J Intell Manuf (2018) 29:1559–1580
https://doi.org/10.1007/s10845-016-1199-9

Engineering design optimization using an improved local search based epsilon differential evolution algorithm

Wenchao Yi[1] · **Yinzhi Zhou**[2] · **Liang Gao**[1] · **Xinyu Li**[1] · **Chunjiang Zhang**[1]

Received: 16 June 2015 / Accepted: 16 January 2016 / Published online: 29 January 2016

Abstract Many engineering problems can be categorized into constrained optimization problems (COPs). The engineering design optimization problem is very important in engineering industries. Because of the complexities of mathematical models, it is difficult to find a perfect method to solve all the COPs very well. ε constrained differential evolution (εDE) algorithm is an effective method in dealing with the COPs. However, εDE still cannot obtain more precise solutions. The interaction between feasible and infeasible individuals can be enhanced, and the feasible individuals can lead the population finding optimum around it. Hence, in this paper we propose a new algorithm based on ε feasible individuals driven local search called as ε constrained differential evolution algorithm with a novel local search operator (εDE-LS). The effectiveness of the proposed εDE-LS algorithm is tested. Furthermore, four real-world engineering design problems and a case study have been studied. Experimental results show that the proposed algorithm is a very effective method for the presented engineering design optimization problems.

Keywords Constrained optimization problems · Constraint handling technique · ε Constrained differential evolution · Local search operator · Engineering design optimization

✉ Xinyu Li
lixinyu@hust.edu.cn; lixinyu@mail.hust.edu.cn

Wenchao Yi
janety1989@gmail.com

Yinzhi Zhou
leitezhou@gmail.com

Liang Gao
gaoliang@mail.hust.edu.cn

Chunjiang Zhang
zhchj1989@gmail.com

1 The State Key Laboratory of Digital Manufacturing Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, Hubei, China

2 The School of Mechanical and Aerospace Engineering, Nanyang Technological University, Singapore 639798, Singapore

Introduction

Differential evolution (DE) algorithm is one of the most efficient evolutionary algorithms, which was firstly proposed by Storn and Price (1997). During the past decade, numerous competitive DE-based algorithms were presented to solve the constrained optimization problems (COPs). Most optimization problems in real world which are subjected to constraints can be categorized into COPs, such as scheduling (Artigues and Lopez 2014; Naber and Kolisch 2014; Brajevic and Tuba 2013), engineering design optimization (Kanagaraj et al. 2014; Flager et al. 2014), optimal control of systems (Ellis and Christofides 2014) and etc. As a significant portion of engineering design optimization problems is under the category of COP, this paper focuses on solving COPs using evolutionary algorithm. A general COPs can be stated as follows:

$$\begin{aligned} &\min f(\vec{x}) \\ &s.t.\ g_j(\vec{x}) \le 0, \quad j = 1, \ldots, q \\ &\quad\ h_j(\vec{x}) = 0, \quad j = q+1, \ldots, m \end{aligned} \tag{1}$$

where $\vec{x} = (x_1, \ldots, x_n)$ is generated within the range $L_i < x_i < U_i$. L_i and U_i denote the lower and upper bound in each dimension. $g_j(\vec{x})$ denotes the jth inequality constraint and $\boldsymbol{h}_j(\vec{x})$ denotes the $(j-q)$th equality constraint. Solving COPs is difficult especially when the feasible region is small and there are numerous constraint conditions. So the researches based on COPs will never end.

Many methods have been proposed for COPs. Generally, they can be divided into three main categories (Mezura-Montes and Coello 2011), namely, the method based on transforming the COPs into unconstrained optimization problems, the method based on multi-objective techniques, and the method based on adding extra rules or operator.

(1) The method based on transforming COPs into unconstrained optimization problems.
The representative of this kind of method is penalty function method. Penalty function method is a simple but effective method in dealing with COPs. Although this method is simple to implement, it is difficult to find a good balance between objective and penalty functions. Since the penalty function method proposed, many researchers have proposed several improved versions. Huang et al. (2007) proposed a co-evolutionary DE algorithm, in which a special adaptive penalty function was proposed to deal with the constraints. Although dynamic penalty factor setting method (Puzzi and Carpinteri 2008; Montemurro et al. 2013), adaptive penalty function method (Tessema and Yen 2009) have been proposed, it is still difficult to set the best penalty function factor for all the objectives.
(2) The method based on multi-objective technique.
Wang and Cai (2012) introduced a multi-objective technique with DE algorithm to solve COPs. An infeasible solution replacement mechanism based on multi-objective approach is proposed. The infeasible solution replacement mechanism is a Pareto-dominance-like method to compare the objective and constraints violations between the solutions. The method mainly focuses on guiding the population moving towards to the promising and feasible region more efficiently. Gong and Cai (2008) proposed a multi-objective technique based DE for COPs, in which multi-objective technique based constraint handling technique was proposed. Though the above methods are highly effective in solving COPs, however, it is still difficult to design an effective framework for using multi-objective techniques to tackle COPs.
(3) The method by adding extra rules or operator.
Storn (1999) proposed a constraint adaptive method, which firstly makes all the individuals as feasible ones by relaxing the constraints then decreases the relaxation till reaching the original constraints. Deb's rule (Deb 2000) is an effective method in dealing with COPs, but it may be over-penalization, since the infeasible solutions are always better than the feasible ones. Domínguez-Isidro et al. (2013) proposed the memetic DE algorithm, which implements the mathematical programming method named Powell's conjugate direction as a local search operator. Although the method is effective, the time-consuming of the mathematical programming may be huge. Among these researches, εDE method proposed by Takahama and Sakai (2006) is a very effective one, in which ε constraint handling technique is used to deal with the constraints. The experimental results showed that the εDE not only could find the feasible solutions rapidly, but also could achieve excellent successful performance.

The εDE algorithm, as a promising representative of the last category, is an effective method in dealing with COPs and many researchers have made improvement on it. Takahama and Sakai (2010a) proposed the improved εDE with an archive and gradient-based mutation, In this method, a local search based on the information of first-order derivative was proposed. However, it is usually difficult to calculate the first-order derivative. Also, the calculation of first-order derivative is time-cost. Based on this, Takahama and Sakai (2013) proposed εDE with rough approximation using kernel regression. However, this method still suffers from the huge time-cost of the approximation process. Rather than improving the constraint handling technique in εDE, another direction to improve the performance of the εDE is to improve the DE algorithm. Takahama and Sakai (2010b) proposed an adaptive DE based εDE algorithm and then the rank-based DE based εDE (Takahama and Sakai 2012). Comparing with the constraint handling technique in εDE, the algorithm engine has a limited influence on the performance of εDE. Hence the motivation of our research is focusing on designing a novel mutation operator that drives the population toward epsilon feasible region.

A variety of DE variants for COPs have been proposed in past years. Mutation operator, as an important component of DE algorithm, has great influence on the efficiency in solving the COPs. The motivation of our method is mainly utilizing the information of the feasible and the infeasible individuals, which focuses on guiding the infeasible individuals to move along the directions of the feasible individuals. In this way can we lead the infeasible individuals moving into the feasible region and then find the optimum effectively. The preliminary idea had been published in Yi et al. (2015). We propose a novel mutation operator that is specially designed for COPs, which serve as the local search engine for the εDE algorithm. Wang et al. (2013) proposed predatory search strategy based on particle swarm optimization (PSO-PSS). In the method, they use the Euclidean distance from each individual to the best individual to define the neighborhood of the search. When the search begins, it starts with the minimum distance neighborhood, if no better solution finds, then search with the second to the minimum distance neighborhood. Once the better solution found, the Euclidean distance should be updated. The proposed algorithm differs from the PSO-PSS algorithm from the following aspect: First, we do not use the Euclidean distance based neighborhood. Instead, the "DE/current-to-feasible/2"operator is proposed, which randomly search the area around the current individual with the multi-direction combined that pointed to the possible less constraint violation area. Second, the proposed algorithm does not need to updated the Euclidean distance once a better solution is find. Instead, when all individuals are evaluated after one iteration ends, we can get the updated feasible individual set without extra computation.

In order to evaluate the performance of the proposed algorithm, 24 famous benchmark test functions collected from the special session on the constrained real-parameter optimization of the 2006 IEEE Congress on Evolutionary Computation (IEEE CEC2006). Four real-world engineering design optimization problems and a case study on car side impact design are adopted in this article and the comparisons among proposed method and state-of-the-art algorithms are also conducted. Experimental results show that the proposed algorithm can achieve good solutions on these engineering design problems.

This article is organized as follows. We firstly give a general introduction of DE algorithm and εDE in "DE and εDE algorithm" section. In "The proposed εDE-LS algorithm" section, the proposed εDE-LS algorithm is introduced in detail, which contains the framework. The experimental results and the comparisons are presented in "Experimental results" section. "Case study: car side impact design" section presents a case study in engineering design optimization. Finally, conclusions and future work are given in "Conclusion and future work" section.

DE and εDE algorithm

DE algorithm

DE algorithm is an efficient but simple evolutionary algorithm (EA), which can be divided into four phases, which are the initialization, mutation, crossover and selection. During the initialization phase, the *NP* (number of population) *n*-dimensional individuals $x_i^g = (x_{i,1}^g, \ldots, x_{i,n}^g)$, $i = 1, \ldots, NP$ are generated. g denotes the generation number. Then the mutation phase is adopted to generate the mutation vectors. Several mutation operators have been proposed in past years. The DE/rand/1/*exp* is the commonly used one, where *exp* denotes the exponential crossover operator. The mutation vector can be calculated as follows:

$$v_i^g = x_{r1}^g + F * \left(x_{r2}^g - x_{r3}^g\right) \tag{2}$$

where F denotes the predefined scale parameter, r_1, r_2, and r_3 are three mutually different generated indexes which should be different from index i within the range [1, *NP*]. Then a check will be made to ensure all the elements in the generated v_i^g are within the boundaries, which can be described as follows:

$$v_{i,j}^g = \begin{cases} \min\left\{U_j, 2 * L_j - v_{i,j}^g\right\}, & \text{if } v_{i,j}^g < L_j \\ \max\left\{L_j, 2 * U_j - v_{i,j}^g\right\}, & \text{if } v_{i,j}^g > U_j \end{cases} \tag{3}$$

where L_j and U_j denote the lower and upper bound in jth dimension.

The exponential crossover operator makes the trail vector contains a consecutive sequence of the component taken from the mutation vector. The exponential crossover operator can be given as follow:

$$u_{i,j}^g = \begin{cases} v_{i,j}^g, & if\ j \in \left\{k, \langle k+1\rangle_n, \ldots, \langle k+L-1\rangle_n\right\} \\ x_{i,j}^g, & otherwise \end{cases}, \quad j = 1, \ldots, \mathrm{n} \tag{4}$$

where $L, k \in [1, \mathrm{n}]$ are both random indexes. $\langle j\rangle$ is j if $j < n$ and $j = j - n$ if $j > n$.

During the selection phase, a better individual between the trail vector u_i^g and target vector x_i^g will be chosen according to their objective function value:

$$x_i^{g+1} = \begin{cases} u_i^g, & if\ f\left(u_i^g\right) < f\left(x_i^g\right) \\ x_i^g, & else \end{cases} \tag{5}$$

εDE algorithm

In the εDE algorithm, the constraint violation $\Phi(\mathrm{x}_i^g)$ is defined as the sum of all constraints:

$$\Phi\left(\mathrm{x}_i^g\right) = \sum_{j=1}^{q} \max\left\{0, g_j\left(\mathrm{x}_i^g\right)\right\} + \sum_{j=q+1}^{m} h_j\left(\mathrm{x}_i^g\right) \tag{6}$$

After generating the new target vector through the DE algorithm. The ε level comparison is used in εDE algorithm to help deciding which individual is better. The comparison can be given as follows:

$$(f_1, \Phi_1) <_\varepsilon (f_2, \Phi_2) \Leftrightarrow \begin{cases} f_1 < f_2, & if\ \Phi_1, \Phi_2 \le \varepsilon \\ f_1 < f_2, & if\ \Phi_1 = \Phi_2 \\ \Phi_1 < \Phi_2, & otherwise \end{cases} \tag{7}$$

where f_1 and f_2 are the objective fitness functions values, and the ε level is dynamically decreased along the generation number increases until it reaches to zero. We can see that for comparison between individuals whose constraint violation is small than ε, we treat them as generalized feasible individuals by just comparing their objective fitness functions values and we can note them by ε-feasible individuals for short. The value of the ε is set as formula given below:

$$\varepsilon(g) = \begin{cases} \Phi\left(x_\theta\right), & g = 0 \\ \varepsilon(0) * (1 - g/T_c)^{cp}, & 0 < g < T_c \\ 0, & g \geq T_c \end{cases} \tag{8}$$

where x_θ is the top θth individual in the initialization population and often set $\theta = 0.2 * N$ in the article. T_c is a predefined generation number. cp is the control parameter in ε level comparison and is set as 5 in the article.

