+03	2.2662E−17	0
SR	1	1	1	1
G19 32.655593				
Best	122,700	93,400	32.655593	32.655593
Median	132,100	101,000	32.655593	32.655593
Mean	132,236	101,512	32.655593	32.655593
Worst	141,800	112,500	32.655593	32.655593
SD	5.3654E+03	4.3234E+03	2.1018E−14	2.0254E−14
SR	1	1	1	1

Table 4 continued

Func. and optimal value	FES_D	FES_M	FIT_D	FIT_M
G21 193.724510				
Best	–	–	193.820262	193.751473
Median	–	–	194.380616	194.182448
Mean	–	–	194.870943	194.562136
Worst	–	–	197.500734	198.659273
SD	–	–	1.0309E+00	1.0875E+00
SR	0	0	0	0
G23 −400.0551				
Best	–	–	−1771.5292	−374.8644
Median	–	–	−1351.9079	−298.4791
Mean	–	–	−1313.5235	−295.9368
Worst	–	–	−61.2065	−212.6887
SD	–	–	3.1775E+01	4.8617E+01
SR	0	0	0	0
G24 −5.5080				
Best	5100	2200	−5.5080	−5.5080
Median	7100	2600	−5.5080	−5.5080
Mean	8784	2612	−5.5080	−5.5080
Worst	15,100	3000	−5.5080	−5.5080
SD	3.1676E+03	1.9858E+02	9.9119E−15	3.8073E−15
SR	1	1	1	1

FES_D and FES_M stand for the FES to achieve the success condition ($f(\vec{x}) - f(\vec{x}^*) \leq 0.0001$ and $f(\vec{x})$ is feasible) with corresponding CHTs while FIT_D and FIT_M stands for the fitness values of the successful solution with corresponding CHTs. SR means the success rate. "–" means no successful solutions were found. Values in boldface mean that the obtained result is much better with respect to the CHTs compared

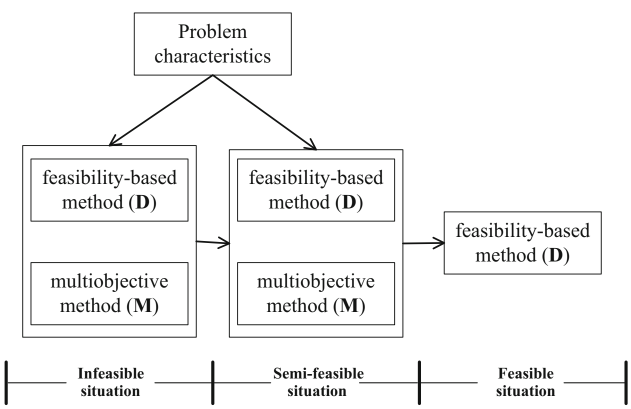

Fig. 2 Illustration of the basic idea

The basic idea of the combining strategy, the framework of CCHF and the implementation of the corresponding CHT choosing are illustrated in Figs. 2, 3 and 4 respectively.

As shown in Fig. 2, in the infeasible and semi-feasible situation, both Deb's feasibility-based method and multi-objective method are ready in the CHT pools. During an evolution, the problem characteristics will determine which CHT will be adopted, as shown in Fig. 4. After choosing the corresponding CHT, the population will be ranked and the best *NP* individuals will be selected to form the next population.

It is important to note that as to the multiobjective method, different pareto front levels will be used to help select the best individuals.

It should be pointed out that CCHF can also be seen as an ensemble method, in which the problem characteristics and different situations are considered when designing the corresponding relationship. This makes it different from other ensemble methods, such as ECHT [11], DECV [22], and other methods based on these three situations, such as ICDE [16], CMODE [7].

6 Experimental study

6.1 Experimental settings

As mentioned in Sect. 4, 22 benchmark functions [32] were used in our experiment. The details of these benchmark functions are reported in Table 1, where n is the number of decision variables, $\rho = |F|/|S|$ is the estimated ratio between the feasible region and the search space, *LI, NI, LE, NE* is the number of linear inequality constraints, nonlinear

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 Combined swarm evolution model

Step 2.1 Update P_t using DE model to create offspring. These *NP* offspring form the offspring population Q_t .

