Available online at www.sciencedirect.com
ScienceDirect
Information Sciences 178 (2008) 3043–3074
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Differential evolution with dynamic stochastic selection for constrained optimization

Min Zhang, Wenjian Luo, Xufa Wang *

Nature Inspired Computation and Applications Laboratory, Department of Computer Science and Technology, University of Science and Technology of China, 230037 Hefei, Anhui, China

Received 30 May 2007; received in revised form 22 November 2007; accepted 4 February 2008

Abstract

How much attention should be paid to the promising infeasible solutions during the evolution process is investigated in this paper. Stochastic ranking has been demonstrated as an effective technique for constrained optimization. In stochastic ranking, the comparison probability will affect the position of feasible solution after ranking, and the quality of the final solutions. In this paper, the dynamic stochastic selection (DSS) is put forward within the framework of multimember differential evolution. Firstly, a simple version named DSS-MDE is given, where the comparison probability decreases linearly. The algorithm DSS-MDE has been compared with two state-of-the-art evolution strategies and three competitive differential evolution algorithms for constrained optimization on 13 common benchmark functions. DSS-MDE is also evaluated on four well-studied engineering design examples, and the experimental results are significantly better than current available results. Secondly, other dynamic settings of the comparison probability for DSS-MDE are also designed and tested. From the experimental results, DSS-MDE is effective for constrained optimization. Finally, DSS-MDE with a square root adjusted comparison probability is evaluated on the 22 benchmark functions in CEC'06, and the experimental results on most functions are competitive.

Keywords: Constrained optimization; Differential evolution; Dynamic stochastic selection; Stochastic ranking

1. Introduction

Evolutionary Algorithms have been extensively used to solve many types of optimization problems [1,2,35]. However, they are mainly unconstrained search techniques that lack an explicit mechanism to bias the search in feasible regions, although it is common to face a large number of constrained optimization problems in many science and engineering fields. Without loss of generality, minimization considered in this paper, the general nonlinear programming (NLP) problem can be formulated as

* Corresponding author. Tel.: +86 0551 3602824 2.
E-mail addresses: zhangmin@mail.ustc.edu.cn (M. Zhang), wjluo@ustc.edu.cn (W. Luo), xfwang@ustc.edu.cn (X. Wang).

0020-0255/$ - see front matter © 2008 Published by Elsevier Inc.
doi:10.1016/j.ins.2008.02.014

$$\text{minimize} \quad f(\mathbf{x}), \qquad \mathbf{x} = (x_1, x_2, \ldots, x_n) \in R^n, \tag{1}$$

where $f(\mathbf{x})$ is the objective function, $\mathbf{x} \in S \cap F$, and S is a n-dimension rectangle space in R^n bounded by the parametric constraints as

$$l(i) \leqslant x_i \leqslant u(i), \quad 1 \leqslant i \leqslant n. \tag{2}$$

The feasible region F is defined by a set of m additional liner or nonlinear constraints ($m \geqslant 0$)

$$F = \{\mathbf{x} \in R^n | g_j(\mathbf{x}) \leqslant 0,\ h_k(\mathbf{x}) = 0,\ 1 \leqslant j \leqslant q < k \leqslant m\}, \tag{3}$$

where q is the number of inequality constraints and $m - q$ is the number of equalities. For an inequality constraint which satisfies $g_j(\mathbf{x}) = 0$ $(1 \leqslant j \leqslant q)$ at any point at $\mathbf{x}$ in region F, we will say it is active at $\mathbf{x}$. And all the equality constraints are considered active at all points in feasible region F.

It is arguably said that obtaining a feasible solution takes precedence of optimizing the objective function when we try to solve the real-world constrained optimization problems [16]. The main challenge in constrained optimization is how to balance the search between feasible and infeasible regions effectively, i.e., to design an efficient constraint handling to locate the global optimum in the feasible region. So how to handle constraints in evolutionary optimization has been concerned by many researchers. Numerous methods have been proposed for handling constraints, including penalty function method, special representations and operators, multiobjective method, repair method, etc. [3].

Among the above methods, special representations and operators, and repair method are very effective to locate feasible solutions, but are limited to certain kind of problems [3]. As for the multiobjective method, constraints are treated as one or several objectives. Although the problem of choosing penalty weights is avoided in the multiobjective method, the computation complexity increases and the problem difficulty does not decrease [16]. According to the analyses in [4], the multiobjective method could not be a good choice for constrained optimization. Nevertheless, the Pareto dominance concept has been successfully used in the approach by Cai and Wang [17].

The penalty function method, due to its simplicity to implementation, has been most widely studied and used so far. Generally, constrained optimization problem is transformed into series of unconstrained ones by introduction of a penalty term, such as

$$\psi(\mathbf{x}) = f(\mathbf{x}) + r_G \phi(\mathbf{x}), \tag{4}$$

where $\phi \geqslant 0$ is a real-valued function constructed from the constraint violations and imposes a "*penalty*" controlled by a sequence of penalty weights $\{r_G\}_0^{MAX\_GEN}$, where G is the generation counter and MAX_GEN is the number of maximum generation in evolution. In particular, the following constraint violation definition has been often used as the penalty function [4,13]:

$$\phi(\mathbf{x}) = \sum_{i=1}^{m} g_i^+(\mathbf{x}), \tag{5}$$

where $g_i^+(\mathbf{x})(1 \leqslant i \leqslant m)$ is the constraint violation for the ith constraint in (3) and its definition is as follows:

$$g_i^+(\mathbf{x}) = \begin{cases} \max[0, g_i(\mathbf{x})], & 1 \leqslant i \leqslant q, \\ \max[0, |h_i(\mathbf{x})| - \delta], & q + 1 \leqslant i \leqslant m, \end{cases} \tag{6}$$

where δ is a small positive constant. So the constrained optimization becomes how to achieve a satisfying compromise between feasible and infeasible regions by handling the objective function in (1) and the penalty function in (5).

As the infeasible solutions are penalized by the function in (4), the search ability in feasible regions will be enhanced. In this method, the most important thing is to set appropriate penalty weights r_G, i.e., to achieve a comfortable balance between feasible and infeasible regions during evolution. In order to get a satisfying result, the penalty weights should be tuned during search or should be reset for different kinds of optimization problems. However, how to decide an optimal value for r_G turns out to be a very difficult problem [6]. So the main drawback of this method is that the penalty weights are difficult to choose and related with the problems [4].

In order to overcome this drawback, many state-of-the-art approaches for constraint handling have been proposed in recent years. Powell and Skolnick [24] suggested a penalty function method in which feasible solutions are always superior to infeasible ones in selection and no penalty weights are used. Although the feasible solutions could be located quickly, the deficiency will occur in problems with disconnected feasible components, because in such cases the genetic algorithm (GA) may be stuck within one of the feasible components and never get to explore [16]. Deb [5] gave three comparison criteria in the tournament select operator for constrained optimization:

1. Any feasible solution is preferred to any infeasible one.
2. Between two feasible solutions, the one having better objective value is preferred.
3. Between two infeasible solutions, the one with smaller constraint violation is preferred.

The comparison criteria do not need any penalty weights but always prefers feasible solutions in the tournament selection. Runarsson and Yao proposed the stochastic ranking method with a comparison probability P_f for the comparison of feasible and infeasible solutions [6], and suggested search biases for constrained optimization [4]. A novel search biases selection strategy was given by Zhang et al. [7]. Barbosa and Lemonge [36] presented a parameter-less adaptive penalty scheme for genetic algorithm to handle constraints. Ray and Liew [22] adopted intra and intersociety interactions within a formal society and the civilization model, and Farmani and Wright [23] applied a two-stage penalty to infeasible solutions with self-adaptive fitness formulation. In addition, Mezura-Montes and Coello [13] presented a simple multimember evolution strategy for constrained optimization, and Venkatraman and Yen [16] established a generic, two-phase framework for solving constrained optimization problems by genetic algorithms.

Differential evolution (DE) proposed by Storn and Price [8] is a relatively new simple evolutionary algorithm, which is an effective adaptive approach to global optimization. It neither employs binary encoding like simple GA nor utilizes a probability density function to adapt its parameters like Evolution Strategy (ES) [9]. Since the differential evolution was suggested for constrained optimization by Lampinen [10], more and more algorithms within the DE framework have been put forward [9,14,18–21]. To the best of our knowledge, among these methods, Deb's three comparison criteria [5] have been mainly adopted in the selection operator for constraint handling. However, the promising infeasible solutions could be discarded during the selection procedure because the feasible ones are always preferred. As a current state-of-the-art selection method, stochastic ranking [6] can maintain the promising infeasible solutions, which is quite beneficial to the problems with disconnected feasible regions and the ones with the optimal in the feasible region boundaries. The satisfying results in [6] have provided an intuitional evidence for this.

In this paper, stochastic ranking is applied to multimember DE framework [11] that is different from the most current DE frameworks for constrained optimization. After analyzing locations of the feasible solution after the stochastic ranking with different comparison probability P_f, and the attention required to the promising infeasible solutions during evolution, we propose a novel dynamic stochastic selection (DSS) here for the multimember DE to solve constrained problems. The simple version named DSS-MDE, where the comparison probability decreases linearly, has been compared with state-of-the-art algorithms on 13 common benchmark functions and four well-studied engineering design examples. Especially, the experimental results on four well-studied engineering design examples are significantly better than current available results. Different dynamic settings for the comparison probability P_f are discussed, and tested on 13 common benchmark functions. From the experimental results, DSS-MDE is effective for constrained optimization. In addition, with a square root adjusted comparison probability, DSS-MDE is evaluated on the 22 benchmark functions in CEC'06, and the experimental results on most functions are competitive.

The rest of the paper is organized as follows. In Section 2, a novel dynamic stochastic selection for multimember DE is proposed by analyzing the different impacts of promising infeasible solutions during evolution. A simple implementation named DSS-MDE with the linearly decreasing comparison probability is given. In Section 3, 13 common benchmark functions and four well-studied engineering design examples are used to test the performance of DSS-MDE, and different crossover rates are also investigated. Section 4 discusses and compares different dynamic settings of the comparison probability. Moreover, DSS-MDE with a square root adjusted comparison probability is also evaluated on the 22 benchmark functions in CEC'06. Finally, the whole paper is summarized with a brief conclusion in Section 5.

2. Dynamic stochastic selection for DE

In this section, a brief introduction of stochastic ranking [6] and multimember DE [12] will be presented before the novel dynamic stochastic selection (DSS) proposed for constraint handling within the framework of multimember DE.

2.1. Stochastic ranking and multimember DE

In constrained optimization, how to balance the objective function and constraint violation is always a fundamental problem [4]. Runarsson and Yao [6] analyzed the relationship between the objective function and constraint violation, and suggested stochastic ranking for constraint handling. The description of stochastic ranking is illustrated in Fig. 1, where the f and Φ denote the objective function value and constraint violation, respectively.

The main ideas of stochastic ranking are: the ranking is based on f value when two adjacent individuals are feasible, otherwise based on f value with probability of 0.45 or Φ value with probability of 0.55. Therefore, the ranking of the whole population is achieved by a bubble-sort-like procedure, which is halted when the rank ordering do not change within a complete sweep. Such ranking ensures that the good feasible solutions as well as promising infeasible ones will be ranked in the top of population, i.e., a satisfying balance between f and Φ could be achieved.

Multimember DE was suggested by Storn [11] and applied to solve constrained optimization problems by Mezura-Montes et al. [9]. Unlike the common DE [8], multimember DE generates M ($M > 1$) children for each individual with three random selected distinct individuals in current generation, and then only one of the $M + 1$ individuals will survive in the next generation.

2.2. DSS: dynamic stochastic selection

In Section 2.1, both stochastic ranking and multimember DE are introduced. It seems that a straightforward method for constrained optimization is to apply stochastic ranking [6] in the selection operator of multimember DE. However, if stochastic ranking [6] is directly applied to the selection operator of multimember DE, the performance is not competitive according to the following experimental results.

```
I_j = j ∀j ∈ {1,L ,λ}
for i = 1 to   do
  for j = 1 to λ -1 do
    sample u ∈ U(0,1)
    if (Φ(I_j)=Φ(I_{j+1})=0) or (u<0.45) then
      if f(I_j) > f(I_{j+1}) then
        swap(I_j, I_{j+1})
      end
      else
      if Φ(I_j) >Φ(I_{j+1}) then
        swap(I_j, I_{j+1})
      end
    end
  end
  if no swap done then break end
end
```

Fig. 1. Stochastic ranking algorithm in [6] where $U(0,1)$ is a uniform random number generator.

Thirteen benchmark functions taken from [4] are used to test this straightforward method, in which the value of M is set to 5 as recommended in [9]. In the experiment, 30 independent runs are performed for each function, where the population size is set to 50 and the number of maximum function evaluations is set to 350,000. The experimental results are listed in Table 1.

In Table 1, the statistical features (best, median and worst values) of the best feasible solutions obtained in 30 runs for each function are used to evaluate the performance of this straightforward method, and "NA" stands for not available for the corresponding value because no feasible solution has been found in all runs. The infeasible times in the 30 runs are illustrated in the last column (*Inf.*). As illustrated in Table 1, the method has only obtained optimum values on functions g08, g11 and g12, while no optimum values have been found on the left 10 functions, and the results are extremely bad on g01, g05 and g13 for almost no feasible solutions have been found during the 350,000 function evaluations for all 30 runs. However, within the framework of evolution strategy [6], the stochastic ranking has obtained satisfying results on the 13 functions. So the potential factors that affect the performance should be investigated in order to find a simple and effective technique for constrained optimization.

As stated before, the most important problem in constrained optimization is how to balance the search between feasible and infeasible regions [4]. As for stochastic ranking, the balance could be achieved by setting appropriate value for the comparison probability P_f (i.e. 0.45) according to the analyses and numerical results in [6]. However, this situation might change because of the different mechanisms in ES and multimember DE. To check whether this value is still effective in the multimember DE condition, an experiment is designed as follows.

Two arrays of f and Φ with the same size $M+1$ are generated uniformly in $(0,1)$, denoting the objective function values and constraint violations of $M+1$ infeasible solutions, respectively. One solution a randomly selected from $M+1$ infeasible solutions is converted to a feasible one by setting the corresponding value in array Φ as 0, i.e. $\Phi(a)=0$. As for this feasible solution, $f(a)$ may be larger or smaller than other values in array f. In other words, the rank of $f(a)$ in array f is uniformly distributed in $[1, M+1]$. Notably, after stochastic ranking, only the solution occupying the first rank will survive in the next generation as for multimember DE. So the feasible solution a will be selected if and only if it has occupied the first rank after stochastic ranking. Now we turn to analyze the selection probability of the feasible solution a with different comparison probability P_f.

