Modified differential evolution based on global competitive ranking for engineering design optimization problems

Md. Abul Kalam Azad and Edite M.G.P. Fernandes

Algoritmi R&D Center, School of Engineering
University of Minho, 4710-057 Braga, Portugal
{akazad,emgpf}@dps.uminho.pt

Abstract. Engineering design optimization problems are formulated as large-scale mathematical programming problems with nonlinear objective function and constraints. Global optimization finds a solution while satisfying the constraints. Differential evolution is a population-based heuristic approach that is shown to be very efficient to solve global optimization problems with simple bounds. In this paper, we propose a modified differential evolution introducing self-adaptive control parameters, modified mutation, inversion operation and modified selection for obtaining global optimization. To handle constraints effectively, in modified selection we incorporate global competitive ranking which strikes the right balance between the objective function and the constraint violation. Sixteen well-known engineering design optimization problems are considered and the results compared with other solution methods. It is shown that our method is competitive when solving these problems.

Keywords: engineering design, constraints handling, ranking, differential evolution, global optimization

1 Introduction

In real-world engineering design optimization problems are formulated as large-scale mathematical programming problems involving mixed variables with linear/nonlinear objective function and constraints. The constraints can be inequality and/or equality type. The design problems can often be formulated as constrained nonlinear programming problems as follows:

$$\begin{array}{rll} \text{minimize} & f(\mathbf{x}) & \\ \text{subject to} & g_k(\mathbf{x}) \le 0 & k = 1, 2, \ldots, m_1 \\ & h_l(\mathbf{x}) = 0 & l = 1, 2, \ldots, m_2 \\ & lb_j \le x_j \le ub_j & j = 1, 2, \ldots, n \end{array} \tag{1}$$

where, $f, g_k, h_l : \mathbb{R}^n \longrightarrow \mathbb{R}$ are real valued functions with feasible set $\mathcal{F} = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{g}(\mathbf{x}) \le 0, \mathbf{h}(\mathbf{x}) = 0 \text{ and } \mathbf{lb} \le \mathbf{x} \le \mathbf{ub}\}$. $\mathbf{x}$ can be mixed types of discrete, integer and continuous. Problems (1) can be described only by nonlinear relationships, which introduce the possibility of multiple local minima. The task of the global optimization is to find a solution where the objective function obtains its most extreme value, the global minimum while satisfying the constraints.

In the last decades, many stochastic solution methods with different constraints handling techniques have been proposed to solve (1). Stochastic methods involve random sample of solutions and the subsequent manipulation of the sample to find good local (and hopefully global) minima. Stochastic methods can be based on a point-to-point search or on a population-based search. Most of the existing population-based stochastic methods try to make the solution feasible by repairing the infeasible one or penalizing an infeasible solution with the penalty function method. In penalty function method constrained problem is transformed into an unconstrained one by penalizing the objective function when the constraints are violated and then minimize the penalty function. Deb and Goyal proposed a genetic adaptive search (GeneAS) [7], Parsopoulos and Vrahatis proposed an unified particle swarm optimization (UPSO) [18] and Tomassetti proposed a cost-effective algorithm with particle swarm optimization (CPSO) [28] based on the penalty function method to solve engineering design optimization problems. But in penalty function method it is not an easy task to find an appropriate penalty parameter. Deb proposed an efficient constraints handling technique for genetic algorithm (GA) [8] based on the feasibility and dominance rules. In this technique a penalty function is used that does not require any penalty parameter and the advantage of this technique is the objective function is not evaluated for infeasible points. This technique is suitable for solving (1). Based on this technique Bernardino et al. proposed a hybrid genetic algorithm with artificial immune system (HGA) [2], Cagnina et al. proposed a simple constrained particle swarm optimizer (SPSO) [4] and Rocha and Fernandes proposed a hybrid electromagnetism-like algorithm with descent search (HEM) [23]. Another constraints handling technique is multilevel Pareto ranking based on the constraints matrix [1, 19, 21]. This technique is based on the concepts of Pareto nondominance in multiobjective optimization. Akhtar et al. proposed a socio-behavioural simulation algorithm (SBS) [1], Ray and Tai proposed an evolutionary algorithm with a multilevel pairing strategy (EA) [19] and Ray and Liew proposed a society and civilization algorithm based on the simulation of social behaviour (SCA) [21]. Runarsson and Yao proposed stochastic ranking and global competitive ranking for constrained nonlinear programming based on the evolution strategy (ES) [24, 25]. In these methods the ranking is based on the objective function as well as the constraint violation. Wang and Yin proposed a ranking selection-based particle swarm optimizer (RPSO) [29] and Y. Wang et al. proposed a hybrid evolutionary algorithm with adaptive constraints handling technique (HEA) [30] for engineering design optimization problems. He et al. proposed an improved particle swarm optimizer (IPSO) [11] for solving (1). In their method a fly-back mechnism is used to move the infeasible particles to the previous feasible region. Hedar and Fukushima proposed a filter simulated annealing method (FSA) [12] for constrained optimization problems. Here they used the filter method rather than the penalty method to handle the constraints effectively. Coello Coello used multiobjective technique by treating the constraints as objectives for single-objective evolutionary optimization [5]. Liu proposed a fuzzy proportional-derivative controller (FPDC) [17] and Lee and Geem proposed a harmony search algorithm (HS) [16] for engineering design optimization problems.