From the experimental results obtained by Takahama and Sakai (2006), we can concluded the mechanism in εDE algorithm that expanding the feasible region at first and then narrowing the region into the original one along the evolutionary process is highly effective. The core of the mechanism lets the less constraint-violated individuals guide the evolutionary directions. However, the feasible ones are important to achieve the optimum and hence we can take advantage of the guidance of feasible individuals and make improvement on it.

The proposed εDE-LS algorithm

Usually, in the COPs, the surrounding region of feasible individuals could have higher chance to be feasible. So the feasible individuals can guide the infeasible ones moving towards to the feasible region. "DE/rand/1" mutation operator is the most commonly used operator, in which three vectors are mutually different from each other. So the "DE/rand/1" operator shows no bias to any search directions because the direction is randomly chosen. "DE/rand/2" operator adds more perturbation than "DE/rand/1" by adding one more difference vector. "DE/current-to-rand/1"operator starts from the current individual, it can avoid individual not being selected as starting point. "DE/current-to-rand/1" can ensure each individual can serve as the starting point in an iteration. "DE/current-to-rand/2" operator adds one more difference vector, which may search more region than "DE/current-to-rand/1" operator by adding more perturbation. While for operators like "DE/best/1" and "DE/best/2" start their search direction at the best individual every time may lead to the early convergence or may not be effective in solving multimodal problems.

DE/rand/1

$$v_i^g = x_{r1}^g + F * \left(x_{r2}^g - x_{r3}^g\right)$$

DE/rand/2

$$v_i^g = x_{r1}^g + F * \left(x_{r2}^g - x_{r3}^g\right) + F * \left(x_{r4}^g - x_{r5}^g\right)$$

DE/current-to-rand/1

$$v_i^g = x_i^g + F * \left(x_{r2}^g - x_{r3}^g\right)$$

DE/current-to-rand/2

$$v_i^g = x_i^g + F * \left(x_{r2}^g - x_{r3}^g\right) + F^* \left(x_{r4}^g - x_{r5}^g\right)$$

DE/best/1

$$v_i^g = x_{best}^g + F * \left(x_{r2}^g - x_{r3}^g\right)$$

DE/best/2

$$v_i^g = x_{best}^g + F * \left(x_{r2}^g - x_{r3}^g\right) + F^* \left(x_{r4}^g - x_{r5}^g\right)$$

So based on the above analysis on the mutation operators, "DE/current-to-rand/2" operator is served as the prototype of the proposed mutation operator. As researches shown (Gong and Cai 2013; Zhou et al. 2013) that the terminal point of the difference vector with better value can help the DE algorithm gain better results. Motivated by these researches and the interaction between the feasible and infeasible individuals, we set the terminal point of the difference vector as feasible ones in the local search. So we design a novel local search operator "DE/current-to-feasible/2" to improve the performance of εDE algorithm. The "DE/current-to-feasible/2" is a transformation version of "DE/current-to-rand/2" mutation operator. It can be presented as follow:

$$v_i^g = x_i^g + a * \left(x_{feas\_r_1}^g - x_i^g\right) + b * \left(x_{feas\_r_2}^g - x_i^g\right) \tag{9}$$

where $feas\_r_1$ and $feas\_r_2$ are two random indexes chosen from the feasible individual set Q. So the number of feasible individuals must be more than 2. One point should be noted is that the feasible individual here we mention actually is ε-feasible individual, which is considered as generalized feasible ones in εDE algorithm. The reason that we set ε-feasible as feasible ones is for some complex problems it is difficult to find real feasible ones for the whole evolutionary process, Another reason is that in εDE algorithm the ε-feasible individuals are less constraint violated individuals and by using the proposed mutation operator it can help guide the individuals moving towards to the feasible region. a and b are two random generated numbers within the range [0, 1], in these way can we have more perturbation on the search direction. If any dimension in v_i^g exceeds the boundary, then randomly choose a feasible individual and make the specific dimension in v_i^g be equal to the related dimension in

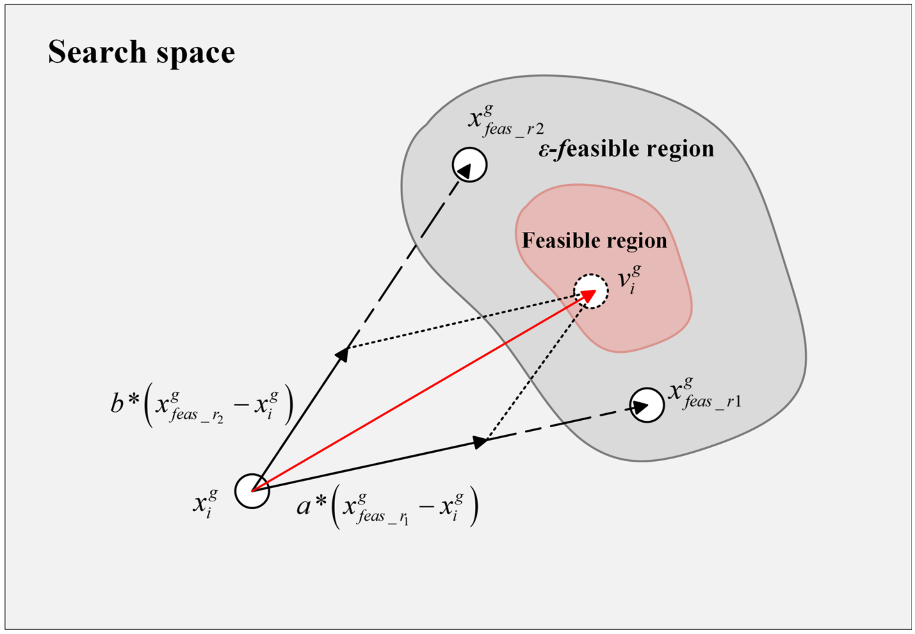


Fig. 1 The "DE/current-to-feasible/2" mutation operator

Fig. 2 The pesudocode of εDE-LS algorithm

```
εDE-LS algorithm
Initialize the individuals {x_1^0,...,x_NP^0}
Initialize the ε level value, ε=ε(0)
While FES<maxFES
{
    Mutation phase
    Crossover phase
    Using ε level comparison in selection phase
    FES=FES+NP;
    Store the ε-feasible ones in Q
    If the number of feasible individuals in Q is bigger than 2
        For i=1: popsize
            Using DE/current-to-feasible/2 to generate V'_i^g
            For j=1:n
                If V'_{i,j}^g exceeds the boundary
                    V'_{i,j}^g = x_{feas_r3,j}^{g+1}
                End If
            End For
            Choose better one between V'_i^g and x_i^{g+1} to survive into next generation
            FES=FES+1;
        End for
    End If
    Update ε level value;
}
```

chosen feasible individual. The mutation operator can also be described in the Fig. 1.

So in the proposed εDE-LS, DE algorithm is used to generate offspring and ε constrained method is used to choose better individual to survive into next generation, and the proposed "DE/current-to-feasible/2" plays as the local search engine to search the area along to the feasible individual direction. For those infeasible solutions, if the number of feasible individuals is <2, the local search phase is skipped. If the number of feasible individuals is more than 2, then for each infeasible individual, local search operator is used. Then the ε level comparison is adopted to choose a better one between the individual and offspring generated by local search operator. The framework of the proposed εDE-LS algorithm can be given as follows (Fig. 2):

Table 1 Parameter settings of εDE-LS algorithm

Popsize	100
MaxFES	5×10^5
θ	0.2
T_c	0.2 ∗ *MaxFES/popsize*
cp	5
F	[0.5, 1.0]
CR	[0.9, 1.0]

Experimental results

Parameter settings

The 24 famous benchmark test functions collected from CEC 2006 (Liang et al. 2006) are adopted in evaluating the performance of the proposed algorithm. The detailed information about the benchmark can be referred to Liang et al. (2006). The parameter settings of the proposed algorithm are shown in Table 1.

Performance of εDE-LS algorithm

Twenty-five independent runs are conducted for the test benchmark functions with 5×10^3, 5×10^4, 5×10^5 *FES*, respectively. The torlerance value δ for the equality constraints is set as 0.0001. The best, median, worst, mean and standard deviation of the error value $(f(\vec{x}) - f(\vec{x}^*))$, where $f(\vec{x}^*)$ is the best objective fitness function value for each benchmark test function that ever known. c is the number of the violated constraints at the median solution: the three numbers refers to the constraints bigger than 1, between 0.01 and 1.0 and between 0.0001and 0.01, respectively. v is mean value of the violations of all the constraints at the median solution. The number in parentheses after best, median and worst solutions is the number of violated constraints.

As shown in Tables 2, 3, 4 and 5, within 5×10^4 *FES*, the functions G02, G03, G04, G09, G10, G12, G19, and G24 can obtain feasible solutionin at least one run. In spite of the test functions G20, G21, and G22, the proposed algorithm can obtatin feasible solution for other twenty-one test benchmark functions in each run within 5×10^5 *FES*. Compared with the best known solution, the proposed algorithm can obtain equal to or better than the best known solution for at least one run for functions G01, G03, G04, G05, G06, G07, G09, G10, G11, G12, G13, G15, and G17. In conclusion, the proposed algorithm can effectively solve the test functions (except for G20–G21) within 5×10^5 *FES*.

In Table 7, we present the number of *FES* needed in each run for each test benchmark function when satisfying the success condition: $f(\vec{x}) - f(\vec{x}^*) \leq 1.0E{-}04$ and $\vec{x}$ is feasible solution. The best, median, worst, mean and SD denote the least, median, most, mean and standard deviation FES when meets the success condition during the 25 independent runs. The feasible rate is the ratio between the feasible solutions and 25 achieved solutions within 5×10^5 *FES*. The success rate is the ratio between the number of success runs and 25 runs within 5×10^5 *FES*. The success performance is the mean number of *FES* for successful runs multiplied by the total runs and divided by the number of successful runs.

From Table 6, we can conclude that 19 out of 24 test benchmark functions can achieve 100 % success rate within 5×10^5 *FES*. εDE-LS algorithm achieves 100 % feasible solutions for 21 out of 24 test benchmark functions. In terms of success performance, εDE-LS algorithm obtained the least *FES* in test benchmark function G01, G04, G07, G14, G15, and G18 comparing with other six state-of-the-art algorithms. As success performance indicate that the proposed εDE-LS requires $<1 \times 10^4$ *FES* for four test benchmark functions, $<5 \times 10^4$ *FES* for 14 test benchmark functions, $<5.0 \times 10^5$ *FES* for 21 test benchmark functions to obtain the require accuracy.

To make a vivid description, we give the convergence curve for test benchmark function G01–G24. The convergence curve of $f(\vec{x}) - f(x^*)$ in Figs. 3, 4, 5, 6 and 7. It is particularly to note that the points with $f(x) - f(x^*) \leq 0$ are not plotted in Figs. 3, 4, 5, 6 and 7.

We can see from Figs. 3, 4, 5, 6 and 7 that the proposed algorithm can meet the satisfying condition within 5×10^5 *FES* for the majority of the test fuctions. The majority of the fucntions can converge to its optimum within 3×10^5 *FES*. Especially for functions G01, G03, G04, G05, G06, G09, G11, G12, G13 G15, and G17, the algorithm can obtain a better solution than the best known solution within 3×10^5 *FES*.

In terms of the limitation of the proposed algorithm, we can see from the above tables and figures that it fails to solve the G20, G21, and G22. These three functions are all linear functions with relatively small feasible region and lots of equality constraints. The optimums are achieved on the boundary of the feasible region also indicate that these function are consist of strong constraints and are difficult to solve.The feasible solution is usually difficult to find. The proposed algorithm is not efficient enough for these three algorithms, which mainly due to the fact that it should utilize the feasible solutions to guide the searching process (Fig. 8).