However, it is very hard to adopt theoretical analyses to obtain the selection probabilities of the feasible solution. The numerical analysis is used instead. In this experiment, the value of M is set to 5 and the probability P_f is set from 0.45 to 0.00 with an interval of 0.05. The probabilities are shown in Table 2, in which each cell contains the mean value in 100,000 independent times.

As shown in Table 2, when $P_f = 0.45$, if the objective function value of the feasible solution a is the best, i.e. the rank of $f(a)$ before stochastic ranking is 1, its selection probability is 1.00000. However, if the rank of $f(a)$ before stochastic ranking is 3, its selection probability is only 0.62708 when $P_f = 0.45$.

Table 1
Results of using stochastic ranking in multimember DE

Fcns	Optimum	Best	Median	Worst	Inf.
g01	−15	−3.6296	−2.4075	−0.0084	21
g02	0.803619	0.8016	0.7921	0.7354	0
g03	1	0.7239	0.2967	0.0328	0
g04	−30665.539	−30637.803	−30620.826	−30602.309	0
g05	5126.4981	NA	NA	NA	30
g06	−6961.814	−6961.774	−6961.362	−6960.621	0
g07	24.306	25.0867	25.8067	26.5107	0
g08	0.095825	−0.095825	−0.095825	−0.095825	0
g09	680.630	680.743	680.828	680.975	0
g10	7049.3307	8590.212	9972.450	11883.278	0
g11	0.75	0.7499	0.7500	0.7525	0
g12	1	1	1	1	0
g13	0.053950	NA	NA	NA	30

Table 2
Selection probabilities of the feasible solution

P_f	Rank of $f(a)$ before stochastic ranking					
	1	2	3	4	5	6
0.45	1.00000	0.77601	0.62708	0.51416	0.42547	0.35318
0.40	1.00000	0.83124	0.70267	0.60236	0.51707	0.44935
0.35	1.00000	0.87314	0.76859	0.68339	0.61000	0.54633
0.30	1.00000	0.90938	0.83032	0.76228	0.70032	0.64287
0.25	1.00000	0.93989	0.88313	0.83209	0.78259	0.73600
0.20	1.00000	0.96128	0.92562	0.88945	0.85506	0.82055
0.15	1.00000	0.97851	0.95749	0.93638	0.91643	0.89414
0.10	1.00000	0.99060	0.98058	0.97135	0.96111	0.95097
0.05	1.00000	0.99780	0.99516	0.99255	0.99100	0.98792
0.00	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000

In Table 2, the value is the selection probability for the feasible solution in just one generation. Suppose there is only one feasible solution in current $(M+1)$ solutions, and the problem is heavily constrained so that it is impossible to generate other feasible solutions during next T generations. Therefore, the selection probability $P(T)$ for the feasible solution after T generations is

$$P(T) = \prod_{i=1}^{T} p_i, \tag{7}$$

where p_i is the selection probability in the ith generation.

From Table 2 it can be seen that the probability of a feasible solution surviving in the evolution is relatively lower when both the comparison probability P_f and its objective rank are relatively larger, i.e., the selection probability p_i is relatively smaller. For example, let $T = 10$ and $p_i \leqslant 0.6$, then $P(T) \leqslant 0.6^{10} \approx 0.006$. Therefore, for heavily constrained optimization, the feasible solution may be lost quickly. But the most important and necessary thing in constrained optimization is to find out feasible solutions. So we should need larger p_i in order to maintain the feasible solution, especially in the late stage of evolution. Contrarily, in the early stage of evolution, it could be benefited from the promising infeasible solutions to locate the optimum, i.e. smaller p_i. Summarily, the promising infeasible solutions do not have the same effect during the different stages of evolution. These will be explained by the example in Fig. 2.

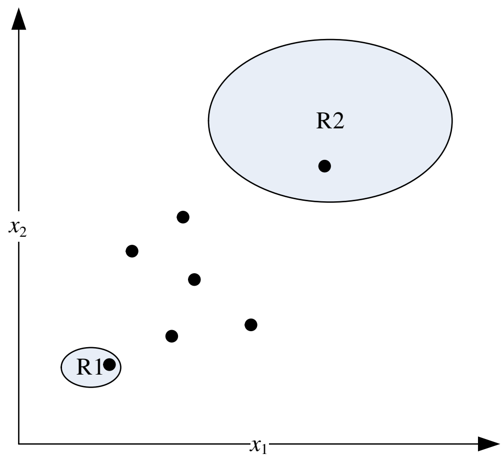

Fig. 2. A simple 2D example of constrained optimization.

Fig. 2 is a 2D constrained optimization problem where only R1 and R2 regions are feasible and the optimum is in R1. Because the region R1 is quite small, the feasible individuals of initial population will have much larger probability located in R2 than R1. In order to locate the optimum, there should be feasible solutions in R1 firstly. So the promising infeasible individuals, i.e., points between regions R1 and R2 can act as the bridge to connect both regions when there are no solutions in R1. However, when individuals enter into R1, the promising infeasible ones do not have the same effect as before. This is because that the optimum can be located by utilizing the feasible solutions in R1 and the promising infeasible solutions might slow down the speed of locating the optimum, in despite of their relatively lower objective values. From the above analysis, it can be obtained that the "*promising*" infeasible solutions are not always "*promising*" during the entire evolution, and they could be effective before population enter into the optimum regions, i.e., early stage of the evolution. On the other hand, if the region R1 is too small or the distance between R1 and R2 are too large, the population might never enter into R1 during the entire evolution under limited evaluations. Although the promising infeasible solutions are always promising from the point of locating optimum view, it is much better to find out feasible solutions than infeasible ones for many types of constrained optimization problems, especially in the real-world conditions. This is because infeasible solutions will have no usable value in real-world applications. In other words, in the late stage of evolution, the best solution in R2, i.e., a local optimum could be a much better alternative than infeasible or any other solutions in R2.

From the above example and corresponding analysis, we have known that the promising infeasible solutions should be paid more attention in the early stage of evolution, such as Runarsson and Yao's stochastic ranking [6], and the same for the feasible solutions in the late stage, such as Deb's three comparison criteria [5]. From the description of stochastic ranking in Fig. 1, the comparison strategy will be similar to Deb's three comparison criteria if the comparison probability P_f is set to 0. And from the results in Table 2, for the stochastic ranking in multimember DE, the probability P_f can be dynamically adjusted during the evolution to achieve the above goals.

The dynamic adjusting of comparison probability is to decrease P_f gradually during the entire evolution, and a simple implementation is as follows:

$$P_f(G) = 0.45 \cdot (1 - G/MAX\_GEN), \tag{8}$$

where G is number of the current generation, MAX_GEN is the maximal generation number in the evolution, and the initial value is set to 0.45 according to the analyses in [6] (the more feasible solutions are more preferred anyway). Furthermore, these are also other types of dynamic settings, which will be compared and discussed in Section 4.

2.3. DSS-MDE: multimember DE with DSS

The dynamic stochastic selection with Eq. (8) is applied to the multimember DE, and the description of the whole algorithm denoted as DSS-MDE is demonstrated in Fig. 3. In Fig. 3, CR and D are the crossover constant (between 0 and 1) and the dimension of parameter variables respectively, and the first element of the subpopulation (composed of one parent and M children) after stochastic ranking will survive in the next generation. To keep solutions within the parametric bounds defined by the problem, our strategy is very simple: When an offspring is generated outside the bounds, the out bound part will be replaced by random numbers between the bounds of the corresponding dimensions.

In Stochastic Ranking Algorithm [6] illustrated in Fig. 1, the ranking is achieved by a bubble-sort-like procedure. Therefore, the number of comparison among individuals in worst case is $\lambda(\lambda - 1)$, where λ is the size of population. And in our algorithm DSS-MDE, the stochastic ranking is adopted for the selection in $M + 1$ individual, so the number of comparison among individuals in worst cast is $(M + 1)MN$ where N is the size of population, which is much less than in Stochastic Ranking Algorithm [6] as $M \ll \lambda$ in the implementation. However, it should be pointed out that although the number of comparison among individuals in DSS-MDE in worst cast is $M + 1$ times of the one in traditional DE approaches [10] under the same number of function evaluations, M is a small const in the experiment ($M = 5$) and the majority of computation time is spent on the evaluation of the objective function, especially for the complex real-world constrained problems.

```
G=0
Initialize population x_{i,G}, i = 1,…,N
Evaluate f(x_{i,G}), φ(x_{i,G}), i = 1,…,N
for G=1 to MAX_GEN do
   for k=1 to N do
      F=rand[0.3,0.9]
      for i=1 to M do
         select randomly r_1 ≠ r_2 ≠ r_3 ∈ [1, N]
         rnbr=rand(1,D)
         for j=1 to D do
            if rand[0,1]<CR or j= =rnbr then
               child_i(j) = x_{r_3,G}(j) + F · (x_{r_1,G}(j) − x_{r_2,G}(j))
            else
               child_i(j) = x_{k,G}(j)
            end if
         end for
      end for
      Evaluate f(child_l), φ(child_l), l = 1,…,M
      P_f = 0.45 · (1 − G / MAX_GEN)
      sp = {x_{k,G}, child_l; 1 ≤ l ≤ M}
      sr_index=StochaticRanking(f(sp),Φ(sp),P_f)
      x_{k,G+1} = sp(sr_index(1))
   end for
end for
```

Fig. 3. Description of the algorithm of DSS-MDE with dynamic stochastic selection.

3. Experimental results

In order to evaluate the performance of the algorithm DSS-MDE, 13 common benchmark functions taken from [6] are used in the experiments. These test functions contain the characteristics which are representative of what could be considered "difficult" constrained optimization problems for an evolutionary algorithm, and their expressions are provided in the Appendix.

To get a measure of the difficulty of solving each of these problems, a ρ metric suggested by [12] is computed by the following expression:

$$\rho = |F|/|S|, \tag{9}$$

where $|F|$ is the number of feasible ones in the total number of $|S|$ randomly generated solutions. In this work, $S = 1{,}000{,}000$ random solutions. This metric and others [6] of each function are listed in Table 3 as described in [7], where n is the number of decision variables, LI is the number of linear inequalities, NE is the number of nonlinear equalities, NI is the number of nonlinear inequalities and a is the number of active constraints at the global optimum.

3.1. Comparison with current state-of-the-art and competitive DE approaches

In the 13 benchmark functions, g02, g03, g08 and g12 are maximization problems and are transformed into minimization ones by using $-f(\boldsymbol{x})$, and the tolerance of violation δ for equality constraints is set to 0.0001.

The DSS-MDE is compared with two types of approaches: (1) the current state-of-the-art approaches, (2) competitive approaches based on DE. To the best of our knowledge, the improved version of Stochastic Ranking approach (ISR) by Runarsson and Yao [6] and the Simple Multimember Evolution Strategy (SMES) by Mezura-Montes and Coello [13] are chosen here as the first type. And three competitive approaches are

Table 3
Summary of main properties of the benchmark functions [7]

fcn	n	$f(x)$ type	ρ (%)	*LI*	*NE*	*NI*	*a*
g01	13	quadratic	0.0003	9	0	0	6
g02	20	nonlinear	99.9962	1	0	1	1
g03	10	polynomial	0.0002	0	1	0	1
g04	5	quadratic	26.9089	0	0	6	2
g05	4	cubic	0.0000	2	3	0	3
g06	2	cubic	0.0065	0	0	2	2
g07	10	quadratic	0.0001	3	0	5	6
g08	2	nonlinear	0.8488	0	0	2	0
g09	7	polynomial	0.5319	0	0	4	2
g10	8	linear	0.0005	3	0	3	3
g11	2	quadratic	0.0099	0	1	0	1
g12	3	quadratic	4.7452	0	0	9^3	0
g13	5	exponential	0.0000	0	3	0	3

selected as the second type, which are the DE approach with re-insertion (RDE) by Mezura-Montes et al. [14], the extended DE approach (EXDE) by Lampinen [10], and the diversity DE approach (DDE) by Mezura-Montes et al. [9].

The parameters of our approach DSS-MDE are set as follows: population size $N = 50$, crossover rate $CR = 0.9$ and the number of children generated by each parent $M = 5$. The number of generation is chosen as 900 and 1400, i.e., 225,000 and 350,000 evaluations of objective function to make fair comparisons, because the different evaluations are used in the above selected approaches. And the approach DSS-MDE with the two different maximum generations is denoted as DSS-MDE-1 and DSS-MDE-2, respectively. The main parameter settings of the six approaches are illustrated in Table 4, where a dynamic tolerance decrease is used in SMES, and the tolerances for g03, g11 and g12 functions are 0.001 in RDE.

The approach DSS-MDE is implemented in C++, and the source code may be obtained from the author upon request. Both the DSS-MDE-1 and DSS-MDE-2 are performed 100 independent runs for each function, and the statistical features of the best solutions obtained by these approaches are presented in Table 5.

In Table 5, because the std. dev. values of RDE and EXDE are not available, we have only done the approximate two-sample t-tests [37] between the approaches ISR, SMES, DDE and our approach DSS-MDE according to

$$t_0 = \frac{\bar{y}_1 - \bar{y}_2}{\sqrt{S_1^2/n_1 + S_2^2/n_2}}, \tag{10}$$

where $\bar{y}_1$ and $\bar{y}_2$, S_1 and S_2 denote the mean values and the standard deviations of the results obtained by the two approaches, and n_1 and n_2 are the independent runs of two approaches, respectively. The value of degrees of freedom is calculated as follows:

Table 4
Main parameter settings of the approaches in comparison

Algorithms	Evaluations	Runs	Tolerance δ	Population[a]
ISR [6]	350,000	100	0.0001	400
SMES [13]	240,000	30	0.0004	300
RDE [14]	348,000	30	0.0001	60
EXDE [10]	10,000–12,000,000	1000	0.001	15–120
DDE [9]	225,000	100	0.0001	450
DSS-MDE-1	225,000	100	0.0001	250
DSS-MDE-2	350,000	100	0.0001	250

[a] This value equals the number of offspring generated by the parent population.

