Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

A relative feasibility degree based approach for constrained optimization problems*

Cheng-gang CUI, Yan-jun LI, Tie-jun WU†‡

(*Department of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China*)

†E-mail: tjwu@zju.edu.cn

Received Feb. 10, 2009; Revision accepted June 20, 2009; Crosschecked Mar. 1, 2010

Abstract: Based on the ratio of the size of the feasible region of constraints to the size of the feasible region of a constrained optimization problem, we propose a new constraint handling approach to improve the efficiency of heuristic search methods in solving the constrained optimization problems. In the traditional classification of a solution candidate, it is either a feasible or an infeasible solution. To refine this classification, a new concept about the relative feasibility degree of a solution candidate is proposed to represent the amount by which the 'feasibility' of the solution candidate exceeds that of another candidate. Relative feasibility degree based selection rules are also proposed to enable evolutionary computation techniques to accelerate the search process of reaching a feasible region. In addition, a relative feasibility degree based differential evolution algorithm is derived to solve constraint optimization problems. The proposed approach is tested with nine benchmark problems. Results indicate that our approach is very competitive compared with four existing state-of-the-art techniques, though still sensitive to the intervals of control parameters of the differential evolution.

Key words: Constrained optimization, Evolutionary computation, Relative feasibility degree (RFD), Evolution differential algorithm

doi:10.1631/jzus.C0910072 **Document code:** A **CLC number:** TP18

1 Introduction

Evolutionary computation (EC) techniques have been used to effectively solve constrained optimization problems (COPs) recently. However, EC techniques are normally used as 'blind heuristics' in the sense that no specific domain knowledge is used or required (Back *et al*., 1997). Several researchers have proposed different mechanisms to incorporate constraints into the fitness function of an EC technique.

Penalty functions, popular in conventional methods for constrained optimization, are most common approaches to handling constraints with evolutionary algorithms (Back *et al*., 1997). There are different types of penalty functions, some of which are discussed as follows. In the static penalty method, the penalty is a weighted sum of the constraint violations where the weights are called penalty factors. Therein the penalty factors are independent of the current generation number and remain constant during the entire search process. In contrast, the success of the static penalty method depends on the proper penalty factors. In the dynamic penalty method, the penalty assigned to each individual relies on the generation number, but the difficulty of tuning parameters for the dynamic penalty method has significantly limited its applicability. Hadj-Alouane and Bean (1997) developed a penalty function which takes a feedback from the search process; however, the choice of the generational gap that provides reasonable information to guide the search is very difficult in this method. Runarsson and Yao (2000) introduced a stochastic ranking method to combine the objective and penalty functions, which could produce

‡ Corresponding author

* Project (No. 2006AA04Z184) supported by the National High-Tech Research and Development Program (863) of China

only feasible solutions in some problems. Farmani and Wright (2003) proposed a self-adaptive penalty function, which uses a two-stage penalty function to replace the explicit definition of any parameter. Wang *et al*. (2008) developed an adaptive tradeoff model to determine the penalty function; however, the performance of this method is determined by the reduction of the initial step size in evolution strategies.

The methods based on the preference of feasible solutions over infeasible ones are another category of constraint handling techniques (Powell and Skolnick, 1993; Hinterding and Michalewicz, 1998; Deb, 2000). Accordingly, a heuristic rule was proposed to process infeasible solutions. However, iterations in the search process may stagnate in the infeasible region if the region is very small compared with the size of its search space.

The methods based on multi-objective optimization techniques are also a category of constraint handling techniques. Coello (2000) proposed a method using multi-objective optimization techniques to handle each of the constraints of COPs as an objective; however, this method tends to generate trade-off solutions, and is hard to reach the global optimum efficiently. Cai and Wang (2006) proposed a method based on multi-objective optimization techniques; however, the performance depends on the parameters of the crossover operator.

From the analyses of the algorithms previously proposed to solve COPs, we notice that no specific domain knowledge is used. However, the constraints provide cheap specific domain knowledge which can be used to augment domain independent search methods in COPs (Coello, 2002). Therefore, incorporating domain knowledge can considerably improve the performance of an EC technique and accelerate evolution by reducing the search space. Several researchers have proposed different mechanisms to use domain knowledge to solve COPs. Kowalczyk (1997) proposed a method based on constraint consistency to avoid the generation of variable instantiations that are not consistent with the constraints of a COP. The main drawback of this method is the extra computational cost of constraint propagation, which may become more expensive than the optimization. Schoenauer and Xanthakis (1993) handled constraints sequentially (one by one), but did not bring out how to order the constraints of a problem. The performance of this method is affected greatly by the particular order of processed constraints (Michalewicz, 1996). Chung and Reynolds (1996) proposed a cultural algorithm with GENOCOP (Michalewicz and Nazhiyath, 1995) to solve COPs. As required to build a map of the search space, this approach is sensitive to high dimensionality.

According to the relationship between the feasible region of a COP and that of constraints, a relative feasibility degree (RFD) based approach is proposed in this paper to solve COPs. The RFD of a constraint and the RFD of a solution candidate are defined to guide the search process of searching for feasible solutions. The former RFD represents the probability of a solution candidate that satisfies this constraint being a feasible solution and the latter RFD represents the amount by which the 'feasibility' of the solution candidate exceeds that of another candidate. According to these concepts, a set of RFD based selection rules are proposed to make solution candidates accelerate the search process of reaching the feasible region.

The proposed constraint handling approach is a generic framework for solving COPs, suitable for all EC techniques. In this paper, only a differential evolution (DE) algorithm is implemented as an example of this constraint handling approach.