Step 2.2 Evaluate $f(\vec{x}_{i,t})$, $G(\vec{x}_{i,t})$ ($i = 1, \cdots, NP$).

Step 2.3 Compute the feasibility percent f_p of the combined population H_t (i.e., $H_t = P_t \cup Q_t$).

Step 2.4 Determinate the current situation of H_t according to the value of f_p .

Step 2.5 Choose the corresponding CHT according to the problem's characteristics and the current situation (see Fig.3. for details).

Step 2.6 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

Fig. 3 Framework of CCHF

Fig. 4 Implementation of the corresponding CHT choosing

Begin

  If current situation =infeasible situation **then**

    If <Prob. Char.>=only inequality constraints **then**

      Choose the Deb's feasibility-based rule

    Else

      Choose the multi-objective optimization technique

    EndIf

  Elseif current situation =semi-feasible situation **then**

    If <Prob. Char.>= only inequality constraints with the nonlinear objective function **then**

      Choose the Deb's feasibility-based rule

    Else

      Choose the multi-objective optimization technique

    EndIf

  Else

    Choose Deb's feasibility-based rule

  EndIf

End

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. These benchmark functions are also classified into different groups as shown in Table 2.

The parameters in DE are set as follows [7]: the population size (*NP*) is set to 100; the scaling factor (F) is randomly chosen between 0.5 and 0.6, and the crossover control parameter (*Cr*) is randomly chosen between 0.9 and 0.95. The same settings of these CHTs were used as in Sect. 4 to keep consistency.

6.2 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. [32]. Additionally, the tolerance value δ for the equality constraints was set to 0.0001.

Table 5 lists the results of CCHF, including best, median, worst, mean, standard deviation values, the feasible rate (the percentage of runs where at least one feasible solution is found in MAX_FES, denoted as FR), the success rate (the percentage of runs where the algorithm finds a solution that satisfies the success condition, denoted as SR). Here, the

Table 5 Results of CCHF, including best, median, worst, mean and standard deviation values

Prob.	Best	Median	Worst	Mean	SD	Feasible rate (%)	Success rate (%)
g01	−15.0000	−15.0000	−15.0000	−15.0000	0.0000E+00	100	100
g02	−0.803619	−0.803619	−0.785267	−0.801646	4.8400E−03	100	84
g03	−1.000500	−1.000500	−1.000500	−1.000500	2.2662E−16	100	100
g04	−30,665.5387	−30,665.5387	−30,665.5387	−30,665.5387	3.7130E−12	100	100
g05	5126.496714	5186.443925	5517.389512	5237.477027	1.0639E+02	100	4
g06	−6961.813876	−6961.813876	−6897.930384	−6956.805311	1.7339E+01	100	92
g07	24.306209	24.306209	24.306209	24.306209	4.7159E−09	100	100
g08	−0.09582504	−0.09582504	−0.09582504	−0.09582504	1.4164E−17	100	100
g09	680.630057	680.630057	680.630057	680.630057	3.3106E−09	100	100
g10	7049.248020	7049.248020	7049.248023	7049.248021	6.2341E−07	100	100
g11	0.749900	0.749900	0.838891	0.753460	1.7798E−02	100	96
g12	−1.0000	−1.0000	−1.0000	−1.0000	0	100	100
g13	0.88003034	0.99455287	0.99990801	0.97670142	3.4283E−02	100	0
g14	−47.764888	−47.764888	−47.764888	−47.764888	2.9001E−14	100	100
g15	961.715022	961.721578	964.283914	962.130169	7.5030E−01	100	44
g16	−1.905155	−1.905155	−1.905155	−1.905155	7.2661E−16	100	100
g17	8859.753007	8941.072424	8961.105710	8924.605594	3.3991E+01	100	0
g18	−0.866025	−0.866025	−0.866025	−0.866025	3.3362E−09	100	100
g19	32.655593	32.655593	32.655593	32.655593	2.1610E−14	100	100
g21	193.798125	194.664646	329.889655	244.498875	6.2859E+01	100	0
g23	−394.394784	−363.853711	−224.653630	−350.670946	4.3000E+01	100	0
g24	−5.508013	−5.508013	−5.508013	−5.508013	9.0649E−16	100	100