Differential evolution (DE) proposed by Storn and Price [27] is a population-based heuristic approach that is very efficient to solve derivative free global optimization problems with simple bounds. DE's performance largely depends on the amplification factor of differential variation and crossover control parameter. Hence self-adaptive control parameters ought to be implemented in DE. Sometimes it is required to improve the local search and quality of the solution. A local search starts from a candidate solution and then iteratively moves to a neighbour solution. Typically, every candidate solution has more than one neighbour solutions and the choice of movement depends only on the information about the solutions in the neighbourhood of the current one. An efficient constraints handling technique is also desirable in the solution method. In this paper, we propose a modified differential evolution (mDE) introducing self-adaptive control parameters, modified mutation, inversion operation and modified selection for solving problems (1) for obtaining global solutions. To handle the constraints effectively, in modified selection we incorporate the global competitive ranking to find the fitness of all individuals. Since the design variables can be mixed types, we give short description to handle these variables in the solution method.

The organization of this paper is as follows. We describe the constraints handling technique with global competitive ranking in Section 2. In Section 3 the modified differential evolution is outlined. Section 4 describes the experimental results and finally we draw the conclusions of this study in Section 5.

2 Constraints Handling Technique

Stochastic solution methods are mostly developed for global optimization over unconstrained problems. Finally, they are extended to constrained problems with the modification of solution procedures or by applying penalty functions. To handle the constraints effectively in engineering design optimization problems (1), firstly it is required to calculate the degree of the average constraint violation of an individual point in a population by

$$\phi(\mathbf{x}_i) = \frac{1}{m}\left(\sum_{k=1}^{m_1} \max\{0, g_k(\mathbf{x}_i)\} + \sum_{l=1}^{m_2} |h_l(\mathbf{x}_i)|\right), \quad i = 1, 2, \ldots, NP, \qquad (2)$$

where $m = m_1 + m_2$ is the total number of constraints and NP represents the number of individuals in the population. For a given value of an individual point $\mathbf{x}_i$, if all the constraints are satisfied then it returns zero, otherwise it returns the average constraint violation. In this paper, we take an individual point as a feasible one if $\phi(\mathbf{x}_i) \leq \delta$, where δ is a very small positive number. An alternative method to transform equality to inequality constraints that can be found in literature is $|\mathbf{h}(\mathbf{x})| - \delta \leq 0$. So all the constraints become inequalities.

In constrained optimization, it is very important to right balance between the objective function and the average constraint violation. Nowadays, there are many constraints handling techniques. In Table 1 some constraints handling techniques found in literature are listed.

Table 1. Different constraints handling techniques

Constraints Handling Technique	Reference
Tournament selection based on dominance and feasibility	[2, 4, 8, 22, 23]
Penalty function approach	[7, 15, 18, 28]
Multilevel Pareto ranking scheme	[1, 19–21]
Stochastic ranking & Global competitive ranking	[24, 25]
Multiobjective technique	[5]
Genotypic-based distances to move from infeasible to feasible	[6]
Fly-back mechanism from infeasible to previous feasible	[11]
Filter method	[12]
Generalized reduced gradient	[13]
Search new harmony until feasible harmony	[16]
Fuzzy proportional-derivative free controller	[17]
Ranking based selection	[29]
Adaptive constraints handling	[30]
Gradient repair & constraint fitness priority-based ranking	[32]

In the following, the constraints handling technique based on the global competitive ranking method that is considered in this paper for solving engineering design optimization problems in the general form (1) is described briefly. We applied four constraints handling techniques in our other work for constrained nonlinear optimization problems and found that the global competitive ranking method gave better performance. So this is the reason for choosing this method.