2 Feasibility degree of a solution candidate

In general, a constrained optimization problem can be expressed as follows:

$$\begin{aligned} &\min\ f(\boldsymbol{x}),\ \ \boldsymbol{x}=(x_1,x_2,...,x_n)\in\mathbb{R}^n,\\ &\text{s.t.}\quad g_j(\boldsymbol{x})\le 0,\quad j=1,2,...,m, \end{aligned} \tag{1}$$

where $f(\boldsymbol{x})$ is the objective function, and m is the number of inequality constraints. Let $S\subset\mathbb{R}^n$ represent the search space bounded by the parametric constraints $\underline{x}_i \le x_i \le \overline{x}_i$, $i\in\{1,2,\ldots,n\}$, where $\underline{x}_i$ and $\overline{x}_i$ are the lower bound and the upper bound of x_i, respectively. We define the feasible region of the jth constraint $g_j(\boldsymbol{x})\le 0$ by $F_j=\{\boldsymbol{x}\in\mathbb{R}^n|g_j(\boldsymbol{x})\le 0\}$, and then the feasible region of the problem Eq. (1) can be expressed by $F=\bigcap_{j=1}^{m}F_j$. Thus, a vector $\boldsymbol{x}\in F$ is a feasible solution of the problem Eq. (1).

It is well known that the COPs with such characteristics as non-differentiable objective functions (and perhaps even non-differentiable constraints), non-convex objective functions, and disjoint feasible regions, are difficult to solve using traditional mathematical programming techniques. In contrast, EC techniques such as genetic algorithms (Holland, 1962; 1992), ant colony algorithms (Dorigo *et al.*, 1991; Dorigo, 1992; Colorni *et al.*, 1992), and particle swarm algorithms (Eberhart and Kennedy, 1995; Kennedy and Eberhart, 1995) are global optimization algorithms using nature-inspired mechanisms. The EC techniques are widely applied to solve COPs since no well-posted problem models are required in using an EC algorithm (Coello, 2002).

In constraint handling approaches used with EC techniques, a number of solution candidates are commonly generated randomly in the search space and set as the initial population. Three phases are needed in the search process using EC techniques for solving COPs:

1. Reproduction phase: new offspring of the current population are generated based on the original candidates by some heuristic rules, such as recombination and mutation.

2. Evaluation phase: the offspring are evaluated by a fitness function.

3. Selection phase: good or bad candidates are distinguished in the new offspring, and candidates that do not satisfy constraints or have a poor objective value are removed.

In the selection phase of the search process, enormous efforts are needed to remove infeasible candidates, especially when the feasible region is very small with respect to the whole search space. This will decrease the efficiency of the search process. Some researchers have developed special operators or decoders to preserve the feasibility of solution candidates in the reproduction phase. However, these methods are not general-purpose tools in all problem domains since problem-specific knowledge is used to design special operators or decoders.

The ability to find feasible solutions is essential for solving the COPs that have smaller feasible regions compared with their search spaces. Therefore, some new approaches based on infeasible candidates were proposed to accelerate the search process. Richardson *et al*. (1989) proposed a method based on the number of violated constraints for solving COPs, but their method is not likely to produce any solution if there are only a small number of feasible solutions in the COPs. Deb (2000) suggested simple feasibility rules to compare two candidates by drawing upon a preference of feasible solutions to infeasible ones; however, this approach requires a niching technique to maintain diversity in the population.

We propose a new approach in this paper based on the solution candidates satisfying a part of the constraints of a COP. The effect of these candidates on constraint handling approaches is evaluated and the results are incorporated as heuristic knowledge into the search process. In the traditional classification of solution candidates, a candidate must be either feasible or infeasible. To extend this classification, a new concept of RFD is proposed in this section to measure the amount by which in terms of 'feasibility' the solution candidate exceeds another. A solution candidate with a larger RFD means a greater probability of being a feasible solution than that of another. Therefore, heuristic rules based on the RFD of a solution candidate can obviously improve the performance of EC techniques if they can increase the RFD of the solution candidate. Therefore, the RFD of a solution candidate can be used as new heuristic knowledge for solving COPs.

Fig. 1 shows the relationship among the feasible region F, the search space S, and the feasible regions F_i (i=1, 2, 3) of constraints, where the rectangular region represents the search space S and the three circular regions indicate the feasible regions F_1, F_2, and F_3, respectively. If a candidate $\boldsymbol{x}$ is randomly generated in the feasible region F_i, the probability of the candidate $\boldsymbol{x}$ being located in the feasible region F is proportional to the ratio of the size of feasible region, $|F|$, to the size of the feasible region of the constraint, $|F_i|$.

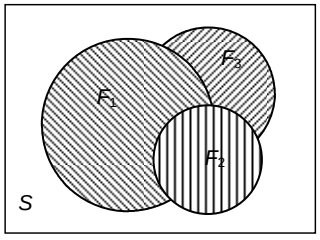

Fig. 1 Relationship bee feasible region and the search space

In order to measure the probability of a solution candidate being a feasible solution when using a search process to solve the problem Eq. (1), the RFD of a constraint is given as follows:

Definition 1 (Relative feasibility degree of a constraint) Given the COP Eq. (1), the ratio of the size of the feasible region F to the size of the feasible region F_i of the ith constraint is called the relative feasibility degree of the constraint, denoted by $\rho_i=|F|/|F_i|$.

The RFD of a constraint measures the probability of a solution candidate satisfying the constraint of becoming a feasible solution. This concept can be extended to the RFD of a solution candidate, describing the amount by which as regards 'feasibility' the solution candidate exceeds another.

Definition 2 (Relative non-common satisfied constraint set) Given the COP Eq. (1), let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ denote two solution candidates of the problem, G represent the constraint set of the problem, G_1 and $\bar{G}_1$ refer to the satisfied constraint set and the violated constraint set of $\boldsymbol{x}_1$ respectively, and G_2 and $\bar{G}_2$ indicate the satisfied constraint set and the violated constraint set of $\boldsymbol{x}_2$ respectively. The intersection of G_1 and $\bar{G}_2$ denoted as $Q(\boldsymbol{x}_1\rightarrow\boldsymbol{x}_2)$ is called the relative non-common satisfied constraint set (RNSCS) of $\boldsymbol{x}_1$ with respect to $\boldsymbol{x}_2$, defined as

