Information Sciences 502 (2019) 201–217

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A feasible-ratio control technique for constrained optimization

Ruwang Jiao[a], Sanyou Zeng[a,*], Changhe Li[b,c,*]

[a] School of Mechanical Engineering and Electronic Information, China University of Geosciences, Wuhan 430074, China
[b] School of Automation, China University of Geosciences, Wuhan 430074, China
[c] Hubei Key Laboratory of Advanced Control and Intelligent Automation for Complex Systems, China University of Geosciences, Wuhan 430074, China

ARTICLE INFO

Article history:
Received 29 January 2019
Revised 7 June 2019
Accepted 10 June 2019
Available online 10 June 2019

Keywords:
Evolutionary computation
Constrained optimization
Constraint-handling
Constrained multi-objective optimization

ABSTRACT

In constrained optimization problems (COPs), a crucial issue is that most constraint-handling evolutionary algorithms (EAs) approach the optimum either mainly from feasible regions or mainly from infeasible regions. This may result in bias in search of feasible and infeasible solutions. To address this issue, we propose a feasible-ratio control technique which controls the ratio of feasible solutions in the population. By using the control technique, an EA can maintain the search balance from feasible and infeasible regions. Based on this technique, we propose a constraint-handling EA, named FRC-CEA. It consists of two-stage optimization. In the first stage, an enhanced dynamic multi-objective evolutionary algorithm (DCMOEA) with the feasible-ratio control technique is adopted to handle constraints. In the second stage, a commonly used differential evolution (DE) is used to speed up the convergence. The performance of the proposed method is evaluated and compared with six state-of-the-art constraint-handling algorithms on two sets of benchmark test suites. Experimental results suggest that the proposed method outperforms or is highly competitive against the compared algorithms on most test problems.

1. Introduction

Most real-world optimization problems in the field of science and engineering are subject to constraints. These problems can be categorized into constrained optimization problems (COPs). The presence of constraints changes the structure of the search space, which makes it hard to solve. Traditional mathematical programming methods adopted to tackle COPs have certain limitations when handling the general nonlinear programming problem [4]. Due to the efficiency and simplicity, evolutionary algorithms (EAs) have attracted a growing interest and have been widely employed for solving optimization problems, which results in a variety of constraint-handling EAs. In the last two decades, several constraint-handling techniques combined with EAs have been developed, and the most commonly used include penalty function, separation of objectives and constraints, multi-objective methods and ensemble of constraint handling methods [9].

Coello [4] points out that keeping every individual feasible is not a good choice because the population might plunge into some locally feasible optima. On the contrary, keeping every individual infeasible is unwise because the population cannot reach the feasible region. Notably, the global optimum of most optimization problems lies on the boundary of the feasible region [14,30]. Ideally, the search for the global optimum should be from both feasible and infeasible two regions.

* Corresponding authors.
E-mail addresses: sanyouzeng@gmail.com (S. Zeng), changhe.lw@gmail.com (C. Li).

https://doi.org/10.1016/j.ins.2019.06.030
0020-0255/

Our previous study [17] has initially demonstrated that searching the global optimal solution from both feasible and infeasible regions can greatly improve the performance in handling constraints. However, most of the research works focus on approaching the global optimum either mainly from feasible regions or infeasible regions.

To tackle the above issue, we propose a feasible-ratio control technique for handling constraints. A constrained EA based on the feasible-ratio control technique (FRC-CEA) is then proposed. FRC-CEA consists of two stages. In the first stage, an enhanced dynamic constrained multi-objective evolutionary algorithm (DCMOEA) framework [42] maintains a trade-off between minimizing objective and satisfying constraints by Pareto dominance and the feasible-ratio control technique until searching to the approximate constraint boundary. Notably, the feasible-infeasible ratio is controlled by selecting promising individuals from both feasible and infeasible regions with a niching method. In the second stage, the feasibility rule [7] based DE is utilized to speed up the convergence. Two sets of benchmark test suites collected from IEEE CEC2006 [19] and IEEE CEC2010 [21] are adopted to demonstrate the effectiveness of FRC-CEA. Experimental results show that the performance of FRC-CEA is highly competitive with that of six state-of-the-art optimizers. The effectiveness of the feasible-ratio control technique is also investigated.

In this paper, the proposed FRC-CEA is based on our previous DCMOEA [42]. However, they are essentially different. The primary difference lies in the search direction. DCMOEA mainly searches from infeasible regions to approach the global optimum due to the multi-objective selection restriction. By contrast, FRC-CEA can search the global optimum from both feasible and infeasible regions since it can strike a sound-balance between feasible and infeasible solutions. The search from disparate directions of the feasible and infeasible regions is advantageous since it is easier to locate the global optima solution than that of one region.

The paper is organized as follows: Section 2 briefly reviews related work. Section 3 gives a brief introduction of the DCMOEA framework. Section 4 describes the proposed method in detail and presents the proposed algorithm. Section 5 carries out experimental studies. Section 6 provides corresponding discussions. Finally, Section 7 concludes this paper.

2. Related work

A COP can be defined as the minimization of an objective function subject to a set of constraints:

$$\begin{aligned} \min y &= f(\boldsymbol{x}) \\ \text{st}: \boldsymbol{g}(\boldsymbol{x}) &= (g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \dots, g_m(\boldsymbol{x})) \leq \boldsymbol{0} \\ \text{where} \boldsymbol{x} &= (x_1, x_2, \dots, x_n) \in \mathbf{X} \\ \mathbf{X} &= \{\boldsymbol{x} | \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u}\} \\ \boldsymbol{l} &= (l_1, l_2, \dots, l_n), \boldsymbol{u} = (u_1, u_2, \dots, u_n), \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ is the solution vector, each x_k is bounded by lower and upper limits $l_k \leq x_k \leq u_k$ $(k = 1, \dots, n)$ which defines the search space $\mathbf{X}$. $\boldsymbol{g}(\boldsymbol{x}) \leq \boldsymbol{0}$ is the vector of constraints and $\boldsymbol{0}$ denotes the constraint boundary. To handle an equality constraint $h(\boldsymbol{x}) = 0$, it is transformed into an inequality constraint, i.e., $|h(\boldsymbol{x})| - \delta \leq 0$ $(\delta = 0.0001)$ [19,21].

Constraint-handling EAs can be categorized into four types based on which constraint handling techniques they employ: penalty function methods, methods that optimize the objective and tackle the constraint separately, methods based on multi-objective optimization technique, and methods based on ensemble of constraint handling techniques.

2.1. Penalty function

Penalty function converts a COP to an unconstrained one by adding the degree of constraint violation multiplied by a predefined penalty factor λ to the objective [3], which is the simplest and the earliest constraint-handling technique. However, setting a suitable penalty factor λ in advance is a complicated task since it is usually problem-dependent. If the penalty is too low, the algorithm might be unable to approach feasible regions. By contrast, if the penalty is too severe, solutions might move to feasible regions quickly but cannot locate optima that lie on the boundary between feasible and infeasible regions.

In recent years, researchers often combine the penalty function with other techniques to tackle complex COPs. For example, Sun and Jonathan [32] developed a multi-cycled sequential framework, where an over-penalized approach is regarded as the selection strategy. Ghasemishabankareh and Li [11] proposed an improved augmented Lagrangian method to deal with constraints. They employed the cooperative co-evolutionary framework that co-evolves two populations, one for decision variables, and the other for Lagrangian multipliers. Wang and Li [36] extended a teaching-learning-based optimization algorithm referred as TLBO. In TLBO, a weighted penalty function is introduced to tradeoff the constraints and objective function.

2.2. Separation of objectives and constraints

Feasibility rule [7] is another efficient constraint-handling technique which does not require users to define parameters in advance, and it is easy to implement. In feasibility rule, one solution $\boldsymbol{x}_i$ is regarded as superior to $\boldsymbol{x}_j$ under the following three conditions:

- $\boldsymbol{x}_i$ is feasible, and $\boldsymbol{x}_j$ is infeasible;
- $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are both feasible, but the objective value of $\boldsymbol{x}_i$ is better than that of $\boldsymbol{x}_j$;
- $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are both infeasible, but the degree of constraint violation of $\boldsymbol{x}_i$ is smaller than that of $\boldsymbol{x}_j$.

Feasibility rule is usually employed as the selection strategy when comparing a parent individual with a child one. For instance, to select appropriate control parameters in DE, a DE algorithm with dynamic selection of three control parameters (DPDE) [28] was proposed for solving COPs. Noha [13] proposed a novel DE mutation strategy which aims to give the combined effect of both a fitness improvement search operator and constraint consensus. The above two algorithms all adopt feasibility rule as the selection strategy. Nevertheless, it is well-known that the main weakness of the feasibility rule is it prefers constraints to objective function. Some infeasible solutions carried important information may be ignored during the selection process, which could cause an EA easy to trap on a local optima when the global optimum located in the feasible region edge or in the feasible regions have multiple disjoint areas. To alleviate the greediness of the feasibility rule for constraints, Wang et al. [39] presented a replacement mechanism which incorporates the promising objective function information into the feasibility rule. Xu et al. [40] integrated a cluster-replacement-based operator into the feasibility rule to alleviate the greediness.

Due to the fact that it can introduce the information of objective function, the ε-constraint method [33] and the stochastic ranking method [27] can remedy the weakness of the feasibility rule to some degree. The ε-constraint method first relaxes the constraint boundary and then gradually shrinks the relaxation space by using the ε parameter. By dynamically controlling the epsilon level ε, some infeasible solutions can be conditionally accepted as feasible solutions so that the objective information of these solutions can be utilized, which is beneficial to tackle COPs with thin feasible regions. The epsilon level ε is a critical parameter. In the case of ε=0, ε-constraint method is equivalent to feasibility rule. For any two solutions, both feasibility rule and ε-constraint method compare the degree of constraint violation before the objective. Stochastic ranking differs from feasibility rule and ε-constraint method based on the order of comparison. In the stochastic ranking, a probability parameter *Pf* is introduced to determine whether the comparison order is based on the objective value or the degree of constraint violation. The comparison will be based on the objective value if a random value is less than *Pf*. On the contrary, the comparison is based on the degree of constraint violation. When *Pf*=0, stochastic ranking also degenerates into the feasibility rule. Therefore, how to set the proper probability parameter *Pf* in the stochastic ranking method and epsilon level ε in the ε-constraint method has a significant impact on the performance of a constraint-handling EA.

2.3. Multi-objective methods

In recent years, much effort has been done in the field of multi-objective evolutionary algorithms (MOEAs) (e.g., some of the recent ones [2,24–26]), which paves a new road to deal with COPs. The key to solving COPs is how to balance the objective function and constraints. In multi-objective-based approach, the degree of constraint violation or each constraint is treated as an objective, which can maintain the population diversity when optimizing objective function and constraints simultaneously. A survey on using MOEAs for single-objective COPs can be found in [29].

Wang and Cai combined multi-objective optimization with DE named CMODE [38] to deal with COPs. In CMODE, a novel infeasible solution replacement mechanism based on multi-objective optimization was proposed, to guide the population toward promising solutions and the feasible regions simultaneously. The above approach adopts Pareto dominance to compare solutions. Jiao and Li [15] introduced a self-adaptive selection method based on the multi-objective technique, which intends to exploit infeasible solutions with a low constraint violation. Datta and Deb [5] combined a bi-objective evolutionary method with the penalty function to tackle COPs. In their approach, the penalty parameters and relaxed constraints with a constant boundary are dynamically tuned. Gao and Yen [10] proposed a co-evolutionary dual-population DE (DPDE) which divides the population into dual subpopulations based on their feasibility. In DPDE, the COP is treated as a bi-objective optimization problem. Meanwhile, the dual subpopulations communicate through an information-sharing strategy.