Global Competitive Ranking
Runarsson and Yao [25] proposed a constraints handling technique for constrained problems in a population-based stochastic method in order to strike the right balance between the objective function and the average constraint violation. This method is called global competitive ranking and is deterministic. In this method, an individual point is ranked by comparing it against all other members of the population. It is assumed that either the objective function or the average constraint violation is used in deciding an individual point's rank.

In this ranking method, at first the objective function f is evaluated and the average constraint violation ϕ is calculated for all the individuals in a population. Then for all individuals, the f and ϕ are sorted separately in ascending order since we consider the minimization problem and given rank. Special consideration is given to the *tied individuals*. In the case of tied individuals the same higher rank will be given. For example, suppose there are eight individuals and in ascending order based on some value, these are $\langle 6, (5, 8), 1, (2, 4, 7), 3\rangle$ (individuals in parentheses have same value). So for ranking these individuals, it becomes $I(6) = 1, I(5) = I(8) = 2, I(1) = 4, I(2) = I(4) = I(7) = 5, I(3) = 8$, where I represents the rank. After the ranking of all the individuals based on the objective function f and the average constraint violation ϕ (separately), the fitness function of an individual point is calculated by

$$\Phi(\mathbf{x}_i) = P_f \frac{I_f(i) - 1}{NP - 1} + (1 - P_f) \frac{I_\phi(i) - 1}{NP - 1} \tag{3}$$

where, $I_f(i)$ and $I_\phi(i)$ are the ranks of an individual point $\mathbf{x}_i$ based on the objective function and the average constraint violation, respectively. P_f indicates the probability that fitness is calculated based on the rank of objective function. It is clear from the above that P_f can be used easily to bias the calculation of fitness according to the objective function or the average constraint violation. The probability should take a value $0.0 < P_f < 0.5$ in order to guarantee that a feasible solution may be found. From the above fitness function, the fitness of an individual point will be $0.0 \leq \Phi \leq 1.0$. So the best individual point in a population has the lowest fitness value.

3 Modified Differential Evolution

Differential evolution (DE) is a simple yet powerful population-based evolutionary algorithm for global optimization over continuous spaces [27]. The DE algorithm has become more popular and has been used in many practical cases, mainly because it has demonstrated good convergence properties and is easy to understand. DE is a floating point encoding that creates a new candidate point by adding the weighted difference between two individuals to a third one in the population. This operation is called mutation. The mutant point's components are then mixed with the components of target point to yield the trial point. This mixing of components is referred to as crossover. In selection, a trial point replaces a target point in the following generation only if it has better or equal fitness. DE has three parameters: amplification factor of differential variation F, crossover control parameter CR, and population size NP.

It is not an easy task to set the appropriate parameters since these depend on the nature and size of the optimization problems. Hence, self-adaptive control parameters ought to be implemented. In original DE, three points are chosen randomly for mutation and the base point is then chosen at random within the three. This has an exploratory effect but it slows down the convergence of DE. In this paper, we propose a modified differential evolution (mDE) for engineering design optimization problems (1) that includes the modifications proposed by Brest et al. [3] for calculating control parameters F and CR, and Kaelo and Ali [14] for modified mutation. We also implement the inversion operation and introduce a modified selection based on the global competitive ranking that is capable to handle the constraints of problems (1) in mDE. The modified differential evolution is outlined below.