$$Q(\boldsymbol{x}_1 \rightarrow \boldsymbol{x}_2) = G_1 \cap \bar{G}_2.$$

Definition 3 (Relative feasibility degree of a solution candidate) Given the COP Eq. (1), let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be two solution candidates of the problem, $Q(\boldsymbol{x}_1\rightarrow\boldsymbol{x}_2)$ the RNSCS of $\boldsymbol{x}_1$ with respect to $\boldsymbol{x}_2$, and ρ_i, $i\in Q(\boldsymbol{x}_1\rightarrow\boldsymbol{x}_2)$, the RFD of constraint i. $R(\boldsymbol{x}_1\rightarrow\boldsymbol{x}_2)$ is called the RFD of $\boldsymbol{x}_1$ with respect to $\boldsymbol{x}_2$, defined as

$$R(\boldsymbol{x}_1 \rightarrow \boldsymbol{x}_2) = \begin{cases} 0, & Q(\boldsymbol{x}_1 \rightarrow \boldsymbol{x}_2) = \varnothing, \\ 1, & Q(\boldsymbol{x}_1 \rightarrow \boldsymbol{x}_2) = \{1,\ 2,\ ...,\ m\}, \\ \max_{i\in Q(\boldsymbol{x}_1\rightarrow\boldsymbol{x}_2)} \rho_i, & \text{otherwise.} \end{cases} \quad (2)$$

We call the constraint with the highest RFD in the RNSCS the maximal non-common satisfied constraint.

The RFD of a solution candidate defined by Eq. (2) has the following properties (for convenience, we classify infeasible solutions into complete infeasible solutions (satisfying no constraints) and incomplete infeasible solutions (partially satisfying the constraints in COPs)):

1. The RFD of a solution candidate with respect to another is 0 if the satisfied constraint set of the former is included in that of the latter. For instance, we can obtain that (1) the RFD of a complete infeasible solution with respect to any solution candidate is 0; (2) the RFD of any type of infeasible solutions with respect to any feasible solution is 0; (3) the RFD of a feasible solution with respect to any feasible solution is 0; (4) the RFD of a solution candidate with respect to another is 0 if the two solution candidates have the same satisfied constraint set.

2. On the contrary, the RFD of any feasible solution with respect to any complete infeasible solution is 1.

3. Except for conditions 1 and 2, the RFD of a solution candidate compared to another is between 0 and 1. In this case the RFD is determined by the maximal non-common satisfied constraint, which is the one with the largest RFD in the RNSCS of the former with respect to the latter. The greater the RFD of the maximal non-common satisfied constraint, the greater the RFD of the solution candidate with respect to another; it indicates that, the probability of the solution candidate in the feasible region of the original optimization problem is much larger than that of another solution candidate.

As implied in Definition 3, since the RFD of a solution candidate is defined by the RFD of a constraint, it is necessary to evaluate the RFD of all the constraints in a COP before the RFD of a solution candidate can be obtained. A method based on randomly selected candidates is used to evaluate the RFD of a constraint since the feasible region is unknown. Michalewicz *et al*. (2000) proposed a method to evaluate the ratio of the size of the feasible region to the size of the search space in a COP. It can also be used to evaluate the RFD of a constraint; i.e., given the COP Eq. (1), generate s randomly selected candidates by a uniform n-dimensional probability distribution in the search space, with k the number of candidates in the feasible region F and k_j the number of candidates in the feasible region F_j of the constraint $g_j(\boldsymbol{x})\leq 0$. Then, according to Definition 1, the RFD of the constraint $g_j(\boldsymbol{x})\leq 0$ is $\rho_j=k/k_j$.

In the real world, there are many COPs in which the ratio of the size of the feasible region to the size of the search space is very low, so the cost of generating a feasible solution is very high. It makes Michalewicz's method inefficient. Actually, as heuristic knowledge in a search process, the exact value of the RFD of a candidate is not necessary. The rank of the RFD can also provide heuristic knowledge to improve the performance of an EC algorithm. The size of the feasible region of a constraint is constant for a given COP. Therefore, the rank of ρ_j can be determined simply by k_j, j=1, 2, …, m; i.e., if $k_i>k_j$, then $\rho_i<\rho_j$.

Experiments show that, only a small number of randomly selected candidates are needed in practice when calculating the rank of the RFDs of constraints. For example, only 100 randomly selected candidates are needed when we calculate the rank ordering of RFD of the 6 constraints in the benchmark problem g10 (the ratio of the size of the feasible region to the size of the search space is only 2.0×10^{-5}) (Runarsson and Yao, 2000). As is shown in Table 1, the results given by evaluating 100 randomly selected candidates are nearly the same as that by 1 000 000 randomly selected candidates.

Table 1 The rank of RFD of constraints evaluated by different numbers of random variables

Constraint	k_j		Rank	
	n=100	n=1×10^6	n=100	n=1×10^6
$g_1(\boldsymbol{x})\le0$	47	499 528	3	3
$g_2(\boldsymbol{x})\le0$	50	504 075	4	4
$g_3(\boldsymbol{x})\le0$	76	779 026	6	6
$g_4(\boldsymbol{x})\le0$	39	421 367	2	2
$g_5(\boldsymbol{x})\le0$	3	73 497	1	1
$g_6(\boldsymbol{x})\le0$	64	595 348	5	5

n: number of candidates

The evaluations of the RFDs of constraints can be done before the search process. Thus, the RFD of a solution candidate can be evaluated by the RFDs of constraints in descending order.

3 Relative feasibility degree based selection rules

The RFD of a solution candidate is defined in the former section as the amount by which the 'feasibility' of the solution candidate exceeds that of another. By this definition, the solution candidate with a larger RFD can be selected as a better candidate in the search process for solving COPs. In this way, the RFD knowledge can be conveniently incorporated into the selection phase of an EC technique to reduce the search space and accelerate the search process.

Deb (2000) worked out a set of simple feasibility rules to compare two solution candidates in the tournament selection of GAs by giving a preference for feasible solutions. It can be stated as follows:

1. A feasible solution is always preferred to an infeasible one.

2. Between two feasible solutions, the one with a better objective function value is preferred.

3. If both solution candidates are infeasible, the one with a smaller constraint violation is preferred.

However, only feasible and infeasible solutions based on the traditional classification of solution candidates are compared in Deb's rules. Therefore, iterations in the searching process may stagnate in the infeasible region if the feasible region is very small as opposed to the size of its searching space. Based on Deb's rules, a set of RFD based selection rules is developed in the following to reduce the search space of COPs.