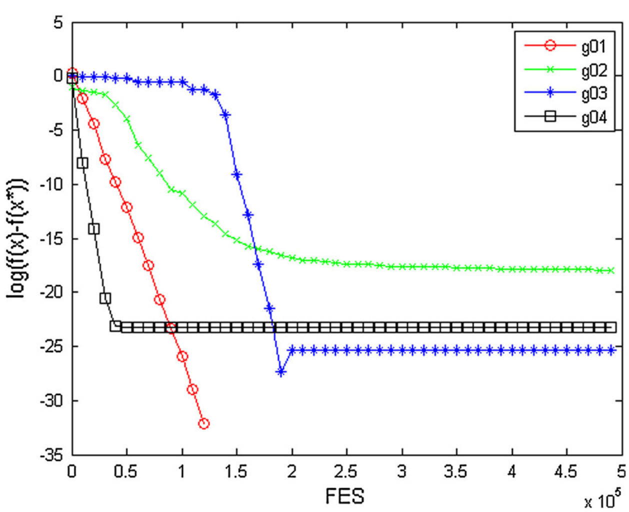

Fig. 5 Convergence graph for g01–g04

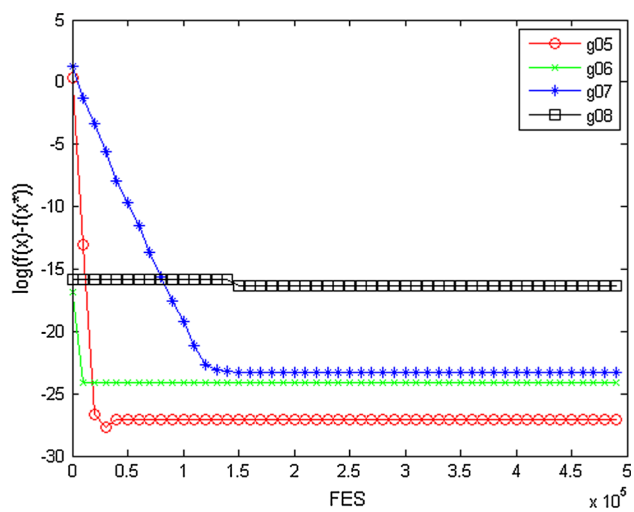

Fig. 6 Convergence graph for g05–g08

success condition is $f(\vec{x}) - f(\vec{x}^*) \leq 0.0001$ and $f(\vec{x})$ is feasible. The results show that CCHF can always find feasible solutions in all functions, but it can not get the known optimal values in g13, g17, g21 and g23. The mainly reason is that this CCHF takes the simplest form of the CHTs. And the main purpose of CCHF is to illustrate the practicability of the idea.

The convergence graphs of $\log(f(\vec{x}) - f(\vec{x}^*))$ over FES at the best run are plotted in Figs. 5, 6, 7, 8, 9 and 10. Since test functions g13, g17, g21 and g23 can not reach the optimal value, their convergence graphs are plotted in Fig. 10.