The target point of mDE is defined by $\mathbf{x}_{i,z} = (x_{i1,z}, x_{i2,z}, \ldots, x_{in,z})$, where z is the index of generation and $i = 1, 2, \ldots, NP$. NP does not change during

the optimization process. The initial population is chosen randomly and should cover the entire component spaces.
Self-adaptive control parameters: We use self-adaptive control parameter for F and CR proposed by Brest et al. [3] by generating a different set (F_i, CR_i) for each point in the population. The new control parameters for next generation $F_{i,z+1}$ and $CR_{i,z+1}$ are calculated by

$$\begin{aligned} F_{i,z+1} &= \begin{cases} F_l + \lambda_1 \times F_u, & \text{if } \lambda_2 < \tau_1 \\ F_{i,z}, & \text{otherwise} \end{cases} \\ CR_{i,z+1} &= \begin{cases} \lambda_3, & \text{if } \lambda_4 < \tau_2 \\ CR_{i,z}, & \text{otherwise,} \end{cases} \end{aligned} \tag{4}$$

where $\lambda_k \sim \mathrm{U}[0,1], k = 1, \ldots, 4$ and $\tau_1 = \tau_2 = 0.1$ represent probabilities to adjust parameters F_i and CR_i, respectively. $F_l = 0.1$ and $F_u = 0.9$, so the new $F_{i,z+1}$ takes a value from $[0.1, 1.0]$ in a random manner. The new $CR_{i,z+1}$ takes a value from $[0,1]$. $F_{i,z+1}$ and $CR_{i,z+1}$ are obtained before the mutation is performed. So, they influence the mutation, crossover and selection operations of the new point $\mathbf{x}_{i,z+1}$.
Modified mutation: We use the mutation proposed by Kaelo and Ali [14] in mDE. After choosing three points randomly the best point based on the fitness value is selected for the base point and the remaining two points are used as differential variation, i.e., for each target point $\mathbf{x}_{i,z}$, a mutant point is created according to

$$\mathbf{v}_{i,z+1} = \mathbf{x}_{r_3,z} + F_{i,z+1}(\mathbf{x}_{r_1,z} - \mathbf{x}_{r_2,z}), \tag{5}$$

where r_1, r_2, r_3 are randomly chosen from the set $\{1, 2, \ldots, NP\}$, mutually different and different from the running index i and r_3 is the index with the best fitness value. This modification has a local effect when the points of the population form a cluster around the global minimizer. In mDE, we also propose a modification in the above mutation. After every B generations the best point found so far is used as the base point and two randomly chosen points are used as differential variation, i.e., $\mathbf{v}_{i,z+1} = \mathbf{x}_{\text{best}} + F_{i,z+1}(\mathbf{x}_{r_1,z} - \mathbf{x}_{r_2,z})$. These modifications allow mDE to maintain its exploratory feature as well as explore the region around each best and at the same time expedite the convergence.
Crossover: In order to increase the diversity of the perturbed component points, crossover is introduced. To this end, the crossover point $\mathbf{u}_{i,z+1}$ is formed, where

$$u_{ij,z+1} = \begin{cases} v_{ij,z+1} & \text{if } (r_j \le CR_{i,z+1}) \text{ or } j = z_i \\ x_{ij,z} & \text{if } (r_j > CR_{i,z+1}) \text{ and } j \ne z_i \end{cases} \tag{6}$$

In (6), $r_j \sim \mathrm{U}[0,1]$ performs the mixing of jth component of points, z_i is randomly chosen from the set $\{1, 2, \ldots, n\}$ and ensures that $\mathbf{u}_{i,z+1}$ gets at least one component from $\mathbf{v}_{i,z+1}$.
Inversion: Since in mDE a point has n-dimensional real components, inversion can easily be applicable. With the inversion probability ($p_{\text{inv}} \in [0,1]$), two positions are chosen on the point $\mathbf{u}_i$, the point is cut at those positions, and the cut segment is reversed and reinserted back into the point to create the trial point

$\mathbf{u}'_i$. In practice, mDE with the inversion has been shown to give better results than those obtained without the inversion.
Bounds check: When generating the mutant point, some components can be generated outside the search spaces. So, in mDE after inversion the bounds of each component should be checked.
Modified selection: After calculating the fitness value of all target and trial points all together, in order to decide whether or not it should become a member of generation $z+1$, the trial point $\mathbf{u}'_{i,z+1}$ is compared to the target point $\mathbf{x}_{i,z}$ using the greedy criterion in the following way

$$\mathbf{x}_{i,z+1} = \begin{cases} \mathbf{u}'_{i,z+1} & \text{if } \Phi(\mathbf{u}'_{i,z+1}) \leq \Phi(\mathbf{x}_{i,z}) \\ \mathbf{x}_{i,z} & \text{otherwise.} \end{cases}$$

Termination condition: Let $G_{\max}$ be the maximum number of generations. If $f_{\max,z}$ and $f_{\min,z}$ are the maximum and minimum objective function values attained at z then our mDE algorithm terminates if ($z > G_{\max}$ or $(f_{\max,z} - f_{\min,z}) \leq \eta$), for a very small positive number η.