Comparison with other state-of-the-art algorithms

In order to further verify the effectiveness of the proposed algorithm, several state-of-the-art algorithms are chosen to make a fair comparison on CEC2006 with 5×10^5 *FES* for each test function. ε constrained differential evolution (εDE) proposed by Takahama and Sakai (2006), combin-

Table 2 Function error values achieved when $FES = 5 \times 10^3$, $FES = 5 \times 10^4$ and $FES = 5 \times 10^5$ for function G01-G06

FES	Prob.					
	G01	G02	G03	G04	G05	G06
5×10^3						
Best	−8.4954E+00(5)	4.4970E−01(0)	9.3252E−01(1)	4.5096E+01(0)	−1.7554E+03(3)	−1.0109E+03(1)
Median	−6.3354E+01(7)	4.7380E−01(0)	−3.4992E+00(1)	1.1660E+02(0)	−3.2212E+03(3)	−1.0111E+03(1)
Worst	−7.5423E+01(7)	4.8010E−01(0)	−4.5255E+01(1)	1.6139E+02(0)	−3.9841E+03(3)	−1.0111E+03(1)
c	7, 0, 0	0, 0, 0	1, 0, 0	0, 0, 0	3, 0, 0	1, 0, 0
v	3.8813E+01	0.0000E+01	1.0812E+00	0.0000E+00	2.8533E+02	1.0973E+01
Mean	−5.7660E+01	4.5360E−01	−1.0552E+01	1.1474E+02	−3.1787E+03	−1.0111E+03
SD	1.7380E+01	3.1400E−02	1.0323E+01	3.0230E+01	4.6543E+02	7.3659E−02
5×10^4						
Best	4.6881E+00(1)	1.0004E−01(0)	2.1606E−02(0)	1.7000E−07(0)	−1.1370E+02(3)	−1.0112E+03(1)
Median	6.8690E−01(3)	1.3425E−01(0)	−1.1981E−01(1)	5.4650E−07(0)	−1.3442E+02(3)	−1.0112E+03(1)
Worst	−4.6530E−01(3)	1.6272E−01(0)	−1.0062E−02(1)	−1.4082E−01(1)	−1.8415E+02(3)	−1.0112E+03(1)
c	3, 0, 0	0, 0, 0	0, 1, 0	0, 0, 0	3, 0, 0	1, 0, 0
v	3.2125E+00	0.0000E+00	3.7135E−02	0.0000E+00	1.1530E+01	1.1000E+01
Mean	1.6174E+00	1.3473E−01	1.1744E−02	−5.6322E−03	−1.4474E+02	−1.0112E+03
SD	2.0103E+00	2.0086E−02	1.9339E−01	2.8164E−02	3.2062E+01	6.9619E−13
5×10^5						
Best	0.0000E+00(0)	2.5228E−09(0)	−2.6645E−15(0)	−3.6380E−12(0)	−1.8190E−12(0)	−1.6371E−11(0)
Median	0.0000E+00(0)	7.4951E−09(0)	−2.4425E−15(0)	−3.6380E−12(0)	−1.8190E−12(0)	−1.6371E−11(0)
Worst	0.0000E+00(0)	4.1143E−08(0)	−2.4425E−15(0)	−3.6380E−12(0)	−1.8190E−12(0)	−1.6371E−11(0)
c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
v	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Mean	0.0000E+00	9.9339E−09	−2.5313E−15	−3.6380E−12	−1,8190E−12	−1.6371E−11(0)
SD	0.0000E+00	8.2351E−09	1.1102E−16	0.0000E+00	0.0000E+00	0.0000E+00

ing multi-objective optimization with differential evolution (CMODE) proposed by Wang and Cai (2012), improved $(\mu + \lambda)$-constrained differential evolution (ICDE) proposed by Jia et al. (2013), improved electromagnetism-like mechanism algorithm (ICEM) proposed by Zhang et al. (2013), and multi-objective optimization based reverse strategy with differential evolution algorithm (MRS-DE) proposed by Gao et al. (2015) are selected. Due to the fact that none of these mentioned algorithms can achieve feasible solution for G20, so the results of G20 is not included in the following Table 10. The comparison results are presented in Tables 7, 8, 9 and 10, in which the boldface indicates the best results among these algorithms. All the results are adopted from the original article mentioned above.

Only comparing with εDE algorithm, we can find that the εDE-LS obtains eleven better solutions in terms of best, median, worst, mean solution over the 25 independent runs for functions G02–G09, G13–G16, and G24. Three similar solutions for three functions G01, G11, and G12, which are already the global optimum reported by Liang et al. (2006). For functions G07, G18, G21–G23, less competitive are obtained by εDE-LS and the reason may mainly be the relatively the proposed local search are less effective than gradient-based mutation using gradient matrix when facing with small feasible region. Partly better solutions are achieved for functions G10, G14, G17, and G19. We can conclude that the proposed local search is effective for the majority of the functions.

Comparing with all the selected state-of-the-art algorithm, we can conclude from Tables 7, 8, 9 and 10 that the εDE-LS can achieve the best results for functions G01, G02, G04–G06, G08, G11, G12, G14–G16 in terms of best, median, worst, mean solution over the 25 independent runs. For function G17, the εDE-LS can achieve the best results in terms of best and median solutions. From the comparison with other state-of-the-art algorithms, εDE-LS can achieve promising results on the majority of the test functions.

Then the ranking is given on 23 problems based on mean results obtained by each algorithm. The average ranking is given in Fig. 9 for each algorithm. The p value computed by the Friedman test for mean results is 0.126, which indicates there are no significant differences for the compared

Table 3 Function error values achieved when $FES = 5 \times 10^3$, $FES = 5 \times 10^4$ and $FES = 5 \times 10^5$ for function G07–G12

FES	Prob.					
	G07	G08	G09	G10	G11	G12
5×10^3						
Best	1.3011E+01(2)	−1.1027E+02(1)	3.7169E+01(1)	−3.9862E+03(1)	−7.4017E−02(1)	1.5167E−06(0)
Median	−1.6283E+01(3)	−1.1034E+02(1)	−6.4335E+00(1)	−4.1915E+03(1)	−1.4962E−01(1)	5.7376E−05(0)
Worst	−1.0375E+01(4)	−1.9970E+02(1)	−2.5051E+02(1)	−4.8075E+03(2)	−2.0839E−01(1)	6.1171E−03(0)
c	3, 0, 0	1, 0, 0	1, 0, 0	1, 0, 0	0, 1, 0	0, 0, 0
v	4.6087E+01	4.1174E+00	3.2629E+02	3.0866E+00	1.6724E−01	0.0000E+00
Mean	−1.2813E+00	−1.3448E+02	−1.0075E+02	−4.2920E+03	−1.5820E−01	3.4397E−04
SD	1.4805E+01	6.1299E+01	8.2330E+01	2.9604E+02	2.9831E−02	1.2081E−03
5×10^4						
Best	−6.9784E+00(2)	1.0254E−02(1)	4.0482E+00(0)	1.0924E+03(0)	−1.1910E−03(1)	0.0000E+00(0)
Median	−1.3593E+01(2)	−4.7397E−01(1)	−1.3085E+01(1)	−4.9486E+03(1)	−5.6889E−03(1)	0.0000E+00(0)
Worst	−2.3401E+01(3)	−1.2561E+01(1)	−1.9139E+01(1)	−4.9492E+03(2)	−8.4350E−03(1)	0.0000E+00(0)
c	2, 0, 0	0, 1, 0	1, 0, 0	1, 0, 0	0, 0, 1	0, 0, 0
v	1.1729E+01	3.4910E−01	1.4804E+01	1.2087E+00	6.5396E−03	0.0000E+00
Mean	−1.3664E+01	−8.7003E−01	4.6065E−01	−3.8487E+03	−5.5739E−03	0.0000E+00
SD	6.4618E+00	2.4413E+00	1.6714E+01	2.3647E+03	2.1063E−03	0.0000E+00
5×10^5						
Best	−3.5527E−14(0)	2.7756E−17(0)	−2.2737E−13(0)	−6.3665E−12(0)	0.0000E+00(0)	0.0000E+00(0)
Median	5.4001E−13(0)	2.7756E−17(0)	−2.2737E−13(0)	−3.6380E−12(0)	0.0000E+00(0)	0.0000E+00(0)
Worst	2.0570E−12(0)	2.7756E−17(0)	−1.1369E−13(0)	1.2619E−07(0)	0.0000E+00(0)	0.0000E+00(0)
c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
v	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Mean	6.6152E−13	2.7756E−17(0)	−1.7735E−13	9.8452E−09	0.0000E+00	0.0000E+00
SD	5.5955E−12	0.0000E+00	5.7596E−14	2.7051E−08	0.0000E+00	0.0000E+00

algorithms in all functions. We can conclude that the εDE-LS algorithm obtains the lowest ranking, which indicates the proposed algorithm has the best overall performance.

In addition, considering the importance of multiple problem statistical analysis introduced by Derrac et al. (2011), we present the results of the Wilcoxon signed ranks test results in Table 11, in which the p value is recorded. "*" means that the proposed method shows an improvement over the compared algorithms with upper diagonal of level significance at $\alpha = 0.1$, and lower diagonal level of $\alpha = 0.05$. It can be concluded from Table 11 that the proposed algorithm shows an improvement over CMODE, with a level of significance $\alpha = 0.05$, over ICEM, with $\alpha = 0.1$.

Comparison of εDE-LS and other state-of-the-art algorithms on engineering optimization problems

To further verify the effectiveness of the εDE-LS, four real-world engineering optimization problems are selected which are the three bar truss design problem, pressure vessel design problem, speed reducer design problem and tension spring design problem. Except that the pressure vessel design problem is a hybrid continuous variable and discrete variable problem, the rest problems are continuous variable problems. For the three bar truss problem, the swarm algorithm introduced by Ray and Saini (2001), the method presented by Tsai (2005), cuckoo search algorithm presented by Gandomi et al. (2013) are selected for the comparison. For the last three engineering optimization problems, PSO-DE (Liu et al. 2010), ABC (Karaboga and Basturk 2007), CMODE (Wang and Cai 2012), and DELC (Wang and Li 2010) are chosen for the comparison. The selected compared algorithms in the last three engineering optimization problems have not been used in solving the three bar truss design problem, and hence we choose other three state-of-the-art algorithm instead. The comparison results of these four engineering optimization problems are presented in Tables 11, 12, 13 and 14.

All the compared results of other state-of-the-art algorithms are adopted from the article mentioned above. Twenty-five independent runs are conducted for each problems, the best, median, worst, mean and standard deviation are given in the following tables. Since the *FES* is an important index

Table 4 Function error values achieved when $FES = 5 \times 10^3$, $FES = 5 \times 10^4$ and $FES = 5 \times 10^5$ for function G13–G18

FES	Prob.					
	G13	G14	G15	G16	G17	G18
5×10^3						
Best	−3.0383E−02(3)	−5.2549E+02(3)	−4.3953E+01(2)	−7.1436E−01(3)	−8.8346E+03(4)	−1.1531E+01(8)
Median	−5.5071E−02(3)	−6.6528E+02(3)	−5.2918E+01(2)	−7.7771E−01(3)	−8.8438E+03(4)	−6.9048E+01(8)
Worst	−5.2842E−02(3)	−6.4095E+02(3)	−7.2555E+01(2)	−7.8084E−01(3)	−8.8474E+03(4)	−7.2455E+01(7)
c	2, 1, 0	3, 0, 0	2, 0, 0	3, 0, 0	4, 0, 0	8, 0, 0
v	1.8481E+00	1.7825E+01	3.3432E+01	4.8600E+01	1.3759E+02	6.8620E+01
Mean	1.9749E−01	−6.4085E+02	−5.4146E+01	−7.7483E−01	−8.8401E+03	−6.2684E+01
SD	3.8811E−01	4.2435E+01	7.4146E+00	1.3269E−02	1.3240E+01	1.6131E+01
5×10^4						
Best	−7.9068E−03(2)	−1.1729E+01(3)	−3.7849E−01(2)	−1.5633E−01(1)	−2.1443E+02(4)	9.0719E−01(3)
Median	−9.4001E−03(2)	−2.2262E+01(2)	−1.7991E+00(2)	−7.9326E−01(3)	1.10361E+03(4)	−7.0958E−01(6)
Worst	−1.2545E−02(2)	−2.3858E+01(3)	−2.7148E+00(2)	−7.9326E−01(3)	−8.8535E+03(4)	−2.5365E+00(6)
c	0, 1, 1	1, 1, 0	2, 0, 0	3, 0, 0	4, 0, 0	3, 3, 0
v	1.5494E−01	1.0581E+00	1.1288E+00	4.9070E+01	1.2492E+01	1.6841E+00
Mean	−9.2826E−03	−2.1148E+01	−1.7909E+00	−6.9875E−01	−2.6533E+03	2.7006E−02
SD	2.5490E−03	3.8670E+00	5.2461E−01	2.1440E−01	3.8235E+03	1.0000E+00
5×10^5						
Best	−2.2204E−16(0)	1.4211E−14(0)	−1.1369E−13(0)	3.7748E−15(0)	−1.8190E−12(0)	1.7130E−11(0)
Median	−2.2204E−16(0)	1.4211E−14(0)	−1.1369E−13(0)	3.7748E−15(0)	−1.8190E−12(0)	1.8764E−10(0)
Worst	−1.9429E−16(0)	2.1316E−14(0)	−1.1369E−13(0)	3.7748E−15(0)	7.4058E+01(0)	7.7298E−10(0)
c	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0	0, 0, 0
v	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
Mean	−2.1982E−16	1.6485E−14	−1.1369E−13	3.7748E−15	1.1849E+01	1.9762E−10
SD	7.6852E−18	3.3829E−15	0.0000E+00	0.0000E+00	2.7710E+01	1.8283E−10

for each compared algorithm, the total *FES* of each problem for each algorithm is also given in the Tables as below. Due to the small number of *FES*, the population size for the engineering optimization is set as 40.