Table 5
Statistical features of the best solutions obtained by ISR, SMES, RDE, EXDE, DDE, and DSS-MED

Function and optimum	Statistical features	Approaches for constrained optimization						
		ISR [6]	SMES [13]	RDE [14]	EXDE [10]	DDE [9]	DSS-MDE-1	DSS-MDE-2
g01	Best	**−15.000**	**−15.000**	**−15.000**	**−15.000**	**−15.000**	**−15.000**	**−15.000**
	Mean	**−15.000**	**−15.000**	**−14.792**	**−15.000**	**−15.000**	**−15.000**	**−15.000**
−15.000	Worst	**−15.000**	**−15.000**	**−12.743**	**−15.000**	**−15.000**	**−15.000**	**−15.000**
	Std. dev.	5.8E−14	**0**	NA	NA	1.0E−09	1.3E−10	**0**
g02	Best	**0.803619**	0.803601	**0.803619**	NA	**0.803619**	**0.803619**	**0.803619**
	Mean	0.782715	0.785238	0.746236	NA	**0.798079**	0.786970	0.788011
0.803619	Worst	0.723591	0.751322	0.302179	NA	**0.751742**	0.728531	0.744690
	Std. dev.	2.2E−02	1.7E−02	NA	NA	**1.01E−02**	1.5E−02	1.5E−02
g03	Best	**1.001**	**1.000**	**1.000**	**1.0252**	**1.000**	**1.0005**	**1.0005**
1.000	Mean	**1.001**	**1.000**	0.640	**1.0252**	**1.000**	**1.0005**	**1.0005**
	Worst	**1.001**	**1.000**	0.029	**1.0252**	**1.000**	**1.0005**	**1.0005**
	Std. dev.	8.2E−09	2.1E−04	NA	NA	**0**	1.9E−08	2.7E−09
g04	Best	**−30665.539**	**−30665.539**	**−30665.539**	**−31025.6**	**−30665.539**	**−30665.539**	**−30665.539**
−30665.539	mean	**−30665.539**	**−30665.539**	−30592.154	**−31025.6**	**−30665.539**	**−30665.539**	**−30665.539**
	Worst	**−30665.539**	**−30665.539**	−29986.214	**−31025.6**	**−30665.539**	**−30665.539**	**−30665.539**
	Std. dev.	1.1E−11	**0**	NA	NA	**0**	2.7E−11	2.7E−11
g05	Best	**5126.497**	5126.599	**5126.497**	**5126.484**	**5126.497**	**5126.497**	**5126.497**
	Mean	**5126.497**	5174.492	5218.729	**5126.484**	**5126.497**	**5126.497**	**5126.497**
5126.498	Worst	**5126.497**	5304.167	5502.410	**5126.484**	**5126.497**	**5126.497**	**5126.497**
	Std. dev.	7.2E−13	5.0E+01	NA	NA	**0**	**0**	**0**
g06	Best	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**
−6961.814	mean	**−6961.814**	−6961.284	−6367.575	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**
	Worst	**−6961.814**	−6952.482	−2236.950	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**
	Std. dev.	1.9E−12	1.9E+00	NA	NA	**0**	**0**	**0**
g07	Best	**24.306**	24.327	**24.306**	**24.306**	**24.306**	**24.306**	**24.306**
24.306	Mean	**24.306**	24.475	104.599	**24.306**	**24.306**	**24.306**	**24.306**
	Worst	**24.306**	24.843	1120.541	24.307	**24.306**	**24.306**	**24.306**
	Std. dev.	6.3E−05	1.3E−01	NA	NA	**8.22E−09**	7.5E−07	7.0E−08
g08	Best	**0.095825**	**0.095825**	**0.095825**	**0.095825**	**0.095825**	**0.095825**	**0.095825**
0.095825	Mean	**0.095825**	**0.095825**	0.091292	**0.095825**	**0.095825**	**0.095825**	**0.095825**
	worst	**0.095825**	**0.095825**	0.027188	**0.095825**	**0.095825**	**0.095825**	**0.095825**
	Std. dev.	2.7E−17	**0**	NA	NA	**0**	4.0E−17	3.9E−17
g09	Best	**680.630**	680.632	**680.630**	**680.630**	**680.630**	**680.630**	**680.630**
	Mean	**680.630**	680.643	692.472	**680.630**	**680.630**	**680.630**	**680.630**
680.630	Worst	**680.630**	680.719	839.78	**680.630**	**680.630**	**680.630**	**680.630**
	Std. dev.	3.2E−13	1.6E−02	NA	NA	**0**	2.9E−13	2.5E−13
g10	Best	**7049.248**	7051.903	**7049.248**	**7049.248**	**7049.248**	**7049.248**	**7049.248**
	Mean	7049.250	7253.047	8442.66	**7049.248**	7049.266	7049.249	**7049.248**
7049.248	Worst	7049.270	7638.366	15580.37	**7049.248**	7049.617	7049.255	7049.249
	Std. dev.	3.2E−03	1.4E+02	NA	NA	4.45E−02	1.4E−03	**3.1E−04**
g11	Best	**0.75**	**0.75**	**0.75**	**0.75**	**0.75**	**0.7499**	**0.7499**
	Mean	**0.75**	**0.75**	0.76	**0.75**	**0.75**	**0.7499**	**0.7499**
0.75	Worst	**0.75**	**0.75**	0.87	**0.75**	**0.75**	**0.7499**	**0.7499**
	Std. dev.	1.1E−16	1.5E−04	NA	NA	**0**	**0**	**0**
g12	Best	**1.000**	**1.000**	**1.000**	NA	**1.000**	**1.000**	**1.000**
1.000	Mean	**1.000**	**1.000**	**1.000**	NA	**1.000**	**1.000**	**1.000**
	Worst	**1.000**	**1.000**	**1.000**	NA	**1.000**	**1.000**	**1.000**
	Std. dev.	1.2E−09	**0**	NA	NA	**0**	**0**	**0**

(*continued on next page*)

Table 5 (*continued*)

Function and optimum	Statistical features	Approaches for constrained optimization						
		ISR [6]	SMES [13]	RDE [14]	EXDE [10]	DDE [9]	DSS-MDE-1	DSS-MDE-2
g13	Best	**0.053942**	0.053986	**0.053866**	NA	**0.053941**	**0.053942**	**0.053942**
	Mean	0.06677	0.166385	0.747227	NA	0.069336	**0.053942**	**0.053942**
0.053950	Worst	0.438803	0.468294	2.259875	NA	0.438803	**0.053942**	**0.053942**
	Std. dev.	7.0E−02	1.8E−01	NA	NA	7.58E−02	1.0E−13	**8.3E−17**

A result in boldface indicates the best result or the global optimum. NA means no available.

$$f = \left\lfloor \frac{1}{k^2/n_1 + (1-k)^2/n_2} \right\rfloor, \tag{11}$$

where $k = \frac{S_1^2/n_1}{S_1^2/n_1 + S_2^2/n_2}$. The *t*-tests are not done on the functions where both approaches have obtained the optimum in all runs, and the values of t_0 and degree of freedom are listed in Table 6.

ISR is one of current most competitive approaches for constrained optimization, and it can be observed from Table 5 that it has found out the optimum in each run on all functions except g02, g10 and g13. From the results of DSS-MDE-2 in Table 5, DSS-MDE is better than ISR in mean, st. dev., and worst values in g02 and g10, and converges to the optimum of g13 in all 100 runs. For the left 10 functions, not only DSS-MDE finds the optimum in each run, but also the std. dev. values are much less than ISR on g01, g05, g06, g07, g11 and g12 and the both performances are similar on the other functions. For the other state-of-the-art algorithm SMES, from the results in Tables 5 and 6, our algorithm is distinctly better than SMES on g05, g07, g09, g10 and g13 functions, and the std. dev. values are smaller than SMES on g02, g03, g06 and g11. Only on g01 SMES is slightly better in std. dev. values.

From the results of RDE and DSS-MDE-2 (columns 5 and 9 in Table 5), it can be seen our approach DSS-MDE-2 has distinct superiority over all the functions except g12, on which the results of both approaches are similar. And for EXDE (column 6 in Table 5), the best solutions are smaller than the optimum or the ones found by DSS-MDE-2 on some functions, such as g04 and g05, because a more relaxed tolerance (0.001) is adopted for the equality constraint transformation. And the approach EXDE used different parameters for each function, so it is difficult to make a fair comparison between EXDE and DSS-MDE-2. But we can see that our approach is slightly better on function g07, and is similar on the left 9 functions except g10, on which EXDE has slight superiority in the mean and worst values. The DDE approach by Mezura-Montes et al. [9] adopted the multimember mechanism and allowed promising infeasible solutions remained in population to maintain the diversity. Like ISR, the approach has found the optimum in all 100 runs except g02, g10 and g13, and the statistical features on g02 are the best among these approaches. From the results of DDE and DSS-MDE-1 in Tables 5 and 6, our approach is significantly better than DDE on g10 and g13, and is similar on the other functions except g02 where DDE is significantly better. Noted that a relatively larger population is used in DDE, but its number of function evaluation is one of the smallest.

Table 6
Results of the approximate two-sample *t*-tests between ISR, SMES, DDE and DSS-MDE

Approaches	Features of *t*-test	g02	g05	g06	g07	g09	g10	g13
DSS-MDE-2 vs. ISR	t_0	2.0^+	–	–	–	–	−6.2[a]	−1.8
	f	176	–	–	–	–	101	100
DSS-MDE-1 vs. SMES	t_0	0.5	−5.3[a]	−1.5	−7.1[a]	−4.5[a]	−8.0[a]	−3.4[a]
	f	44	30	30	30	30	30	30
DSS-MDE-1 vs. DDE	t_0	−6.1[a]	–	–	–	–	−3.8[a]	−2.0[a]
	f	175	–	–	–	–	100	100

"–" means the both approaches have obtained the optimum in all runs on the given functions.
[a] The value of t_0 with the corresponding degrees freedom is significant at $\alpha = 0.05$ by an approximate two-sample *t*-test.

Table 7
Statistic features of our approach DSS-MDE with different *CR* values

function & optimum	Statistical features	Parameter *CR*							
		0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8
g02	Best	0.800214	0.803136	0.803385	0.803463	0.803463	0.803529	0.803551	**0.803596**
0.803619	Mean	0.789199	**0.796703**	0.795759	0.795502	0.793504	0.791831	0.781255	0.759052
	Worst	0.763478	**0.764532**	0.735505	0.751141	0.740907	0.727034	0.680007	0.602831
	Inf.	0	0	0	0	0	0	0	0
g03	Best	0.969537	0.957920	0.947137	0.928927	0.923560	0.909603	0.949990	**0.980481**
1.000	Mean	0.889757	0.866486	0.833995	0.798943	0.767924	0.767003	0.807954	**0.907068**
	Worst	**0.795364**	0.753024	0.628312	0.647051	0.629014	0.536221	0.648440	0.753018
	Inf.	0	0	0	0	0	0	0	0
g04	Best	−30,665.537	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**
−30,665.539	Mean	−30,665.528	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**
	Worst	−30,665.478	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**	**−30,665.539**
	Inf.	0	0	0	0	0	0	0	0
g05	Best	5171.284	5126.505	**5126.497**	**5126.497**	**5126.497**	**5126.497**	**5126.497**	**5126.497**
5126.498	Mean	5314.752	5140.384	5129.886	5127.390	5126.612	**5126.497**	**5126.497**	**5126.497**
	Worst	5576.237	5192.265	5148.682	5150.675	5137.682	**5126.497**	**5126.497**	**5126.497**
	Inf.	85	24	24	3	0	0	0	0
g07 24.306	Best	24.497	24.337	24.316	24.317	24.317	24.313	24.307	**24.306**
	Mean	25.015	24.492	24.379	24.343	24.337	24.327	24.309	**24.306**
	Worst	26.084	24.855	24.652	24.412	24.394	24.354	24.313	**24.306**
	Inf.	0	0	0	0	0	0	0	0
g09	Best	680.943	680.641	680.633	680.631	**680.630**	**680.630**	**680.630**	**680.630**
	Mean	681.313	680.667	680.640	680.634	680.631	**680.630**	**680.630**	**680.630**
680.630	Worst	681.712	680.722	680.650	680.639	680.631	**680.630**	**680.630**	**680.630**
	Inf.	0	0	0	0	0	0	0	0
g10	Best	7698.179	7310.151	7160.946	7117.518	7076.217	7054.855	7049.407	**7049.248**
7049.248	Mean	9373.436	7994.709	7288.168	7238.495	7127.478	7062.606	7049.847	**7049.249**
	Worst	14106.125	10031.795	7914.944	7313.452	7200.413	7079.203	7050.702	**7049.260**
	Inf.	14	0	0	0	0	0	0	0
g13	Best	0.205384	0.065386	0.054091	0.053947	**0.053942**	**0.053942**	**0.053942**	**0.053942**
0.053950	Mean	0.975714	0.374323	0.074257	0.055626	0.054002	0.053946	**0.053942**	**0.053942**
	Worst	0.999990	0.915269	0.162143	0.103306	0.054867	0.054363	**0.053942**	**0.053942**
	Inf.	62	82	83	58	1	0	0	0

A result in boldface indicates the best result or the global optimum.

3.2. Analyses on crossover rate parameter CR

As stated in [8], the convergence of DE algorithm is related with the crossover rate parameter CR. This means that the DE related algorithms are sensitive to the value of CR under the same number of function evaluations. In this section, we will investigate the performance of our approach DSS-MDE with linear adjusting under different CR values (0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7 and 0.8). The values of remaining parameters are kept as before in Section 3.1, and our approach DSS-MDE is performed 100 independent runs for each CR value under 350,000 function evaluations. In Table 7, we present the statistic features of the best solutions found by DSS-MDE under different CR values (best, mean, worst of feasible solutions, and the number of infeasible solutions (Inf.)). We omit g01, g06, g08, g11 and g12 functions, because the experimental results are exactly the same as the DSS-MDE-2 in Table 7.

From the results in Table 7, it can be seen that with larger CR values, more feasible solutions or more accurate feasible solutions could be obtained by DSS-MDE. So the convergence of our algorithm DSS-MDE will be sped up with larger CR values under the same number of function evaluations.

3.3. Engineering design examples

In order to study the performance of our algorithm DSS-MDE on real-world constrained optimization problems, four well-studied engineering design examples that are chosen from [22] and presented in the Appendix have been solved. And the best results obtained by our algorithm with linear adjusting over 50 independent runs have been compared with those reported in literature. Each example has been solved using a population size N of $3n$–$5n$ (n the dimension of the variable space for each problem) and it has been set to evolve over 300 generations (400 for the Spring Design Example). We have done the approximate two-sample t-tests by Eq. (10) between the approach by Ray and Liew [22] and our approach DSS-MDE (the t_0 values are listed in the parenthesis after the std. dev. values), and have also reported the results of DSS-MDE on these four design examples with much fewer evaluations.

(1) *Welded beam design* [22]: this example is a well-studied constrained single objective optimization problem. It aims to minimize the cost of beam subject to constraints on shear stress, bending stress, bucking load, and the end deflection. Four continuous design variables are the thickness of the beam x_1, width of the bean x_2, length of the weld x_3, and the weld thickness x_4.

A population size N of 16 and a maximum generation MAX_GEN of 300 have been used to solve this problem. The results are reported based on 50 independent runs, and the best, worst, median, mean and standard deviation are illustrated in Table 8. For the best objective value obtained, the values of constraint functions are $[0.000000, -0.000000, -0.000000, -0.234241, -0.000000]$. The details of best solutions by different approaches are presented in Table 9. From the comparisons, it is clear that the results obtained by our approach are better than those by FSA [25], Ray and Liew [22], Deb [5,27], and are competitive with that by He et al. [26].