2.4. Ensemble of constraint handling techniques

The ensemble of constraint handling techniques (ECHT) [20] has been proposed to make use of the advantages of different constraint-handling approaches. Four constraint handling approaches were integrated into the ECHT framework. Rodrigues and Lima [6] proposed a balanced ranking method which gathers the advantages of existing constraint-handling techniques, such as penalty function, feasibility rule, stochastic ranking, and so on. Yu and Wang [41] presented an improved constrained teaching-learning-based optimization algorithm to handle constraints. Three different constraint handling techniques are adopted for three cases during the evolution process: infeasible case, semi-feasible case, and feasible case, respectively. However, among the existing constraint-handling techniques, the ensemble of constraint handling techniques often suffer from high design complexities and computational burden with respect to single constraint handling methods.

2.5. Discussions

To summarize, the above constraint-handling techniques all try to address one issue: how to find the feasible optimum for a COP. The global optima for most COPs is on or near the constraint boundary. It is beneficial to maintain some infeasible solutions in the population, because feasible children of infeasible parent solutions will tend to near the optimum [18]. Some attempts have been made to maintain both feasible and infeasible solutions in the population [14,18,30]. In these algorithms, the search is initiated from both sides of the search space, which aims at driving feasible solutions to feasible regions with better fitness and driving infeasible solutions towards areas with less constraint-violation. Following this idea, in this paper, we propose a two-stage optimization algorithm to deal with COPs. It has two advantages: 1) A feasible-ratio control technique is used to balance the proportion of feasible and infeasible solutions; 2) a dynamic niching technique can guarantee the proper diversity from both feasible and infeasible regions.

It is worthwhile to note that most existing two-stage constraint-handling EAs (e.g.,[35,44]) all have similar characteristics. In the first stage, constraint satisfaction as the primary goal to guide the population to enter the feasible region. Once a certain number of feasible solutions are found, the search will enter the second stage in which the algorithm mainly aims at improving the fitness value in the feasible region. However, our proposed two-stage method has distinct purposes in different stages compared with [35,44]. In the first stage, FRC-CEA intends to narrow the search from the entire search space to a small promising region where the global optimum is located from both feasible and infeasible regions rather than just find a certain number of feasible solutions. After the first stage of optimization, the final population contains both feasible and infeasible solutions, but the infeasible solutions are very close to the feasible region. The purpose of the second-stage of optimization is to force infeasible solutions to enter the feasible region quickly and then approach the global optimum.

3. DCMOEA framework

Significant progress has been made in the recent two decades in the development of MOEAs for solving multi-objective optimization problems (MOPs). Existing MOEAs usually perform well on unconstrained or weakly constrained MOPs. They encounter difficulties in highly constrained optimization problems. To remedy this issue, we have proposed the dynamic constrained multi-objective evolutionary algorithm (DCMOEA) framework [16,42,43] which converts a COP into an equivalent dynamic constrained MOP (DCMOP) with weakly constraints. Any existing MOEAs can be integrated into the DCMOEA framework to deal with COPs. Since the proposed method of this paper is based on our previous DCMOEA [42], in this section, we provide a brief overview of DCMOEA.

3.1. Conversion of the COP to the DCMOP

The idea of DCMOEA is to adopt the constraint-violation and niche-count as two new objectives to solve COPs. It recasts a COP as a tri-objective optimization problem to minimize the original objective function, the degree of constraint violation and the niche-count simultaneously. For the sake of clarity, let $F(\boldsymbol{x}) = (f(\boldsymbol{x}), cv(\boldsymbol{x}), nc(\boldsymbol{x}|\mathbf{U}, \sigma))$.

3.1.1. Constraint-violation

A solution's constraint-violation is defined as the mean of the normalized violations of all constraints:

$$cv(\boldsymbol{x}) = \frac{1}{m}\sum_{i=1}^{m}\frac{G_i(\boldsymbol{x})}{\max\limits_{\boldsymbol{x}\in\mathbf{P}(0)}\{G_i(\boldsymbol{x})\}}, \tag{2}$$

where $\mathbf{P}(0)$ represents the initial population, $G_i(\boldsymbol{x})$ is evaluated by $G_i(\boldsymbol{x}) = \max\{g_i(\boldsymbol{x}), 0\}$, $i = 1, 2, \ldots, m$.

3.1.2. Niche-count

Given $\mathbf{U} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L\}$ (the union of the parent and offspring populations) and the niche radius σ, the niche-count [12] of $\boldsymbol{x} \in \mathbf{U}$ is:

$$nc(\boldsymbol{x}|\mathbf{U}, \sigma) = \sum_{i=1, \boldsymbol{x}_i\neq\boldsymbol{x}}^{L} sh(\boldsymbol{x}, \boldsymbol{x}_i), \tag{3}$$

where $sh(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is the sharing function between two solutions $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$:

$$sh(\boldsymbol{x}_1, \boldsymbol{x}_2) = \begin{cases} 1 - (\frac{d(\boldsymbol{x}_1, \boldsymbol{x}_2)}{\sigma}) & d(\boldsymbol{x}_1, \boldsymbol{x}_2) \leq \sigma \\ 0 & otherwise, \end{cases} \tag{4}$$

where $d(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is the Euclidean distance.

3.1.3. DCMOP

A constrained MOP (CMOP) can be obtained as follows:

$$\begin{aligned} & min(f(\boldsymbol{x}), cv(\boldsymbol{x}), nc(\boldsymbol{x}|\mathbf{U}, \sigma)) \\ & st: \mathbf{g}(\boldsymbol{x}) \leq \mathbf{0} \end{aligned} \tag{5}$$

The CMOP defined by Eq. (5) is not equal to the initial COP in Eq. (1), although the optimum of Eq. (1) is part of the Pareto set of Eq. (5). The constrained MOEA (CMOEA) is confronted with the same constraint-handling difficulty as a single-objective constrained EA does. To remedy this issue, the CMOP is recast as the DCMOP in the form of:

$$CMOP^{(s)} \begin{cases} min(f(\boldsymbol{x}), cv(\boldsymbol{x}), nc(\boldsymbol{x}|\mathbf{U}, \sigma^{(s)})) \\ st : \boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{\varepsilon}^{(s)}, \end{cases} \tag{6}$$

where $\sigma^{(0)} > \sigma^{(1)} > \cdots > \sigma^{(S)} = 0$, $\boldsymbol{\varepsilon}^{(s)} = (\varepsilon_1^{(s)}, \cdots, \varepsilon_m^{(s)})$, $s = 0, 1, \cdots, S$. $\boldsymbol{\varepsilon}^{(0)} > \boldsymbol{\varepsilon}^{(1)} > \cdots > \boldsymbol{\varepsilon}^{(S)} = \mathbf{0}$, s indicates the environmental state. $\boldsymbol{\varepsilon}^{(s)}$ and $\sigma^{(s)}$ represent the dynamic constraint boundary and dynamic niche radius, respectively. s to $s+1$ is an environmental change which represents the shrink of the constraint boundary and the niche radius. S is the maximum number of environmental changes. Actually, the dynamic reducing the constraint boundary is similar to the barrier method [23] and ε-constraint method [33].

At the last state S, the dynamic constraint boundary shrinks to the original constraint boundary: $\boldsymbol{\varepsilon}^{(S)} = \mathbf{0}$, $cv(\boldsymbol{x}) = 0$, and $\sigma^{(S)} = 0$, $nc(\boldsymbol{x}, \mathbf{U}|0) = 0$. Then the final problem $CMOP^{(S)}$ of DCMOP in Eq. (6) can be simplified as:

$$min(f(\boldsymbol{x}), 0, 0) nonumber \tag{7}$$

$$st : \boldsymbol{g}(\boldsymbol{x}) \le \mathbf{0}. \tag{7}$$

The following two concepts are introduced for DCMOPs:

Definition 1. ($\boldsymbol{\varepsilon}$-feasible, $\boldsymbol{\varepsilon}$-infeasible) A solution $\boldsymbol{x}$ is regarded as $\boldsymbol{\varepsilon}$-feasible if $\boldsymbol{g}(\boldsymbol{x}) \le \boldsymbol{\varepsilon}$; else, it is treated as $\boldsymbol{\varepsilon}$-infeasible.

The $\boldsymbol{\varepsilon}$-constrained Pareto dominance principle is used to compare pairwise solutions:

Definition 2. ($\boldsymbol{\varepsilon}$-Constrained Pareto Dominance Principle)

- When two $\boldsymbol{\varepsilon}$-feasible solutions are compared, the one Pareto dominating the other is better;
- When an $\boldsymbol{\varepsilon}$-feasible solution is compared with an $\boldsymbol{\varepsilon}$-infeasible solution, the $\boldsymbol{\varepsilon}$-feasible solution is better;
- When two $\boldsymbol{\varepsilon}$-infeasible solutions are compared, the one with a smaller constraint-violation is better.

The shrinkage of the constraint boundary $\boldsymbol{\varepsilon}$ and the niche radius σ adopts two exponential functions:

$$\varepsilon_i^{(s)} = \varepsilon_i^{(0)} e^{-(\frac{\sqrt{ln\varepsilon_i^{(0)}}s}{S})^2} \tag{8}$$

$$\sigma^{(s)} = \sigma^{(0)} e^{-(\frac{\sqrt{ln\sigma^{(0)}}s}{S})^2} \tag{9}$$

where $i = 1, 2, \cdots, m$, m is the number of constraints. For the sake of achieving an $\boldsymbol{\varepsilon}$-feasible initial population $\mathbf{P}(0)$, the maximum violation of a constraint in $\mathbf{P}(0)$ is selected as the initial relaxed constraint boundary $\varepsilon_i^{(0)}$. Therefore, the initial relaxed constraint boundary $\varepsilon_i^{(0)} = \max_{\boldsymbol{x} \in \mathbf{P}(0)} \{G_i(\boldsymbol{x})\}$, $i = 1, 2, \cdots, m$ at $s = 0$ for $G_i(\boldsymbol{x})$, and the shrunk constraint boundary at final state $s = S$ is $\boldsymbol{\varepsilon}^{(S)} = \boldsymbol{\delta}$, where δ is a positive close-to-zero number for the precision requirement ($\delta = 1e-8$). The average space a solution takes in the initial population is regarded as the initial niche size, i.e., $\prod_{i=1}^{n}(u_i - l_i)/L$. Therefore, the initial niche radius $\sigma^{(0)} = \frac{1}{2}\sqrt[n]{\frac{2n\prod_{i=1}^{n}(u_i - l_i)}{L\pi}}$ can be obtained at $s = 0$, and the shrunk niche radius at final state $s = S$ is $\sigma^{(S)} = \delta$ ($\delta = 1e-8$).

4. The proposed feasible-ratio control technique

4.1. Motivation

DCMOEA utilizes the multi-objective technique for dealing with COPs. One primary drawback of DCMOEA is that it mainly searches the global optimal solution from infeasible regions, which may result in the bias against feasible solutions, and the useful information about the optimal direction of feasible regions are not fully used.