The RFD based selection rules can be formulated as follows:

Given two solution candidates $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$:

1. If $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are both feasible solutions, the one with a better objective is preferred.

2. Otherwise, (1) if $R(\boldsymbol{x}_1)\neq R(\boldsymbol{x}_2)$, the one with a larger RFD wins; and (2) if $R(\boldsymbol{x}_1)=R(\boldsymbol{x}_2)$, the one with a smaller violation of their maximal non-common satisfied constraint is preferred.

Note that violation of a constraint is defined by $V(\boldsymbol{x})=\max(0, g(\boldsymbol{x}))$ (for details, see Back *et al*., 1997).

Following these rules, the search process can reach the feasible region of a COP quickly. When two infeasible solutions with the same RFD are compared, the one whose maximal non-common satisfied constraint bears a smaller violation is preferred. This rule forces solution candidates into satisfying their maximal non-common satisfied constraint. When two infeasible solutions with different RFDs are compared, preference would be given to the solution candidate with a larger RFD. This rule makes solution candidates satisfy the constraint with a larger RFD first. When two feasible solutions are compared,

preference would be taken up by the solution candidate with a better objective. Following this rule, the search process needs to find solution candidates with the best objective. Therefore, the solution candidates selected based on these rules will satisfy the constraints in descending order of RFDs and reach the global optimum finally.

4 Implementation with differential evolution

To illustrate validity of the RFD based selection rules, we introduce them to a DE algorithm as an example. The DE algorithm proposed by Storn and Price (1997) is a heuristic method for real parameter optimization problems.

Let $\boldsymbol{x}_t^i$ denote an individual in the population of the DE algorithm and NP the size of the population, where i indicates the index of the individual, j the index of the variable, and t the current generation. A new mutated individual $v_{j,t+1}^i$ is generated according to the following equation:

$$v_{j,t+1}^i = x_{j,t}^{d_3} + \eta(x_{j,t}^{d_1} - x_{j,t}^{d_2}), \tag{3}$$

where the random indexes d_1, d_2, $d_3 \in [0, \text{NP}]$ are mutually different integers and also different from the running index i, and $\eta \in (0, 1]$ is called the scaling factor or the amplification factor.

According to Eq. (5), a crossover operator is used to generate the trial individual $u_{j,t+1}^i$ based on the original individual $x_{j,t}^{d_3}$ and the new individual $v_{j,t+1}^i$.

$$u_{j,t+1}^i = \begin{cases} v_{j,t+1}^i, & \text{if Rand}[0,1) \le \text{CR} \\ & \text{or } j = \text{randint}(1, D), \\ x_{j,t}^i, & \text{otherwise,} \end{cases} \tag{4}$$

where Rand[0, 1) is a function that returns a real number between 0 and 1, randint(min, max) is a function that returns an integer between min and max, $\text{CR} \in [0, 1]$ is a crossover factor. The probability of the mutated individuals being preserved in the next generation is determined by the crossover factor CR.

A selection operator is used to choose an individual for the next generation (t+1) according to the following rule:

$$x_{t+1}^i = \begin{cases} u_{t+1}^i, & \text{if } u_{t+1}^i \text{ is better than } x_t^i, \\ x_t^i, & \text{otherwise,} \end{cases} \tag{5}$$

where u_{t+1}^i and x_t^i are compared by the RFD based selection rules.

In this way, an individual will replace the one with a lower RFD with respect to it; an individual will replace the one with the same RFD depending on different conditions, where an infeasible individual will replace the one with a larger violation of the maximal non-common satisfied constraint and a feasible individual will replace the one with a worse objective with respect to it, respectively. Therefore, the EC techniques using the RFD based selection rules can reduce the search space and find the optimal solution.

The pseudo code of the DE with the RFD based selection rules is given in Fig. 2. As can be seen, the rules keep the operators of DE algorithms unchanged. Therefore, the RFD based selection rules are a general method and can be used not only in ES but also in other DE algorithms.

```
Begin
  t=0;
  Create NP random solutions for the initial population;
  Evaluate all individuals;
  For t=1 to MAX_GENERATION Do
    For i=1 to NP Do
      Select randomly d1≠d2≠d3;
      If (Rand[0, 1]≤CR or j=randint(1, D)) Then
        u^i_{j,t+1} = v^i_{j,t+1};
      Else
        u^i_{j,t+1} = x^i_{j,t};
      End If
    End For
    Compare u^i_{j,t+1} and x^i_{j,t} by the RFD based selected
      rules;
    If u^i_{t+1} is better than x^i_t Then
      x^i_{t+1} = u^i_{t+1};
    Else
      x^i_{t+1} = x^i_t;
    End If
    t=t+1;
  End For
End
```

Fig. 2 Pseudo code of the DE using the RFD based selection rules

The capability of finding the global minimum and a fast convergence speed of DE are both highly sensitive to the control parameters CR and η (Qin and Suganthan, 2005). Therefore, a self-adaptive approach is developed to adjust these parameters based on the success rate φ_t, where φ_t is defined by the percentage of original individuals replaced by trial individuals in the population at every generation, through the following updating law:

$$\eta_t = \begin{cases} \eta_l + \text{rand}_1 \eta_u, & \phi_t \le 0.5, \\ \eta_{t-1}, & \text{otherwise}, \end{cases} \tag{6}$$

$$\text{CR}_t = \begin{cases} \text{rand}_2, & \phi_t \le 0.5, \\ \text{CR}_{t-1}, & \text{otherwise}, \end{cases} \tag{7}$$

where η_t and CR_t are the scaling factor η and the crossover factor CR at generation t, respectively; rand_1 and rand_2 are uniformly distributed random numbers in [0, 1]; η_l=0.1, η_u=0.9. The updating of η_t and CR_t is conducted before the mutation is performed. Eqs. (6) and (7) ensure that $\eta_t \in [0.1, 1] \subset (0, 1]$, $\text{CR}_t \in [0, 1]$, $\forall t$, included in the defined ranges of CR and η. Note that, although the parameters are self-adaptively adjusted in the search process, the upper and lower boundaries of the scaling factor are still defined empirically.