(2) *Spring design* [22]: a tension/compression spring design considered is to minimize the weight subject to constraints on minimum deflection, shear stress, surge frequency, and limits on outside diameter, which has three continuous variables and four nonlinear inequality constraints.

We have used a population size N of 12 and a maximum generation MAX_GEN of 400 to solve this problem. The best, worst, median, mean and standard deviation are shown in Table 10. For best value obtained by

Table 8
Statistical results of the welded beam example

Algorithms	Pop size	Gen	Best	Median	Mean	Worst	Std. Dev.	Evaluations
DSS-MDE	16	300	**2.38095658**	**2.38095658**	**2.38095658**	**2.38095658**	3.19E−10	24,000
FSA [25]	NA[a]	NA	2.381065	NA	2.404166	2.488967	NA	56,243
Ray and Liew [22]	40	1000	2.3854347	3.2551371	3.0025883	6.3996785	0.959078 (4.6[b])	33,095
Deb [5]	80	500	2.38119	2.39289	NA	2.64583	NA	40,080
Deb [27]	80	4000	2.38119	2.39203	NA	2.64583	NA	320,080

[a] The FSA in [25] is an algorithm based on simulated annealing.
[b] The value of t with 50 degrees of freedom is significant at $\alpha = 0.05$ by a two-sample t-test.

Table 9
Best solutions for the welded beam example

	DSS-MDE	FSA [25]	He et al. [26]	Ray and Liew [22]	Deb [5]	Deb [27]
x_1	0.2443689758	0.24435257	0.244369	0.2444382760	NA	0.2489
x_2	6.2175197152	6.2157922	6.217520	6.2379672340	NA	6.1730
x_3	8.2914713905	8.2939046	8.291471	8.2885761430	NA	8.1789
x_4	0.2443689758	0.24435258	0.244369	0.2445661820	NA	0.2533
Best	**2.38095658**	2.381065	**2.380957**	2.3854347	2.38119	2.43
Evaluations	24,000	56,243	30,000	33,095	40,080	4,500

Table 10
Statistical results of the spring example

Algorithms	Pop size	Gen	Best	Median	Mean	Worst	Std. Dev.	Evaluations
DSS-MDE	12	400	**0.012665233**	**0.012665304**	0.012669366	0.012738262	1.25E−05	24,000
FSA [25]	NA[a]	NA	0.012665258	NA	**0.012665299**	**0.012665338**	NA	49,531
Ray and Liew [22]	30	1000	0.01266924934	0.012922669	0.012922669	0.016717272	5.92E−04 (3.0[b])	25,167
Coello [28]	NA	NA	0.01270478	0.01275576	0.01276920	0.01282208	NA	900,000

[a] The FSA in [25] is an algorithm based on simulated annealing.
[b] The value of t with 50 degrees of freedom is significant at $\alpha = 0.05$ by a two-sample t-test.

Table 11
Best solutions for the spring example

	DSS-MDE	FSA [25]	He et al. [26]	Ray and Liew [22]	Coello [28]	Ray and Saini [29]
x_1	0.3567177469	0.35800478345599	0.356750	0.368158695	0.351661	0.321532
x_2	0.0516890614	0.05174250340926	0.051690	0.0521602170	0.051480	0.050417
x_3	11.2889653382	11.21390736278739	11.287126	10.6484422590	11.632201	13.979915
Best	**0.012665233**	0.012665258	**0.012665**	0.01266924934	0.0127047834	0.013060
Evaluations	24,000	49,531	15,000	25,167	900,000	1291

our approach, the values of constraint functions are $[0.000000, 0.000000, -4.053786, -0.727729]$. The details of best solutions by different approaches are given in Table 11. From Table 12, we can see that our approach has found out the best solution when compared with FSA [25], Ray and Liew [22], Coello [28], and Ray and Saini [29]. Although in Table 13 the mean and worst of the best results obtained by our approach in 50 runs is slightly worse than FSA [25], only about half of function evaluations are used in our approach.

(3) *Spring reducer design*: In this constrained optimization problem, the weight of speed reducer is to be minimized subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts, and stresses in the shafts. The variables x_1–x_7 represent the face width, length of the first shaft between bearings, lengths of the second shaft between bearings, and the diameter of first and second shafts respectively. This is an example of a mixed integer programming problem. The third variable x_3 (number of teeth) is of integer value while all left variables are continuous. And there are 11 constraints in the problem, so it is very hard even to locate a feasible solution [32] (the solution reported in [32] is infeasible).

We have used a population size N of 20 and a maximum generation MAX_GEN of 300 to solve this problem. Table 12 illustrates the best, worst, median, mean and standard deviation. For the best objective value by DDS-MDE, the values of constraint functions are $[-0.073915, -0.197999, -0.499172, -0.904644, -0.000000,$

Table 12
Statistical results of the spring reducer example

Algorithms	Pop size	Gen	Best	Median	Mean	Worst	Std. Dev.	Evaluations
DSS-MDE	20	300	**2994.471066**	**2994.471066**	**2994.471066**	**2994.471066**	3.58E−12	30,000
Montes et al. [30]	60	80	2996.356689	NA	2996.367220	NA	8.2E−03	24,000
Ray & Liew [22]	70	1,000	2994.744241	3001.758264	3001.758264	3009.964736	4.00914232 (12.9[a])	54,456
Akhtar et al. [31]	100	200	3008.08	NA	3012.12	3028.28	NA	19,154

[a] The value of t with 50 degrees of freedom is significant at $\alpha = 0.05$ by a two-sample t-test.

Table 13
Best solutions for the spring reducer example

	DSS-MDE	Montes et al. [30]	Ray and Liew [22]	Akhtar et al. [31]	Kuang et al. [32]
x_1	3.5000000000	3.500010	3.50000681	3.506122	3.6
x_2	0.7000000000	0.700000	0.70000001	0.700006	0.70
x_3	17	17	17	17	17
x_4	7.3000000000	7.300156	7.32760205	7.549126	7.3
x_5	7.7153199115	7.800027	7.71532175	7.859330	7.8
x_6	3.3502146661	3.350221	3.35026702	3.365576	3.4
x_7	5.2866544650	5.286685	5.28665450	5.289773	5.0
Best	**2994.471066**	2996.356689	2994.744241	3008.08	2876.117623
Evaluations	30,000	24,000	54,456	19,154	NA

$0.000000, -0.702500, -0.000000, -0.583333, -0.051326, -0.000000]$. Table 13 shows a detail comparison of the best results, where the solutions located by our approach are the best compared with the other approaches.

(4) *Three-bar truss design*: the last example considered deals with the design of a three-bar truss structure where the volume is to minimized subject to stress constraints [33]. A population size N of 10 and a maximum generation MAX_GEN of 300 have been used to solve this problem. The best, worst, median, mean and standard deviation are reported in Table 14, and the constraint values of the best solution obtained are $[0.000000, -1.464102, -0.535898]$. A detail comparison is presented in Table 15. From Table 15, it can be seen that the solution found out by our approach is the best and the function evaluations of our approach is slightly less than Ray and Liew [22].

Furthermore, in order to validate the effectiveness of our approach DSS-MDE under limited number of function evaluations, much fewer evaluations have been used to solve the four engineering design examples. The corresponding parameter settings for each problem and the statistical features of the best results obtained in 50 independent runs are presented in Table 16. From the results in Table 16, it can be seen that our approach DSS-MDE is quite competitive on the four well-studied engineering design examples under fewer function evaluations, and could be effective for solving the constrained optimization problems when the cost of function evaluation is expensive.

Table 14
Statistical results of the three-bar truss example

Algorithms	Pop size	Gen	Best	Median	Mean	Worst	Std. Dev.	Evaluations
DSS-MDE	10	300	**263.8958434**	**263.8958434**	**263.8958436**	**263.8958498**	9.72E−07	15,000
Ray and Liew [22]	20	1000	263.8958466	263.8989	263.9033	263.96975	1.26E−02 (4.2[a])	17,610

[a] The value of t with 50 degrees of freedom is significant at $\alpha = 0.05$ by a two-sample t-test.

Table 15
Best solutions for the three-bar truss example

	DSS-MDE	Ray and Liew [22]	Ray and Saini [29]	Hernendez [33]
x_1	0.7886751359	0.7886210370	0.795	0.788
x_2	0.4082482868	0.4084013340	0.395	0.408
Best	**263.8958434**	263.8958466	264.3	263.9
Evaluations	15,000	17,610	2712	NA

Table 16
Results of the four engineering design examples for DSS-MDE with fewer function evaluations

Problems	Pop size	Gen	Best	Median	Mean	Worst	Std. Dev.	Evaluations
Weld Beam	16	70	2.3809695	2.3812005	2.41746052	3.33179718	1.45E−01	4200
Spring	12	100	0.012665233	0.012669666	0.012701463	0.013448139	1.14E−04	6000
Speed reducer	14	120	2994.471144	2994.471524	2994.472357	2994.505702	4.83E−03	8400
Three-bar truss	8	120	263.8958434	263.8958434	263.8981518	263.95226	9.19E−03	4800

4. Other dynamic settings for comparison probability

4.1. Experimental comparison of two dynamic settings

The dynamic setting for comparison probability P_f in Section 3.1 adopts the Eq. (8), where the probability decreases linearly with the generation. In this section, another dynamic setting can be used as follows:

$$P_f(G) = 0.45 \cdot \left(1 - \sqrt{G/MAX\_GEN}\right), \tag{11}$$

where the probability P_f is decreases linearly with the square root of generation.

In order to distinguish the two settings, the probability P_f in (8) and (11) are denoted as PL_f and PS_f, respectively. From the definitions in (8) and (11), it can be obtained that $PL_f > PS_f$ in the all generations except MAX_GEN. Therefore, the promising infeasible solutions in (11) will have less importance than in (8), and the probability of striking into local optimum will become larger. However, the speed of convergence to the optimum will be accelerated by (11) if the optimum region has been located. For example, in Fig. 2, if the region R1 has been located by both the (8) and (11), then the (11) will converge to the optimum more quickly, because the feasible solutions are more emphasized in (11). However, the (11) will have more risks to strike into the local optimum in region R2 for the same reason. These are consistent with the No Free Lunch theory [15], which will be illustrated in the following experiments.

In order to compare the (8) and (11), the 13 benchmark functions in Section 3.1 are tested here, and all the parameter settings are the same as before except the comparison probability P_f. The algorithms DSS-MDE with the (8) and (11) settings are denoted as L and S respectively. Both L and S with the maximum generation 900 and 1400 are performed 100 independent runs respectively, and the statistical features including the median and std. dev. values of the best solutions and the median generation of locating the optimum G_m are investigated here. Besides, the current state-of-the-art algorithm ISR by Runarsson and Yao [6] is chosen here for comparison. And because of the different population sizes in these algorithms, the G_m multiplied by the number of evaluations in every generation named NE_m is used instead. The results are shown in Table 17, where L-1 and L-2 represent the L with 900 and 1400 generations respectively, and it is the same for S-1 and S-2.

From the results of ISR, L-2 and S-2 (columns 3, 6 and 7 in Table 17) where 350,000 function evaluations are all used, we can see that both L-2 and S-2 can locate the optimum with much less NE_m than ISR on g04, g05, g06, g09, g11 and g12 functions, and the values of NE_m are almost equal on the left functions. Meanwhile, the median values of the three approaches have all reached the optimum except on function g02. So both the L-2 and S-2 approaches are quite competitive, although the number of function evaluations is relatively less.

Now, we compare the results of L and S approaches with different MAX_GEN settings. Both S-1 and S-2 converge to the optimum more quickly than L-1 and L-2 on functions g04, g05, g06, g09 and g11, and the values of std. dev. are less or equal except on g13. Because the promising infeasible solutions are treated less important in (11) than (8), so once any individuals enter the optimum region, the convergence to the optimum will be accelerated by (11) more than (8), which is consistent with our above analysis. However, if the problem is heavily constrained such as g13, it is harder for (11) to enter the optimum region, and a local optimum may be struck into because the promising infeasible solutions are paid less attention than actually required. The experimental results on function g13 have provided an evidence for this. Anyway, although the attention actually required to the promising infeasible solution is very hard to determine without any prior knowledge and may be problem-dependent, the approach DSS-MDE with linear adjusting in (8) is quite effective to solve constrained optimization problems.

4.2. Discussion

Although it is very hard to determine the attention actually required to the feasible and infeasible solutions, the dynamic setting for the comparison probability P_f could be altered to fulfill the requirement as illustrated in Fig. 4.

In Fig. 4, five different dynamic settings listed from S_1–S_5 could be adopted for different types of constrained optimization problems. S_3 is the liner adjusting of comparison probability and has been studied

Table 17
Statistical features of the best solutions found by ISR, L and S

Function and optimum	Statistical features	Approaches for constrained optimization				
		ISR	L-1	S-1	L-2	S-2
g01	Median	**−15.000**	**−15.000**	**−15.000**	**−15.000**	**−15.000**
−15.000	Std. dev.	5.8E−14	1.3E−10	1.0E−13	**0.0E+00**	**0.0E+00**
	NE_m	350,000	225,000	**224,750**	322,750	253,000
g02	Median	**0.793082**	0.792607	0.792607	0.792608	0.792608
0.803619	Std. dev.	2.2E−02	1.6E−02	**1.5E−02**	**1.5E−02**	1.9E−02
	NE_m	349,600	**224,750**	**224,750**	349,750	349,750
g03	Median	**−1.001**	**−1.0005**	**−1.0005**	**−1.0005**	**−1.0005**
1.000	Std. dev.	8.2E−09	1.9E−08	4.5E−09	2.7E−09	**4.3E−10**
	NE_m	349,200	**225,000**	**225,000**	350,000	350,000
g04	Median	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**	**−30665.539**
−30665.539	Std. dev.	**1.1E−11**	2.7E−11	2.7E−11	2.7E−11	2.7E−11
	NE_m	192,000	110,000	**79,500**	133,000	84,000
g05	Median	**5126.497**	**5126.497**	**5126.497**	**5126.497**	**5126.497**
5126.498	Std. dev.	7.2E−13	**0.0E+00**	**0.0E+00**	**0.0E+00**	**0.0E+00**
	NE_m	195,600	115,000	**79,250**	139,750	84,750
g06	Median	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**	**−6961.814**
−6961.814	Std. dev.	1.9E−12	**0.0E+00**	**0.0E+00**	**0.0E+00**	**0.0E+00**
	NE_m	168,800	39,250	**23,750**	48,750	24,750
g07	Median	**24.306**	**24.306**	**24.306**	**24.306**	**24.306**
24.306	Std. dev.	6.3E−05	7.5E−07	2.3E−07	7.0E−08	**1.6E−08**
	NE_m	350,000	**225,000**	**225,000**	350,000	350,000
g08	Median	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**	**−0.095825**
0.095825	Std. dev.	**2.7E−17**	4.0E−17	3.9E−17	3.9E−17	3.9E−17
	NE_m	160,000	15,000	**14,000**	15,000	15,000
g09	Median	**680.630**	**680.630**	**680.630**	**680.630**	**680.630**
680.630	Std. dev.	3.2E−13	2.9E−13	2.8E−13	**2.5E−13**	**2.5E−13**
	NE_m	271,200	138,500	**118,000**	215,750	171,500
g10	Median	**7049.248**	**7049.248**	**7049.248**	**7049.248**	**7049.248**
7049.248	Std. dev.	3.2E−03	1.0E−03	8.5E−04	3.1E−04	**1.3E−04**
	NE_m	348,800	**225,000**	**225,000**	350,000	350,000
g11	Median	**0.750**	**0.7499**	**0.7499**	**0.7499**	**0.7499**
0.75	Std. dev.	1.1E−16	**0.0E+00**	**0.0E+00**	**0.0E+00**	**0.0E+00**
	NE_m	137,200	51,000	**24,000**	62,750	26,000
g12	Median	**−1.000000**	**−1.000000**	**−1.000000**	**−1.000000**	**−1.000000**
1.000	Std. dev.	1.2E−09	**0.0E+00**	**0.0E+00**	**0.0E+00**	**0.0E+00**
	NE_m	33,600	11,000	11,000	**10,750**	11,250
g13	Median	**0.053942**	**0.053942**	**0.053942**	**0.053942**	**0.053942**
0.053950	Std. dev.	7.0E−02	1.0E−13	4.5E−02	**8.3E−17**	1.4E−10
	NE_m	223,600	162,750	**158,250**	251,750	225,250

A result in boldface indicates the best result or the global optimum.

and compared in detail in Section 3. Compared with S_3, the attention to promising infeasible solution is more in S_1 and S_2 because the comparison probability P_f is relative larger except for the two endpoints, while the attention is less in S_4 and S_5 for the similar reason.