Fig. 1 presents the distribution of solutions in a population based on multi-objective techniques. The vertical axis represents the degree of constraint violation, and the horizontal axis is the fitness value. The union of parent and offspring population consists of four parts: 1) The non-dominated infeasible; 2) the dominated infeasible; 3) the best feasible; 4) the dominated feasible. Part one and part two are in infeasible regions, while part three and part four are in feasible regions.

As shown in Fig. 1, DCMOEA can only preserve at most one feasible solution in the non-dominated front, that is the best feasible one. The other feasible solutions are discarded since they are dominated by the best feasible one. This way, DCMOEA searches almost from infeasible regions. It should be noted that the discarded dominated feasible solutions locate at the feasible regions. Keeping the promising dominated feasible solutions in the population will be beneficial to search the global optimum from feasible regions, especially when multiple solutions are positioned within the feasible region where the global optimum is.

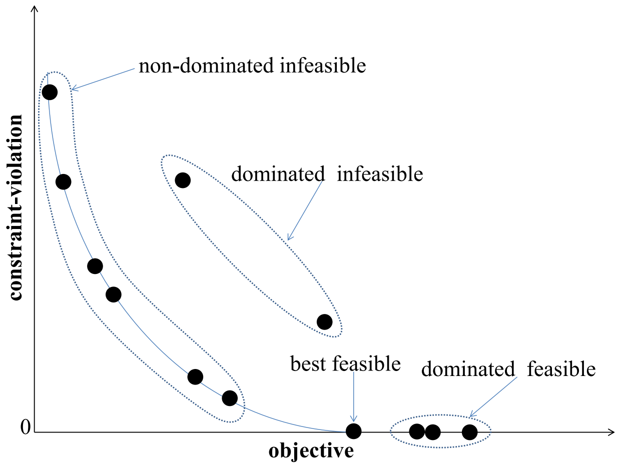


Fig. 1. The union of parent and offspring population consisting of four parts.

4.2. Controlling feasible ratio

Since constraints split the search space into feasible and infeasible regions, the way to design a reasonable selection strategy, which takes full advantage of the information in such regions, naturally becomes a key issue of handling constraints. To realize it, we design a selection strategy by controlling the feasible-infeasible ratio of the population.

Algorithm 1 shows the pseudo code of the selection strategy with feasible-infeasible control technique, where P_t and

Algorithm 1 Select the next parent population.

Input: N: population size; M: number of feasible solutions to be reserved.
Output: P_{t+1}: population at generation $t+1$.

1: $R_t = P_t \bigcup Q_t$. /* Q_t represents the offspring population at generation t */
2: $P_{t+1} = \emptyset$.
3: $S_{fea} = \{ x|x$ is feasible, $x \in R_t \}$. /* S_{fea} denotes the feasible set */
4: **if** $|S_{fea}| \leq M$ **then**
5: $\quad P_{t+1} = P_{t+1} \bigcup S_{fea}$.
6: **else**
7: $\quad$ Apply non-dominated sorting and pruning methods [8] to truncate the feasible set S_{fea} of size M.
8: $\quad P_{t+1} = P_{t+1} \bigcup S_{fea}$.
9: **end if**
10: Remove P_{t+1} from R_t.
11: Apply non-dominated sorting and pruning methods to truncate the R_t of size $N - |P_{t+1}|$.
12: $P_{t+1} = P_{t+1} \bigcup R_t$.

Q_t are the parent and offspring populations at generation t, respectively, N is the population size and M is the number of feasible solutions to be reserved. The main idea is to select feasible and infeasible solutions with a certain ratio to generate the next population. We first select M feasible solutions based on non-dominated sorting and pruning methods [8] and add them to the population of the next generation. The selection of feasible solutions is actually a two-objective selection problem: The original objective $f(\mathbf{x})$ and the niche-count objective $nc(\mathbf{x}|\mathbf{U}, \sigma)$. Optimizing the original objective is beneficial to the improvement of the fitness value in feasible regions, and optimizing the niche-count can maintain the diversity in feasible regions. The rest of the $2N - M$ solutions also undergo the non-nominated sorting and pruning methods (see line 11 of Algorithm 1). The process of this step is a tri-objective selection procedure, and the rest of $2N - M$ solutions may contain many infeasible solutions. This process represents the search from infeasible regions. Note that the final number of selected feasible solutions may be a little over M due to the tri-objective selection procedure. However, it will not influence the performance of the proposed feasible-ratio control mechanism.

Fig. 2(a) and Fig. 2(b) show the comparison of the selection of the next parent population (in the objective space) for DCMOEA and FRC-CEA, respectively. In the selection process, from Fig. 2(a), we can see that point A is first discarded as it is ε-infeasible. Point G locates in the intersection between the non-dominated front and the horizontal axis, thus it is the feasible non-dominated solution which is the best one in the current population. Points on the horizontal axis are feasible solutions. For DCMOEA, points H, I and J are discarded during the selection process because they are not in the non-dominated front, although they can provide valuable information about the optimal direction of improvement for the

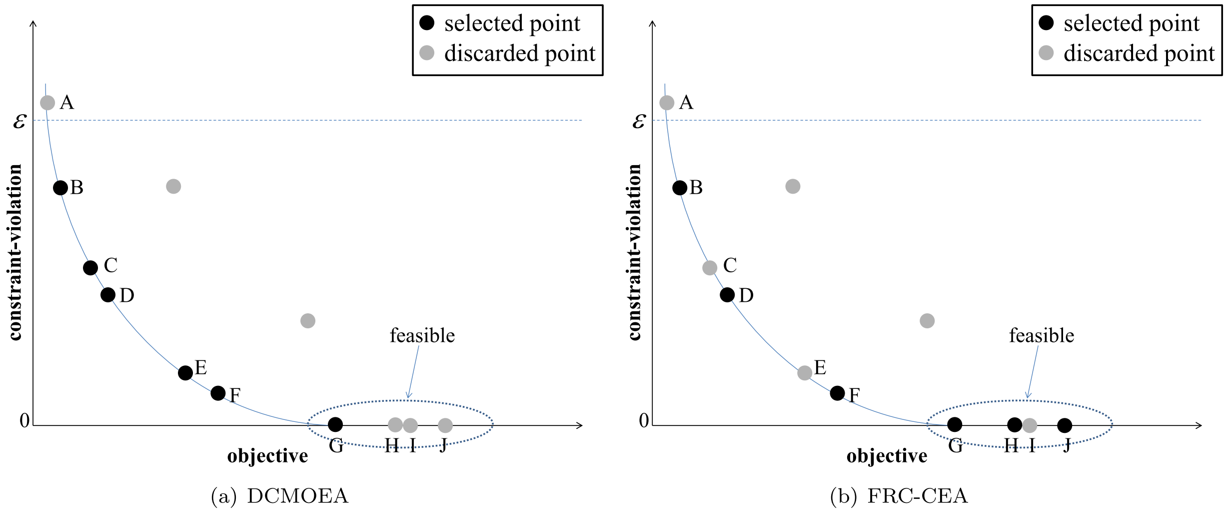


Fig. 2. Graph representation for selecting the next parent population (objective space).

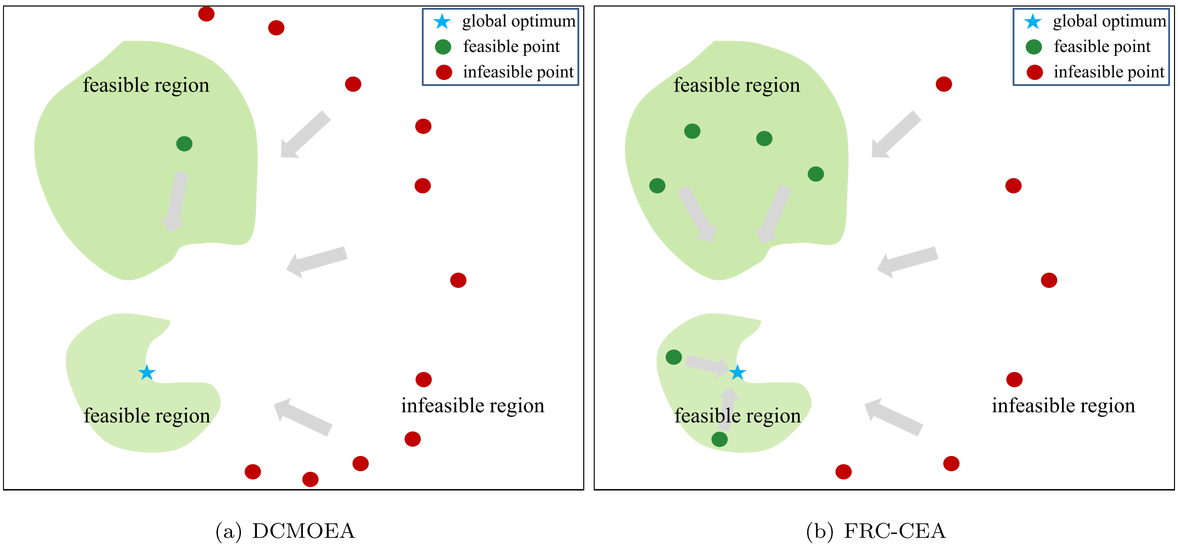


Fig. 3. Schematic diagram illustrating the tendency of movement of solutions (decision space).

objective value from feasible regions. By contrast, from Fig. 2(b), suppose M is 3, among all the feasible points which lie in the horizontal axis, FRC-CEA first selects 3 feasible points G, H and J based on the trade-off among their objective and niche-count values, and directly add these three promising feasible points to the next parent population. Then, the remaining points will be selected based on the trade-off among objective, constraint-violation and niche-count values. Through the above selection procedure, we can see that selected points can maintain the proper diversity both in feasible and infeasible regions.

Fig. 3(a) and Fig. 3(b) illustrate the search behavior of DCMOEA and FRC-CEA in decision space, respectively. As shown in Fig. 3(a), DCMOEA searches the global optimum mainly from infeasible regions where most of the solutions are located. On the contrary, for FRC-CEA in Fig. 3(b), a part of promising feasible solutions are reserved; thus the search is from both feasible and infeasible regions. It is obvious that the population from disparate directions of both feasible and infeasible regions of FRC-CEA is easier to locate the global optimum than one single direction of feasible regions of DCMOEA.

4.3. Proposed algorithm

As shown in Algorithm 2, FRC-CEA consists of two stages for handling COPs with different aims. In the first stage, an enhanced DCMOEA with a feasible-ratio control technique aims to maintain a trade-off between the infeasible search and feasible search. While in the second stage, the feasibility rule is employed to speed up the convergence.

Algorithm 2 The proposed FRC-CEA framework.

```
Initialize a parent population P_0 and set the generation number t = 0.
if t < α × maxT    /*maxT is the maximum generation number*/
    The first-stage optimization.    /* Algorithm 3*/
else
    The second-stage optimization.
end if
t = t + 1.
Output the best individual x_best.
```

4.3.1. The first stage

Algorithm 3 shows the framework of the first stage optimization. The Pareto dominance principle is replaced by $\boldsymbol{\varepsilon}$-constrained Pareto dominance principle. Since any existing MOEAs can be integrated into the algorithm framework [42], so a classical and representative MOEA: NSGA-II [8] is selected to instantiate the algorithm. $DE/rand/1/bin$ operator [31] is used to generate offspring owing to its simplicity and efficiency [1,2,36,37,39]. In line 6 of Algorithm 3, the selection of the next parent population P_{t+1} is Algorithm 1.