As shown in Figs. 5, 6, 7, 8 and 9, all test functions (except g02 and g08), can reach the error accuracy level with −20 with $<2 \times 10^5$ FES. The test functions g02 and g08 can reach

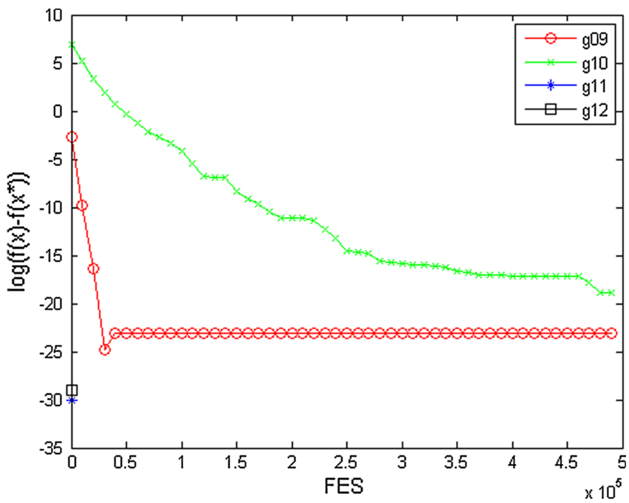

Fig. 7 Convergence graph for g09–g12

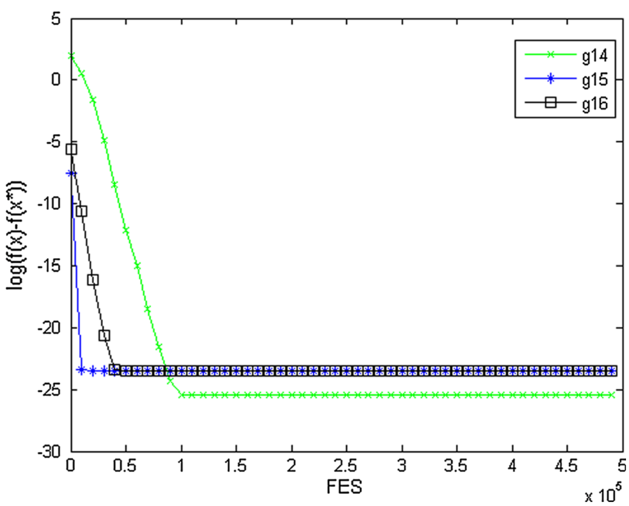

Fig. 8 Convergence graph for g14–g16

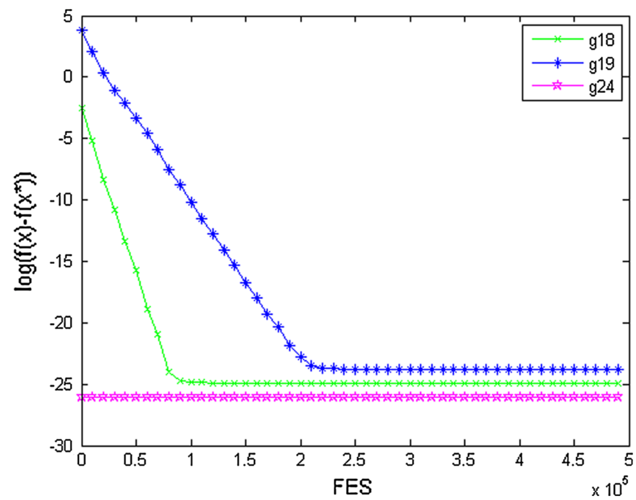

Fig. 9 Convergence graph for g18, g19, and g24

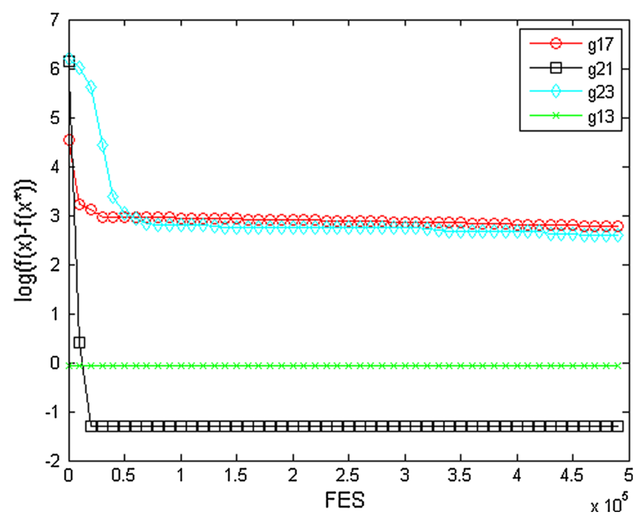

Fig. 10 Convergence graph for g13, g17, g21, and g23

the error accuracy level with −15. It is also important to note that g11 and g12 can reach the optimal value at the first generation.