In mDE we also incorporate the elitism to preserve the best point found so far throughout the entire generations. Since engineering design optimization problems have mixed variables, so we are having attention to handle discrete and integer variables. For discrete variables we randomly generate values from an appropriate discrete set in initialization and mutation. For integer variables we use rounding off to the nearest integer at evaluation stages.

4 Experimental Results

We code mDE in C with AMPL [9] interfacing and compile with Microsoft Visual Studio 9.0 compiler in a PC having 2.5 GHz Intel Core 2 Duo processor and 4 GB RAM. We set the value of parameters $NP = \min(100, 10n)$, $B = 10$, $p_{\text{inv}} = 0.05$, $\delta = 10^{-5}$, $P_f = 0.45$ and $\eta = 10^{-6}$. We consider 16 benchmark problems found in literature. The first problem is a classical benchmark problem in constrained nonlinear optimization. Remaining 15 engineering design optimization problems are commonly used for test problems. 30 independent runs for all problems were performed and the obtained results were compared with other solution methods found in literature. Values in “bold” in tables represent the best obtained in the listed comparisons. We model six of the selected problems in AMPL modeling language. These and the remaining ten problems can be made available from http://www.norg.uminho.pt/emgpf/problems.htm.

Himmelblau’s Function
This is a common benchmark function for constrained nonlinear optimization problems proposed by Himmelblau [13]. This problem has five design variables and six inequality constraints and details of the problem are described in [8, 11, 24]. For fair comparison, we set $G_{\max} = 3000$ and maximum number of function evaluations, $nfe_{\max} = 90000$ according to He et al. [11] for termination condition rather than termination condition discussed in Section 3. We compared

Table 2. Comparative results of Himmelblau's function

Values	Best solution found					
	mDE	GA [8]	IPSO [11]	GRG [13]	HS [16]	ES [24]
x_1	78.000000	–	78.000000	78.000000	78.000	–
x_2	33.000000	–	33.000000	33.000000	33.000	–
x_3	29.995123	–	29.995256	29.995256	29.995	–
x_4	45.000000	–	45.000000	45.000000	45.000	–
x_5	36.775724	–	36.775813	36.775813	36.776	–
$f(\mathbf{x})$	**-30665.587237**	-30665.539	-30665.539	-30665.539	-30665.500	-30665.539

– Not available

the obtained results from our mDE with other solution methods such as GA, IPSO, generalized reduced gradient, GRG [13], HS and ES. The comparative results based on the best objective function value are shown in Table 2. It is shown that our mDE is rather competitive for solving Himmelblau's function.

Heat Exchanger Design

The heat exchanger design problem is also a common benchmark function for constrained nonlinear optimization problems, and is described in [8, 16, 24]. This problem has eight design variables and six inequality constraints. We set $G_{\max} = 2000$ and $nfe_{\max} = 150000$. We compared the obtained results from our mDE with GA, HS, FPDC, HEM and ES. The comparative results based on the best objective function value are shown in Table 3. From the table, it is shown that our mDE is rather competitive when solving this problem.

Table 3. Comparative results of heat exchanger design problem

Values	Best solution found					
	mDE	GA [8]	HS [16]	FPDC [17]	HEM [23]	ES [24]
x_1	579.315	–	500.004	951.8	607.211	–
x_2	1361.100	–	1359.311	1529.5	1560.399	–
x_3	5108.084	–	5197.960	4807.3	5303.680	–
x_4	182.018	–	174.726	206.6	173.324	–
x_5	295.647	–	292.082	307.9	287.951	–
x_6	217.982	–	224.705	193.4	205.448	–
x_7	286.372	–	282.645	298.7	284.110	–
x_8	395.647	–	392.082	407.8	387.925	–
$f(\mathbf{x})$	**7048.499**	7060.221	7057.274	7288.8	7471.290	7054.316