Three bar truss design problem

Three bar truss design problem is firstly introduced by Nowcki (1973). Its objective is to optimize the volume of a statistically loaded three bar truss. The three bar truss is subjected to vertical and horizontal forces. The volume of the three bar truss is subjected to stress constraints. The mathematical model and figure of the problem can be given as follows:

Minimize

$$f(\vec{x}) = \left(2\sqrt{2}A_1 + A_2\right) \times l$$

Subject to

$$g_1(\vec{x}) = \frac{\sqrt{2}A_1 + A_2}{\sqrt{2}A_1^2 + 2A_1A_2} P - \sigma \le 0$$

$$g_2(\vec{x}) = \frac{A_2}{\sqrt{2}A_1^2 + 2A_1A_2} P - \sigma \le 0$$

$$g_3(\vec{x}) = \frac{1}{A_1 + \sqrt{2}A_2} P - \sigma \le 0$$

where

$$0 \le A_1 \le 1, \quad 0 \le A_2 \le 1, \quad l = 100\,\text{cm},$$
$$P = 2\,\text{KN/cm}^2, \quad and \quad \sigma = 2\,\text{KN/cm}^2$$

The best result obtained by εDE-LS is (0.788675135051261, 0.408248289172833) with objective function value 263.8958433764684. The constraints violation are (0, −1.464101616605413, 0.535898383394587). The constraint violation is equal to or smaller than zero means the solution do not violate the constraint, which is the same for the following three problems. In Table 11, the comparison results are presented. The best results are in boldface.

Pressure vessel design problem

The problem is to design a cylindrical vessel with hemispherical heads as we present in Fig. 10. The problem is firstly proposed by Kannan and Kramer (1994). The objective is to

Table 5 Function error values achieved when $FES = 5 \times 10^3$, $FES = 5 \times 10^4$ and $FES = 5 \times 10^5$ for function G19–G24

FES	Prob.					
	G19	G20	G21	G22	G23	G24
5×10^3						
Best	2.4288E+02(0)	4.3957E+00(19)	−1.9372E+02(6)	−2.2173E+02(20)	−2.2354E+03(4)	1.9502E−04(0)
Median	4.5531E+02(0)	5.6806E+00(19)	−1.9371E+02(6)	−2.2835E+02(20)	−2.7287E+03(4)	9.9761E−04(0)
Worst	6.0366E+02(0)	5.8927E+00(19)	−1.9372E+02(6)	−2.3590E+02(20)	−2.9216E+03(4)	4.0004E−03(0)
c	0, 0, 0	2, 17, 0	6, 0, 0	20, 0, 0	4, 0, 0	0, 0, 0
v	0.0000E+00	5.5923E+00	1.5010E+02	3.3733E+08	5.1464E+01	0.0000E+00
Mean	4.4309E+02	5.3656E+00	−1.9371E+02	−2.3023E+02	−2.6621E+03	1.4072E−03
SD	8.9504E+01	6.8153E−01	3.3636E−02	9.2511E+00	2.0269E+02	1.0186E−03
5×10^4						
Best	3.8756E+00(0)	2.7461E−03(19)	−1.9372E+02(6)	−2.3643E+02(20)	−1.7076E+03(5)	3.2863E−14(0)
Median	6.1062E+00(0)	2.7582E−03(19)	−1.9372E+02(6)	−2.3643E+02(20)	−1.7143E+03(5)	3.2863E−14(0)
Worst	8.5809E+00(0)	3.4253E−03(19)	−1.9372E+02(6)	−2.3643E+02(20)	−1.7289E+03(5)	3.2863E−14(0)
c	0, 0, 0	1, 13, 5	4, 2, 0	19, 1, 0	3, 2, 0	0, 0, 0
v	0.0000E+00	1.9401E−01	8.9673E+00	1.4405E+07	1.6286E+00	0.0000E+00
Mean	6.3342E+00	2.9203E−03	−1.9372E+02	−2.3643E+02	−1.7121E+03	3.2863E−14
SD	1.2517E+00	4.6739E−04	2.9008E−14	0.0000E+00	1.0262E+01	0.0000E+00
5×10^5						
Best	1.9510E−07(0)	2.0237E−01(14)	−1.9372E+02(6)	−2.3643E+02(20)	4.2501E−07(0)	3.2863E−14(0)
Median	4.0445E−07(0)	2.0098E−01(16)	−1.9372E+02(6)	−2.3643E+02(20)	7.7408E−06(0)	3.2863E−14(0)
Worst	1.1072E−06(0)	1.9777E−01(16)	−1.9372E+02(6)	−2.3643E+02(20)	5.8830E−03(0)	3.2863E−14(0)
c	0, 0, 0	0, 1, 11	3, 3, 0	19, 1, 0	0, 0, 0	0, 0, 0
v	0.0000E+00	1.0169E−02	7.5953E+00	1.4839E+02	0.0000E+00	0.0000E+00
Mean	4.4233E−07	2.01411E−01	−1.9372E+02	−2.3643E+02	3.9960E−04	3.2863E−14
SD	2.1203E−07	1.3191E−03	2.9008E−14	0.0000E+00	1.2591E−03	0.0000E+00

minimize the total cost, which includes the cost of material, forming and welding. Four variables are needed to be optimized, that is T_s (x_1, Thickness of the shell), T_h (x_2, thickness of the head), R (x_3, inner radius), and L (x_4, length of the cylindrical section of the vessel, not including the head). It is worth mentioning that T_s and T_h should be the integer multiples of 0.0625 in. The mathematical model of this problem can be given as follows:

Minimize

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

Subject to

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \le 0$$

$$g_2(\vec{x}) = -x_2 + 0.00954x_3 \le 0$$

$$g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1{,}296{,}000 \le 0$$

$$g_4(x) = x_4 - 240 \le 0$$

$$1 * 0.0625 \le x_1 \le 99 * 0.0625, \quad 1 * 0.0625 \le x_2 \le 99 * 0.0625, \quad 10 \le x_3 \le 200, \quad 10 \le x_4 \le 200$$

The best feasible solution obtained by εDE-LS is (0.8125, 0.4375, 42.09844455958549, 176.636598424394). The constraint violation are (0, −0.035880829015544, 0, −63.363404157560581) with objective function value 6059.714335048436. Table 12 presents the results obtained by εDE-LS and other state-of-the-art algorithms. The best results are in boldface.

Tension compression spring design problem

The problem is firstly proposed by Arora (1989) as we present in Fig. 11. The objective is to optimize the weight of the tension compression spring. There are three variables that are needed to be optimized: the wire diameter $d(x_1)$, the mean coil diameter $D(x_2)$, and the number of active coils $N(x_3)$. The constraints include the minimum deflection, shear stress, surge frequency, constraint, outside diameter and bounds on

Table 6 The success performance, feasible rate and success rate of the εDE-LS algorithm

Prob.	Best	Median	Worst	Mean	SD	Feasible rate (%)	Success rate (%)	Success performance
G01	119,800	123,300	126,000	123,236	1684	100	100	123,236
G02	244,100	266,900	295,100	267,272	10,609	100	100	267,272
G03	97,000	102,800	110,600	102,924	3129	100	100	102,924
G04	35,100	37,900	93,200	39,764	11,207	100	100	39,764
G05	100,900	101,900	106,100	102,084	1048	100	100	102,084
G06	102,500	105,800	111,300	106,064	2009	100	100	106,064
G07	201,200	218,700	225,900	217,580	6073	100	100	217,580
G08	56,200	95,100	101,000	92,196	11,189	100	100	92,196
G09	104,900	109,600	112,100	109,228	1862	100	100	109,228
G10	245,300	287,800	426,500	311,892	60,015	100	100	311,892
G11	39,400	73,600	81,800	70,692	10,413	100	100	70,692
G12	1700	4900	6400	4648	1453	100	100	4648
G13	82,600	84,400	87,100	84,632	1418	100	100	84,632
G14	165,000	174,500	182,100	174,044	4784	100	100	174,044
G15	86,700	90,100	92,000	89,984	1310	100	100	89,984
G16	98,100	111,800	121,000	111,116	5985	100	100	111,116
G17	190,300	300,300	406,700	279,085	58,503	100	80	348,856
G18	202,600	227,100	255,200	229,152	11,953	100	100	229,152
G19	315,300	343,800	365,500	342,780	12,863	100	100	34,2780
G20	NA	NA	NA	NA	NA	0	0	NA
G21	NA	NA	NA	NA	NA	0	0	NA
G22	NA	NA	NA	NA	NA	0	0	NA
G23	383,500	455,100	491,300	448,100	28,923	100	84	533,452
G24	6100	7300	8100	7320	454	100	100	7320

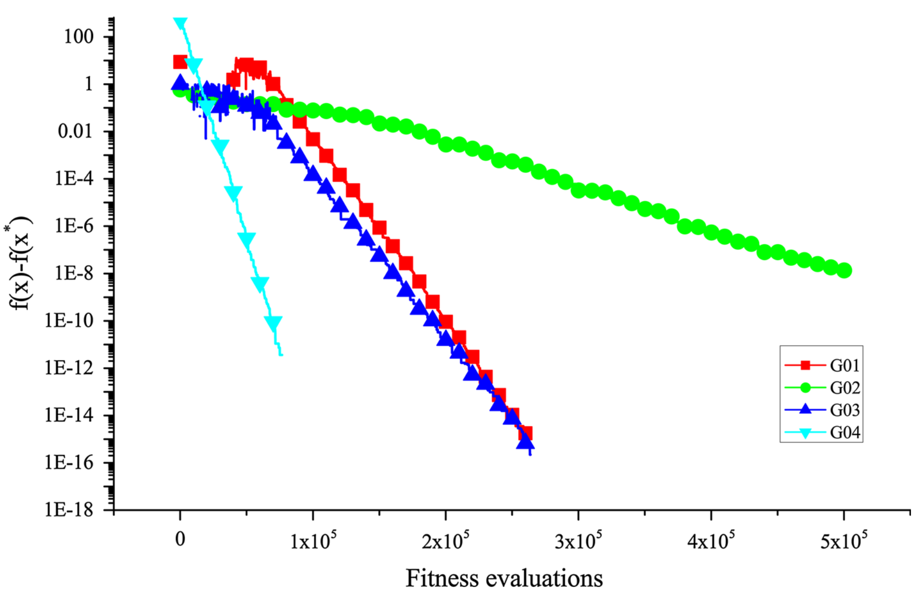


Fig. 3 Convergence curve of $f(\vec{x}) - f(x^*)$ for G01–G04

variables. The mathematical model of the problem can be summarized as follows (Fig. 12):

$$\text{Minimize } f(\vec{x}) = (x_3 + 2)\, x_2 x_1^2$$

Subject to

$$g_1(\vec{x}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$

$$g_2(\vec{x}) = \frac{4x_2^2 - x_1 x_2}{12566\left(x_2 x_1^3 - x_1^4\right)} + \frac{1}{5108 x_1^2} - 1 \le 0$$