5 Experiments and results

Nine benchmark problems described in Runarsson and Yao (2000) were used to evaluate the performance of the RFD based differential evolution (RFDDE) algorithm proposed in this paper. The characteristics of the benchmark problems are shown in Table 2, where n is the number of decision variables, m_l the number of linear inequalities, m_n the number of nonlinear inequalities, m_le the number of linear equalities, m_ne the number of nonlinear equalities, and $\hat{r}$ the ratio of the size of the feasible region to the size of the search space of the benchmark problems evaluated by Michalewicz *et al*. (2000)'s method using 1 000 000 random candidates. As shown in Table 2, $\hat{r}$ of the remaining problems is fairly low except for problems g02 and g04, while $\hat{r}$ of problems g02 is nearly 100%. Therefore, the capability to find feasible solutions with different $\hat{r}$ can be tested using these benchmark problems.

Table 2 Characters of the nine benchmark problems chosen

Problem	n	Type of function	$\hat{r}$ (%)	m_l	m_n	m_le	m_ne
g01	13	quadratic	0.0003	9	0	0	0
g02	20	nonlinear	99.9973	1	1	0	0
g04	5	quadratic	27.0079	0	6	0	0
g05	4	nonlinear	0.0000	2	0	0	3
g06	2	nonlinear	0.0057	0	2	0	0
g07	8	quadratic	0.0000	3	5	0	0
g08	2	nonlinear	0.8581	0	2	0	0
g09	7	nonlinear	0.5199	0	4	0	0
g10	6	linear	0.0020	6	0	0	0

n: number of decision variables; m_l: number of linear inequalities; m_n: number of nonlinear inequalities; m_le: number of linear equalities; m_ne: number of nonlinear equalities

We performed 30 independent runs for each benchmark problem. Equality constraints were transformed into inequalities using a tolerance value of 0.0001. The parameters were set the same as those of Mezura-Montes *et al*. (2004): NP=60, MAX_GENERATIONS=5800. The control parameters CR and η were adjusted using a self-adaptive method. Our approach was implemented in C/C++ and tested on a Pentium IV 2.8 GHz PC.

The statistical results of RFDDE are summarized in Table 3.

We compared our approach against four state-of-the-art approaches: the stochastic ranking (SR) algorithm (Runarsson and Yao, 2000), the simple multimembered evolution strategy (SMES) algorithm (Mezura-Montes and Coello, 2005), the adaptive tradeoff model evolution strategy (ATMES) algorithm (Wang *et al.*, 2008), and the constraint handling differential evolution (CHDE) algorithm (Mezura-Montes *et al*., 2004). The best results, the mean results, the worst results, and the standard deviations obtained by each approach are shown in Table 4. The results provided by these approaches were taken from the original references for each method.

6 Discussion

6.1 General performance of the proposed approach

As described in Table 3, our approach was able to find the global optimum in nine benchmark

Table 3 Statistical results obtained by RFDDE for the 9 benchmark problems over 30 independent runs

Problem	Optimal	Best	Mean	Median	Worst	SD
g01	−15.000	**−15.000**	**−15.000**	**−15.000**	**−15.000**	0
g02	0.803619	**−0.803 619**	−0.801783	**−0.803619**	−0.792608	4.2E−3
g04	−30665.539	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	2.2E−11
g05	5126.4967	**5126.4967**	5149.7695	5135.2497	5448.4213	5.8E+01
g06	−6961.8140	**−6961.8140**	**−6961.8140**	**−6961.8140**	**−6961.8140**	0
g07	24.306	**24.306**	**24.306**	**24.306**	**24.306**	2.7E−05
g08	0.095825	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	4.21E−17
g09	680.6300	**680.6300**	**680.6300**	**680.6300**	**680.6300**	1.2E−13
g10	7049.248	**7049.248**	**7049.248**	**7049.248**	7049.253	1.3E−03

Results in boldface indicate the global optimum (or best known solution). SD: standard deviation

Table 4 Comparison of the best, the mean solutions, the worst solutions, and the standard deviations found by our RFDDE against SR, SMES, ATMES, and CHDE

Problem	Optimal		Statistics				
			SR	SMES	ATMES	CHDE	RFDDE
g01	−15.000	Best	**−15.000**	**−15.000**	**−15.000**	**−15.000**	**−15.000**
		Mean	**−15.000**	**−15.000**	**−15.000**	−14.792	**−15.000**
		Worst	**−15.000**	**−15.000**	**−15.000**	−12.743	**−15.000**
		SD	0.0E+00	0	1.6E−14	4.0E−01	0
g02	−0.803619	Best	−0.803515	−0.803601	−0.803388	**−0.803619**	**−0.803619**
		Mean	−0.781975	−0.785238	−0.790148	−0.746236	−0.801783
		Worst	−0.726288	−0.751322	−0.756986	−0.302179	−0.792608
		SD	2.0E−02	1.7E−02	1.3E−02	8.1E−02	4.2E−03
g04	−30665.539	Best	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**
		Mean	**−30665.539**	**−30665.539**	**−30665.539**	−30592.154	**−30665.539**
		Worst	**−30665.539**	**−30665.539**	**−30665.539**	−29986.214	**−30665.539**
		SD	2.0E−05	0	7.4E−12	1.1E+02	2.2E−11
g05	5126.497	Best	**5126.497**	5126.599	5126.498	**5126.497**	**5126.497**
		Mean	5128.881	5174.492	5127.648	5218.729	5149.769
		Worst	5142.472	5304.167	5135.256	5502.410	5448.421
		SD	3.5E+00	5.0E+01	1.8E+00	7.6E+01	5.8 E+01
g06	−6961.814	Best	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**
		Mean	−6875.940	−6961.284	**−6961.814**	−6367.575	**−6961.814**
		Worst	−6350.262	−6952.482	**−6961.814**	−2236.950	**−6961.814**
		SD	1.6E+02	1.9E+00	4.6E−12	7.7E+02	0
g07	24.306	Best	24.307	24.327	**24.306**	**24.306**	**24.306**
		Mean	24.374	24.475	24.316	104.599	**24.306**
		Worst	24.642	24.843	24.359	1120.541	**24.306**
		SD	6.6E−02	1.3E−01	1.1E−02	1.8E+02	2.7E−05
g08	−0.095825	Best	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**
		Mean	**−0.095825**	**−0.095825**	**−0.095825**	−0.091292	**−0.095825**
		Worst	**−0.095825**	**−0.095825**	**−0.095825**	−0.027188	**−0.095825**
		SD	2.6E−17	0	2.8E−17	0.012	0
g09	680.630	Best	**680.630**	680.632	**680.630**	**680.630**	**680.630**
		Mean	680.656	680.643	680.639	692.472	**680.630**
		Worst	680.763	680.719	680.673	839.783	**680.630**
		SD	3.4E−02	1.6E−02	1.0E−02	23.575	1.2E−13
g10	7049.248	Best	7054.316	7051.903	7052.253	**7049.248**	**7049.248**
		Mean	7559.192	7253.047	7250.437	8442.657	**7049.248**
		Worst	8835.655	7638.366	7560.224	15580.370	7049.253
		SD	5.3E+02	1.4E+02	1.2E+02	2.2E+03	1.3E−03