As to Fig. 10, these test functions can not get the optimal values, with the error accuracy level with −1 to −3. The main reason is also the simple form of CHTs.

6.3 Comparison with some state-of-the-art approaches

In this part, five latest "dynamic" or "ensemble" approaches: COMDE [33], DECV [22], DSS-MDE [13], ATMES [12], and ECHT [11], are selected to compare with CCHF.

Table 6 presents the statistically results of t test (h values) for the different approaches. Numerical values −1, 0, 1 represent that CCHF is inferior to, equal to and superior to other approaches respectively.

CCHF performs better than the other five approaches in g01, g02, g07 and g10, and it presents a worse performance in g06 and g13 than the other five approaches. All the six approaches have the same or similar performance in g04, g08 and g12.

As for g03 and g09, CCHF performs similar with DSS-MDE and ECHT-EP, DSS-MDE, COMDE and DECV, but superior than the other algorithms. As for g05 and g11, CCHF performs similar with COMDE and ATMES, ATMES and DECV respectively, but inferior to the other approaches.

Overall, CCHF is superior to, equal to and inferior to other approaches in 25, 25 and 15 cases, respectively out of the 65 cases. The worse cases are mainly from g06 and g13.

Therefore, CCHF shows a comparable overall performance with the other five approaches. This also verifies the effectiveness of the proposed method in solving COPs.

Table 6 Comparison of CCHF with state-of-the-art "dynamic" or "ensemble" approaches

Func. and optimal value	DSS-MDE [13]	COMDE [33]	ECHT-EP [11]	ATMES [12]	DECV [22]	CCHF
G01 −15.0000						
Best	−15.000	−15.000000	−15.0000	−15.000	−15.000	−15.0000
Median	NA	−15.000000	−15.0000	−15.000	NA	−15.0000
Mean	−15.000	−15.000000	−15.0000	−15.000	−14.855	−15.0000
Worst	−15.000	−15.000000	−15.0000	−15.000	−13.0000	−15.0000
SD	1.39E−10	1.97E−13	0.00E+00	1.6E−14	4.59E−01	0.00E+00
h	1	1	1	1	1	–
G02 −0.803619						
Best	−0.803619	−0.803619	−0.8036191	−0.803388	−0.704009	−0.803619
Median	NA	−0.803616	−0.8033239	−0.792420	NA	−0.803619
Mean	−0.786970	−0.801238	−0.7998220	−0.756986	−0.569458	**−0.801646**
Worst	−0.728531	−0.785265	−0.7851820	−0.790148	−0.238203	−0.785267
SD	1.5E−02	5.0E−03	6.29E−03	1.3E−02	9.51E−02	4.84E−03
h	1	1	1	1	1	–
G03 −1.0005						
Best	−1.0005	−1.000000049	−1.0005	−1.000	−0.461	−1.000500100
Median	NA	−1.000000039	−1.0005	−1.000	NA	−1.000500100
Mean	−1.0005	−1.000000027	−1.0005	−1.000	−0.134	−1.000500100
Worst	−1.0005	−0.99999994	−1.0005	−1.000	−0.002	−1.000500100
SD	1.9E−08	3.026E−08	0.0E+00	5.9E−05	1.17E−01	2.2662E−16
h	0	1	0	1	1	–
G04 −30,665.5387						
Best	−30,665.539	−30,665.539	−30,665.5387	−30,665.539	−30,665.539	−30,665.5387
Median	NA	−30,665.539	−30,665.5387	−30,665.539	NA	−30,665.5387
Mean	−30,665.539	−30,665.539	−30,665.5387	−30,665.539	−30,665.539	−30,665.5387
Worst	−30,665.539	−30,665.539	−30,665.5387	−30,665.539	−30,665.539	−30,665.5387
SD	2.7E−11	0.00E+00	0.0E+00	7.4E−12	1.56E−06	3.7130E−12
h	0	0	0	0	0	–
G05 5126.4967						
Best	5126.497	5126.4981094	5126.4967	5126.498	5126.497	5126.4967140
Median	NA	5126.4981094	5126.4967	5126.776	NA	5186.4439258
Mean	5126.497	5126.4981094	5126.4967	5135.256	5126.497	5237.4770279
Worst	5126.497	5126.4981094	5126.4972	5127.648	5126.497	5517.3895124
SD	0	0.00E+00	0.0E+00	1.8E+00	0	1.0639E+02
h	−1	0	−1	0	−1	–
G06 −6961.8139						
Best	−6961.814	−6961.813875	−6961.8139	−6961.814	−6961.814	−6961.813876
Median	NA	−6961.813875	−6961.8139	−6961.814	NA	−6961.813876
Mean	−6961.814	−6961.813875	−6961.8139	−6961.814	−6961.814	−6956.805311
Worst	−6961.814	−6961.813875	−6961.8139	−6961.814	−6961.814	−6897.930384
SD	0	0.00E+00	0.00E+00	4.6E−12	0	1.7339E+01
h	−1	−1	−1	−1	−1	–
G07 24.3062						
Best	24.306	24.306209	24.3063	24.306	24.306	24.306209
Median	NA	24.306209	24.3078	24.313	NA	24.306209
Mean	24.306	24.306209	24.3090	24.359	24.794	24.306209
Worst	24.306	24.306211	24.3166	24.316	29.511	**24.306209**