Algorithm 3 The first-stage optimization.

```
Input: The initial population: P_0;
Output: The last population: P_{t+1}.
1: Initialize the niche radius σ = σ^(0) and the constraint boundary ε = ε^(0), and set the problem state s = 0.
2: while the halting criterion is not satisfied do
3:     Reduce σ = σ^(s+1) and ε = ε^(s+1).
4:     Update population ε-feasibility, s = s + 1.
5:     Generate the offspring population Q_t.
6:     Select the next parent population P_{t+1} .    /* Algorithm 1*/
7:     t = t + 1.
8: end while
9: Output P_{t+1}.
```

In the first stage, the feasible-ratio control technique guides the search for global optimum not only from the infeasible space containing solutions with good objective values and low constraint violation, but also exploit the feasible regions containing solutions with good objective values and diversity.

4.3.2. The second stage

For the second-stage optimization, we have the following considerations: the search of the first stage is a trade-off between infeasible search (i.e., $cv(\boldsymbol{x})$ in Eq. (2)) and feasible search (i.e., $f(\boldsymbol{x})$ in Eq. (1)). The termination criterion of the first stage is that the relaxation constraint boundary gradually shrunk into the original constraint boundary in Eq. (6), which pushes the population toward the feasible regions but does not mean that the population converges to the global optimum. To this end, a second-stage optimization is needed for the population converging to the global optimum.

In the first stage, the population is composed of both infeasible solutions and feasible solutions. The constraint difficulty is addressed by the first-stage optimization. In the second stage, the feasibility rule serves as the constraint-handling technique to compare pairwise solutions due to its advantages: easy to implementation, no additional parameters and the ability to rapidly motivate the population toward feasible regions. Comparison of infeasible solutions based on the overall constraint violation aims to push infeasible solutions to feasible regions, while the comparison of two feasible solutions on the objective value improves the overall solution.

The final population of stage one is regarded as the initial population of stage two. After the first-stage optimization, the population may reach a narrow and promising region, so convergence will be regarded as the primary goal of stage two. Due to its powerful search ability [22], and also to unify the search algorithm both in the first and second stages, we still use DE as the search algorithm in the second stage. $DE/best/2$ is selected as the mutation strategy:

$$v_i = x_{best} + F \times (x_{r1} - x_{r2}) + F \times (x_{r3} - x_{r4})$$

where $r1$, $r2$, $r3$ and $r4$ are distinct individual indexes randomly selected, x_{best} is the best individual in the current population based on the feasibility rule. This strategy usually has poor exploration ability but converges quickly. Since the population has approached a small and potential area, $DE/best/2$ is more appropriate than random-solution-based strategies in stage two.

4.4. Time complexity

The computational time complexity of the FRC-CEA is divided into two stages. In the first stage, the computational complexity is governed mainly by three parts:

(1) Considering the running time of FRC-CEA for one generation, the complexity is $O(N^2)$ where N is the population size.
(2) The calculation of the constraint violation with the time complexity $O(mN)$, where m is the number of constraints.
(3) The calculation of the niche-count with the time complexity of $O(nN^2)$, where n is the number of dimensions.

In the second stage, the complexity of feasibility rule based DE is $O(nN)$.

Therefore, the overall computational complexity is $O(N^2 + nN^2 + mN)$ for FRC-CEA. Compared with the DCMOEA [42], FRC-CEA does not impose any severe burden for solving COPs.

5. Experimental studies

In this section, experimental studies are carried out on two well-known benchmark test sets collected from IEEE CEC 2006 [19] and IEEE CEC 2010 [21]. The first problem set contains 24 benchmark test functions with different numbers of variables, while the second problem set consists of 18 benchmark test functions of dimensionality 10 and 30 (D=10 and D=30). The characteristics of these test functions can be found in [19] and [21], respectively.

The performance of FRC-CEA is compared with six state-of-the-art constraint-handling EAs: ICTLBO [41], CMODE [38], DCNSGA-II [42], CACDE [40], ITLBO [36] and AIS [44]. In these competitors, ICTLBO and ITLBO both employ the teaching-learning-based optimization method. DCNSGA-II is an instantiation of the DCMOEA which adopts a dynamic multi-objective technique. CMODE uses a multi-objective selection method and an infeasible solution replace mechanism. CACDE is a very recent work, which utilizes feasibility rule and certain mutation strategies to handle constraints. AIS is a two-stage EA which uses an immune-based algorithm to solve COPs. The results of ICTLBO, CACDE, ITLBO, and AIS were directly taken from the original papers for a fair comparison. The results of CMODE and DCNSGA-II were obtained from the authors' homepage.

5.1. Parameter settings

For a fair comparison, some parameter settings of FRC-CEA are the same as used in [42]:

- The population size: $N = 100$.
- The number of feasible solutions reserved: $M = 50\% \times N$=50 (this will be discussed in Section 6.2).
- The settings of the DE operator: Crossover probability $CR = 0.9$. In the first stage, the scaling factor F is randomly selected due to the need of diversity: $F = rndReal(0, 1)$ [42]. In the second stage, a suggested value $F = 0.5$ [31] is used for the fast convergence purpose.
- The proportion of function evaluations (FEs) of the first stage: $\alpha = 0.9$ (this will be discussed in Section 6.3).
- The number of independent runs: 25 [19,21].
- The number of the maximal number of function evaluations is based on the suggestions in [19,21]: $FEs = 240,000$ for CEC 2006 problems, $FEs = 200,000$ and $FEs = 600,000$ for 10D and 30D of CEC 2010 problems, respectively.

To detect the statistically significant differences, the nonparametric statistical test: Friedmans test is performed to sort the compared algorithms and Wilcoxons rank sum test at significance level $\alpha = 0.05$ is also performed for all algorithms.

5.2. Experiments on CEC 2006 benchmark test functions

The mean and standard deviation of the objective values for each algorithm are shown in Table 1. The best mean result for each problem is highlighted in bold. Note that the $G20$ problem has no feasible solutions, and the $G22$ problem is very hard to obtain a feasible solution. Most of the literature does not take these two functions into consideration. Thus, we also exclude these two problems and mainly focus on the remaining 22 functions. From Table 1, we can see that ICTLBO, CMODE, DCNSGA-II, CACDE, ITLBO, AIS and FRC-CEA can achieve the true global optimum for 16, 17, 18, 20, 15, 15 and 19 problems, respectively. With regard to the mean values, FRC-CEA performs better than ICTLBO, CMODE, DCNSGA-II, CACDE, ITLBO and AIS on 3, 2, 3, 0, 5 and 6 test functions, respectively. In contrast, these six competitors outperform FRC-CEA on 3, 3, 1, 3, 1 and 3 functions, respectively.

The Wilcoxon test results regarding the mean objective values are presented in Table 2. R^+ and R^- represent the sum of ranks. $R^+ > R^-$ means the proposed algorithm is better than the compared method and vice versa. The "+", "−" and "≈" denote that the proposed method is significantly better than, worse than, and statistically equivalent to (the difference is not statistically significant) the compared algorithm. Based on the Wilcoxon test results reported in Table 2, there is no statistical difference between FRC-CEA and the other state-of-the-art algorithms. However, compared with DCNSGA-II, ITLBO, and AIS, FRC-CEA has better R^+ values than R^- values in terms of the objective values.

Based on the mean objective values in Table 1, the average ranks of the seven algorithms according to the Friedman test are shown in Fig. 4, where FRC-CEA is ranked slightly higher than CACDE and CMODE, and lower than ICTLBO, DCNSGA-II, ITLBO, and AIS, which means that FRC-CEA performs a little worse than CACDE and CMODE, and better than ICTLBO, DCNSGA-II, ITLBO, and AIS on CEC2006 benchmark problems.

Table 1
The average results of 7 algorithms after reaching 240,000 FEs of CEC2006 test functions.

Fun		ICTLBO [41]	CMODE [38]	DCNSGA-II [42]	CACDE [40]	ITLBO [36]	AIS [44]	FRC-CEA
G01	Mean	**−15.000**	**−15.000**	**−15.000**	**−15.000**	**−15.000**	**−15.000**	**−15.000**
	Std	0.00E+00	0.00E+00	7.78E−10	0.00E+00	0.00E+00	0.00E+00	2.38E−10
G02	Mean	−0.799622	−0.803179	**−0.803619**	**−0.803619**	−0.80226	−0.802193	**−0.803619**
	Std	5.17E−03	2.20E−03	7.20E−08	7.99E−08	3.26E−03	5.19E−10	7.57E−08
G03	Mean	**−1.0005**	**−1.0005**	**−1.0005**	**−1.0005**	**−1.0005**	**−1.0005**	**−1.0005**
	Std	5.17E−03	5.92E−10	6.01E−07	0.00E+00	2.58E−09	1.77E−11	6.48E−10
G04	Mean	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**
	Std	7.43E−12	3.71E−12	2.56E−08	3.71E−12	3.71E−12	3.69E−13	7.80E−12
G05	Mean	**5126.4967**	**5126.4967**	**5126.4967**	**5126.4967**	**5126.4967**	5126.4981	**5126.4967**
	Std	1.86E−12	2.78E−12	3.41E−07	2.66E−12	2.78E−12	1.70E−02	7.43E−13
G06	Mean	**−6961.8139**	**−6961.8139**	**−6961.8139**	**−6961.8139**	**−6961.8139**	**−6961.8139**	**−6961.8139**
	Std	3.71E−12	0.00E+00	5.59E−08	0.00E+00	0.00E+00	1.90E−12	2.73E−12
G07	Mean	**24.3062**	**24.3062**	**24.3062**	**24.3062**	**24.3062**	24.3557	**24.3062**
	Std	5.40E−14	6.32E−15	2.11E−04	7.64E−15	1.51E−05	8.20E−03	3.64E−05
G08	Mean	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**
	Std	0.00E+00	0.00E+00	0.00E+00	3.59E−07	0.00E+00	0.00E+00	0.00E+00
G09	Mean	**680.630**	**680.630**	**680.630**	**680.630**	**680.630**	**680.630**	**680.630**
	Std	4.64E−13	3.64E−13	1.56E−06	3.60E−13	3.36E−13	0.00E+00	2.49E−13
G10	Mean	7049.313	**7049.248**	7049.249	**7049.248**	7049.249	7049.570	**7049.248**
	Std	8.39E−02	3.26E−12	8.24E−04	2.72E−12	4.29E−05	4.50E−04	1.40E−04
G11	Mean	**0.7499**	**0.7499**	**0.7499**	**0.7499**	**0.7499**	**0.7499**	**0.7499**
	Std	0.00E+00	0.00E+00	1.61E−07	7.46E−05	0.00E+00	1.40E−08	0.00E+00
G12	Mean	**−1.000**	**−1.000**	**−1.000**	**−1.000**	**−1.000**	**−1.000**	**−1.000**
	Std	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
G13	Mean	0.207886	**0.053942**	**0.053942**	**0.053942**	0.054008	**0.053942**	**0.053942**
	Std	1.92E−01	0.00E+00	1.55E−07	3.98E−12	3.30E−04	7.80E−10	0.00E+00
G14	Mean	**−47.7649**	**−47.7649**	**−47.7649**	**−47.7649**	**−47.7649**	**−47.7649**	**−47.7649**
	Std	2.10E−08	2.34E−14	2.43E−04	2.24E−14	3.80E−05	1.00E−12	3.05E−04
G15	Mean	**961.7150**	**961.7150**	**961.7150**	**961.7150**	**961.7150**	**961.7150**	**961.7150**
	Std	4.64E−13	5.80E−13	1.71E−07	3.42E−11	5.80E−13	0.00E+00	4.55E−13
G16	Mean	**−1.9052**	**−1.9052**	**−1.9052**	**−1.9052**	**−1.9052**	**−1.9052**	**−1.9052**
	Std	2.79E−15	0.00E+00	7.04E−06	0.00E+00	0.00E+00	0.00E+00	0.00E+00
G17	Mean	8880.5953	8853.6182	8895.0034	**8853.5340**	8859.8	8853.5397	8872.1108
	Std	3.69E+01	4.22E−01	3.92E+01	3.14E−04	3.77E+01	1.90E−09	3.12E+01
G18	Mean	**−0.866025**	**−0.866025**	**−0.866025**	**−0.866025**	**−0.866025**	**−0.866025**	**−0.866025**
	Std	1.48E−13	0.00E+00	1.57E−06	0.00E+00	1.68E−05	1.30E−15	4.25E−10
G19	Mean	32.6570	32.6558	32.6572	**32.6556**	32.662	**32.6556**	32.6614
	Std	1.56E−03	6.05E−05	1.80E−03	5.79E−10	1.06E−02	0.00E+00	1.28E−02
G21	Mean	**193.7245**	230.3984	**193.7245**	**193.7245**	222.22	196.7245	**193.7245**
	Std	6.00E−11	6.00E+01	1.27E−07	3.31E−11	4.84E+01	1.10E+00	4.56E−10
G23	Mean	−400.0372	**−400.0547**	−393.1349	−399.9922	−256.4	−399.8743	−398.2812
	Std	7.44E−02	6.05E−04	1.86E+01	3.15E−01	1.42E+02	2.00E+00	6.27E+00
G24	Mean	**−5.5080**	**−5.5080**	**−5.5080**	**−5.5080**	**−5.5080**	**−5.5080**	**−5.5080**
	Std	5.66E−15	0.00E+00	3.58E−11	0.00E+00	0.00E+00	0.00E+00	0.00E+00
+		**3**	**2**	**3**	**0**	**5**	**6**	\
−		**3**	**3**	**1**	**3**	**3**	**3**	\
≈		**16**	**17**	**18**	**19**	**16**	**13**	\