– Not available

Welded Beam Design

The design of a welded beam is the most commonly used test problem for engineering design optimization problems to check the effectiveness of a solution method. The objective is to minimize the cost of a welded beam, subject to

Table 4. Comparative results of welded beam design problem

Values	Best solution found						
	mDE	HGA [2]	IPSO [11]	FSA [12]	SCA [21]	HEM [23]	HEA [30]
x_1	0.244429	0.244386	0.244369	0.244353	0.244438	0.243532	0.244369
x_2	6.215393	6.218304	6.217520	6.217592	6.237967	6.167268	6.217518
x_3	8.291471	8.291165	8.291471	8.293904	8.288576	8.377163	8.291477
x_4	0.244369	0.244387	0.244369	0.244353	0.244566	0.243876	0.244369
$f(\mathbf{x})$	**2.380810**	2.381217	2.380956	2.381065	2.385435	2.386269	2.380957

the constraints on the shear stress, bending stress, buckling load on the bar, end deflection of the beam and side constraints. The problem has four design variables and seven inequality constraints, and is described in [1, 11]. We set $G_{\max} = 1000$ and $nfe_{\max} = 30000$ as in [11]. We compared the obtained results from our mDE with other solution methods such as HGA, IPSO, FSA, SCA, HEM and HEA. The comparative results are shown in Table 4. The best solution obtained by mDE is better than other solutions.

Spring Design 1

This is a real-world optimization problem involving discrete, integer and continuous design variables. The objective is to minimize the volume of a compression spring under static loading. The design problem has three variables and eight

Table 5. Comparative results of spring design 1 problem

Values	Best solution found					
	mDE	GeneAS [7]	IPSO [11]	DE [15]	in [26]	RPSO [29]
x_1	0.283	0.283	0.283	0.283	0.283	0.283
x_2	1.223021	1.226	1.223041	1.223041	1.180701	1.223041
x_3	9	9	9	9	10	9
$f(\mathbf{x})$	**2.65852**	2.665	2.65856	2.65856	2.7995	2.65856

inequality constraints [7, 11, 15]. We set $G_{\max} = 500$ and $nfe_{\max} = 15000$ [11]. We compared the obtained results from our mDE with GeneAS, IPSO, differential evolution, DE [15], solution method proposed in [26] and RPSO. The comparative results are shown in Table 5. The best solution obtained by mDE is better than other solutions.

Spring Design 2

This problem aims to minimize the weight of a tension/compression spring. This problem has three continuous variables and four constraints [2, 11, 28]. We set $G_{\max} = 500$ and $nfe_{\max} = 15000$ [11]. The comparative results are shown in Table 6 where the best solution obtained by mDE is better than other solutions.

Table 6. Comparative results of spring design 2 problem

Values	Best solution found						
	mDE	HGA [2]	IPSO [11]	FSA [12]	HEM [23]	CPSO [28]	HEA [30]
x_1	0.051689	0.051661	0.051690	0.051743	0.051557	0.051644	0.051689
x_2	0.356734	0.356032	0.356750	0.358005	0.353534	0.355632	0.356729
x_3	11.287348	11.329555	11.287126	11.213907	11.479520	11.353040	11.288294
$f(\mathbf{x})$	**0.012664**	0.012666	0.012665	0.012665	0.012667	0.012665	0.012665

Pressure Vessel Design
The design of a cylindrical pressure vessel with both ends capped with a hemispherical head is to minimize the total cost of fabrication [1, 11]. The problem has four design variables and four inequality constraints. This is a mixed variables problem where x_1 and x_2 are discrete of integer multiples of 0.0625 inch., and other two are continuous. We set $G_{\max} = 1000$ and

Table 7. Comparative results of pressure vessel design problem

Values	Best solution found						
	mDE	SBS [1]	HGA [2]	IPSO [11]	HEM [23]	CPSO [28]	RPSO [29]
x_1	0.8125	0.8125	0.8125	0.8125	0.8125	0.8125	0.8125
x_2	0.4375	0.4375	0.4375	0.4375	0.4375	0.4375	0.4375
x_3	42.1000	41.9768	42.0950	42.0984	42.0700	42.0984	42.0984
x_4	176.6173	182.2845	176.6797	176.6366	177.3762	176.6366	176.6366
$f(\mathbf{x})$	**6059.525**	6171.000	6060.138	6059.714	6072.232	6059.714	6059.714

$nfe_{\max} = 30000$ [11]. The comparative results from our mDE, with SBS, HGA, IPSO, HEM, CPSO and RPSO, are shown in Table 7. From the table, mDE is competitive with other methods.