Table 6 continued

Func. and optimal value	DSS-MDE [13]	COMDE [33]	ECHT-EP [11]	ATMES [12]	DECV [22]	CCHF
SD	7.5E−07	4.7E−07	3.0E−03	1.1E−02	1.37E+00	**4.7159E−09**
h	1	1	1	1	1	–
G08 −0.09582504						
Best	−0.095825	−0.095825	−0.09582504	−0.095825	−0.095825	−0.09582504
Median	NA	−0.095825	−0.09582504	−0.095825	NA	−0.09582504
Mean	−0.095825	−0.095825	−0.09582504	−0.095825	−0.095825	−0.09582504
Worst	−0.095825	−0.095825	−0.09582504	−0.095825	−0.095825	−0.09582504
SD	4.0E−17	9.00E−18	0.0E+00	2.8E−17	4.23E−17	1.4164E−17
h	0	0	0	0	0	–
G09 680.630057						
Best	680.630	680.630057	680.630057	680.630	680.630	680.630057
Median	NA	680.630057	680.630057	680.633	NA	680.630057
Mean	680.630	680.630057	680.630057	680.673	680.630	680.630057
Worst	680.630	680.630057	680.630060	680.639	680.630	680.630057
SD	2.9E−13	4.071E−13	2.0E−04	1.0E−02	3.45E−07	3.3106E−09
h	0	0	1	1	0	–
G10 7049.2480						
Best	7049.248	7049.248020	7049.2487	7052.253	7049.248	7049.248020
Median	NA	7049.248020	7049.3456	7215.357	NA	7049.248020
Mean	7049.249	7049.248077	7049.4342	7560.224	7103.548	7049.248021
Worst	7049.255	7049.248615	7050.3902	7250.437	7808.980	**7049.248023**
SD	1.4E−03	1.5E−04	2.00E−01	1.2E+02	1.48E+02	**6.2341E−07**
h	1	1	1	1	1	–
G11 0.7499						
Best	0.7499	0.749999	0.7499	0.75	0.75	0.749900
Median	NA	0.749999	0.7499	0.75	NA	0.749900
Mean	0.7499	0.749999	0.7499	0.75	0.75	0.753460
Worst	0.7499	0.749999	0.7499	0.75	0.75	0.838891
SD	0	0.00E+00	0.0E+00	3.4E−04	1.12E−16	1.7798E−02
h	−1	−1	−1	0	0	–
G12 −1.0000						
Best	−1.000	−1.000000	−1.0000	−1.000	−1.000	−1.0000
Median	NA	−1.000000	−1.0000	−1.000	NA	−1.0000
Mean	−1.000	−1.000000	−1.0000	−0.994	−1.000	−1.0000
Worst	−1.000	−1.000000	−1.0000	−1.000	−1.000	−1.0000
SD	0	0.00E+00	0.0E+00	1.0E−03	0	0
h	0	0	0	0	0	–
G13 0.05394151						
Best	0.053942	0.0539415	0.053941514	0.053950	0.059798	0.88003034
Median	NA	0.0539415	0.053941514	0.053952	NA	0.99455287
Mean	0.053942	0.0539415	0.053941514	0.053999	0.382401	0.97670142
Worst	0.053942	**0.0539415**	0.053941514	0.053959	0.999094	0.99990801
SD	8.3E−17	1.4E−17	6.50E−12	1.3E−05	2.68E−01	3.4283E−02
h	−1	−1	−1	−1	0	–