Results in boldface indicate the obtained global optimum (or best known solution). SD: standard deviation

problems. For problems g01, g04, g06, g07, g08, and g09, the optimal solutions were consistently found in all 30 runs. For problems g02, g05, and g10, the near-optimal solutions were found in all 30 runs. For problem g05, the optimal solutions were not consistently found since the ratio of the size of the feasible region to the size of the search space was very small. For problems g02 and g10, the optimal solutions were found in most of the runs. Furthermore, feasible solutions were continuously found for all the benchmark problems in 30 runs. These results reveal that RFDDE has the substantial capability to deal with various kinds of COPs.

6.2 Comparison with four state-of-the-art approaches

The performance of RFDDE was compared in detail with four state-of-the-art techniques using the selected performance metrics (Table 4). For benchmark problems g01, g04, and g08, RFDDE, SR, SMES, and ATMES consistently found the optimal solutions in all 30 runs. For problem g06, the optimal solutions were consistently found by RFDDE and ATMES in all 30 runs. For problem g05, SR, SMES, and ATMES found better 'mean' and 'worst' results than RFDDE. However, RFDDE was also able to find the optimal solution in 30 runs and the 'mean' results were very close to the optimal solution. For all the other 5 problems, RFDDE found better 'best', 'mean', and 'worst' results than SR, SMES, and ATMES. As against CHDE, our approach found 'similar' best results in all the problems, and furthermore located better 'mean' and 'worst' results in all the problems.

In summary, we can conclude that RFDDE outperforms or has similar performances to SR, SMES, ATMES, and CHDE in all the problems.

6.3 Advantages of our approach

After corroborating the effectiveness of our approach, we want to verify the advantages of our approach. Two comparisons of RFDDE and CHDE were designed as follows as there are only two differences between the two algorithms: (1) RFDDE uses the RFD based rules while CHDE uses simple feasibility rules, and (2) the control parameters of RFDDE are set using a self-adaptive method while those of CHDE are set empirically.

6.3.1 Reaching the feasible region

In the real world, it is normally desirable to find feasible (even if not optimal) solutions with the lowest possible number of fitness function evaluations (FFEs) for COPs. To study this issue, we compared RDFDE with CHDE on the percentage of feasible solutions in the population at every generation.

For problems g04, g06, g07, g08, g09, and g10, the percentages of feasible solutions in the population of RFDDE were larger than those of CHDE at the same generation (Fig. 3). The statistical results showed that a lower number of FFEs is needed for RFDDE than for CHDE to obtain a same percentage of feasible solutions in the population; i.e., RFDDE can reach the feasible region more quickly than CHDE.

There were 9 constraints in problem g01. The RFD selection rules make RFDDE satisfy the constraints in descending order of RFDs. Thus, the more constraints were located in a COP, the more FFEs were required for RFDDE to satisfy all the constraints. Therefore, the percentages of feasible solutions in the population of RFDDE were larger only than those of CHDE after 1500 FFEs. The ratio of the size of the feasible region to the size of the search space was very high in problem g02 ($\hat{r}$ = 99.973%). Therefore, the percentages of feasible solutions in the population at generation 1 were both 100% for RFDDE and CHDE. Analysis of each constraint in problem g05 shows that: the RFDs of three equality constraints were all near 0 in problem g05. Since the RFD of a constraint was used as the heuristic knowledge in the RFD based selection rules, the effect of these rules will be weakened when the RFDs of constraints are similar.

The experiment revealed that RFDDE can reach the feasible region quickly and find more feasible solutions than CHDE.

6.3.2 Convergence

In order to demonstrate the effectiveness of our approach, we compared RFDDE with CHDE on their convergence (Fig. 4). It can be found that the convergence speed and the accuracy of RFDDE were much higher than those of CHDE in most cases. For problems g01, g02, g04, g06, g08, g09, and g10, RFDDE showed a significantly fast convergence and could find solutions very close to the optimal solutions before 50 000 FFEs. For problem g07, although