Therefore, if we want to locate the global optimum with larger probability, S_1 or S_2 are better choices for the promising infeasible solutions that would be helpful to escape from local optimums are more emphasized. However, the risk of no feasible solutions located would be increased, which has been analyzed in Section 2.

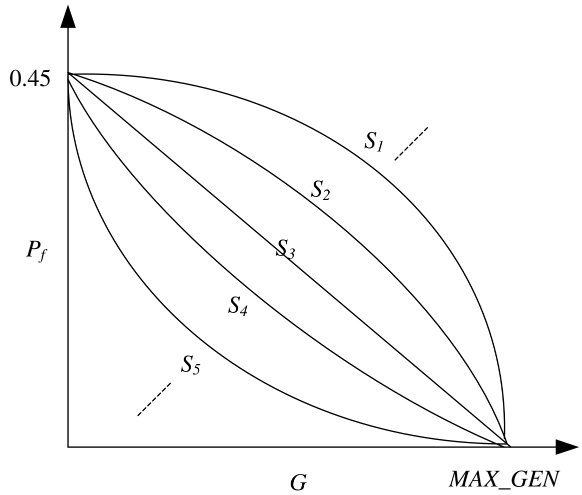

Fig. 4. Different dynamic settings for comparison probability P_f.

On the other hand, S_4 or S_5 could be relatively more effective to locate feasible solutions than S_3 because more attention is paid to the feasible solutions. However, the probability of getting into local optimum would be raised when the attention to promising infeasible solutions does not reach the actually required.

A litter surprisingly, from the above analysis, it is still hard to choose the appropriate dynamic setting for any given problems without any prior knowledge, which accords with the No Free Lunch theory [15]. However, we can utilize different dynamic settings to solve any kind of constrained optimization problems under different requirements. When we want to locate the global optimum, the settings such as S_1 and S_2 would be preferred, and the following dynamic setting could be adopted.

$$P_f(G) = 0.45 \cdot (1 - (G/MAX\_GEN)^r), \quad r > 1. \tag{12}$$

When we want to find out a satisfying feasible solution as soon as possible, our approach DSS-MDE with the adjusting settings such as S_4 and S_5 would be more effective, and a practicable dynamic setting could be

$$P_f(G) = 0.45 \cdot (1 - (G/MAX\_GEN)^r), \quad 1 > r \geqslant 0. \tag{13}$$

Nevertheless, our approach DSS-MDE with linear adjusting is an effective alternative for constrained optimization from the experimental results on the 13 common test functions and four engineering design examples.

4.3. Benchmark functions in CEC'06

In this section, we test our algorithm DSS-MDE on 22 benchmark functions in CEC'06 [34], which are extensions to the 13 test functions used in Section 3.1 and are also described in the Appendix. Moreover, although 24 benchmark functions are defined in [34], two of the functions are not used to evaluate our algorithm DSS-MDE because no feasible optimal solutions ever been found for these two problems (g20 and g22) so far.

As the number of maximum function evaluations is set to 5×10^5 according to the settings in [34], we just set the number of maximum generation to 2000 and keep the other parameters the same as in Section 3.1. Because the maximum generation is larger than in Section 3.1, the adjusting in Eq. (11) is adopted for DSS-MDE. The experimental results are summarized in Tables 18–22 according to the guidelines given in [34]. Each problem is run for 25 independent trials. For each trial the following procedure is followed:

(1) The function error value ($f(x) - f(x^*)$) after 5×10^3, 5×10^4, and 5×10^5 function evaluations (FES) is recorded respectively. Then the function error values for the 25 trials are compared and the best, median, mean, worst and the standard deviation (std) values are reported in Tables 18–21. The numbers in the parenthesis after the error values indicate the number of violated constraints at the corresponding values.

Table 18
Error values achieved when FES = 5×10^3, FES = 5×10^4, FES = 5×10^5 for problems g01–g06

FES		g01	g02	g03	g04	g05	g06
5×10^3	Best	1.4033E+01(1)	4.4088E−01(0)	6.8534E−01(0)	8.7597E+01(0)	3.8827E+02(3)	4.3519E+00(0)
	Median	8.4731E+00(5)	5.1743E−01(0)	8.9436E−01(1)	1.5243E+02(0)	−1.9626E+02(3)	5.1345E+01(0)
	Worst	2.3628E−01(7)	5.6740E−01(0)	9.9180E−01(1)	2.6092E+02(0)	7.7389E+00(3)	1.9741E+02(0)
	$c(v)$	2,5,5 (2.8E−01)	0,0,0 (0.0E+00)	0,0,1 (4.6E−05)	0,0,0 (0.0E+00)	1,3,3 (8.3E+00)	0,0,0 (0.0E+00)
	Mean	6.6623E+00	5.1338E−01	9.1985E−01	1.61E+02	2.38E+02	6.27E+01
	Std.	3.2816E+00	2.5982E−02	9.7029E−02	5.28E+01	3.05E+02	4.28E+01
5×10^4	Best	2.5336E+00(0)	9.6993E−04(0)	8.5068E−03(0)	5.2081E−06(0)	1.8745E−07(0)	4.5475E−11(0)
	Median	4.3261E+00(0)	2.9059E−02(0)	2.4476E−02(0)	2.3845E−05(0)	4.3724E−06(0)	4.5475E−11(0)
	Worst	9.9493E+00(0)	1.4734E−01(0)	7.1999E−02(0)	3.7240E−04(0)	5.8856E−05(0)	4.5475E−11(0)
	$c(v)$	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)
	Mean	4.8693E+00	4.7962E−02	3.1317E−02	4.31E−05	9.86E−06	4.55E−11
	Std.	2.1115E+00	4.4334E−02	1.6678E−02	7.21E−05	1.27E−05	6.60E−27
5×10^5	Best	0.0000E+00(0)	2.5857E−08(0)	−1.0003E−11(0)	7.2760E−11(0)	−1.8190E−12(0)	4.5475E−11(0)
	Median	0.0000E+00(0)	1.1011E−02(0)	−5.0862E−12(0)	7.6398E−11(0)	−1.8190E−12(0)	4.5475E−11(0)
	Worst	0.0000E+00(0)	3.7412E−02(0)	2.7170E−10(0)	7.6398E−11(0)	−1.8190E−12(0)	4.5475E−11(0)
	$c(v)$	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)
	Mean	0.0000E+00	1.2572E−02	2.6777E−11	7.48E−11	−1.82E−12	4.55E−11
	Std.	0.0000E+00	1.1835E−02	7.0167E−11	1.84E−12	4.12E−28	6.60E−27

Table 19
Error values achieved when FES = 5×10^3, FES = 5×10^4, FES = 5×10^5 for problems g07–g012

FES		g07	g08	g09	g10	g11	g12
5×10^3	Best	4.3508E+01(0)	8.7263E−11(0)	6.9037E+00(0)	9.1037E+03(2)	2.0304E−03(0)	2.7187E−08(0)
	Median	7.5209E+01(0)	2.6181E−10(0)	1.3531E+01(0)	1.1182E+04(3)	9.1974E−02(0)	5.4009E−07(0)
	Worst	2.0513E+02(0)	3.4932E−09(0)	2.8992E+01(0)	2.2637E+03(2)	2.4799E−01(1)	9.5265E−06(0)
	$c(v)$	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,3,3 (2.4E−02)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)
	Mean	8.20E+01	5.57E−10	1.45E+01	8.33E+03	8.52E−02	1.32E−06
	Std.	3.47E+01	7.35E−10	5.18E+00	3.80E+03	8.14E−02	2.19E−06
5×10^4	Best	2.0025E−02(0)	8.1964E−11(0)	7.4143E−07(0)	5.6124E+01(0)	0.0000E+00(0)	0.0000E+00(0)
	Median	4.9032E−02(0)	8.1964E−11(0)	2.2851E−06(0)	1.2492E+02(0)	0.0000E+00(0)	0.0000E+00(0)
	Worst	1.1336E−01(0)	8.1964E−11(0)	8.0683E−06(0)	6.8439E+02(0)	0.0000E+00(0)	0.0000E+00(0)
	$c(v)$	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)
	Mean	5.23E−02	8.20E−11	2.73E−06	1.57E+02	0.00E+00	0.00E+00
	Std.	2.06E−02	1.32E−26	1.91E−06	1.24E+02	0.00E+00	0.00E+00
5×10^5	Best	9.7472E−11(0)	8.1964E−11(0)	−9.8225E−11(0)	6.2755E−11(0)	0.0000E+00(0)	0.0000E+00(0)
	Median	7.4772E−10(0)	8.1964E−11(0)	−9.8225E−11(0)	4.2866E−06(0)	0.0000E+00(0)	0.0000E+00(0)
	Worst	5.0180E−08(0)	8.1964E−11(0)	−9.8225E−11(0)	2.4593E−04(0)	0.0000E+00(0)	0.0000E+00(0)
	$c(v)$	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)
	Mean	4.25E−09	8.20E−11	−9.82E−11	3.16E−05	0.00E+00	0.00E+00
	Std.	1.01E−08	1.32E−26	2.64E−26	5.56E−05	0.00E+00	0.00E+00

(2) As for $c(v)$ in Tables 18–21, c denotes the number of violated constraints (including the number of violations more than 1, 0.01, and 0.0001, respectively) at the median solution, and v denotes the mean violations at the median solution.
(3) In Table 22 the best, median, worst, mean, and the standard deviation of the number of FES to achieve a fixed accuracy level ($f(x) - f(x^*) \leqslant 0.0001$) are reported. Meanwhile, the Feasible Rate (rate of runs where at least one feasible solution is obtained), the Success Rate (rate of runs where the required accuracy is satisfied) and the Success Performance (the mean FES of the success runs divided by the Success Rate) are also shown. In addition, "NA" in Table 22 means no available value.

From Tables 18–21, it can be seen that our algorithm DSS-MDE is able to find feasible solutions for all the 22 benchmark functions in all the 25 runs. For problems g03, g05, g09 and g23 the best solutions found are

Table 20
Error values achieved when FES = 5×10^3, FES = 5×10^4, FES = 5×10^5 for problems g13–g18

FES		g13	g14	g15	g16	g17	g18
5×10^3	Best	7.5178E−01(3)	−1.4137E+02(3)	3.4958E+00(2)	6.4448E−02(0)	6.9202E+01(4)	1.4480E+00(9)
	Median	4.0885E−01(3)	−2.7234E+02(3)	2.0490E+00(2)	1.3362E−01(0)	1.0406E+02(4)	−2.2616E+00(9)
	Worst	7.9681E−01(3)	−3.8273E+02(3)	8.6793E+00(2)	2.5198E−01(0)	−2.0431E+01(4)	2.0748E+00(11)
	$c(v)$	0,3,3 (1.1E−01)	3,3,3 (2.7E+00)	0,2,2 (2.4E−01)	0,0,0 (0.0E+00)	4,4,4 (3.4E+00)	7,9,9 (1.7E+00)
	Mean	5.90E−01	−2.97E+02	3.68E+00	1.33E−01	8.08E+01	1.88E−01
	Std.	2.71E−01	4.85E+01	2.84E+00	5.28E−02	4.23E+01	1.83E+00
5×10^4	Best	4.5568E−07(0)	−1.2404E+02(3)	6.3324E−11(0)	1.7131E−02(0)	3.1506E−04(0)	1.4870E−02(0)
	Median	2.3113E−06(0)	−1.4874E+02(3)	2.1873E−10(0)	1.7131E−02(0)	1.3167E−02(0)	6.0042E−02(0)
	Worst	3.8415E−05(2)	−1.7675E+02(3)	3.6879E−08(0)	1.7234E−02(0)	1.0051E+02(2)	1.5574E−01(0)
	$c(v)$	0,0,0 (0.0E+00)	3,3,3 (1.4E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)
	Mean	1.66E−05	−1.44E+02	2.88E−09	1.71E−02	4.08E+00	6.52E−02
	Std.	5.08E−05	2.24E+01	7.72E−09	2.10E−05	2.01E+01	2.93E−02
5×10^5	Best	4.1898E−11(0)	2.4527E−10(0)	6.0822E−11(0)	1.7131E−02(0)	8.1855E−11(0)	1.5561E−11(0)
	Median	4.1898E−11(0)	3.5451E−08(0)	6.0822E−11(0)	1.7131E−02(0)	8.1855E−11(0)	1.5561E−11(0)
	Worst	4.1898E−11(0)	1.6888E+00(0)	6.0822E−11(0)	1.7233E−02(0)	8.1855E−11(0)	1.5561E−11(0)
	$c(v)$	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)
	Mean	4.19E−11	7.47E−02	6.08E−11	1.71E−02	8.19E−11	1.56E−11
	Std.	6.60E−27	3.37E−01	1.32E−26	2.08E−05	1.32E−26	0.00E+00