"+", "-", and "≈" represent that the performance of the FRC-CEA is better than, worse than, and statistically equivalent to that of the corresponding method, respectively.

In summary, we can conclude that the FRC-CEA provides highly competitive results compared with the state-of-the-art constraint-handling EAs in the CEC 2006 test suite.

5.3. Experiments on CEC 2010 benchmark test functions with 10-D and 30-D

Compared with CEC 2006 test suite, IEEE CEC 2010 test suite [21] is much more complicated. 12 out of 18 test problems have more than one equality constraints, and 13 out of 18 test problems contain extremely tiny feasible regions. As a result, solving these test problems are much more challenging. The detailed results of all these seven algorithms on CEC 2010 problems are shown in Tables 3 and 4, where "*" represents that feasible solutions cannot be consistently found by the corresponding algorithm in all runs. Note that a value less than 1E-15 is treated as 0.

In the case of D = 10, we can observe that a significant difference can be detected on most of the test instances. Particularly, Table 3 shows that the number of the test problems where FRC-CEA significantly outperforms ICTLBO, CMODE, DCNSGA-II, CACDE, ITLBO, and AIS is 13, 14, 6, 7, 6 and 11 at a 0.05 significance level, respectively, which is larger than the number of problems where the performance of the FRC-CEA is worse than its peer algorithms. The Friedman's test summa-

Table 2
Results of the Multiple-Problem Wilcoxon's test for CEC2006.

Algorithm	R^+	R^-	$\alpha = 0.05$
FRC-CEA vs ICTLBO	8.0	13.0	$\approx$
FRC-CEA vs CMODE	6.0	9.0	$\approx$
FRC-CEA vs DCNSGA-II	8.0	2.0	$\approx$
FRC-CEA vs CACDE	0.0	6.0	$\approx$
FRC-CEA vs ITLBO	17.0	4.0	$\approx$
FRC-CEA vs AIS	26.0	19.0	$\approx$

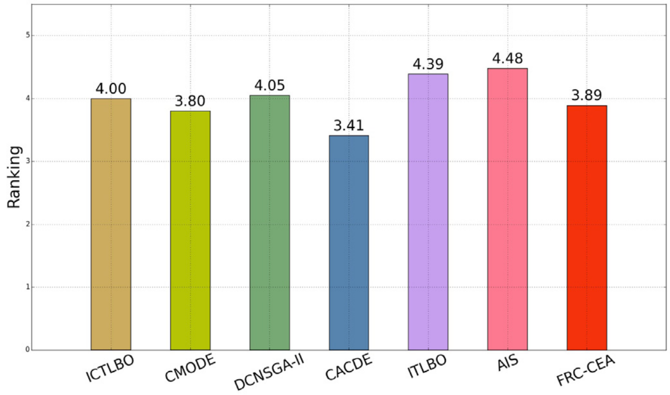


Fig. 4. Average ranks of 7 algorithms for CEC2006 by Friedman's test.

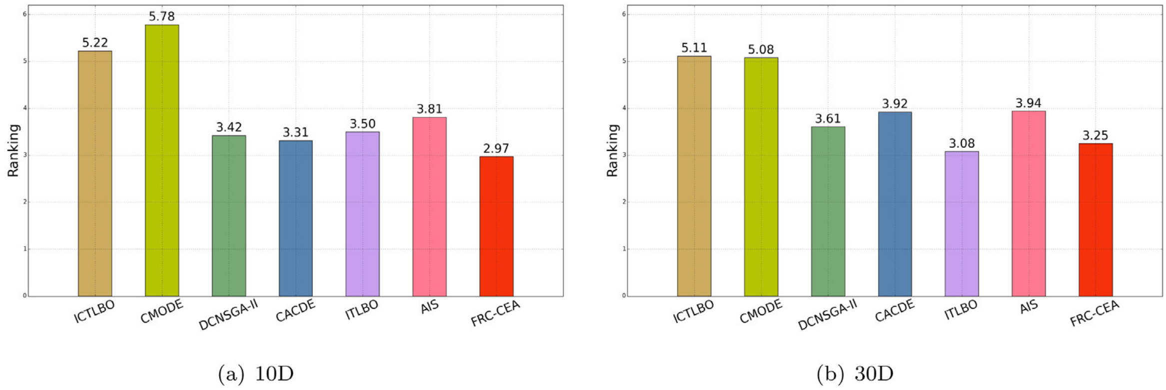


Fig. 5. Average ranks of 7 algorithms for CEC2010 with 10D and 30D by Friedman's test.

rized in Fig. 5(a) shows that FRC-CEA achieves the first rank among all the peer algorithms. For the statistical results of the Wilcoxon-test, as shown in Table 5, FRC-CEA is significantly better than ICTLBO and CMODE at $\alpha = 0.05$.

In the case of D = 30, from Table 4, FRC-CEA outperforms ICTLBO, CMODE, DCNSGA-II, CACDE, ITLBO, and AIS on 15, 11, 6, 13, 6 and 10 test functions at a 0.05 significance level, respectively. Nevertheless, ICTLBO, CMODE, DCNSGA-II, CACDE, ITLBO, and AIS outperform FRC-CEA on 2, 4, 5, 5, 10 and 8 test functions, respectively. From Table 5, it is obvious that FRC-CEA provides higher R^+ values than R^- values in all the cases except for ITLBO. In addition, with respect to the average rankings of different algorithms by the Friedman test, we can observe from Fig. 5(b) that FRC-CEA gets the second ranking among all the competitors.

For problem C01, the fitness landscape contains multiple uneven optima, and the feasible regions are very spacious. The global optimum locates on the boundary between feasible and infeasible regions. As expected, good results are obtained by FRC-CEA on C01 (10D and 30D) because a certain proportion of feasible and infeasible solutions are preserved, respectively. In addition, dynamically reducing the niche radius can maintain proper diversity in both feasible and infeasible regions, and helps the population jump out of the local optima. For a COP with equality constraints, like C03, after an equality constraint function $h(\boldsymbol{x})=0$ is transformed into an inequality constraint function $|h(\boldsymbol{x})| - \delta \leq 0$, the feasible regions become a hyperplane. The feasible-ratio control technique can preserve some solutions on the hyperplane, which could help the population avoid some locally feasible optima.

Table 3
The average results of 7 algorithms for CEC2010 test functions at D=10.