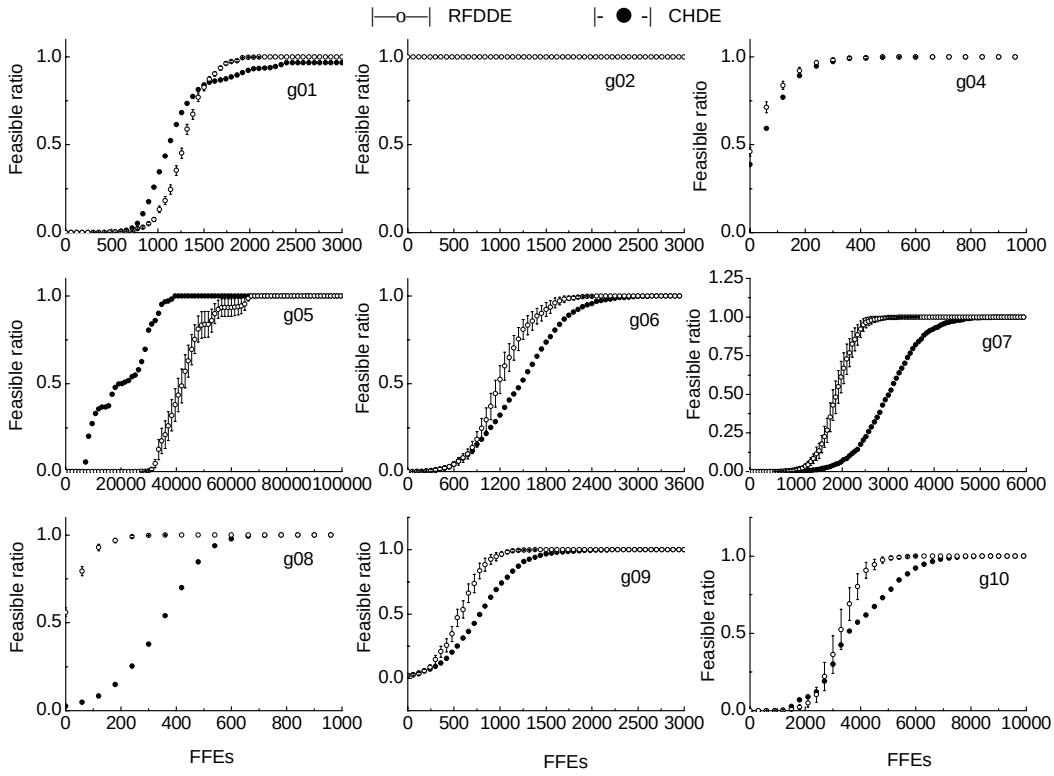

Fig. 3 Comparison of RFDDE and CHDE on the percentage of feasible solutions in the population (error bars indicate standard deviation in population)

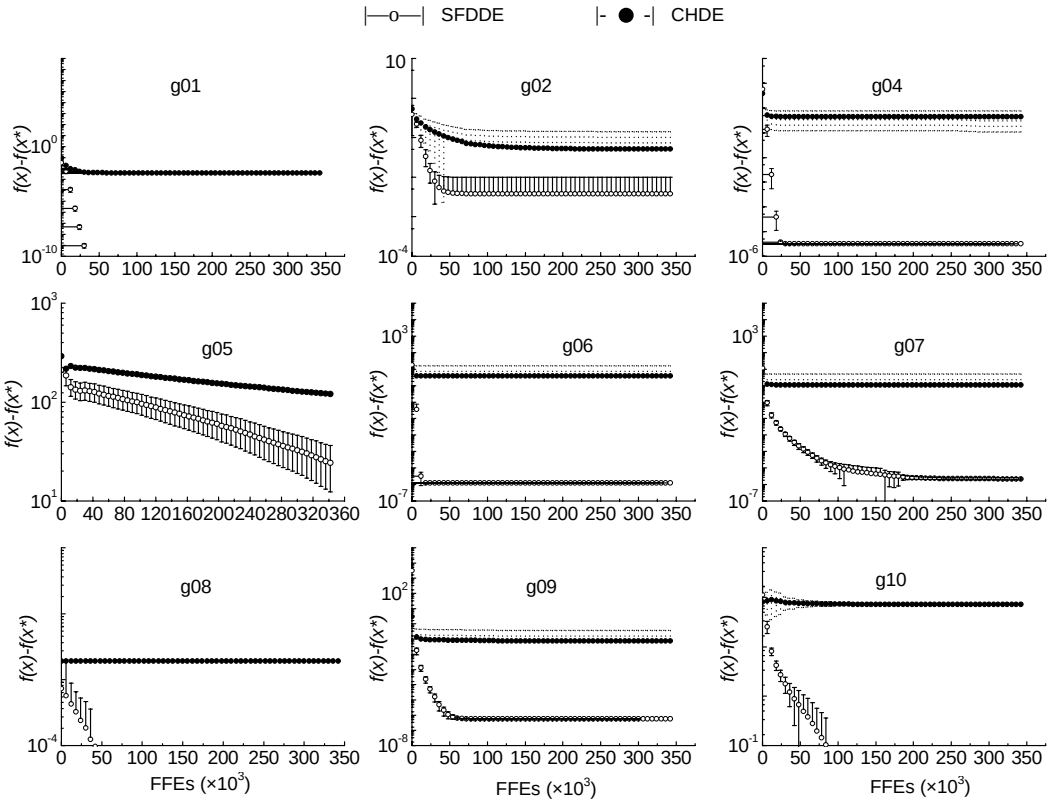

Fig. 4 Comparison of RFDDE and CHDE on the convergence (error bars indicate standard deviation in population)

CHDE had a faster convergence speed, RFDDE outperformed CHDE in terms of solution quality, i.e., achieving better mean solutions. The ultra small feasible region of problem g05 ($\hat{r}$ = 0.0000%) will result in poor performance of many EC techniques on it, which was also true for RFDDE and CHDE. In other words, RFDDE and CHDE both had poor performances and slow convergence speed in these two problems.

As shown in the two experiments, the RFD based selection rules can make DE reach the feasible region more quickly; that is, RFDDE can accelerate the convergence of DE.

7 Conclusion

In order to extend the traditional classification of a solution candidate, we proposed a novel approach based on the feasibility of a solution candidate for solving COPs. Herein new concepts of the RFD of a constraint and the RFD of a solution candidate were derived on the ratio of the size of the feasible region of constraints to the size of the feasible region of a COP to measure the amount by which the 'feasibility' of the solution candidate exceeds that of another. Accordingly, RFD based selection rules were proposed to select the solution candidates. EC techniques using these rules can reach the feasible region of a COP quickly. RFDDE, an implementation of this approach with a DE in this paper, is compared with CHDE on the percentage of feasible solutions in the population at every generation and the convergence. This new approach reaches the feasible region more quickly and bears a more rapid convergence speed than CHDE in most of the benchmark problems. However, the ranges of the crossover factor and the scaling factor of RFDDE are still defined empirically. Since the effect of our approach may be decreased when the RFDs of constraints are similar, our future work will focus on accelerating the search process in these conditions.