Table 21
Error values achieved when FES = 5×10^3, FES = 5×10^4, FES = 5×10^5 for problems g19–g24

FES		g19	g21	g23	g24
5×10^3	Best	1.8186E+02(0)	7.4178E+02(5)	−1.0815E+01(5)	2.3193E−04(0)
	Median	2.7081E+02(0)	1.8515E+02(5)	1.0887E+02(5)	1.0106E−03(0)
	Worst	3.7029E+02(0)	5.3677E+02(5)	−3.0121E+02(5)	4.0713E−03(0)
	$c(v)$	0,0,0 (0.0E+00)	2,5,5 (1.1E+00)	2,5,5 (5.0E−01)	0,0,0 (0.0E+00)
	Mean	2.77E+02	3.15E+02	−3.45E+02	1.09E−03
	Std.	4.87E+01	1.89E+02	3.98E+02	8.21E−04
5×10^4	Best	2.3009E+00(0)	2.6737E+02(5)	3.9029E+02(4)	4.6372E−12(0)
	Median	3.7603E+00(0)	7.6699E+01(5)	−5.9191E+02(5)	4.6372E−12(0)
	Worst	5.2783E+00(0)	−2.9455E+01(5)	−7.2933E+02(5)	4.6425E−12(0)
	$c(v)$	0,0,0 (0.0E+00)	0,3,5 (5.9E−02)	0,5,5 (1.6E−01)	0,0,0 (0.0E+00)
	Mean	3.78E+00	7.70E+01	−5.02E+02	4.64E−12
	Std.	8.87E−01	1.87E+02	3.30E+02	1.47E−15
5×10^5	Best	5.9860E−07(0)	1.1036E−06(0)	−1.7053E−13(0)	4.6372E−12(0)
	Median	3.8665E−05(0)	1.2601E−05(0)	1.9278E+01(0)	4.6372E−12(0)
	Worst	4.2340E−02(0)	1.1236E+02(0)	3.0236E+02(0)	4.6425E−12(0)
	$c(v)$	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)	0,0,0 (0.0E+00)
	Mean	3.46E−03	1.13E+01	3.84E+01	4.64E−12
	Std.	1.00E−02	2.91E+01	6.93E+01	1.47E−15

even better than the optimal values reported in [34]. For problems g01, g03, g04, g05, g06, g07, g08, g09, g11, g12, g13, g15, g17, g18 and g24, the best solutions obtained by our algorithm DSS-MDE are very approximate to or equal to the optimal values reported in [34] in all the 25 runs. And DSS-MDE only fails to locate the optimal value for problem g16. Therefore, our algorithm DSS-MDE is very effective to find feasible optimal solutions.

The success rate in Table 22 shows that our approach DSS-MDE is able to achieve 100% success for finding the required accuracy level for 15 of the 22 problems (g01, g03, g04, g05, g06, g07, g08, g09, g11, g12, g13, g15, g17, g18 and g24). The success rate greater than 50% is achieved on 4 test problems (g10, g14, g19 and g21). The success rate less than 50% is only on 3 problems (g02, g16 and g23). Nevertheless, our approach DSS-MDE is able to achieve 100% feasible rate for all the 22 test functions. This is because our approach DSS-MDE will focus on finding feasible solutions in the late stage of evolution.

Table 22
Number of FES to achieve the fixed accuracy level ($f(x) - f(x^*) \leqslant 0.0001$), success rate, feasible rate and success performance

Functions	Best	Median	Worst	Mean	Std.	Feasible rate (%)	Success rate (%)	Success performance
g01	111,034	118,771	130,458	118,919	4682.5511	100	100	118,919
g02	71,404	88,984	109,107	90,261	12,138.6824	100	36	250,724
g03	83,476	101,847	121,467	103,585	8743.9548	100	100	103,585
g04	43,348	46,981	53,444	46,852	2065.6766	100	100	46,852
g05	40,904	45,390	48,832	45,550	1857.4193	100	100	45,550
g06	13,834	15,529	19,076	15,776	1184.3736	100	100	15,776
g07	99,423	113,009	129,200	113,672	7070.3948	100	100	113,672
g08	855	2470	3101	2313	536.5258	100	100	2313
g09	34,995	37,605	42,496	38,226	2208.7409	100	100	38,226
g10	164,119	204,100	498,533	244,002	95,451.22	100	92	265,220
g11	9160	19,593	30,615	19,343	5400.5371	100	100	19,343
g12	808	3244	4061	2968	839.073	100	100	2968
g13	34,088	44,206	53,073	44,243	4100.4946	100	100	44,243
g14	211,862	275,862	479,584	313,390	85,671.1356	100	84	373,084
g15	28,204	34,722	39,992	34,580	2637.499	100	100	34,580
g16	NA	NA	NA	NA	NA	100	0	NA
g17	53,453	58,214	68,984	58,887	3422.1876	100	100	58,887
g18	100,454	111,947	138,025	113,959	8603.644	100	100	113,959
g19	258,825	382,792	488,370	380,668	81,728.4825	100	68	559,806
g21	119,496	133,016	214,865	146,211	28,317.6681	100	72	203,070
g23	292,693	322,553	401,084	334,721	49,010.0431	100	16	2,092,005
g24	5116	6632	8744	6679	778.4924	100	100	6679

5. Conclusion

The effect of the promising infeasible solutions in different stages of evolution is discussed and analyzed firstly in this paper. Then a novel dynamic stochastic selection method is proposed by the stochastic ranking within the framework of multimember differential evolution. We have implemented the algorithm DSS-MDE with a linear dynamic adjusting and compared it with the two current state-of-the-art and three competitive DE approaches. The experimental results on the common 13 benchmark functions have shown the effectiveness of our approach DSS-MDE.

We have also investigated different dynamic adjusting settings for comparison probability by analyses and experiments. It can be seen that although the convergence is speed up by the dynamic setting of the square root Eq. (11), the probability of striking into local optimum will be enlarged on some heavily constrained problems such as g13, which is consistent with the No Free Lunch theory [15]. Nevertheless, also from the experimental results and comparisons on four well-studied engineering design examples, our algorithm DSS-MDE with linear adjusting is an effective alternative approach for constrained optimization. In addition, when the number of maximum function evaluations is sufficient, i.e., a relatively larger maximum generation is used in evolution, our algorithm DSS-MDE with square root adjusting is also effective according to the experimental results on the 22 benchmark functions in CEC'06. Anyway, how much attention should be paid to the promising infeasible solution becomes the fundamental problem for constrained optimization. How to implement our Dynamic Stochastic Selection from dynamic adjusting to self-adaptive adjusting for constraint handling is our future works, in order to determine the attention exactly required during the evolution.

Acknowledgment

This work is partially supported by the National Natural Science Foundations of China (Grant No. 60428202), China.

Appendix

(1) g01

Minimize $f(x) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$

subject to

$$
\begin{aligned}
g_1(x) &= 2x_1 + 2x_2 + x_{10} + x_{11} - 10 \leqslant 0,\\
g_2(x) &= 2x_1 + 2x_3 + x_{10} + x_{12} - 10 \leqslant 0,\\
g_3(x) &= 2x_2 + 2x_3 + x_{11} + x_{12} - 10 \leqslant 0,\\
g_4(x) &= -8x_1 + x_{10} \leqslant 0,\\
g_5(x) &= -8x_2 + x_{11} \leqslant 0,\\
g_6(x) &= -8x_3 + x_{12} \leqslant 0,\\
g_7(x) &= -2x_4 - x_5 + x_{10} \leqslant 0,\\
g_8(x) &= -2x_6 - x_7 + x_{11} \leqslant 0,\\
g_9(x) &= -2x_8 - x_9 + x_{12} \leqslant 0,
\end{aligned}
$$

where the bounds are $0 \leqslant x_i \leqslant 1$ $(1 \leqslant i \leqslant 9)$, $0 \leqslant x_i \leqslant 100$ $(10 \leqslant i \leqslant 12)$, and $0 \leqslant x_{13} \leqslant 1$. The global optimum is at $x^* = (1,1,1,1,1,1,1,1,1,3,3,3,1)$, where $f(x^*) = 15$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

(2) g02

Maximize $f(x) = \left| \dfrac{\sum_{i=1}^{n} \cos^4(x_i) - 2\prod_{i=1}^{n} \cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}} \right|$

subject to

$$
\begin{aligned}
g_1(x) &= 0.75 - \prod_{i=1}^{n} x_i \leqslant 0,\\
g_2(x) &= \sum_{i=1}^{n} x_i - 7.5n \leqslant 0,
\end{aligned}
$$

where $n = 20$ and $0 \leqslant x_i \leqslant 10$ $(i = 1, \ldots, n)$. The global maximum is unknown, and the best reported solution is [4] $f(x^*) = 0.803\,619$. Constraint g_1 is close to being active $(g_1 = -10^{-8})$.

(3) g03

Maximize $f(x) = \left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i$

subject to

$$
h_1(x) = \sum_{i=1}^{n} x_i^2 - 1 = 0,
$$

where $n = 10$ and $0 \leqslant x_i \leqslant 1$ $(i = 1, \ldots, n)$. The global maximum is at $x_i^* = 1/\sqrt{n}(i = 1, \ldots, n)$ where $f(x^*) = 1$.

(4) g04

Minimize $f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$

subject to

$$g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leqslant 0,$$
$$g_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leqslant 0,$$
$$g_3(x) = 80.51249 + 0.007131x_2x_5 + 0.0029955x_1x_2 + 0.002183x_3^2 - 110 \leqslant 0,$$
$$g_4(x) = -80.51249 - 0.007131x_2x_5 - 0.0029955x_1x_2 - 0.002183x_3^2 + 90 \leqslant 0,$$
$$g_5(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leqslant 0,$$
$$g_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leqslant 0,$$

where $78 \leqslant x_1 \leqslant 102$, $33 \leqslant x_2 \leqslant 45$, $27 \leqslant x_i \leqslant 45$ $(i = 3,4,5)$. The optimum solution is $x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$, where $f(x^*) = -30{,}665.539$. Constraints g_1 and g_6 are active.

(5) g05

Minimize $f(x) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$

subject to

$$g_1(x) = -x_4 + x_3 - 0.55 \leqslant 0,$$
$$g_2(x) = -x_3 + x_4 - 0.55 \leqslant 0,$$

where $0 \leqslant x_{1,2} \leqslant 1200$, $-0.55 \leqslant x_{3,4} \leqslant 0.55$. The best known solution is $x^* = (679.9453, 1026.067, 0.1188764, -0.3962336)$, where $f(x^*) = 5126.4981$.

(6) g06

Minimize $f(x) = (x_1 - 10)^3 + (x_2 - 20)^3$

subject to

$$g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leqslant 0,$$
$$g_2(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \leqslant 0,$$
$$h_3(x) = 1000\sin(-x_3 - 0.25) + 1000\sin(-x_4 - 0.25) + 894.8 - x_1 = 0,$$
$$h_4(x) = 1000\sin(x_3 - 0.25) + 1000\sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0,$$
$$h_5(x) = 1000\sin(x_4 - 0.25) + 1000\sin(x_4 - x_3 - 0.25) + 1294.8 = 0,$$

where $13 \leqslant x_1 \leqslant 100$, $0 \leqslant x_2 \leqslant 100$. The optimum solution is $x^* = (14.095, 0.84296)$, where $f(x^*) = -6961.81388$. Both constraints are active.

(7) g07

Minimize $f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$

subject to

$$g_1(x) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \leqslant 0,$$
$$g_2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \leqslant 0,$$
$$g_3(x) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leqslant 0,$$
$$g_4(x) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leqslant 0,$$
$$g_5(x) = 5x_1^2 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \leqslant 0,$$

$$g_6(x) = x_1^2 + 2(x_2-2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leqslant 0,$$
$$g_7(x) = 0.5(x_1-8)^2 + 2(x_2-2)^2 + 3x_5^2 - 6x_6 - 30 \leqslant 0,$$
$$g_8(x) = -3x_1 + 6x_2 + 12(x_9-8)^2 - 7x_{10} \leqslant 0,$$

where $-10 \leqslant x_i \leqslant 10$ $(i = 1,\ldots,10)$. The global optimum is $x^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644, 9.828726, 8.280092, 8.375927)$, where $f(x^*) = 24.3062091$. Constraints g_1, g_2, g_3, g_4, g_5 and g_6 are active.

(8) g08

$$\text{Maximize} \quad f(x) = \frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1+x_2)}$$

subject to

$$g_1(x) = x_1^2 - x_2 + 1 \leqslant 0,$$
$$g_2(x) = 1 - x_1 + (x_2-4)^2 \leqslant 0,$$

where $0 \leqslant x_{1,2} \leqslant 10$. The global optimum is $x^* = (1.2279713, 4.2453733)$, where $f(x^*) = 0.095{,}825$.

(9) g09

$$\text{Minimize} \quad f(x) = (x_1-10)^2 + 5(x_2-12)^2 + x_3^4 + 3(x_4-11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$$

subject to

$$g_1(x) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leqslant 0,$$
$$g_2(x) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leqslant 0,$$
$$g_3(x) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leqslant 0,$$
$$g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leqslant 0,$$

where $-10 \leqslant x_i \leqslant 10$ $(i = 1,\ldots,7)$. The global optimum is $x^* = (2.330499, 1.951372, -0.4775414, 4.365726, -0.6244870, 1.038131, 1.594227)$, where $f(x^*) = 680.6300573$. Two constraints g_1 and g_4 are active.

(10) g10

$$\text{Minimize} \quad f(x) = x_1 + x_2 + x_3$$

subject to

$$g_1(x) = -1 + 0.0025(x_4 + x_6) \leqslant 0,$$
$$g_2(x) = -1 + 0.0025(x_5 + x_7 - x_4) \leqslant 0,$$
$$g_3(x) = -1 + 0.01(x_8 - x_5) \leqslant 0,$$
$$g_4(x) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \leqslant 0,$$
$$g_5(x) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \leqslant 0,$$
$$g_6(x) = -x_3x_8 + 1250000 + x_3x_5 - 2500x_5 \leqslant 0,$$

where $100 \leqslant x_1 \leqslant 10{,}000$, $1000 \leqslant x_i \leqslant 10{,}000$, $10 \leqslant x_j \leqslant 10{,}000$ $(i = 2,3,\quad j = 4,\ldots,8)$. The global optimum is $x^* = (579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979)$, where $f(x^*) = 7049.3307$. Constraints g_1, g_2 and g_3 are active.

(11) g11

$$\begin{aligned}&\text{Minimize} \quad f(x) = x_1^2 + (x_2 - 1)^2\\&\text{subject to}\\&\qquad h(x) = x_2 - x_1^2 = 0,\end{aligned}$$

where $-1 \leqslant x_{1,2} \leqslant 1$. The optimum is $x^* = \left(\pm 1/\sqrt{2}, 1/2\right)$, where $f(x^*) = 0.75$.