Speed Reducer Design
The weight of the speed reducer is to be minimized subject to the constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stress in the shafts. See description in [1, 2, 4, 28]. There are seven variables and 11 inequality constraints. This is a mixed variables problem, where x_3 is integer (number of teeth) and others are continuous. We set $G_{\max} = 500$ and $nfe_{\max} = 35000$. The comparative results are shown in Table 8 where the best solution obtained by mDE is rather competitive than other solutions.

Three-Bar Truss Design
The design of a three-bar truss is to minimize the volume of the truss subject to the stress constraints [21]. This problem has two design variables representing the cross-sectional areas of two bars (two identical of three-bar) and three inequality constraints. We set $G_{\max} = 500$ and $nfe_{\max} = 10000$. The comparative

Table 8. Comparative results of speed reducer design problem

Values	Best solution found						
	mDE	SBS [1]	HGA [2]	SPSO [4]	HGA [6]	HEM [23]	CPSO [28]
x_1	3.499615	3.506122	3.500000	3.500000	3.500000	3.500062	3.500000
x_2	0.700000	0.700006	0.700000	0.700000	0.700000	0.700000	0.700000
x_3	17	17	17	17	17	17	17
x_4	7.300000	7.549126	7.300003	7.300000	7.300008	7.367704	7.300000
x_5	7.715320	7.859330	7.715322	7.800000	7.715322	7.731763	7.800000
x_6	3.350215	3.365576	3.350215	3.350214	3.350215	3.351341	3.350215
x_7	5.286654	5.289773	5.286654	5.286683	5.286655	5.286937	5.286683
$f(\mathbf{x})$	**2994.320**	3008.080	2994.471	2996.348	2994.342	2995.804	2996.348

results are shown in Table 9 where the best solution obtained by mDE is rather competitive than other solutions.

Table 9. Comparative results of three-bar truss design problem

Values	Best solution found				
	mDE	FPDC [17]	SCA [21]	HEM [23]	HEA [30]
$x_1(=x_3)$	0.788663	0.7511	0.788621	0.788764	0.788680
x_2	0.408242	0.5262	0.408401	0.408000	0.408234
$f(\mathbf{x})$	**263.8919**	265.07	263.8958	263.8960	263.8958

Hydrostatic Thrust Bearing Design

The thrust bearing design problem aims to minimize power loss associated with the bearing. This problem consists of four design variables and seven constraints, and is described in [5, 7, 11]. We set $G_{\max} = 3000$ and $nfe_{\max} = 90000$ according to [11]. The comparative results from different solution methods are shown in Table 10. The best solution obtained by mDE is better than other solutions.

Table 10. Comparative results of hydrostatic thrust bearing design problem

Values	Best solution found				
	mDE	GA [5]	GeneAS [7]	BGA [7]	IPSO [11]
x_1	5.955780	6.271	6.778	7.077	5.956869
x_2	5.389013	12.901	6.234	6.549	5.389175
$x_3(\times 10^{-6})$	5.396500	5.605	6.096	6.619	5.402133
x_4	2.277653	2.938	3.809	4.849	2.301547
$f(\mathbf{x})$	**1631.1716**	1950.2860	2161.6000	2295.1000	1632.2149

Tubular Column Design

The design of a tubular column aims at minimizing the cost of fabrication [17].

This problem has two design variables with two inequality constraints. We set $G_{\max} = 500$ and $nfe_{\max} = 10000$. The comparative results are shown in Table 11. It is shown that the best result obtained by mDE is slightly greater than that of FPDC.