Values in boldface mean that the obtained result is much better with respect to the approaches compared

7 Conclusion

In this paper, a CCHF, which combines promising aspects of different CHTs in different situations with consideration of problem characteristics, was proposed, implemented, and validated. The presented work is distinguished in three scientific contributions. First, the relationship between problem characteristics and CHTs, and the relationship between different CHTs were analyzed; second, the CCHF was developed based on the analysis; third, the 22 benchmark functions collected on constrained real-parameter optimization were utilized to verify the effectiveness of the newly developed CCHF.

The results show that CCHF is comparable to the other five dynamic or ensemble state-of-the-art approaches for constrained optimization, especially when considering that CCHF is simple and easy to realize due to adoption of only the basic CHTs without any variants in this framework.

The problem characteristics summarized in this paper are based on the benchmark functions, but as Z. Michalewicz concluded [17], there is no comparison in terms of complexity between real-world problems and toy problems, and real-world applications usually require hybrid approaches where an 'evolutionary algorithm' is loaded with non-standard features, so how to apply these conclusions to the real-world problems is still challenging and will be our future work.

Acknowledgements CS would like to thank Prof. Dr. Robert Weigel for his great help in the life and research work, and he is grateful to Dr. Guojun Gao for proofreading and valuable suggestions for this paper. CS also appreciates M.S. Chengyu Huang's inspiration on the systematical analysis.