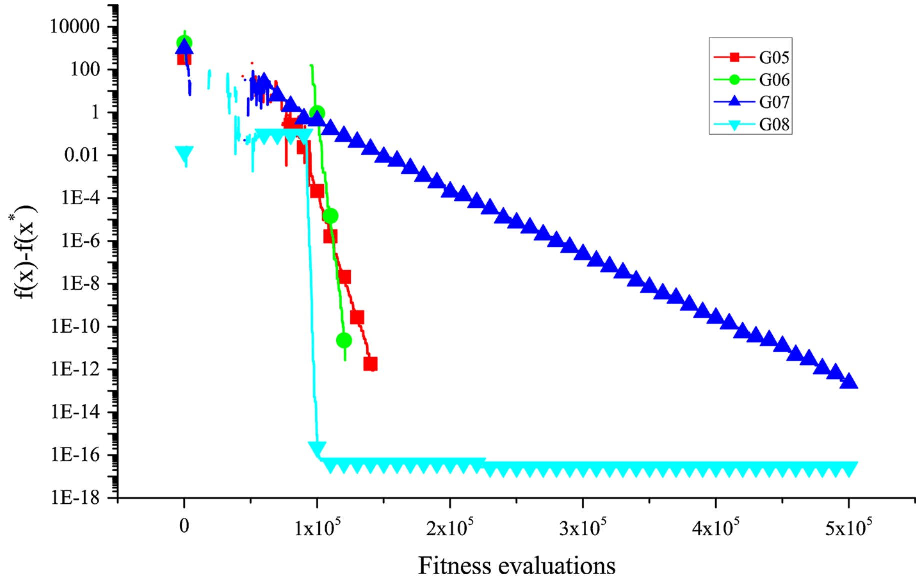


Fig. 4 Convergence curve of $f(\vec{x}) - f(x^*)$ for G05–G08

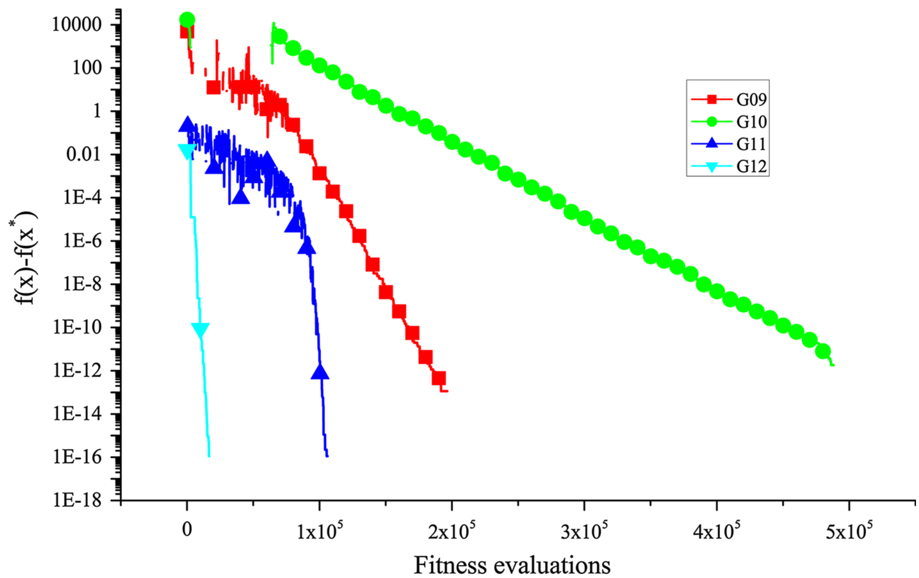


Fig. 5 Convergence curve of $f(\vec{x}) - f(x^*)$ for G09–G12

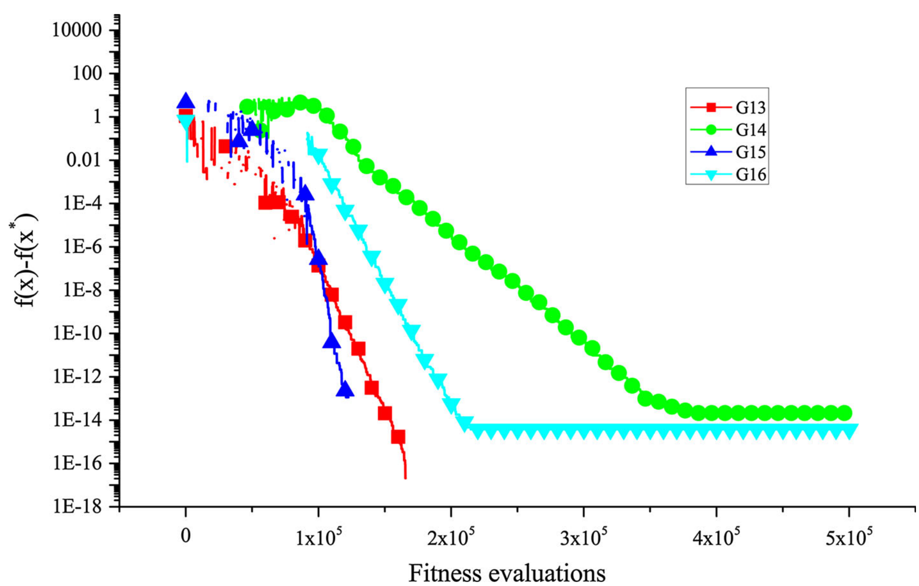


Fig. 6 Convergence curve of $f(\vec{x}) - f(x^*)$ for G13–G16

Fig. 7 Convergence curve of $f(\vec{x}) - f(x^*)$ for G17–G20

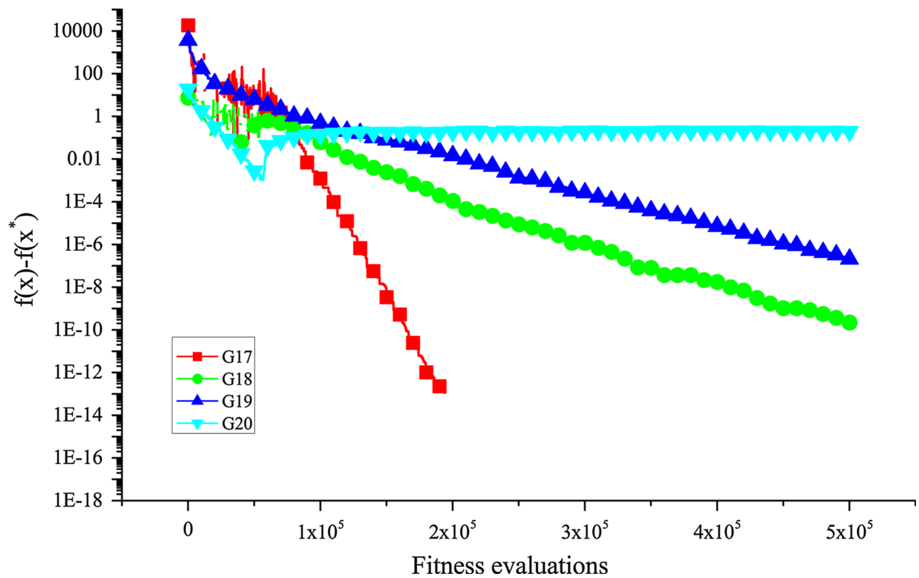
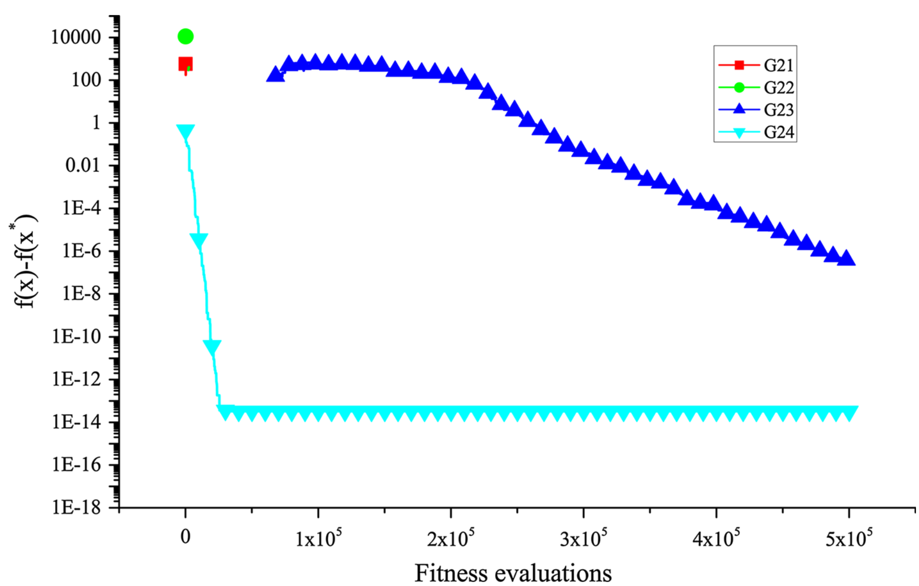


Fig. 8 Convergence curve of $f(\vec{x}) - f(x^*)$ for G21–G24



$$g_3(\vec{x}) = 1 - \frac{140.45x_1}{x_2^2 x_3} \leq 0$$

$$g_4(\vec{x}) = \frac{x_2 + x_1}{1.5} - 1 \leq 0$$

$$0.05 \leq x_1 \leq 2.0, \quad 0.25 \leq x_2 \leq 1.3, \quad 2 \leq x_3 \leq 15$$

The best result obtained by εDE-LS is (0.051689060493998, 0.356717725635278, 11.288966582014742) with objective function value 0.012665232788320. The constraints violation are (−0.000000000000036, −0.000000000000011, −4.053785602361403, −0.727728809247150). Table 13 presents the results obtained by εDE-LS and other state-of-the-art algorithms. The best results are in boldface.

Speed reducer design problem

The problem is to minimize the weight of the speed reducer (Rao 1996). The constraints include the bending stress of the gear teeth, surface stress, and transverse deflections of the shafts. The variables that are need to be optimized are the face width $b(x_1)$, module of teeth $m(x_2)$, number of teeth in the pinion $z(x_3)$, length of the first shaft between bearings $l_1(x_4)$, length of the second shaft between bearings $l_2(x_5)$, the diameter of the first shaft $d_1(x_6)$ and second shaft $d_2(x_7)$. The mathematical model of the problem can be given as follows:

Table 7 The comparison with other state-of-the-art algorithms on function G01–G06

Prob.	Algo.	Best	Median	Worst	Mean	SD
G01	εDE	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	0.0000E+00
	CMODE	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	0.0000E+00
	ICDE	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.00E+00
	ICEM	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	0.0000E+00
	MRS-DE	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	0.0000E+00
	εDE-LS	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	0.0000E+00
G02	εDE	4.0394E−09	3.0933E−08	7.3163E−08	3.0333E−08	1.7523E−08
	CMODE	4.1726E−09	1.1372E−08	1.1836E−07	2.0387E−08	2.4195E−08
	ICDE	1.28E−08	1.39E−07	7.43E−07	2.28E−07	2.06E−07
	ICEM	2.2823E−06	5.4463E−06	1.1014E−02	1.3617E−03	2.6951E−03
	MRS-DE	2.3571E−07	2.5776E−02	1.4238E−01	4.3026E−02	4.0583E−02
	εDE-LS	**2.5228E−09**	**7.4951E−09**	**4.1143E−08**	**9.9339E−09**	8.2351E−09
G03	εDE	−4.4409E−16	−4.4409E−16	−4.4409E−16	−4.4409E−16	2.9682E−31
	CMODE	2.3964E−10	1.1073E−09	2.5794E−09	1.1665E−09	5.2903E−10
	ICDE	**−1.00E−11**	**−1.00E−11**	**−1.00E−11**	**−1.00E−11**	1.63E−16
	ICEM	3.2984E−11	3.3199E−08	1.3832E−06	1.3609E−07	2.7619E−07
	MRS-DE	6.4531E−12	4.3339E−07	7.3502E−06	1.0822E−06	1.7011E−06
	εDE-LS	−2.6645E−15	−2.4425E−15	−2.4425E−15	−2.5513E−15	1.1102E−16
G04	εDE	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	CMODE	7.6398E−11	7.6398E−11	7.6398E−11	7.6398E−11	2.6382E−26
	ICDE	7.64E−11	7.64E−11	7.64E−11	7.64E−11	2.64E−26
	ICEM	7.2760E−11	7.2760E−11	7.6398E−11	7.2905E−11	7.1290E−13
	MRS-DE	**−3.6380E−12**	**−3.6380E12**	**−3.6380E−12**	**−3.6380E−12**	**0.0000E+00**
	εDE-LS	**−3.6380E−12**	**−3.6380E12**	**−3.6380E−12**	**−3.6380E−12**	0.0000E+00
G05	εDE	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	CMODE	**−1.8190E−12**	**−1.8190E−12**	**−1.8190E−12**	**−1.8190E−12**	1.2366E−27
	ICDE	**−1.82E−12**	**−1.82E−12**	**−1.82E−12**	**−1.82E−12**	1.24E−27
	ICEM	**−1.8190E−12**	**−1.8190E−12**	−9.0950E−013	−1.7462E−12	2.4674E−13
	MRS-DE	**−1.8190E−12**	**−1.8190E−12**	**−1.8190E−12**	**−1.8190E−12**	0.0000E+00
	εDE-LS	**−1.8190E−12**	**−1.8190E−12**	**−1.8190E−12**	**−1.8190E−12**	0.0000E+00
G06	εDE	1.1823E−11	1.1823E−11	1.1823E−11	1.1823E−11	0.0000E+00
	CMODE	3.3651E−11	3.3651E−11	3.3651E−11	3.3651E−11	1.3191E−26
	ICDE	3.37E−11	3.37E−11	3.37E−11	3.37E−11	3.37E−11
	ICEM	3.3651E−11	3.3651E−11	3.3651E−11	3.3651E−11	0.0000E+00
	MRS-DE	**−1.6371E−11**	**−1.6371E−11**	**−1.6371E−11**	**−1.6371E−11**	0.0000E+00
	εDE-LS	**−1.6371E−11**	**−1.6371E−11**	**−1.6371E−11**	**−1.6371E−11**	0.0000E+00

$$f(\vec{x}) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right)$$

$$g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0$$

$$g_2(x) = \frac{397.5}{x_1x_2^2x_3^2} - 1 \leq 0$$

$$g_3(x) = \frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \leq 0$$

$$g_4(x) = \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \leq 0$$

$$g_5(x) = \frac{\left(\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6\right)^{1/2}}{110.0x_6^3} - 1 \leq 0$$