(12) g12

$$\begin{aligned}&\text{Maximize} \quad f(x) = \frac{100 - (x_1 - 5)^2 - (x_2 - 5)^2 - (x_3 - 5)^2}{100}\\&\text{subject to}\\&\qquad g(x) = (x_1 - p)^2 + (x_2 - q)^2 + (x_3 - r)^2 - 0.0625 \leqslant 0,\end{aligned}$$

where $0 \leqslant x_i \leqslant 10$ $(i = 1,2,3)$ and p, q, $r = 1,2,\ldots,9$. The feasible region of the search space consists of 9^3 disjointed spheres. A point (x_1, x_2, x_3) is feasible if and only if there exist p, q, r such above inequality holds. The global optimum is located at $x^* = (5,5,5)$, where $f(x^*) = 1$.

(13) g13

$$\begin{aligned}&\text{Minimize} \quad f(x) = e^{x_1 x_2 x_3 x_4 x_5}\\&\text{subject to}\\&\qquad h_1(x) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0,\\&\qquad h_2(x) = x_2 x_3 - 5 x_4 x_5 = 0,\\&\qquad h_3(x) = x_1^3 + x_2^3 + 1 = 0,\end{aligned}$$

where $-2.3 \leqslant x_{1,2} \leqslant 2.3$, $-3.2 \leqslant x_{3,4,5} \leqslant 3.2$. The global optimum is $x^* = (-1.717143, 1.595709, 1.827247, -0.763641\ 3, -0.763645)$, where $f(x^*) = 0.0539498$.

(14) g14

$$\begin{aligned}&\text{Minimize} \quad f(x) = \sum_{i=1}^{10} x_i \left(c_i + \ln \frac{x_i}{\sum_{j=1}^{10} x_j} \right)\\&\text{subject to}\\&\qquad h_1(x) = x_1 + 2x_2 + 2x_3 + x_6 + x_{10} - 2 = 0,\\&\qquad h_2(x) = x_4 + 2x_5 + x_6 + x_7 - 1 = 0,\\&\qquad h_3(x) = x_3 + x_7 + x_8 + 2x_9 + x_{10} - 1 = 0,\end{aligned}$$

where $0 < x_i \leqslant 10$ $(i = 1,\ldots,10)$, and $c_1 = -6.089$, $c_2 = -17.164$, $c_3 = -34.054$, $c_4 = -5.914$, $c_5 = -24.721$, $c_6 = -14.986$, $c_7 = -24.1$, $c_8 = -10.708$, $c_9 = -26.662$, $c_{10} = -22.179$. The best known solution $x^* = (0.0406684113216282, 0.14772124049245, 0.783205732104114, 0.00141433931889084, 0.485293636780388, 0.000693183051556082, 0.0274052040687766, 0.0179509660214818, 0.0373268186859717, 0.0968844604336845)$ where $f(x^*) = -47.7648884594915$.

(15) g15

Minimize $f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$

subject to

$$h_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0,$$
$$h_2(x) = 8x_1 + 14x_2 + 7x_3 - 56 = 0,$$

where $0 \leqslant x_i \leqslant 10$ $(i = 1, 2, 3)$. The best known solution $x^* = (3.51212812611795133, 0.216987510429556135, 3.55217854929179921)$ where $f(x^*) = 961.715022289961$.

(16) g16

Minimize
$$f(x) = 0.000117y_{14} + 0.1365 + 0.00002358y_{13} + 0.000001502y_{16} + 0.0321y_{12} + 0.004324y_5 + 0.0001\frac{c_{15}}{c_{16}} + 37.48\frac{y_2}{c_{12}} - 0.0000005843y_{17}$$

subject to

$$g_1(x) = \frac{0.28}{0.72}y_5 - y_4 \leqslant 0, \qquad g_2(x) = x_3 - 1.5x_2 \leqslant 0,$$
$$g_3(x) = 3496\frac{y_2}{c_{12}} - 21 \leqslant 0, \qquad g_4(x) = 110.6 + y_1 - \frac{62212}{c_{17}} \leqslant 0,$$
$$g_5(x) = 213.1 - y_1 \leqslant 0, \qquad g_6(x) = y_1 - 405.23 \leqslant 0,$$
$$g_7(x) = 17.505 - y_2 \leqslant 0, \qquad g_8(x) = y_2 - 1053.667 \leqslant 0,$$
$$g_9(x) = 11.275 - y_3 \leqslant 0, \qquad g_{10}(x) = y_3 - 35.03 \leqslant 0,$$
$$g_{11}(x) = 214.228 - y_4 \leqslant 0, \qquad g_{12}(x) = y_4 - 665.585 \leqslant 0,$$
$$g_{13}(x) = 7.458 - y_5 \leqslant 0, \qquad g_{14}(x) = y_5 - 584.463 \leqslant 0,$$
$$g_{15}(x) = 0.961 - y_6 \leqslant 0, \qquad g_{16}(x) = y_6 - 265.916 \leqslant 0,$$
$$g_{17}(x) = 1.612 - y_7 \leqslant 0, \qquad g_{18}(x) = y_7 - 7.046 \leqslant 0,$$
$$g_{19}(x) = 0.146 - y_8 \leqslant 0, \qquad g_{20}(x) = y_8 - 0.222 \leqslant 0,$$
$$g_{21}(x) = 107.99 - y_9 \leqslant 0, \qquad g_{22}(x) = y_9 - 273.366 \leqslant 0,$$
$$g_{23}(x) = 922.693 - y_{10} \leqslant 0, \qquad g_{24}(x) = y_{10} - 1286.105 \leqslant 0,$$
$$g_{25}(x) = 926.832 - y_{11} \leqslant 0, \qquad g_{26}(x) = y_{11} - 1444.046 \leqslant 0,$$
$$g_{27}(x) = 18.766 - y_{12} \leqslant 0, \qquad g_{28}(x) = y_{12} - 537.141 \leqslant 0,$$
$$g_{29}(x) = 1072.163 - y_{13} \leqslant 0, \qquad g_{30}(x) = y_{13} - 3247.039 \leqslant 0,$$
$$g_{31}(x) = 8961.448 - y_{14} \leqslant 0, \qquad g_{32}(x) = y_{14} - 26844.086 \leqslant 0,$$
$$g_{33}(x) = 0.063 - y_{15} \leqslant 0, \qquad g_{34}(x) = y_{15} - 0.386 \leqslant 0,$$
$$g_{35}(x) = 71084.33 - y_{16} \leqslant 0, \qquad g_{36}(x) = -140000 + y_{16} \leqslant 0,$$
$$g_{37}(x) = 2802713 - y_{17} \leqslant 0, \qquad g_{38}(x) = y_{17} - 12146108 \leqslant 0,$$

where

$$y_1 = x_2 + x_3 + 41.6, \quad c_1 = 0.024x_4 - 4.62,$$
$$y_2 = \frac{12.5}{c_1} + 12, \quad c_2 = 0.0003535x_1^2 + 0.5311x_1 + 0.08705y_2x_1,$$
$$c_3 = 0.052x_1 + 78 + 0.08705y_2x_1, \quad y_3 = \frac{c_2}{c_3},$$
$$y_4 = 19y_3, \quad c_4 = 0.04782(x_1 - y_3) + \frac{0.1956(x_1 - y_3)^2}{x_2} + 0.6376y_4 + 1.594y_3,$$

$$c_5 = 100x_2, \quad c_6 = x_1 - y_3 - y_4,$$
$$c_7 = 0.950 - \frac{c_4}{c_5}, \quad y_5 = c_6 c_7,$$
$$y_6 = x_1 - y_5 - y_4 - y_3, \quad c_8 = (y_5 + y_4)0.995,$$
$$y_7 = \frac{c_8}{y_1}, \quad y_8 = \frac{c_8}{3798},$$
$$c_9 = y_7 - \frac{0.0663y_7}{y_8} - 0.3153, \quad y_9 = \frac{96.82}{c_9} + 0.321y_1,$$
$$y_{10} = 1.29y_5 + 1.258y_4 + 2.29y_3 + 1.71y_6, \quad y_{11} = 1.71y_1 - 0.452y_4 + 0.580y_3,$$
$$c_{10} = \frac{12.2}{752.3}, \quad c_{11} = (1.75y_2)(0.995x_1),$$
$$c_{12} = 0.995y_{10} + 1998, \quad y_{12} = c_{10}x_1 + \frac{c_{11}}{c_{12}},$$
$$y_{13} = c_{12} - 1.75y_2, \quad y_{14} = 3623 + 64.4x_2 + 58.4x_3 + \frac{146312}{y_9 + x_5},$$
$$c_{13} = 0.995y_{10} + 60.8x_2 + 48x_4 - 0.1121y_{14} - 5095, \quad y_5 = \frac{y_{13}}{c_{13}},$$
$$y_{16} = 148000 - 331000y_{15} + 40y_{13} - 61y_{15}y_{13}, \quad c_{14} = 2324y_{10} - 28740000y_2,$$
$$y_{17} = 14130000 - 1328y_{10} - 531y_{11} + \frac{c_{14}}{c_{12}}, \quad c_{15} = \frac{y_{13}}{y_{15}} - \frac{y_{13}}{0.52},$$
$$c_{16} = 1.104 - 0.72y_{15}, \quad c_{17} = y_9 + x_5$$

and where the bounds are $704.4148 \leqslant x_1 \leqslant 906.3855$, $68.6 \leqslant x_2 \leqslant 288.88$, $0 \leqslant x_3 \leqslant 134.75$, $193 \leqslant x_4 \leqslant 287.0966$ and $25 \leqslant x_5 \leqslant 84.1988$. The best known solution is at $x^* = (705.174537070090537, 68.5999999999999943, 102.89999\ 9999999991, 282.324931593660324, 37.5841164258054832)$ where $f(x^*) = -1.90515525853479$.

(17) g17

Minimize $f(x) = f_1(x_1) + f_2(x_2)$,

where

$$f_1(x_1) = \begin{cases} 30x_1 & 0 \leqslant x_1 < 300, \\ 31x_1 & 300 \leqslant x_1 < 400, \end{cases}$$
$$f_2(x_2) = \begin{cases} 28x_2 & 0 \leqslant x_2 < 100, \\ 29x_2 & 100 \leqslant x_2 < 200, \\ 30x_2 & 200 \leqslant x_2 < 100 \end{cases}$$

subject to

$$h_1(x) = -x_1 + 300 - \frac{x_3x_4}{131.078}\cos(1.48477 - x_6) + \frac{0.90798x_3^2}{131.078}\cos(1.47588),$$
$$h_2(x) = -x_2 - \frac{x_3x_4}{131.078}\cos(1.48477 + x_6) + \frac{0.90798x_3^2}{131.078}\cos(1.47588),$$
$$h_3(x) = -x_5 - \frac{x_3x_4}{131.078}\sin(1.48477 + x_6) + \frac{0.90798x_3^2}{131.078}\sin(1.47588),$$
$$h_4(x) = 200 - \frac{x_3x_4}{131.078}\sin(1.48477 - x_6) + \frac{0.90798x_3^2}{131.078}\sin(1.47588),$$

where $0 \leqslant x_1 \leqslant 400$, $0 \leqslant x_2 \leqslant 1000$, $340 \leqslant x_3 \leqslant 420$, $340 \leqslant x_4 \leqslant 420$, $-1000 \leqslant x_5 \leqslant 1000$ and $0 \leqslant x_6 \leqslant 0.5236$. The best known solution $x^* = (201.784467214523659, 99.9999999999999005, 383.071034852773266, 420, -10.9076584514292652, 0.0731482312084287128)$ where $f(x^*) = 8853.53967480648$.

(18) g18

$$\text{Minimize} \quad f(x) = -0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7)$$

subject to

$$\begin{aligned}
&g_1(x) = x_3^2 + x_4^2 - 1 \leqslant 0,\\
&g_2(x) = x_9^2 - 1 \leqslant 0,\\
&g_3(x) = x_5^2 + x_6^2 - 1 \leqslant 0,\\
&g_4(x) = x_1^2 + (x_2 - x_9)^2 - 1 \leqslant 0,\\
&g_5(x) = (x_1 - x_5)^2 + (x_2 - x_6)^2 - 1 \leqslant 0,\\
&g_6(x) = (x_1 - x_7)^2 + (x_2 - x_8)^2 - 1 \leqslant 0,\\
&g_7(x) = (x_3 - x_5)^2 + (x_4 - x_6)^2 - 1 \leqslant 0,\\
&g_8(x) = (x_3 - x_7)^2 + (x_4 - x_8)^2 - 1 \leqslant 0,\\
&g_9(x) = x_7^2 + (x_8 - x_9)^2 - 1 \leqslant 0,\\
&g_{10}(x) = x_2x_3 - x_1x_4 \leqslant 0,\\
&g_{11}(x) = -x_3x_9 \leqslant 0,\\
&g_{12}(x) = x_5x_9 \leqslant 0,\\
&g_{13}(x) = x_6x_7 - x_5x_8 \leqslant 0,
\end{aligned}$$

where $-10 \leqslant x_i \leqslant 10 \quad (i = 1, \ldots, 8)$ and $0 \leqslant x_9 \leqslant 20$. The best known solution is at $x^* = (-0.657776192427943163, -0.153418773482438542, 0.323413871675240938, -0.946257611651304398, -0.657776194376798906, -0.753213434632691414, 0.323413874123576972, -0.346462947962331735, 0.59979466285217542)$ where $f(x^*) = -0.866025403784439$.

(19) g19

$$\text{Minimize} \quad f(x) = \sum_{j=1}^{5}\sum_{i=1}^{5} c_{ij}x_{10+i}x_{10+j} + 2\sum_{j=1}^{5} d_j x_{10+j}^3 - \sum_{i=1}^{10} b_i x_i$$

subject to

$$g_j(x) = -2\sum_{i=1}^{5} c_{ij}x_{10+i} - 3d_j x_{10+j}^2 - e_j + \sum_{i=1}^{10} a_{ij}x_i \leqslant 0, \quad j = 1, \ldots, 5,$$

where $b = (-40, -2, -0.25, -4, -4, -1, -40, -60, 5, 1)$ and the remaining data is on Table A, and $0 \leqslant x_i \leqslant 10$ $(i = 1, \ldots, 15)$. The best known solution is at $x^* = (1.66991341326291344e-17, 3.95378229282456509e-16, 3.94599045143233784, 1.06036597479721211e-16, 3.2831773458454161, 9.99999999999999822, 1.12829414671605333e-17, 1.2026194599794709e-17, 2.50706276000769697e-15, 2.24624122987970677e-15, 0.3707648474170139 87, 0.278456024942955571, 0.523838487672241171, 0.388620152510322781, 0.298156764974678579)$ where $f(x^*) = 32.6555929502463$.