Fun		ICTLBO [41]	CMODE [38]	DCNSGA-II [42]	CACDE [40]	ITLBO [36]	AIS [44]	FRC-CEA
C01	Mean	−7.44E−01	**−7.47E−01**	−7.38E−01	**−7.47E−01**	**−7.47E−01**	**−7.47E−01**	**−7.47E−01**
	Std	5.45E−03	2.35E−13	3.77E−03	1.88E−03	1.87E−03	1.30E−03	4.49E−13
C02	Mean	−1.72E+00	−1.48E+00*	**−2.28E+00**	−2.26E+00	−2.03E+00	**−2.28E+00**	**−2.28E+00**
	Std	8.14E−01	4.88E−01	2.04E−05	6.75E−02	8.14E−02	2.00E−03	3.76E−03
C03	Mean	1.41E+09	2.84E+00	3.55E−01	**0.00E+00**	**0.00E+00**	3.75E−09	**0.00E+00**
	Std	6.01E+09	4.23E+00	1.74E+00	0.00E+00	0.00E+00	4.81E−04	0.00E+00
C04	Mean	**−1.00E−05**	**−1.00E−05**	**−1.00E−05**	**−1.00E−05**	**−1.00E−05**	−9.97E−06	2.75E−05
	Std	0.00E+00	0.00E+00	3.04E−10	0.00E+00	3.39E−15	4.28E−03	9.14E−05
C05	Mean	−6.83E+01	−4.50E+02*	**−4.84E+02**	**−4.84E+02**	**−4.84E+02**	−4.80E+02	**−4.84E+02**
	Std	3.36E+01	1.61E+02	7.13E−05	3.48E−13	1.11E−11	6.30E+00	3.36E−02
C06	Mean	−5.46E+02	−5.78E+02	−5.79E+02	−5.79E+02	−5.79E+02	**−5.80E+02**	−5.79E+02
	Std	3.30E+01	1.60E−02	1.05E−10	1.68E−02	2.39E−04	7.30E−08	2.63E−05
C07	Mean	**0.00E+00**	6.69E−15	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.17E−08	**0.00E+00**
	Std	0.00E+00	8.95E−15	0.00E+00	0.00E+00	0.00E+00	2.70E+00	0.00E+00
C08	Mean	1.72E+01	8.94E+00	**3.26E+00**	7.01E+00	8.47E+00	4.09E+00	3.30E+00
	Std	2.64E+01	3.98E+00	4.34E+00	5.01E+00	4.09E+00	1.46E+00	4.54E+00
C09	Mean	3.56E+05	2.13E+06*	**0.00E+00**	2.10E+01	**0.00E+00**	2.70E+01	1.39E−01
	Std	1.00E+06	1.04E+07	0.00E+00	3.51E+01	0.00E+00	7.50E+01	1.10E+00
C10	Mean	1.32E+06	1.35E+05*	2.50E+01	6.59E+01	**1.92E−01**	1.62E+03	1.42E+01
	Std	6.59E+06	1.61E+06	2.04E+01	4.40E+01	9.62E−01	5.00E+02	1.95E+01
C11	Mean	**−1.52E−03**	−7.70E−02*	2.51E−03*	**−1.52E−03**	−1.51E−03	−9.20E−04	−1.15E−02*
	Std	4.83E−14	2.85E−02	7.18E−03	1.30E−06	1.30E−05	8.23E−04	4.01E−02
C12	Mean	5.01E+01	−6.14E+02*	−2.77E+02*	−4.34E+02	−2.39E+01	**−4.36E+02**	−2.17E+02*
	Std	1.41E+02	2.74E+02	2.77E+02	2.49E+02	1.14E+02	6.02E+01	2.66E+02
C13	Mean	**−6.84E+01**	−5.79E+01	**−6.84E+01**	−6.72E+01	−6.52E+01	−6.79E+01	**−6.84E+01**
	Std	4.42E−14	4.09E+00	1.68E−10	1.04E+00	1.78E+00	3.11E−01	3.51E−12
C14	Mean	3.19E−01	8.18E−09	2.35E−15	**0.00E+00**	**0.00E+00**	1.22E−04	**0.00E+00**
	Std	1.10E+00	1.64E−08	0.00E+00	0.00E+00	0.00E+00	2.90E−08	0.00E+00
C15	Mean	3.73E+01	1.20E+02	1.62E+00	3.38E+00	3.54E+00	**5.19E−09**	1.51E+00
	Std	9.47E+01	3.48E+02	1.82E+00	1.02E+00	4.97E+00	1.10E−08	1.83E+00
C16	Mean	4.27E−12	6.82E−05	2.19E−03	4.52E−02	2.27E−01	**0.00E+00**	**0.00E+00**
	Std	2.13E−11	1.49E−04	6.57E−03	1.03E−01	3.11E−01	6.27E−15	0.00E+00
C17	Mean	7.68E+00	4.37E−02	**0.00E+00**	**0.00E+00**	3.91E−01	2.93E+00	**0.00E+00**
	Std	3.72E+01	1.12E−01	0.00E+00	0.00E+00	6.71E−01	2.29E+00	0.00E+00
C18	Mean	2.59E+00	5.75E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	1.66E+00	**0.00E+00**
	Std	1.30E+01	2.64E+02	0.00E+00	0.00E+00	0.00E+00	1.27E+00	0.00E+00
+		**13**	**14**	**6**	**7**	**6**	**11**	\
−		**3**	**1**	**3**	**3**	**5**	**5**	\
≈		**2**	**3**	**9**	**8**	**7**	**2**	\

"+", "-", and "≈" represent that the performance of the FRC-CEA is better than, worse than, and statistically equivalent to that of the corresponding method, respectively. Results with * indicates that there are infeasible solutions over 25 independent runs.

Overall, from the above discussions, we can conclude that FRC-CEA is better than or comparable to state-of-the-art algorithms tested in this paper on problems CEC 2010 in both $D = 10$ and $D = 30$ in terms of the solution quality and statistical results.

6. Discussions

In this section, we present some insight discussions and carry out some additional experiments to further investigate the working mechanisms of FRC-CEA.

6.1. Investigation of the search behavior

First, to visualize how FRC-CEA works, we take three instances (*ATF*1, *ATF*2, and *ATF*3) borrowed from [39] as examples to illustrate the search behavior of the FRC-CEA. *ATF*1 contains an equality constraint and two inequality constraints, so the feasible region is very narrow as shown in Fig. 6(a). *ATF*2 has two disjoint feasible regions, and the global optimum lies within the one with a smaller region as shown in Fig. 6(b). *ATF*3 only has one inequality constraint, but the global optimum is located on the boundary of the feasible region as shown in Fig. 6(c). We set the population size 40 as in [39].

6.1.1. For problems with equality constraint

Fig. 7 visualizes the search process of the FRC-CEA at four different generations over a single run on *ATF*1. In Fig. 7(a), the population is randomly generated and widely distributed in the first generation, and all individuals are infeasible due to the narrow feasible region. At the 80th generation, we can see that the feasible region and infeasible region all contain some individuals. In the infeasible region, some individuals fall into the local basins of attraction of the infeasible region due to

Table 4
The average results of 7 algorithms for CEC2010 test functions at D=30.

Fun		ICTLBO [41]	CMODE [38]	DCNSGA-II [42]	CACDE [40]	ITLBO [36]	AIS [44]	FRC-CEA
C01	Mean	−8.18E−01	**−8.21E−01**	−8.06E−01	−8.20E−01	−8.20E−01	−8.20E−01	**−8.21E−01**
	Std	2.97E−03	3.30E−03	9.80E−03	2.67E−03	8.95E−04	3.25E−04	1.88E−03
C02	Mean	−3.83E−01	9.75E−01	**−2.28E+00**	−2.01E+00	−2.03E+00	−2.21E+00	−2.27E+00
	Std	1.72E+00	6.25E+01	2.32E−03	7.78E−02	7.64E−02	2.84E−03	6.06E−03
C03	Mean	2.13E+11	**2.18E+01**	2.87E+01	3.08E+01	7.84E+01	6.68E+01	2.87E+01
	Std	3.35E+11	1.25E+01	6.50E−09	3.50E+01	6.31E+01	4.26E+03	1.22E−07
C04	Mean	1.59E−01	6.72E−04	**9.42E−05**	3.54E+00	1.69E−03	1.98E−03	4.17E−03
	Std	3.24E−01	4.24E−04	6.58E−05	7.62E+00	1.14E−03	1.61E−03	4.17E−03
C05	Mean	−5.98E+01	2.77E+02*	**−4.83E+02**	−3.41E+02	−4.82E+02	−4.36E+02	−4.80E+02
	Std	8.86E+00	2.03E+02	1.70E−01	8.69E+01	1.73E+00	2.51E+01	2.45E+00
C06	Mean	−4.64E+02	−4.96E+02*	**−5.31E+02**	−5.22E+02	−5.30E+02	−4.54E+02	**−5.31E+02**
	Std	9.19E+01	2.15E+02	6.56E−03	2.92E+00	4.80E−01	4.79E+01	1.82E−02
C07	Mean	2.88E+01	**5.24E−05**	4.70E+01	9.57E−01	1.59E−01	1.07E+00	3.88E−01
	Std	4.68E+01	5.89E−05	2.87E+01	1.74E+00	7.97E−01	1.61E+00	1.05E+00
C08	Mean	1.01E+02	**3.68E−01**	3.68E+01	9.76E+00	1.14E+01	1.65E+00	1.43E+01
	Std	1.22E+02	2.62E−01	3.34E+01	3.20E+01	2.79E+01	6.41E−01	3.34E+01
C09	Mean	1.01E+07	1.72E+13*	2.29E+00	9.23E+03	2.86E+00	**1.57E+00**	4.59E+01
	Std	3.52E+07	1.07E+13	6.23E+00	1.26E+04	1.43E+01	1.96E+00	3.75E+01
C10	Mean	6.01E+09	1.60E+13*	9.50E+01	8.20E+10	3.29E+01	**1.78E+01**	9.05E+01
	Std	2.31E+10	7.00E+12	8.65E+01	3.91E+11	1.41E+01	1.88E+01	9.05E+01
C11	Mean	−3.65E−04	9.50E−03*	2.06E−02*	2.99E−03	**−3.86E−04**	−1.58E−04	−3.72E−02*
	Std	4.98E−05	9.70E−03	6.31E−02	7.14E−03	1.14E−05	4.67E−05	4.58E−01
C12	Mean	**−1.99E−01**	−3.46E+00*	−4.48E+02*	**−1.99E−01**	−1.98E−01	4.29E−06	−1.61E+02*
	Std	6.15E−05	7.35E+02	4.31E+02	2.35E−04	2.39E−03	4.52E−04	3.26E+02
C13	Mean	−6.81E+01	−3.89E+01	**−6.84E+01**	−6.77E+01	−5.05E+01	−6.62E+01	**−6.84E+01**
	Std	7.78E−01	2.17E+00	1.62E−08	6.88E−01	1.18E+00	2.27E−01	2.88E−01
C14	Mean	8.02E+00	9.31E+00	4.64E+01	**0.00E+00**	4.78E−01	8.68E−07	6.19E−01
	Std	8.69E+00	2.46E+00	2.61E+01	0.00E+00	1.32E+00	3.14E−07	6.20E−01
C15	Mean	2.91E+01	1.51E+13	2.44E+01	2.17E+01	2.38E+01	3.41E+01	**2.16E+01**
	Std	3.63E+01	8.26E+12	1.55E+01	2.45E−01	2.51E+01	3.82E+01	9.92E−05
C16	Mean	**0.00E+00**	6.30E−02	**0.00E+00**	6.03E−04	**0.00E+00**	8.21E−02	**0.00E+00**
	Std	0.00E+00	2.72E−02	0.00E+00	3.02E−03	0.00E+00	1.12E−01	0.00E+00
C17	Mean	3.29E+01	3.12E+02*	4.75E+00	**8.24E−01**	9.65E−01	3.61E+00	7.02E+00
	Std	1.35E+02	2.75E+02	9.77E+00	6.85E−01	1.73E+00	2.54E+00	1.27E+01
C18	Mean	8.82E−04	7.36E+03	**0.00E+00**	2.35E−05	**0.00E+00**	4.02E+01	**0.00E+00**
	Std	3.22E−03	3.12E+03	0.00E+00	8.46E−05	9.77E+00	1.80E+01	0.00E+00
+		**15**	**11**	**6**	**13**	**6**	**10**	\
−		**2**	**4**	**5**	**5**	**10**	**8**	\
≈		**1**	**3**	**7**	**0**	**2**	**0**	\

"+", "-", and "≈" represent that the performance of the FRC-CEA is better than, worse than, and statistically equivalent to that of the corresponding method, respectively. Results with * indicates that there are infeasible solutions over 25 independent runs.

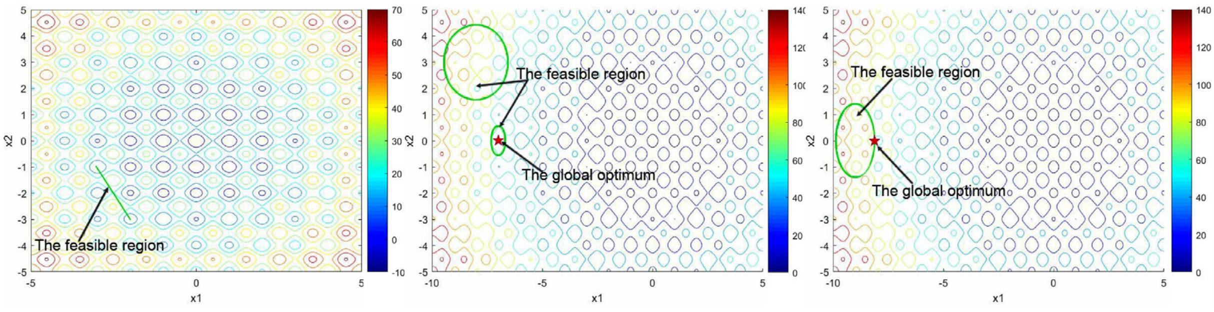
Table 5
Results of the Multiple-Problem Wilcoxon's test for CEC2010 (D = 10 and D = 30).