Table 8 Comparison with other state-of-the-art algorithms for function G07–G12

Prob.	Alg.	Best	Median	Worst	Mean	SD
G07	εDE	−1.8474E−13	**−1.8474E−13**	**−1.7764E−13**	**−1.8360E−13**	2.1831E−15
	CMODE	7.9783E−11	7.9783E−11	7.9783E−11	7.9783E−11	7.6527E−15
	ICDE	7.98E−11	7.98E−11	7.98E−11	7.98E−11	5.26E−15
	ICEM	7.9762E−11	7.9769E−11	7.9787E−11	7.9770E−11	6.0526E−15
	MRS-DE	**−2.3093E−13**	1.3156E−08	1.0777E−06	1.0526E−07	2.4441E−07
	εDE-LS	−3.5527E−14	5.4001E−13	2.0570E12	6.6152E−13	5.5955E−12
G08	εDE	4.1633E−17	4.1633E−17	4.1633E−17	4.1633E−17	1.2326E−32
	CMODE	8.1964E−11	8.1964E−11	8.1964E−11	8.1964E−11	6.3596E−18
	ICDE	8.20E−11	8.20E−11	8.20E−11	8.20E−11	2.78E−18
	ICEM	8.1964E−11	8.1964E−11	8.1964E−11	8.1964E−11	3.8774E−26
	MRS-DE	**2.7756E−17**	**2.7756E−17**	**2.7756E−17**	**2.7756E−17**	0.0000E+00
	εDE-LS	**2.7756E−17**	**2.7756E−17**	**2.7756E−17**	**2.7756E−17**	0.0000E+00
G09	εDE	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	CMODE	**−9.8225E−11**	**−9.8225E−11**	**−9.8225E−11**	**−9.8225E−11**	4.9554E−14
	ICDE	**−9.82E−11**	−9.81E−11	−9.81E−11	**−9.82E−11**	5.76E−14
	ICEM	**−9.8225E−11**	**−9.8225E−11**	**−9.8225E−11**	**−9.8225E−11**	4.4556E−13
	MRS-DE	−2.2737E−13	−2.2737E−13	−1.1368E−13	−2.1373E−13	−7.7487E−12
	εDE-LS	−2.2737E−13	−2.2737E−13	−1.1369E−13	−1.7735E−13	5.7596E−14
G10	εDE	−1.8190E−12	−9.0949E−13	−9.0949E−13	−1.2005E−12	4.2426E−13
	CMODE	6.2755E−11	6.2755E−11	6.3664E−11	6.2827E−11	2.5183E−13
	ICDE	6.18E−11	6.28E−11	6.28E−11	6.27E−11	2.64E−26
	ICEM	6.1846E− 11	6.2755E− 11	6.2755E− 11	6.2391E− 11	7.1290E−13
	MRS-DE	**−8.1855E−12**	**−7.2760E−12**	**−3.6380E−12**	**−7.3487E−12**	8.6760E−13
	εDE-LS	−6.3665E−12	−3.6380E−12	1.2691E−07	9.8452E−09	2.7015E−08
G11	εDE	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	CMODE	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	ICDE	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	ICEM	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	MRS-DE	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	εDE-LS	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
G12	εDE	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	CMODE	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	ICDE	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	ICEM	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	MRS-DE	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**
	εDE-LS	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**	**0.0000E+00**

$$g_6(x) = \frac{\left(\left(\frac{745x_4}{x_2x_3}\right) + 157.5 \times 10^6\right)^{1/2}}{85.0x_7^3} - 1 \le 0$$

$$g_7(x) = \frac{x_2x_3}{40} - 1 \le 0$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \le 0$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \le 0$$

$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0$$

$$2.6 \le x_1 \le 3.6, \quad 0.7 \le x_2 \le 0.8, \quad 17 \le x_3 \le 28, \quad 7.3 \le x_4 \le 8.3, \quad 7.8 \le x_5 \le 8.3, \quad 2.9 \le x_6 \le 3.9, \quad 5.0 \le x_7 \le 5.5$$

Table 9 Comparison with other state-of-the-art algorithms for function G13–G18

Prob.	Alg.	Best	Median	Worst	Mean	SD
G13	εDE	−9.7145E−17	−9.7145E−17	−9.7145E−17	−9.7145E−17	0.0000E+00
	CMODE	4.1897E−11	4.1897E−11	4.1897E−11	4.1897E−11	1.0385E−17
	ICDE	4.19E−11	4.19E−11	4.19E−11	4.19E−11	1.13E−17
	ICEM	4.1898E−11	3.8486E−01	5.3327E−02	3.9090E−01	6.4960E−01
	MRS-DE	**−2.4286E−16**	**−2.2220E−16**	**−2.2220E−16**	**−2.2371E−16**	−2.2371E−16
	εDE-LS	−2.2204E−16	−2.2204E−16	−1.9429E−16	−2.1982E−16	7.6825E−18
G14	εDE	**1.4211E−14**	2.1316E−14	**2.1316E−14**	2.1032E−14	1.3924E−15
	CMODE	8.5123E−12	8.5194E−12	8.5194E−12	8.5159E−12	3.6230E−15
	ICDE	8.51E−12	8.51E−12	8.52E−12	8.51E−12	2.36E−15
	ICEM	8.5052E−12	8.5123E−12	8.5194E−12	8.5123E−12	2.0097E−15
	MRS-DE	2.2365E−09	1.2820E−03	2.1782E−01	1.4900E−02	4.4172E−02
	εDE-LS	**1.4211E−14**	**1.4211E−14**	**2.1316E−14**	**1.6485E−14**	3.3829E−15
G15	εDE	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	CMODE	6.0822E−11	6.0822E−11	6.0822E−11	6.0822E−11	0.0000E+00
	ICDE	6.52E−11	6.52E−11	6.52E−11	6.52E−11	1.32E−26
	ICEM	6.5214E−11	6.5214E−11	6.5214E−11	6.5214E−11	0.0000E+00
	MRS-DE	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00	0.0000E+00
	εDE-LS	**−1.1369E−13**	**−1.1369E−13**	**−1.1369E−13**	**−1.1369E−13**	0.000E+00
G16	εDE	4.4409E−15	4.4409E−15	4.4409E−15	4.4409E−15	1.5777E−30
	CMODE	6.5213E−11	6.5213E−11	6.5213E−11	6.5213E−11	2.6382E−26
	ICDE	6.52E−11	6.52E−11	6.52E−11	6.52E−11	1.32E−26
	ICEM	6.5214E−11	6.5214E−11	6.5214E−11	6.5214E−11	0.0000E+00
	MRS-DE	**3.7748E−15**	**3.7748E−15**	5.5511E−15	4.0590E−15	6.4652E−16
	εDE-LS	**3.7748E−15**	**3.7748E−15**	**3.7748E−15**	**3.7748E−15**	0.0000E+00
G17	εDE*	1.8190E−12	1.8190E−12	1.8190E−12	1.8190E−12	1.2177E−17
	CMODE*	1.8189E−12	1.8189E−12	1.8189E−12	1.8189E−12	1.2366E−27
	ICDE*	−1.82E−11	−1.82E−11	**−1.46E−11**	**−1.78E−11**	9.51E−13
	ICEM	−0.0058	7.4052E+01	8.4353E+01	6.5588E+01	2.4306E+01
	MRS-DE	−0.0058	−0.0058	7.4052E+01	2.3693E+01	3.5259E+01
	εDE-LS	−0.0058	−0.0058	7.4000E+01	1.1269E+01	2.7710E+01
	εDE-LS*	**−1.8190E−12**	**−1.8190E−12**	7.4058E+01	1.1849E+01	2.7710E+01
G18	εDE	3.3307E−16	3.3307E−16	4.4409E−16	3.3751E−16	2.1756E−17
	CMODE	1.5561E−11	1.5561E−11	1.5561E−11	1.5561E−11	6.5053E−17
	ICDE	1.56E−11	1.56E−11	1.56E−11	1.56E−11	6.60E−27
	ICEM	1.5561E−11	1.5561E−11	1.9104E−01	2.2925E−02	6.2082E−02
	MRS-DE	**1.1102E−16**	**2.2204E−16**	**2.2204E−16**	**2.0428E−16**	4.1541E−17
	εDE-LS	1.7130E−11	1.8764E−10	7.7298E−10	1.9762E−10	1.8283E−10

The best result obtained by εDE-LS is (3.500000001834946, 0.700000000010531, 17.000000006024688, 7.300000003292110, 7.715319945129747, 3.350214666835395, 5.286654468646312) with objective function value 2994.471071240866. The constraints violation are (−1.000000000000000, −1.000000000000000, −0.499172248051728, −0.904643903608071, −0.000000000660707, −0.000000002074479, −0.702499999890092, −0.000000000509226, −0.583333333121156, −0.051325753817815, −0.000000003838960). Table 13 presents the results obtained by εDE-LS and other state-of-the-art algorithms. The best results are in boldface.

Discussion on the four engineering optimization problems

From the experimental results of the previous sections, we can conclude that the proposed algorithm can achieve the best results on three bar truss design problem, pressure ves-

Table 10 Comparison with other state-of-the-art algorithms for function G19, G21–G24

Prob.	Alg.	Best	Median	Worst	Mean	SD
G19	εDE	5.2162E−08	5.2162E−08	5.9840E−05	5.3860E−06	1.2568E−05
	CMODE	1.1027E−10	2.1582E−10	5.4446E−10	2.4644E−10	1.0723E−10
	ICDE	**4.63E−11**	**4.63E−11**	**4.63E−11**	**4.63E−11**	5.05E−15
	ICEM	**4.6313E−11**	**4.6313E−11**	**4.6313E−11**	**4.6313E−11**	3.2920E−14
	MRS-DE	2.8358E−07	1.0543E−04	3.9297E−01	2.4252E−02	8.4407E−02
	εDE-LS	1.9510E−07	4.0445E−07	1.1072E−06	4.4233E−07	2.210E−07
G21	εDE	−2.8422E−14	−2.8422E−14	1.4211E−13	−2.1600E−14	3.3417E−14
	CMODE	−3.1237E−10	−2.39436E−10	1.3097E+02	2.6195E+01	5.3471E+01
	ICDE	−3.05E−10	−2.58E−10	−1.68E−10	−2.53E−10	2.80E−11
	ICEM	**−3.4743E−10**	**−3.4731E−10**	**−2.8948E−10**	**−3.3427E−10**	1.2013E−11
	MRS-DE	3.7748E−15	3.7745E−15	5.5511E−15	4.0590E−15	6.4652E−16
	εDE-LS	–	–	–	–	–
G22	εDE	1.9518E+00	1.2332E+01	6.8922E+01	1.8369E+01	1.5690E+01
	CMODE	–	–	–	–	–
	ICDE	2.81E+00	2.04E+01	6.05E+01	2.29E+01	1.32E+01
	ICEM	–	–	–	–	–
	MRS-DE	–	–	–	–	–
	εDE-LS	–	–	–	–	–
G23	εDE	0.0000E+00	0.0000E+00	**5.6843E−14**	**2.2737E−15**	1.1139E−14
	CMODE	1.8758E−12	1.5859E−11	2.8063E−10	4.4772E−11	7.3264E−11
	ICDE	−1.71E−13	5.68E−14	1.08E−12	1.02E−13	2.92E−13
	ICEM	−2.8422E−13	5.6843E−14	1.7224E−11	7.2532E−13	3.3719E−12
	MRS-DE	**−5.6843E−13**	**−1.1369E−13**	3.0001E+02	4.9018E+01	1.1190E+02
	εDE-LS	4.2501E−07	7.7408E−06	5.8830E−03	3.9960E−04	1.2591R-03
G24	εDE	5.7732E−14	5.7732E−14	5.7732E−14	5.7732E−14	2.5244E−29
	CMODE	4.6735E−12	4.6735E−12	4.6735E−12	4.6735E−12	82445E−28
	ICDE	4.67E−12	4.67E−12	4.67E−12	4.67E−12	0.0000E+00
	ICEM	4.6736E−12	4.6736E−12	4.6736E−12	4.6736E−12	0.0000E+00
	MRS-DE	**3.2862E−14**	**3.2862E−14**	**3.2862E−14**	**3.2862E−14**	0.0000E+00
	εDE-LS	3.2863E−14	3.2863E−14	3.2863E−14	3.2863E−14	0.0000E+00