References

Back, T., Fogel, D.B., Michalewicz, Z., 1997. Handbook of Evolutionary Computation. IOP Publishing Ltd., Bristol, UK, p.351-356. [doi:10.1887/0750308958]

Cai, Z., Wang, Y., 2006. A multiobjective optimization-based evolutionary algorithm for constrained optimization. *IEEE Trans. Evol. Comput.*, **10**(6):658-675. [doi:10.1109/TEVC.2006.872344]

Chung, C.J., Reynolds, R.G., 1996. A Testbed for Solving Optimization Problem Using Cultural Algorithms. Evolutionary Programming V: Proc. 5th Annual Conf. on Evolutionary Programming, p.225-236.

Coello, C.A.C., 2000. Treating constraints as objectives for single-objective evolutionary optimization. *Eng. Optim.*, **32**(3):275-308. [doi:10.1080/03052150008941301]

Coello, C.A.C., 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. [doi:10.1016/S0045-78250100323-1]

Colorni, A., Dorigo, M., Maniezzo, V., 1992. Distributed Optimization by Ant Colonies: Toward a Practice of Autonomous Systems. Proc. First European Conf. on Artificial Life, **37**:258-267.

Deb, K., 2000. An efficient constraint handling method for genetic algorithms. *Comput. Methods Appl. Mech. Eng.*, **186**(2-4):311-338. [doi:10.1016/S0045-7825(99)00389-8]

Dorigo, M., 1992. Optimization, Learning and Natural Algorithms. Unpublished Doctoral Thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy.

Dorigo, M., Colorni, A., Maniezzo, V., 1991. Positive Feedback as a Search Strategy. Technical Report, No. 91-016. Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy.

Eberhart, R., Kennedy, J., 1995. A New Optimizer Using Particle Swarm Theory. Proc. 6th Int. Symp. on Micro Machine and Human Science, p.39-43. [doi:10.1109/MHS.1995.494215]

Farmani, R., Wright, J., 2003. Self-adaptive fitness formulation for constrained optimization. *IEEE Trans. Evol. Comput.*, **7**(5):445-455. [doi:10.1109/TEVC.2003.817236]

Hadj-Alouane, A.B., Bean, J., 1997. A genetic algorithm for the multiple-choice integer program. *Oper. Res.*, **45**(1):92-101. [doi:10.1287/opre.45.1.92]

Hinterding, R., Michalewicz, Z., 1998. Your Brains and My Beauty: Parent Matching for Constrained Optimisation. IEEE Int. Conf. on Evolutionary Computation Proc. IEEE World Congress on Computational Intelligence, p.180-185. [doi:10.1109/ICEC.1998.700156]

Holland, J.H., 1962. Outline for a logical theory of adaptive systems. *J. ACM*, **9**(3):297-314. [doi:10.1145/321127.321128]

Holland, J.H., 1992. Adaptation in Natural and Artificial Systems. MIT Press Cambridge, MA, USA.

Kennedy, J., Eberhart, R., 1995. Particle Swarm Optimization. Proc. IEEE Int. Conf. on Neural Networks, p.942-948. [doi:10.1109/ICNN.1995.488968]

Kowalczyk, R., 1997. Constraint Consistent Genetic Algorithms. IEEE Int. Conf. on Evolutionary Computation, p.343-348. [doi:10.1109/ICEC.1997.592333]

Mezura-Montes, E., Coello, C., 2005. A simple multimembered evolution strategy to solve constrained optimization problems. *IEEE Trans. Evol. Comput.*, **9**(1):1-17.

[doi:10.1109/TEVC.2004.836819]

Mezura-Montes, E., Coello, C.A.C., Tun-Morales, E.I., 2004. Simple Feasibility Rules and Differential Evolution for Constrained Optimization. Proc. 3rd Mexican Int. Conf. on Artificial Intelligence, p.707-716. [doi:10.1007/b96521]

Michalewicz, Z., 1996. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, New York.

Michalewicz, Z., Nazhiyath, G., 1995. Genocop III: A Co-evolutionary Algorithm for Numerical Optimization problems with Nonlinear Constraints. Proc. IEEE Int. Conf. on Evolutionary Computation, p.647-651. [doi:10.1109/ICEC.1995.487460]

Michalewicz, Z., Deb, K., Schmidt, M., Stidsen, T., 2000. Test-case generator for nonlinear continuous parameter optimization techniques. *IEEE Trans. Evol. Comput.*, **4**(3):197-215. [doi:10.1109/4235.873232]

Powell, D., Skolnick, M., 1993. Using Genetic Algorithms in Engineering Design Optimization with Non-linear Constraints. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Qin, A.K., Suganthan, P.N., 2005. Self-adaptive Differential Evolution Algorithm for Numerical Optimization. IEEE Congress on Evolutionary Computation, p.1785-1791. [doi:10.1109/CEC.2005.1554904]

Richardson, J.T., Palmer, M.R., Liepins, G., Hilliard, M., 1989. Some Guidelines for Genetic Algorithms with Penalty Functions. Proc. 3rd Int. Conf. on Genetic Algorithms, p.191-197.

Runarsson, T.P., Yao, X., 2000. Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.*, **4**(3):284-294. [doi:10.1109/4235.873238]

Schoenauer, M., Xanthakis, S., 1993. Constrained GA Optimization. Proc. 5th Int. Conf. on Genetic Algorithms, p.573-580.

Storn, R., Price, K., 1997. Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.*, **11**(4):341-359. [doi:10.1023/A:1008202821328]

Wang, Y., Cai, Z., Zeng, W., 2008. An adaptive tradeoff model for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.*, **12**(1):80-92. [doi:10.1109/TEVC.2007.902851]