Algorithm	D = 10			D = 30		
	R^+	R^-	$\alpha = 0.05$	R^+	R^-	$\alpha = 0.05$
FRC-CEA vs ICTLBO	109.0	27.0	+	126.0	27.0	+
FRC-CEA vs CMODE	117.0	3.0	+	104.0	16.0	+
FRC-CEA vs DCNSGA-II	31.0	14.0	≈	44.0	22.0	≈
FRC-CEA vs CACDE	35.0	20.0	≈	110.0	61.0	≈
FRC-CEA vs ITLBO	32.0	34.0	≈	44.0	92.0	≈
FRC-CEA vs AIS	82.0	89.0	≈	70.0	50.0	≈

the multi-modal property. At the 180th generation, the population approaches to a small region, and infeasible individuals toward the feasible region from different directions. At the 300th generation, the second-stage optimization pushes the population to converge to the global optimum.

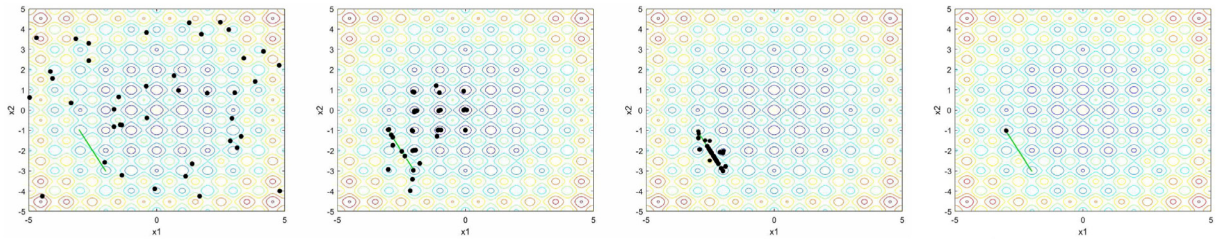
6.1.2. For problems with separate feasible regions

In Fig. 8 on *ATF*2 instance, we can observe that the initial population only contains two feasible individuals, which are not in the feasible region where the global optimum located. At the 100th generation, the population towards the two feasible sub-regions from various directions. As expected, at the 200th generation, the population has been close to the global optimum, and feasible and infeasible individuals are about half of the population size due to the feasible-ratio control mechanism. At the 300th generation, the second-stage optimization drives the population to the global optimum.



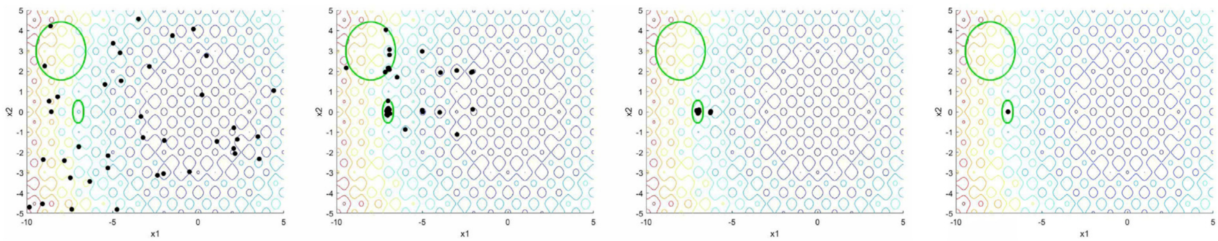
(a) ATF1 (b) ATF2 (c) ATF3

Fig. 6. Visualization of the decision space, the contours of objective function, the feasible region, and the global optimum of three constrained optimization instances.



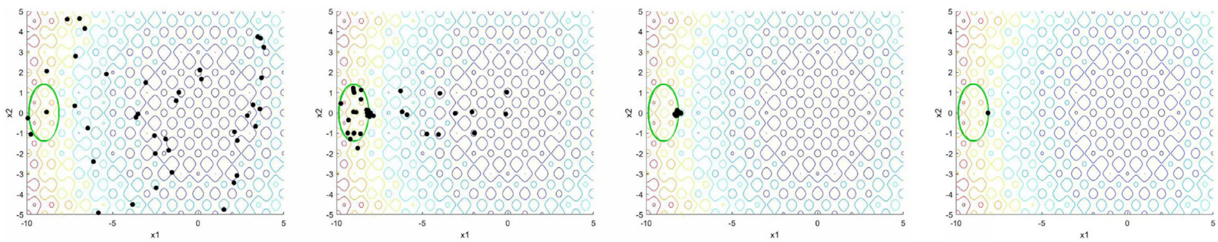
(a) Initial population (g=1) (b) Stage one (g=80) (c) Stage one (g=180) (d) Stage two (g=300)

Fig. 7. Visualization of the FRC-CEA on instance ATF1 over a single run.



(a) Initial population (g=1) (b) Stage one (g=100) (c) Stage one (g=200) (d) Stage two (g=300)

Fig. 8. Visualization of the FRC-CEA on instance ATF2 over a single run.

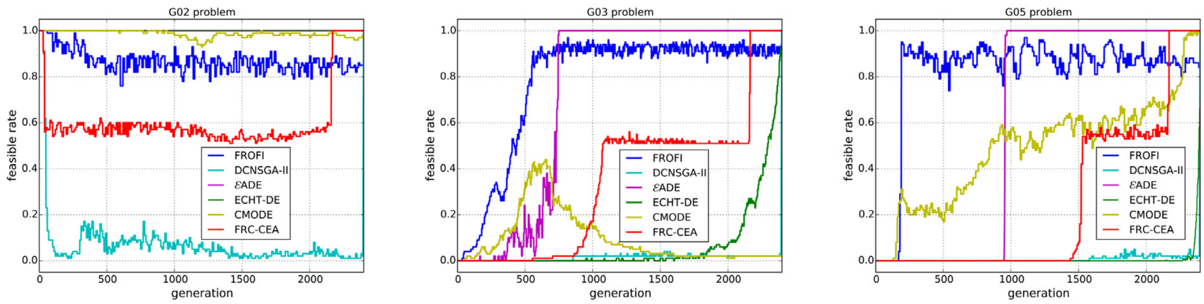


(a) Initial population (g=1) (b) Stage one (g=50) (c) Stage one (g=200) (d) Stage two (g=300)

Fig. 9. Visualization of the FRC-CEA on instance ATF3 over a single run.

6.1.3. For problems with the optimum on the boundary of the feasible region

From Fig. 9(b), we can see that both feasible region and infeasible region contain many points and they have a wide distribution. At the 200th generation from Fig. 9(c), the population is close to the global optimum from both feasible and infeasible regions. At the 300th generation, all individuals converge to the global optimum. This also provides a detailed insight into how the FRC-CEA works during the evolution search process.



(a) G02 for inequality problem (b) G03 for equality problem (c) G05 for inequality&equality problem

Fig. 10. The feasible rate during the optimization process.

Table 6
Summary results of the different percentage of feasible solutions reserved for CEC2010 test functions (10D and 30D).

50% vs	10%	30%	70%	90%	99%
+	11	10	11	11	15
–	8	7	5	7	4
≈	17	19	20	18	17

"+", "–", and "≈" represent that the performance of the FRC-CEA with 50% feasible solutions is better than, worse than, and statistically equivalent to that of the FRC-CEA with the corresponding percentage feasible solutions at a 0.05 significance level, respectively.

6.1.4. Comparison of the feasible ratio

To investigate the search behavior of different algorithms, we compare the feasible ratio of six algorithms: εADE [34], CMODE [38], DCNSGA-II [42], FROFI [39], ECHT-DE [20], and FRE-CEA. We select these algorithms because they represent different types of constraint handling optimiziers. εADE utilizes ε constraint method to handle constraints. CMODE and DCNSGA-II deal with COPs by a multi-objective optimization technique. FROFI is adopted to make a comparison in term of the feasibility rule it used, while ECHT-DE adopts ensemble of constraint handling techniques.

We take three representative problems ($G02$, $G03$, and $G05$ [19]) to investigate the behaviors of different algorithms. $G02$ only has inequality constraints; $G03$ only contains equality constraints; while $G05$ has both inequality and equality constraints. The change of feasible ratio during the run-time of different algorithms on these three problems are shown in Fig. 10. From Fig. 10, FROFI can quickly find feasible solutions because it adopts the feasibility rule to handle constraints. Once it finds feasible solutions, the feasible ratio maintains at a high level which indicates it mainly searches the global optimum from feasible regions. DCNSGA-II and FRC-CEA enter the feasible region slower than FROFI. The reason is that they gradually shrink the relaxation constraint boundary. However, the feasible ratio of DCNSGA-II is significantly low, about 2%-3% even for $G02$ problem, which has 99.99% of feasible regions. This can be understandable due to the multi-objective selection strategy, that is, at most one feasible solution is reserved in each non-dominated front. For FRC-CEA, initially the feasible ratio is very low for $G02$ and $G03$. However, after it reaches feasible regions, the feasible ratio of FRC-CEA increases dramatically. Compared with FROFI, DCNSGA-II, CMODE, and ECHT-DE, the feasible ratio of FRC-CEA is consistently about 50%-60%, which means the search for the global optimum of FRC-CEA are from both feasible and infeasible regions.

6.2. The choice of the feasible ratio

Intuitively, individuals in a population should approach the global optimum from both feasible and infeasible regions with approximately the same size, so that balance the search between the feasible and infeasible regions of the search space. To find out if this is a good choice, FRC-CEA was run with different percentages of feasible solutions: 10%, 30%, 50%, 70%, 90% and 99%. The CEC2010 test suite with 10D and 30D was used as the test suit. Due to the limitation of paper length, only the significance test results are presented in Table 6. Their performance was compared by the Wilcoxon's rank sum test at a 0.05 significance level, where " + ", "–", and "≈" denote that FRC-CEA with about 50% feasible solutions reserved was better than, worse than, and statistically equivalent to that of the FRC-CEA with corresponding percentage feasible solutions, respectively.

The results show that FRC-CEA with about 50% feasible solutions reserved performs better than FRC-CEA-10%, FRC-CEA-30%, FRC-CEA-70%, FRC-CEA-90% and FRC-CEA-99% on most instances. More specifically, FRC-CEA-50% achieves better results on 11, 10, 11, 11 and 15 instances compared to FRC-CEA-10%, FRC-CEA-30%, FRC-CEA-70%, FRC-CEA-90% and FRC-CEA-99%,

Table 7
Summary results of the different α for CEC2010 test functions (10D and 30D).

$\alpha = 0.9$ vs	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.8$	$\alpha = 1.0$
+	19	15	14	14	14
−	6	9	8	3	4
≈	11	12	14	19	18

"+", "−", and "≈" represent that the performance of the FRC-CEA with $\alpha = 0.9$ is better than, worse than, and statistically equivalent to that of the FRC-CEA with corresponding α value at a 0.05 significance level, respectively.

respectively. These results indicate that the FRC-CEA with about 50% feasible solutions reserved is the best choice among FRC-CEA-10%, FRC-CEA-30%, FRC-CEA-70%, FRC-CEA-90% and FRC-CEA-99%.