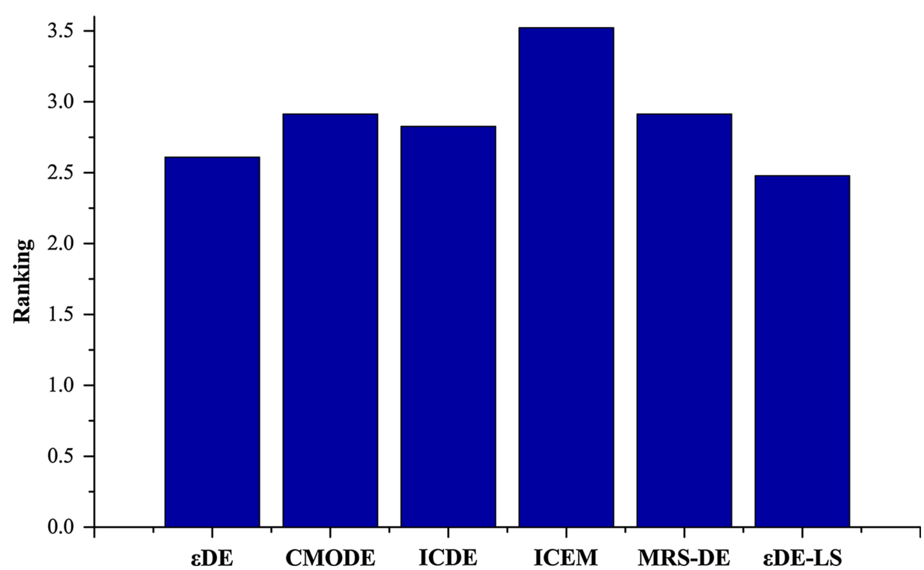


Fig. 9 Average ranking of six algorithms

Table 11 Wilcoxon signed ranks test results

	p value
εDE-LS versus εDE	0.881
εDE-LS versus CMODE	0.040*
εDE-LS versus ICDE	0.176
εDE-LS versus ICEM	0.061*
εDE-LS versus MRS-DE	0.359

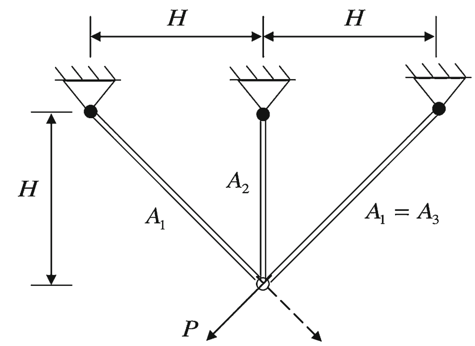


Fig. 10 Three bar truss design

sel design problem, and tension compression spring design problem in terms of best, mean, worst and standard deviation with the relatively small *FES*. In the last speed reducer design problem the εDE-LS algorithm show less competitive than CMODE and DELC algorithms. The reason is that speed reducer design problem consists of eleven constraints, which may be difficult for εDE-LS algorithm to find feasible solutions in relatively small *FES*. Based on the above analysis and experimental results, we can see that the εDE-

Table 12 Comparison with other state-of-the-art algorithms on three bar truss design problem

Three bar truss	Ray and Saini	Tsai	Gandomi et al.	εDE-LS
Solutions				
Best	264.3	263.68	263.9716	**263.895843376468**
Mean	–	–	–	**263.895843376468**
Worst	–	–	–	**263.895843376468**
SD	–	–	–	**2.3206E−14**
FES	–	–	15,000	15,000
Best solutions				
A_1	0.795	0.788	0.78867	0.788675135051261
A_2	0.395	0.408	0.40902	0.408248289172833
g_1	−0.00169	0.00082[a]	−0.00029	0
g_2	−0.26124	−0.2674	−0.26853	−1.464101616605413
g_3	−0.74045	−0.73178	−0.73176	−0.535898383394587

[a] Constraint is violated

Table 13 Comparison with other state-of-the-art algorithms on pressure vessel design problem

	PSO-DE	ABC	CMODE	DELC	εDE-LS
Pressure vessel design					
Best	**6059.7143**	6059.7147	**6059.714335**	**6059.7143**	**6059.7143**
Mean	**6059.7143**	6245.3081	**6059.714335**	**6059.7143**	**6059.7143**
Worst	**6059.7143**	–	**6059.714335**	**6059.7143**	**6059.7143**
SD	1.0E−10	2.1E+02	3.62E−10	2.1E−11	**3.4030E−13**
FES	42,100	30,000	30,000	30,000	20,000

Table 14 Comparison with other state-of-the-art algorithms on tension compression spring design problem

	PSO-DE	ABC	CMODE	DELC	εDE-LS
Tension compression spring design					
Best	0.012665	**0.012665**	**0.012665233**	**0.012665233**	**0.012665233**
Mean	0.012665	0.012709	0.012667168	0.012665267	**0.012665233**
Worst	0.012665	–	0.012676809	0.012665575	**0.012665233**
SD	1.2E−08	1.3E−02	3.09E−06	1.3E−07	**5.0075E−14**
FES	24,950	30,000	24,000	20,000	20,000

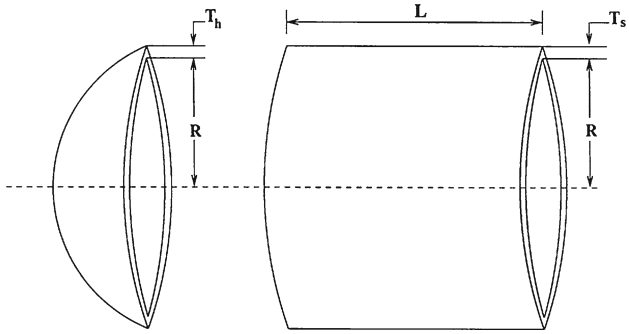


Fig. 11 Schematic of pressure vessel

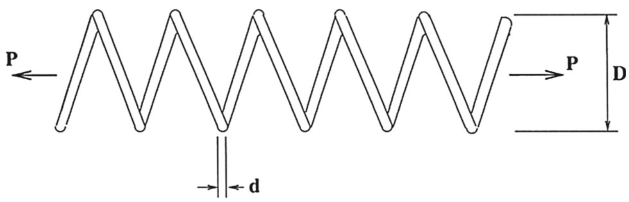


Fig. 12 Schematic of tension compression spring

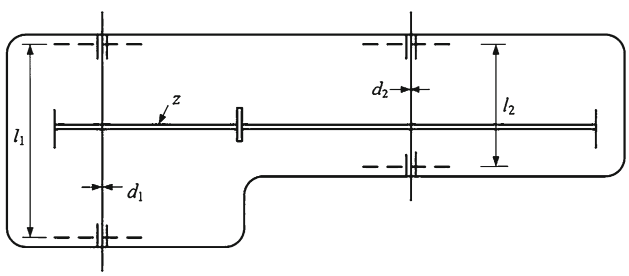


Fig. 13 Schematic of the speed reducer

LS algorithm could be competitive in solving COPs with a small number of constraints within a few *FES*.

The reason that εDE-LS is effective is that the information of the feasible individuals, namely ε -feasible individuals, is useful for guide the population search the region along the direction of the feasible region. The proposed local search, namely "DE/current-to-feasible/2", though simple it is, also effective in finding more precisely solutions for these engineering optimization problems.

Case study: car side impact design

The car side impact design is proposed by Gu et al. (2001). It is extracted from real-world problems and belongs to hybrid continuous variable and discrete variable problem. As we can see from the FEM model of the problem from Fig. 13, the car is exposed to a side-impact on the foundation of the European Enhanced Vehicle-Safty Committee (EEVC) procedures. The objective is to minimize the total weight. The decision variables are thickness of the B-Pillar inner, B-Pillar reinforcement, floor side inner, cross members, door beam, door beltline reinforcement, and roof rail (x_1–x_7), materials of B-Pillar inner and floor side inner (x_8 and x_9) and barrier height, and hitting position (x_{10} and x_{11}). The constraints include load in abdomen (g_1), dummy upper chest (g_2), dummy middle chest (g_3), dummy lower chest (g_4), upper rib deflection (g_5), middle rib deflection (g_6), lower rib deflection (g_7), pubic force (g_8), velocity of V-Pillar at middle point (g_9), and velocity of front door at V-Pillar (g_{10}). The mathematical model of the problem can be formulated as follows:

$$\text{Min } f(X) = Weight = 1.98 + 4.90x_1 + 6.67x_2 + 6.98x_3 + 4.01x_4 + 1.78x_5 + 2.73x_7$$

$$s.t. \begin{cases} g1 = Fa = 1.16 - 0.3717x_2x_4 - 0.00931x_2x_{10} - 0.484x_3x_9 + 0.01343x_6x_{10} \le 1 \\ g2 = VC_u = 0.261 - 0.0159x_1x_2 - 0.188x_1x_8 - 0.019x_2x_7 + 0.0144x_3x_5 + 0.0008757x_5x_{10} + 0.080405x_6x_9 + 0.00139x_8x_{11} + 0.00001575x_{10}x_{11} \le 0.32 \\ g3 = VC_m = 0.214 + 0.00817x_5 - 0.131x_1x_8 - 0.0704x_1x_9 + 0.03099x_2x_6 - 0.018x_2x_7 + 0.0208x_3x_8 + 0.121x_3x_9 - 0.00364x_5x_6 + 0.0007715x_5x_{10} - 0.0005354x_6x_{10} + 0.00121x_8x_{11} \le 0.32 \\ g4 = VC_l = 0.074 - 0.061x_2 - 0.163x_3x_8 + 0.001232x_3x_{10} - 0.166x_7x_9 + 0.227x_2^2 \le 0.32 \\ g5 = \Delta ur = 28.98 + 3.818x_3 - 4.2x_1x_2 + 0.0207x_5x_{10} + 6.63x_6x_9 - 7.7x_7x_8 + 0.32x_9x_{10} \le 32 \\ g6 = \Delta mr = 33.86 + 2.95x_3 + 0.1792x_{10} - 5.057x_1x_2 - 11.0x_2x_8 - 0.0215x_5x_{10} - 9.98x_7x_8 + 22.0x_8x_9 \le 32 \\ g7 = \Delta lr = 46.36 - 9.9x_2 - 12.9x_1x_8 + 0.1107x_3x_{10} \le 32 \\ g8 = Fp = 4.72 - 0.5x_4 - 0.19x_2x_3 - 0.0122x_4x_{10} + 0.009325x_6x_{10} + 0.000191x_{11}^2 \le 4 \\ g9 = VMBP = 10.58 - 0.674x_1x_2 - 1.95x_2x_8 + 0.02054x_3x_{10} - 0.0198x_4x_{10} + 0.028x_6x_{10} \le 9.9 \\ g10 = VFD = 16.45 - 0.489x_3x_7 - 0.843x_5x_6 + 0.0432x_9x_{10} - 0.0556x_9x_{11} - 0.000786x_{11}^2 \le 15.7 \end{cases}$$

$$0.5 \le x_1 – x_7 \le 1.5; \quad x_8, x_9 \in (0.192, 0.345); \quad -30 \le x_{10}, x_{11} \le 30;$$

In this case, the εDE-LS is compared with PSO, DE, GA, firefly algorithm (FA), and cuckoo search algorithm (CS) by Gandomi et al. (2011, 2013). The simulations are conducted with 20,000 *FES* for all the algorithms. Since the number of independent runs are not reported by Gandomi et al. (2011,

Table 15 Comparison with other state-of-the-art algorithms on speed reducer design problem

	PSO-DE	ABC	CMODE	DELC	εDE-LS
Speed reducer design					
Best	2996.348	2997.058	**2994.4710661**	**2994.471066**	2994.471071
Mean	2996.348	2997.058	**2994.4710661**	**2994.471066**	2994.471098
Worst	2996.348	–	**2994.4710661**	**2994.471066**	2994.471137
SD	6.4E−06	0.0E+00	**1.54E−12**	1.90E−12	1.70E−05
FES	54,350	30,000	21,000	30,000	21,000

Table 16 Statistical results of the car side impact design by different algorithms

Method	PSO	DE	GA	FA	CS	εDE-LS
Best	22.84474	22.84298	22.85653	22.84298	**22.84294**	22.84297
x_1	0.50000	0.50000	0.50005	0.50000	0.50000	0.50000
x_2	1.11670	1.11670	1.28017	1.36000	1.11643	1.11637
x_3	0.50000	0.50000	0.50001	0.50000	0.50000	0.50000
x_4	1.30208	1.30208	1.30302	1.20200	1.30208	1.30218
x_5	0.50000	0.50000	0.50001	0.50000	0.50000	0.50000
x_6	1.50000	1.50000	1.50000	1.50000	1.50000	1.50000
x_7	0.50000	0.50000	0.50000	0.50000	0.50000	0.50000
x_8	0.34500	0.34500	0.34499	0.34500	0.34500	0.34500
x_9	0.19200	0.19200	0.19200	0.19200	0.19200	0.19200
x_{10}	−19.54935	−19.54935	10.3119	8.87307	−19.54935	−19.55985
x_{11}	−0.00431	−0.00431	0.00167	−18.99808	−0.00431	−0.00046
Mean	22.89429	23.22828	23.51585	22.89376	22.85858	**22.84297**
Worst	23.21354	24.12606	26.240578	24.06623	23.25998	**22.84297**
SD	0.15017	0.34451	0.66555	0.16667	0.07612	**5.60176E−07**

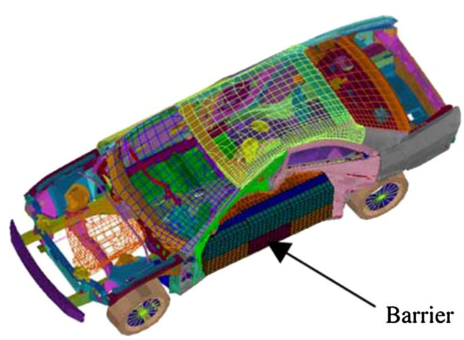


Fig. 14 The FEM model of the car side impact design

2013), 30 independent runs are conducted for εDE-LS. The experimental results are shown in Table 14 and the best results are in boldface (Tables 15, 16; Fig. 14).