Table A
Data set for test problem g19

j	1	2	3	4	5
e_j	−15	−27	−36	−18	−12
c_{1j}	30	20	10	32	−10
c_{2j}	20	39	−6	−31	32
c_{3j}	−10	−6	10	−6	−10
c_{4j}	32	−31	−6	39	−20
c_{5j}	−10	32	−10	−20	30
d_j	4	8	10	6	2
a_{1j}	−16	2	0	1	0
a_{2j}	0	−2	0	0.4	2
a_{3j}	−3.5	0	2	0	0
a_{4j}	0	−2	0	−4	−1
a_{5j}	0	−9	−2	1	−2.8
a_{6j}	2	0	−4	0	0
a_{7j}	−1	−1	−1	−1	−1
a_{8j}	−1	−2	−3	−2	−1
a_{9j}	1	2	3	4	5
a_{10j}	1	1	1	1	1

(20) g21

Minimize $f(x) = x_1$
subject to

$$
\begin{aligned}
&g_1(x) = -x_1 + 35x_2^{0.6} + 35x_3^{0.6} \leqslant 0,\\
&h_1(x) = -300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 = 0,\\
&h_2(x) = 100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 25x_4x_7 - 15536.5 = 0,\\
&h_3(x) = -x_5 + \ln(-x_4 + 900) = 0,\\
&h_4(x) = -x_6 + \ln(x_4 + 300) = 0,\\
&h_5(x) = -x_7 + \ln(-2x_4 + 700) = 0,
\end{aligned}
$$

where $0 \leqslant x_1 \leqslant 1000$, $0 \leqslant x_2, x_3 \leqslant 40$, $100 \leqslant x_4 \leqslant 300$, $6.3 \leqslant x_5 \leqslant 6.7$, $5.9 \leqslant x_6 \leqslant 6.4$ and $4.5 \leqslant x_7 \leqslant 6.25$. The best known solution is at $x^* = (193.724510070034967, 5.56944131553368433\text{e}{-27}, 17.3191887294084914, 100.0478\ \ 97801386839, 6.68445185362377892, 5.99168428444264833, 6.21451648886070451)$ where $f(x^*) = 193.724510070035$.

(21) g23

Minimize $f(x) = -9x_5 - 15x_8 + 6x_1 + 16x_2 + 10(x_6 + x_7)$
subject to

$$
\begin{aligned}
&g_1(x) = x_9x_3 + 0.02x_6 - 0.025x_5 \leqslant 0,\\
&g_2(x) = x_9x_4 + 0.02x_7 - 0.015x_8 \leqslant 0,\\
&h_1(x) = x_1 + x_2 - x_3 - x_4 = 0,\\
&h_2(x) = 0.03x_1 + 0.01x_2 - x_9(x_3 + x_4) = 0,\\
&h_3(x) = x_3 + x_6 - x_5 = 0,\\
&h_4(x) = x_4 + x_7 - x_8 = 0,
\end{aligned}
$$

where $0 \leqslant x_1, x_2, x_6 \leqslant 300$, $0 \leqslant x_3, x_5, x_7 \leqslant 100$, $0 \leqslant x_4, x_8 \leqslant 200$ and $0.01 \leqslant x_9 \leqslant 0.03$. The best known solution is at $x^* = (0.00510000000000259465, 99.9947000000000514, 9.01920162996045897\text{e}{-18},$

99.9999000000000535, 0.000100000000027086086, 2.75700683389584542e−14, 99.9999999999999574, 2000.01-00000100000100008) where $f(x^*) = -400.055099999999584$.

(22) g24

$$\text{Minimize} \quad f(x) = -x_1 - x_2$$

subject to

$$g_1(x) = -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \leqslant 0,$$
$$g_2(x) = -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36 \leqslant 0,$$

where $0 \leqslant x_1 \leqslant 3$, $0 \leqslant x_2 \leqslant 4$. The feasible global minimum is at $x^* = (2.32952019747762, 3.17849307411774)$ where $f(x^*) = -5.50801327159536$.

(23) Weld beam design

$$\text{Minimize} \quad f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

subject to

$$\tau(x) - \tau_{\max} \leqslant 0 \sigma(x) - \sigma_{\max} \leqslant 0 x_1 - x_4 \leqslant 0,$$
$$\delta(x) - \delta_{\max} \leqslant 0 P - P_C(x) \leqslant 0,$$

The other parameters are defined as follows:

$$\tau(x) = \sqrt{(\tau')^2 + \frac{2\tau'\tau''x_2}{2R} + (\tau'')^2}, \quad \tau' = \frac{P}{\sqrt{2}x_1x_2}, \quad \tau'' = \frac{MR}{J},$$
$$M = P\left(L + \frac{x_2}{2}\right), \quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2},$$
$$J = 2\left\{\frac{x_1x_2}{\sqrt{2}}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_2}{2}\right)^2\right]\right\}, \quad \sigma(x) = \frac{6PL}{x_4x_3^2},$$
$$\delta(x) = \frac{4PL^3}{Ex_4x_3^3}, \quad P_C(x) = \frac{4.013\sqrt{EGx_3^2x_4^6/36}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right),$$

where $P = 6000$ lb., $L = 14$, $\delta_{\max} = 0.25$ in., $E = 30 \times 10^6$ psi, $G = 12 \times 10^6$ psi, $\sigma_{\max} = 30{,}000$ psi, $0.125 \leqslant x_1 \leqslant 10.0$, $0.1 \leqslant x_2 \leqslant 10.0$, $0.1 \leqslant x_3 \leqslant 10$, and $0.1 \leqslant x_4 \leqslant 10.0$.

(24) Spring design

$$\text{Minimize} \quad f(x) = (x_3 + 2)x_1x_2^2$$

subject to

$$1 - \frac{x_1^3x_3}{71785x_2^4} \leqslant 0, \quad \frac{4x_1^2 - x_1x_2}{12566(x_1x_2^3 - x_2^4)} + \frac{1}{5108x_2^2} - 1 \leqslant 0,$$
$$1 - \frac{140.45x_2}{x_1^2x_3} \leqslant 0, \quad \frac{x_1 + x_2}{1.5} - 1 \leqslant 0,$$

where $0.25 \leqslant x_1 \leqslant 1.3$, $0.05 \leqslant x_2 \leqslant 2.0$, and $2 \leqslant x_3 \leqslant 15$.

(25) Speed reducer design

$$\text{Minimize} \quad f(x) = (3.3333x_3^2 + 14.9334x_3 - 43.0934) \cdot 0.7854x_1x_2^2 - 1.508x_1(x_6^2 + x_7^2)$$
$$+ (x_6^3 + x_7^3) \cdot 7.4777 + 0.7854(x_4x_6^2 + x_5x_7^2)$$

subject to

$$\frac{27}{x_1x_2^2x_3}-1\leqslant 0,\quad \frac{397.5}{x_1x_2^2x_3^2}-1\leqslant 0,\quad \frac{1.93x_4^3}{x_2x_3x_6^4}-1\leqslant 0,$$

$$\frac{1.93x_5^3}{x_2x_3x_7^4}-1\leqslant 0,\quad \frac{[(745x_4/x_2x_3)^2+16.9\times 10^6]^{1/2}}{110.0x_6^3}-1\leqslant 0,$$

$$\frac{x_2x_3}{40}-1\leqslant 0,\quad \frac{[(745x_5/x_2x_3)^2+157.5\times 10^6]^{1/2}}{85.0x_7^3}-1\leqslant 0,$$

$$\frac{5x_2}{x_1}-1\leqslant 0,\quad \frac{x_1}{12x_2}-1\leqslant 0,$$

$$\frac{1.5x_6+1.9}{x_4}-1\leqslant 0,\quad \frac{1.1x_7+1.9}{x_5}-1\leqslant 0,$$

where $2.6\leqslant x_1\leqslant 3.6$, $0.7\leqslant x_2\leqslant 0.8$, $17\leqslant x_3\leqslant 28$, $7.3\leqslant x_4\leqslant 8.3$, $7.3\leqslant x_5\leqslant 8.3$, $2.9\leqslant x_6\leqslant 3.9$, $5.0\leqslant x_7\leqslant 5.5$.

(26) Three-bar truss design

Minimize $f(x)=\left(2\sqrt{2}x_1+x_2\right)\times l$

subject to

$$\frac{\sqrt{2}x_1+x_2}{\sqrt{2}x_1^2+2x_1x_2}P-\sigma\leqslant 0,\quad \frac{x_2}{\sqrt{2}x_1^2+2x_1x_2}P-\sigma\leqslant 0,$$

$$\frac{1}{x_1+\sqrt{2}x_2}P-\sigma\leqslant 0,$$

where $0\leqslant x_1\leqslant 1$ and $0\leqslant x_2\leqslant 1$; $l=100$ cm, $P=2$ KN/cm^2, and $\sigma=2$ KN/cm^2.

References

[1] C.A.C. Coello, D.A. Van Veldhuizen, G.B. Lamont, Evolutionary Algorithms for Solving Multi-Objective Problems, Kluwer, Norwell, MA, 2002.

[2] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer-Verlag, Berlin, Germany, 2003.

[3] C.A.C. Coello, 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) (2002) 1245–1287.

[4] T.P. Runarsson, X. Yao, Search biases in constrained evolutionary optimization, IEEE Trans. Systems Man Cybernetics – Part C: Appl. Rev. 35 (2) (2005) 233–243.

[5] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Eng. 186 (2–4) (2000) 311–338.

[6] T.P. Runarsson, X. Yao, Stochastic ranking for constrained evolutionary optimization, IEEE Trans. Evol. Comput. 4 (3) (2000) 284–294.

[7] M. Zhang, H. Geng, W. Luo, L. Huang, X. Wang, A novel search biases selection strategy for constrained evolutionary optimization, CEC (2006) 1845–1850.

[8] R. Storn, K. Price, Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341–359.

[9] E. Mezura-Montes, J. Velázquez-Reyes, C.A.Coello Coello, Promising infeasibility and multiple offspring incorporated to differential evolution for constrained optimization, GECCO (2005) 225–232.

[10] J. Lampinen, A constraint handling approach for the differential evolution algorithm, in: Proc. 2002 IEEE Congress on Evolutionary Computation, Honolulu, Hawaii, May 2002, pp. 1468–1473.

[11] R. Storn, System design by constraint adaptation and differential evolution, IEEE Trans. Evol. Comput. 3 (1) (1999) 22–34.

[12] Z. Michalewicz, M. Schoenauer, Evolutionary algorithms for constrained parameter optimization problems, Evol. Comput. 4 (1) (1996) 1–32.

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

[14] E. Mezura-Montes, C.A.C. Coello, E.I. Tun-Morales. Simple feasibility rules and differential evolution for constrained optimization. IMICAI'2004, LNAI 2972, pp. 707–716.

[15] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1) (1997) 67–82.
[16] Sangameswar Venkatraman, Gary G. Yen, A generic framework for constrained optimization using genetic algorithms, IEEE Trans. Evol. Comput. 9 (4) (2005) 424–435.
[17] Z. Cai, Y. Wang, A multiobjective optimization-based evolutionary algorithm for constrained optimization, IEEE Trans. Evol. Comput. 10 (6) (2006) 658–675.
[18] M.F. Tasgetiren, P.N. Suganthan, A multi-populated differential evolution algorithm for solving constrained optimization problem, CEC (2006) 33–40.
[19] S. Kukkonen, J. Lampinen, Constrained real-parameter optimization with generalized differential evolution, CEC (2006) 207–214.
[20] K. Zielinski, R. Laur, Constrained single-objective optimization using differential evolution, CEC (2006) 223–230.
[21] E. Mezura-Montes, J. Velázquez-Reyes, C.A.Coello Coello, Modified differential evolution for constrained optimization, CEC (2006) 25–32.
[22] T. Ray, K.M. Liew, Society and civilization: an optimization algorithm based on the simulation of social behavior, IEEE Trans. Evol. Comput. 7 (4) (2003) 386–396.
[23] R. Farmani, J.A. Wright, Self-adaptive fitness formulation for constrained optimization, IEEE Trans. Evol. Comput. 7 (5) (2003) 445–455.
[24] D. Powell, M. Skolnick, Using genetic algorithms in engineering design optimization with nonlinear constraints, in: Proc. 5th Int. Conf. Genetic Algorithms, 1989, pp. 424–431.
[25] A.R. Hedar, M. Fukushima, Derivative-free filter simulated annealing method for constrained continuous global optimization, J. Global Optim. 35 (4) (2006) 521–649.
[26] S. He, E. Prempain, O.H. Wu, An improved particle swarm optimizer for mechanical design optimization problems, Eng. Optim. 36 (5) (2004) 585–605.
[27] K. Deb, Optimal design of a welded beam via genetic algorithms, AIAA J. 29 (8) (1991) 2013–2015.
[28] C.A.C. Coello, Self-adaptive penalties for GA based optimization, in: Proc. Congress Evolutionary Computation (CEC 1999), vol. 1, 1999, pp. 573–580.
[29] T. Ray, P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, Eng. Optim. 33 (3) (2001) 735–748.
[30] E.M. Montes, C.A.C. Coello, J.V. Reyes, Increasing successful offspring and diversity in differential evolution for engineering design, in: Proceedings of the Seventh International Conference on Adaptive Computing in Design and Manufacture (ACDM 2006), April 2006, pp. 131–139.
[31] S. Akhtar, K. Tai, T. Ray, A socio-behavioural simulation model for engineering design optimization, Eng. Optim. 34 (4) (2002) 341–354.
[32] J.K. Kuang, S.S. Rao, L. Chen, Taguchi-aided search method for design optimization of engineering systems, Eng. Optim. 30 (1998) 1–23.
[33] S. Hernendez, Multiobjective structural optimization, in: S. Kodiyalam, M. Saxena (Eds.), Geometry and Optimization Techniques for Structural Design, Elsevier, Amsterdam, The Netherlands, 1994, pp. 341–362.
[34] J.J. Liang, T.P. Runarsson, E. Mezura-Montes, M. Clerc, P.N. Suganthan, C.A.C. Coello, K. Deb, Problem definitions and evaluation criteria for the CEC2006 special session on constrained real-parameter optimization, 2006. <http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC-06/CEC06.htm>.
[35] Frank Y. Shih, Yi-Ta Wu, Robust watermarking and compression for medical images based on genetic algorithms, Inform. Sci. 175 (3) (2005) 200–216.
[36] Helio J.C. Barbosa, Afonso C.C. Lemonge, A new adaptive penalty scheme for genetic algorithms, Inform. Sci. 156 (3–4) (2003) 215–251.
[37] R. Zhu, Statistical Analysis Methods, China Forestry Publishing House, Beijing, China, 1989 (in Chinese).