In addition, often *a priori* knowledge about test problems is not available, making it hard to know the geometric properties of the test functions (e.g., the shape and the range of feasible regions). So reserving the same number of feasible and infeasible solutions (FRC-CEA-50%) is a desirable and reliable choice, which can provide a trade-off between feasible search and infeasible search.

6.3. Parameter sensitivity analysis

In this subsection, FRC-CEA is tested with different values of α, where α represents the proportion of the number of function evaluations of the first stage optimization to the total number of function evaluations(*FEs*).

A too large α would result in poor convergence while a too small α would run the high risk of falling into local optima. Hence, parameter α should be chosen properly to achieve a tradeoff between diversity and convergence. For this purpose, six values: $\alpha = 0.5$, 0.6, 0.7, 0.8, 0.9 and 1.0 were evaluated on the 36 test functions with 10D and 30D from IEEE CEC 2010. The significance test results of the FRC-CEA with different α values are listed in Table 7, where "+", "−" and "≈" represent that the performance of the FRC-CEA with $\alpha = 0.9$ is better than, worse than, and statistically equivalent to that of the FRC-CEA with corresponding α value, respectively.

Considering the statistical test results reported in Table 7, we can see that FRC-CEA with $\alpha = 0.9$ beats $\alpha = 0.5$, $\alpha = 0.6$, $\alpha = 0.7$, $\alpha = 0.8$ and $\alpha = 1.0$ on 19, 15, 14, 14 and 14 test problems, respectively. Note that in the case of $\alpha = 1.0$, which means FRC-CEA only have the first-stage optimization, the results indicate that the second-stage optimization is essential.

Based on the results, α with a value of 0.9 is recommended in this paper.

7. Conclusion

Ideally, the search of the global optimum for COPs should involve two regions: both feasible and infeasible regions. To this end, this paper proposes a feasible-ratio control technique for COPs via both feasible and infeasible regions. The developed method can utilize both feasible and infeasible solutions efficiently. The retained feasible and infeasible solutions are promising and can maintain the proper diversity in both feasible and infeasible regions. We integrate the proposed feasible-ratio control technique into a constraint-handling EA framework, named FRC-CEA. The performance of the proposed FRC-CEA has been tested on two sets of benchmark test suites collected from IEEE CEC 2006 and IEEE CEC 2010, respectively. Based on experimental results, the performance of the proposed FRC-CEA outperforms or is highly competitive against the referred state-of-the-art constraint-handling algorithms in most cases.

Much work remains to be completed in the future. For instance, the proposed feasible-ratio control technique could be extended for addressing multi-objective optimization problems with constraints. It is also of interest to develop self-adaptive feasible ratio tuning strategies.

The source code of FRC-CEA can be obtained from: http://www.escience.cn/people/RuwangJiao/index.html.

Conflict of interest

The authors declared that they have no conflicts of interest to this work.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under grant 61673355, in part by the Hubei Provincial Natural Science Foundation of China under grant 2015CFA010, in part by the 111 project under grant B17040, in part by the Fundamental Research Funds for National Universities, China University of Geosciences(Wuhan).

References

[1] R.D. Al-Dabbagh, F. Neri, N. Idris, M.S. Baba, Algorithmic design issues in adaptive differential evolution schemes: review and taxonomy, Swarm Evol. Comput. 43 (2018) 284–311.
[2] J. Cheng, G. Zhang, F. Caraffini, F. Neri, Multicriteria adaptive differential evolution for global numerical optimization, Integ. Comput-Aid. Eng. 22 (2) (2015) 103–107.
[3] C.A. Coello Coello, Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art, Comput. Method Appl. M. 191 (11–12) (2002) 1245–1287.
[4] C.A. Coello Coello, Constraint-handling techniques used with evolutionary algorithms, in: Proceedings of the 2018 on Genetic and Evolutionary Computation Conference Companion, ACM, 2018, pp. 773–799.
[5] R. Datta, K. Deb, Individual penalty based constraint handling using a hybrid bi-objective and penalty function approach, in: 2013 IEEE Congress on Evolutionary Computation (CEC), 2013, pp. 2720–2727.
[6] M. de Castro Rodrigues, S. de Lima B.S.L.P., Guimarães, Balanced ranking method for constrained optimization problems using evolutionary algorithms, Inf. Sci. 327 (2016) 71–90.
[7] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Method Appl. M. 186 (2–4) (2000) 311–338.
[8] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: nsga-ii, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.
[9] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E. Goodman, Push and pull search for solving constrained multi-objective optimization problems, Swarm Evol. Comput. 44 (2019) 665–679.
[10] W. Gao, G. Yen, S. Liu, A dual-population differential evolution with coevolution for constrained optimization, IEEE Trans. Cybern. 45 (5) (2015) 1094–1107.
[11] B. Ghasemishabankareh, X. Li, M. Ozlen, Cooperative coevolutionary differential evolution with improved augmented lagrangian to solve constrained optimisation problems, Inf. Sci. 369 (2016) 441–456.
[12] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: International Conference on Genetic Algorithms on Genetic Algorithms and Their Application, 1987, pp. 41–49.
[13] N.M Hamza, D.L. Essam, R.A. Sarker, Constraint consensus mutation-based differential evolution for constrained optimization, IEEE Trans. Evol. Comput. 20 (3) (2016) 447–459.
[14] A. Isaacs, T. Ray, W. Smith, Blessings of maintaining infeasible solutions for constrained multi-objective optimization problems, in: 2008 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2008, pp. 2780–2787.
[15] L. Jiao, L. Li, R. Shang, F. Liu, R. Stolkin, A novel selection evolutionary strategy for constrained optimization, Inf. Sci. 239 (2013) 122–141.
[16] R. Jiao, Y. Sun, J. Sun, Y. Jiang, S. Zeng, Antenna design using dynamic multi-objective evolutionary algorithm, IET Microw. Antennas Propag. 12 (13) (2018) 2065–2072.
[17] R. Jiao, S. Zeng, C. Li, Y. Jiang, Dynamic constrained multi-objective evolutionary algorithms with a novel selection strategy for constrained optimization, in: Proceedings of the 2018 on Genetic and Evolutionary Computation Conference Companion, ACM, 2018, pp. 213–214.
[18] S.O. Kimbrough, G.J. Koehler, M. Lu, D.H. Wood, On a feasible–infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: distance tracing and no free lunch, Eur. J. Oper. Res. 190 (2) (2008) 310–327.
[19] J.J. Liang, T.P. Runarsson, E. Mezura-Montes, M. Clerc, P.N. Suganthan, C.A. Coello Coello, K. Deb, Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization, J. Appl. Mech. 41 (8) (2006).
[20] R. Mallipeddi, P.N. Suganthan, Ensemble of constraint handling techniques, IEEE Trans. Evol. Comput. 14 (4) (2010) 561–579.
[21] R. Mallipeddi, P.N. Suganthan, Problem definitions and evaluation criteria for the cec 2010 competition on constrained real-parameter optimization, Nanyang Technological University (2010).
[22] F. Neri, V. Tirronen, Recent advances in differential evolution: a survey and experimental analysis, Artif. Intell. Rev. 33 (1–2) (2010) 61–106.
[23] J. Nocedal, S.J. Wright, Numerical Optimization, Springer New York, 2006.
[24] S. Rostami, F. Neri, Covariance matrix adaptation pareto archived evolution strategy with hypervolume-sorted adaptive grid algorithm, Integ. Comput-Aid. Eng. 23 (4) (2016) 313–329.
[25] S. Rostami, F. Neri, A fast hypervolume driven selection mechanism for many-objective optimisation problems, Swarm Evol. Comput. 34 (2017) 50–67.
[26] S. Rostami, F. Neri, M. Epitropakis, Progressive preference articulation for decision making in multi-objective optimisation problems, Integ. Comput-Aid. Eng. 24 (4) (2017) 315–335.
[27] T.P. Runarsson, X. Yao, Stochastic ranking for constrained evolutionary optimization, IEEE Trans. Evol. Comput. 4 (3) (2000) 284–294.
[28] R.A. Sarker, S.M. Elsayed, T. Ray, Differential evolution with dynamic parameters selection for optimization problems., IEEE Trans. Evol. Comput. 18 (5) (2014) 689–707.
[29] C. Segura, C.A. Coello Coello, G. Miranda, C. León, Using multi-objective evolutionary algorithms for single-objective constrained and unconstrained optimization, Ann. Oper. Res. 240 (1) (2016) 217–250.
[30] H.K. Singh, A. Isaacs, T. Ray, W. Smith, Infeasibility driven evolutionary algorithm (idea) for engineering design optimization, in: Australasian Joint Conference on Artificial Intelligence, Springer, 2008, pp. 104–115.
[31] R. Storn, K.V. Price, Differential evolution a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.
[32] J. Sun, J.M. Garibaldi, Y. Zhang, A. Al-Shawabkeh, A multi-cycled sequential memetic computing approach for constrained optimisation, Inf. Sci. 340 (2016) 175–190.
[33] T. Takahama, S. Sakai, Constrained optimization by the ε constrained differential evolution with gradient-based mutation and feasible elites, in: 2006 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2006, pp. 1–8.
[34] T. Takahama, S. Sakai, Efficient constrained optimization by the ε constrained adaptive differential evolution, in: 2010 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2010, pp. 1–8.
[35] S. Venkatraman, G.G. Yen, A generic framework for constrained optimization using genetic algorithms, IEEE Trans. Evol. Comput. 9 (4) (2005) 424–435.
[36] B.C. Wang, H.X. Li, Y. Feng, An improved teaching-learning-based optimization for constrained evolutionary optimization, Inf. Sci. 456 (2018) 1245–1287.
[37] J. Wang, G. Liang, J. Zhang, Cooperative differential evolution framework for constrained multiobjective optimization, IEEE Trans. Cybern. 49 (6) (2019) 2062–2072.
[38] Y. Wang, Z. Cai, Combining multiobjective optimization with differential evolution to solve constrained optimization problems, IEEE Trans. Evol. Comput. 16 (1) (2012) 117–134.
[39] Y. Wang, B.C. Wang, H. Li, G.G. Yen, Incorporating objective function information into the feasibility rule for constrained evolutionary optimization, IEEE Trans. Cybern. 46 (12) (2016) 2938–2952.
[40] B. Xu, X. Chen, L. Tao, Differential evolution with adaptive trial vector generation strategy and cluster-replacement-based feasibility rule for constrained optimization, Inf. Sci. 435 (2018) 240–262.
[41] K. Yu, X. Wang, Z. Wang, Constrained optimization based on improved teaching–learning-based optimization algorithm, Inf. Sci. 352 (2016) 61–78.
[42] S. Zeng, R. Jiao, C. Li, X. Li, J.S. Alkasassbeh, A general framework of dynamic constrained multiobjective evolutionary algorithms for constrained optimization, IEEE Trans. Cybern. 47 (9) (2017) 2678–2688.
[43] S. Zeng, R. Jiao, C. Li, R. Wang, Constrained optimisation by solving equivalent dynamic loosely-constrained multiobjective optimisation problem, Int. J. Bio-Insp. Com. 13 (2) (2019) 86–101.
[44] W. Zhang, G.G. Yen, Z. He, Constrained optimization via artificial immune system, IEEE Trans. Cybern. 44 (2) (2014) 185–198.