The statistical results are shown that the proposed εDE-LS algorithm can achieve the best results in terms of mean, worst and SD indexes. Although the proposed algorithm cannot obtain the best results, the little gap between the best and the worst results indicates that it has the best robustness. It can be concluded that the εDE-LS is competitive and robust compared with PSO, DE GA, FA, and CS algorithm in solving this case.

Conclusion and future work

This article proposed the εDE-LS algorithm, in which a novel local search operator designed for engineering design optimization is introduced. The interaction between feasible and infeasible individuals is enhanced by applying the proposed mutation operator. By utilizing the novel mutation operator as the local search engine, we can guide the population moving towards to the feasible region more effective. The effectiveness of the proposed εDE-LS algorithm is demonstrated by 24 famous benchmark functions collected from IEEE CEC2006 special session on constrained real parameter optimization. The experimental results have suggested that εDE-LS algorithm is highly competitive in terms of accuracy and convergent speed. The performance of εDE-LS algorithm is encouraging and competitive as shown in the comparative studies with other state-of-the-art algorithms. As the effectiveness and efficiency of the proposed algorithm demonstrated above, we further conducted the experiments on engineering design optimization problems. The performance on four real-world engineering design problems and the case study have demonstrated the usefulness of the proposed algorithm in solving engineering design optimization problems.

As a part of the future direction, the performance of εDE-LS may be further improved by discovering a more efficient mutation operator. For another future direction, εDE-LS now only consider the feasible individuals in the mutation operator, we could consider how the infeasible individuals affect the searching process in the future. The future applications can be extended to deal with multi-objective constrained optimization (Mavrotas and Florios 2013), complex engineering optimization problems (Rao and Pawar 2010; Moradi and Abedini 2012; Han et al. 2015) and etc.

Acknowledgements The authors would like to thank the editor and anonymous referees whose comments helped a lot in improving this paper. The authors would also like to thank Dr. Senbong Gee and Prof. Kaychen Tan for their constructive and insightful suggestions. This research work is supported by the Natural Science Foundation of China (NSFC) under Grant Nos. 51435009, 51421062 and 61232008.

References

Arora, J. S. (1989). *Introduction to optimum design*. New York: McGraw-Hill.

Artigues, C., & Lopez, P. (2014). Energetic reasoning for energy-constrained scheduling with a continuous resource. *Journal of Scheduling*. doi:10.1007/s10951-014-0404-y.

Brajevic, I., & Tuba, M. (2013). An upgraded artificial bee colony (ABC) algorithm for constrained optimization problems. *Journal of Intelligent Manufacturing*, *24*(4), 729–740.

Deb, K. (2000). An efficient constraint handling method for genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, *186*(2–4), 311–338.

Derrac, J., Garcia, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, *1*(1), 3–18.

Domínguez-Isidro, S., Mezura-Montes, E., & Leguizamon, G. (2013). Memetic differential evolution for constrained numerical optimization problems. In *IEEE congress on evolutionary computation (CEC)*, pp. 2996–3003.

Ellis, M., & Christofides, P. D. (2014). Integrating dynamic economic optimization and model predictive control for optimal operation of nonlinear process systems. *Control Engineering Practice*, *22*, 242–251.

Flager, F., Soremekun, G., Adya, A., Shea, K., Haymaker, J., & Fischer, M. (2014). Fully constrained design: A general and scalable method for discrete member sizing optimization of steel truss structures. *Computers and Structures*, *140*(30), 55–65.

Gandomi, A. H., Yang, X. S., & Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Computers and Structures*, *29*(5), 464–483.

Gandomi, A., Yang, H. X., & Alavi, A. H. (2013). Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems. *Engineering Computing*, *29*, 17–35.

Gao, L., Zhou, Y. Z., Li, X. Y., Pan, Q. K., & Yi, W. C. (2015). Multi-objective optimization based reverse strategy with differential evolution algorithm for constrained optimization problems. *Expert Systems with Applications*, *42*(14), 5976–5987.

Gong, W., & Cai, Z. (2008). A multi-objective differential evolution algorithm for constrained optimization. In *Congress on evolutionary computation (CEC2008), Hong Kong, 1–6, June*, pp. 181–188.

Gong, W. Y., & Cai, Z. H. (2013). Differential evolution with ranking-based mutation operators. *IEEE Transactions on Cybernetics*, *46*(6), 2066–2081.

Gu, L., Yang, R. J., Cho, C. H., Makowski, M., Faruque, M., & Li, Y. (2001). Optimization and robustness for crashworthiness. *International Journal of Vehicle Design*, *26*(4), 348–360.

Han, B., Zhang, W. J., Lu, X. W., & Lin, Y. Z. (2015). On-line supply chain scheduling for single-machine and parallel-machine configurations with a single customer: Minimizing the makespan and delivery cost. *European Journal of Operational Research*, *244*(3), 704–714.

Huang, F. Z., Wang, L., & He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation*, *286*(1), 340–356.

Jia, G., Wang, Y., Cai, Z., & Jin, Y. (2013). An improved ($\mu + \lambda$)-constrained differential evolution for constrained optimization. *Information Sciences, 222*, 302–322.

Kanagaraj, G., Ponnnabalam, S. G., Jawahar, N., & Nilakantan, J. M. (2014). An effective hybrid cuckoo search and genetic algorithm for constrained engineering design optimization. *Engineering Optimization*, *46*(10), 1331–1351.

Kannan, B. K., & Kramer, S. N. (1994). An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*, *116*(2), 318–320.

Karaboga, D., & Basturk, B. (2007). Artificial bee colony optimization algorithm for solving constrained optimization problems. *LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing*, *4529*, 789–798.

Liang, J. J., Runarsson, T. P., Mezura-Montes, E., Clerc, M., Suganthan, P. N., Coello Coello, C. A., et al. (2006). *Problems definitions and evaluation criteria for the CEC' 2006 special session on constrained real-parameter optimization*. http://www.ntu.edu.sg/home/EPNSugan/cec2006/technicalreport.pdf.

Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution algorithm for constrained numerical and engineering optimization. *Applied Soft Computing*, *10*(2), 629–640.

Mavrotas, G., & Florios, K. (2013). An improved version of the augmented ε-constraint method (AUGMECON2) for finding the exact Pareto set in multi-objective integer programming problems. *Applied Mathematics and Computation*, *219*(18), 9652–9669.

Mezura-Montes, E., & Coello, C. A. C. (2011). Constraint-handling in nature-inspired numerical optimization: Past, presnt and future. *Swarm and Evolutionary Computation*, *1*, 173–194.

Mohamed, A. W., & Sabry, H. Z. (2012). Constrained optimization based on modified differential evolution algorithm. *Information Sciences*, *194*, 171–208.

Montemurro, M., Vincenti, A., & Vannucci, P. (2013). The automatic dynamic penalization method for handling constraints with genetic algorithms. *Computer Methods in Applied Mechanics and Engineering*, *256*, 70–87.

Moradi, M. H., & Abedini, M. (2012). A combination of genetic algorithm and particle swarm optimization for optimal DG location and sizing in distribution systems. *International Journal of Electrical Power & Energy Systems*, *34*(1), 66–74.

Naber, A., & Kolisch, R. (2014). MIP models for resource-constrained project scheduling with flexible resource profiles. *European Journal of Operation Research*, *239*(2), 335–348.

Nowcki, H. (1973). Optimization in pre-contract ship design. In Y. Fujida, K. Lind, & T. J. Williams (Eds.), *Computer applications in the automation of shipyard operation and ship design* (Vol. 2, pp. 327–328). New York: Elsevier.

Puzzi, S., & Carpinteri, A. (2008). A double-multiplicative dynamic penalty approach for constraint evolutionary optimization. *Structure Multidiscipline Optimization*, *35*(5), 431–445.

Ray, T., & Saini, P. (2001). Engineering design optimization using a swarm with an intelligent information sharing among individuals. *Engineering Optimization*, *33*(6), 735–748.

Rao, S. S. (1996). *Engineering optimization* (3rd ed.). Hoboken: Wiley.

Rao, R. V., & Pawar, P. J. (2010). Parameter optimization of a multi-pass milling process using non-traditional optimization algorithms. *Applied Soft Computing*, *10*(2), 445–456.

Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, *11*(4), 341–359.

Storn, R. (1999). System design by constraint adaptation and differential evolution. *IEEE Transactions on Evolutionary Computation*, *3*(1), 22–34.

Takahama, T., & Sakai, S. (2006). Constrained optimization by the ε-constrained differential evolution with gradient-based mutation and feasible elites. In *IEEE congress on evolutionary computation (CEC2006), Vancouver, BC, Canada, 16–21 July*, pp. 308–315.

Takahama, T., & Sakai, S. (2010a). Constrained optimization by the ε constrained differential evolution with an archive and gradient-based mutation. In *IEEE congress on evolutionary computation (CEC 2010)*, pp. 1–9.

Takahama, T., & Sakai, S. (2010b). Efficient constrained optimization by the ε constrained adaptive differential evolution. In *IEEE congress on evolutionary computation (CEC 2010)*, pp. 1–8.

Takahama, T., & Sakai, S. (2012). Efficient constrained optimization by the ε constrained rank-based differential evolution. In *IEEE congress on evolutionary computation (CEC 2012)*, pp. 1–8.

Takahama, T., & Sakai, S. (2013). Efficient constrained optimization by the ε constrained differential evolution with rough approximation using kernel regression. In *IEEE congress on evolutionary computation (CEC 2013)*, pp. 1334–1341.

Tessema, B., & Yen, G. (2009). An adaptive penalty formulation for constrained evolutionary optimization. *IEEE Transactions on Systems*, *39*(3), 565–578.

Tsai, J. (2005). Global optimization of nonlinear fractional programming problems in engineering design. *Engineering Optimization*, *37*(4), 399–409.

Wang, J. W., Wang, H. F., Ip, W. H., Furuta, K., & Zhang, W. J. (2013). Predatory search strategy based on swarm intelligence for continuous optimization problems. *Mathematical Problems in Engineering*. doi:10.1155/2013/749256.

Wang, L., & Li, L. (2010). An effective differential evolution with level comparison for constrained engineering design. *Structure and Multidisciplinary Optimization*, *41*, 947–963.

Wang, Y., & Cai, Z. X. (2012). Combining multi-objective optimization with differential evolution to solve constrained optimization problems. *IEEE Transactions on Evolutionary Computation*, *16*(1), 117–134.

Yi, W. C., Li, X. Y., Gao., L., & Zhou, Y. Z. (2015). ε constrained differential evolution algorithm with a novel local search operator for constrained optimization problems. In *Proceedings in adaptation, learning and optimization*, pp. 495–507.

Zhang, C., Li, X. Y., Gao, L., & Wu, Q. (2013). An improved electromagnetism-like mechanism algorithm for constrained optimization. *Expert Systems with Applications*, *40*, 5621–5634.

Zhou, Y. Z., Li, X. Y., & Gao, L. (2013). A differential evolution algorithm with intersect mutation operator. *Applied Soft Computing*, *13*(1), 390–401.

Zou, D. X., Liu, H. K., Gao, L. Q., & Li, S. (2011). A novel modified differential evolution algorithm for constrained optimization problems. *Computers & Mathematics with Applications*, *61*(6), 1608–1623.