References

1. Mezura-Montes E, Coello Coello CA (2005) A simple multi-membered evolution strategy to solve constrained optimization problems. IEEE Trans Evol Comput 9(1):1–17
2. Coello Coello CA (2002) Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Comput Methods Appl Mech Eng 191(11/12):1245–1287
3. Mezura-Montes E, Coello Coello CA (2011) Constraint-handling in nature-inspired numerical optimization: past, present and future. Swarm Evol Comput 1(4):173–194
4. Jordehi AR (2015) A review on constraint handling strategies in particle swarm optimization. Neural Comput Appl 26(6):1265–1275
5. Deb K (2000) An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 186(2–4):311–338
6. Zhou A, Qu B-Y, Li H, Zhao S-Z, Suganthan PN, Zhang Q (2011) Multiobjective evolutionary algorithms: a survey of the state-of-the-art. Swarm Evol Comput 1(1):32–49
7. Wang Y, Cai Z (2012) Combining multiobjective optimization with defferential evolution to solve constrained optimization problems. IEEE Trans Evol Comput 16(1):117–134
8. Tang K, Peng F, Chen G, Yao X (2014) Population-based algorithm portfolios with automated constituent algorithms selection. Inf Sci 279:94–104
9. Gao W, Yen GG, Liu S (2015) A dual-population differential evolution with coevolution for constrained optimization. IEEE Trans Cybern 45(5):1094–1107
10. Shiu Y, Zhang X (2015) On composing an algorithm portfolio. Memet Comput 7:203–214
11. Mallipeddi R, Suganthan PN (2010) Ensemble of constraint handling techniques. IEEE Trans Evol Comput 14(4):561–579
12. Wang Y, Cai Z, Zhou Y, Zeng W (2008) An adaptive tradeoff model for constrained evolutionary optimization. IEEE Trans Evol Comput 12(1):80–92
13. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochasitc selection for constrained optimization. Inf Sci 178(15):3043–3074
14. Tessema B, Yen GG (2009) An adaptive penalty formulation for constrained evolutionary optimization. IEEE Trans Syst Man Cybern A Syst Hum 39(3):565–578
15. Wang Y, Cai Z (2011) Constrained evolutionary optimization by means of $(\mu + \lambda)$-differential evolution and improved adaptive trade-off model. Evol Comput 19(2):249–285
16. Jia G, Wang Y, Cai Z, Jin Y (2013) An improved $(\mu + \lambda)$-constrained differential evolution for constrained optimization. Inf Sci 222:302–322
17. Michalewicz Z (2012) Quo Vadis, evolutionary computation? On a growing gap between theory and practice. In: Liu J et al (eds) Plenary/invited lectures (WCCI 2012). LNCS, vol 7311, pp 98–121
18. Jin Y, Tang K, Yu X, Sendhoff B, Yao X (2013) A framework for finding robust optimal solutions over time. Memetic Comput 5:3–18
19. He J, Chen T, Yao X (2015) On the easiest and hardest fitness functions. IEEE Trans Evol Comput 19(2):295–305
20. Yao X (2012) Unpacking and understand evolutionary algorithms. In: Liu J et al (eds) Plenary/invited lectures (WCCI 2012). LNCS, vol 7311, pp 60–76
21. Tsang E, Kwan A (1993) Mapping constraint satisfaction problems to algorithms and heuristics. Technical Report CSM-198, Department of Computer Science, University of Essex, Colchester, UK
22. Mezura-Montes E, Miranda-Varela ME, Gómez-Ramón RC (2010) Differential evolution in constrained numerical optimization: an empirical study. Inf Sci 180(22):4223–4262
23. Gibbs M, Maier H, Dandy G (2011) Relationship between problem characteristics and the optimal number of genetic algorithm generations. Eng Optim 43(4):349–376
24. Wang Y, Wang B, Li H, Yen GG (2015) Incorporating objective function information into the feasibility rule for constrained evolutionary optimization. IEEE Tran Cybern 46(12):2938–2952
25. Qiu X, Xu J, Tan KC, Abbass HA (2016) Adaptive cross-generation differential evolution operators for multi-objective optimization. IEEE Trans Evol Comput 20(2):232–244
26. Elsayed S, Sarker R (2016) Differential evolution framework for big data optimization. Memet Comput 8(1):17–33
27. Feng L, Ong YS, Lim MH, Tsang IW (2015) Memetic search with interdomain learning: a realization between CVRP and CARP. IEEE Trans Evol Comput 19(5):644–658
28. Si C, Wang L, Wu Q (2012) Mapping constrained optimization problems to algorithms and constraint handling techniques. In: Proceedings of the CEC, pp 3308–3315
29. Das S, Suganthan PN (2011) Differential evolution: a Survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31
30. Zitzler E, Deb K, Thiele L (2000) Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 8(2):173–195 summer

31. Si C, Shen J, Zou X, Duo Y, Wang L, Wu Q (2015) A dynamic penalty function for constrained optimization. In: Tan Y et al (eds) ICSI 2015. LNCS, vol 9141, pp 261–272
32. Liang JJ, Runarsson TP, Mezura-Montes E, Clerc M, Suganthan PN, Coello Coello CA, Deb K (2006) Problem definitions and evaluation criteria for the CEC 2006. Special session on constrained real-parameter optimization. Technical Report, Nanyang Technology University, Singapore
33. Mohamed AW, Sabry HZ (2012) Constrained optimization based on modified differential evolution algorithm. Inf Sci 194(1):171–208