Table 11. Comparative results of tubular column design problem

Method	x_1	x_2	$f(\mathbf{x})$
mDE	5.4512	0.2919	26.5311
FPDC [17]	5.4507	0.2920	**26.5310**
HEM [23]	5.4511	0.2920	26.5323

Tanker Fleet Design

The design of a tanker fleet is to minimize the total cost, which includes the cost of fuel, the cost of hull and the cost of machinery. The details description of this problem can be found in [19]. This is a mixed variables problem having nine design variables and 19 inequality constraints. Variable x_5 is integer (number of ships). We set $G_{\max} = 500$ and $nfe_{\max} = 40000$. We compared the obtained results from mDE with EA and HEM. The comparative results are shown in Table 12. From table it is shown that mDE gave better result of $14,514,897.18$ although the number of ships is 7.

Table 12. Comparative results of tanker fleet design problem

Method	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	$f(\mathbf{x})$
mDE	41.0473	20.2914	78530.3591	284.0244	7	12.3864	0.7284	16.8862	88817.2680	**14,514,897.18**
EA [19]	27.6300	12.0900	15200.0000	165.2000	44	7.4060	0.9280	10.9100	22660.0000	135,500,000.00
HEM [23]	48.3175	19.9585	79071.9980	279.4188	8	12.4186	0.7347	14.4925	118646.5600	21,216,265.00

Gear Train Design

A compound gear train is to be designed to minimize the error between the obtained gear ratio and a required gear ratio of 1/6.931 subject to the ranges on gear teeth [7, 18]. This problem has four design variables and all are strictly integers. We set $G_{\max} = 1000$ and $nfe_{\max} = 40000$. The comparative results are shown in Table 13 where mDE is rather competitive when solving this problem.

I-Beam Design

The design of a simply supported I-beam is a multiobjective optimization problem where the objectives are to minimize the cross-sectional area and the static deflection subject to the stress constraint. This problem has four design

Table 13. Comparative results of gear train design problem

Method	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
mDE	49	19	16	43	2.700857E-12
GeneAS [7]	49	16	19	43	**2.7E-12**
UPSO [18]	–	–	–	–	2.70085E-12
HEM [23]	49	19	16	43	2.700857E-12

– Not available

Table 14. Comparative results of I-beam design problem

Method	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
mDE	80.0000	50.0000	4.4221	5.0000	**809.5464**
PSO [22]	–	–	–	–	127.95–829.57†
HIA [31]	–	–	–	–	127.41–833.04†

– Not available †Range of values in the Pareto front

variables and one inequality constraint, and is described in [22, 31]. We dropped the static deflection objective function and added this to the constraints with maximum allowable deflection 0.006 cm taken from the minimum deflection of the Pareto front [22, 31]. So this problem became a single objective optimization problem. We set $nfe_{\max} = 10000$ as in [22]. We compared the obtained results from our mDE with PSO and hybrid immune algorithm, HIA [31]. The comparative results are shown in Table 14. From the Pareto front, in [22] with minimum deflection of 0.006 cm the cross-sectional area is 829.57 cm^2 and in [31] the cross-sectional area is 833.04 cm^2 whereas by mDE it is 809.5464 cm^2.

Disc Brake Design

This problem deals with the design of a multiple disc brake, and is described in [20] and is a multiobjective optimization problem. The objectives of the design are to minimize the mass of the brake and to minimize the stopping time. This problem has four design variables and five inequality constraints. We dropped the stopping time objective function and added this to the constraints with maximum allowable stopping time 32.0 sec. taken from the maximum stopping time of the Pareto front [20]. We set $G_{\max} = 1000$ and $nfe_{\max} = 30000$. We compared the obtained results from our mDE with swarm metaphor, SM [20] and HEM. The comparative results are shown in Table 15. It is shown that mDE is rather competitive when solving this problem.

Table 15. Comparative results of disc brake design problem

Method	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
mDE	55.00	75.00	1764.42	2.00	**0.1274**
SM [20]	–	–	–	–	0.2–2.7†
HEM [23]	55.00	75.00	1862.87	2.00	**0.1274**

– Not available †Range of values in the Pareto front

Four-Bar Truss Design
The design of a four-bar truss is a multiobjective optimization problem where the objectives are to minimize the volume of the truss and displacement subject to the stress constraints on four design variables which represent the cross-sectional areas [20]. We dropped the displacement objective function and added this to the constraints with maximum allowable displacement 0.04 cm. We set $G_{\max} = 1000$ and $nfe_{\max} = 30000$. The comparative results are shown in Table 16. It is shown that mDE is also capable of solving this problem.

Table 16. Comparative results of four-bar truss design problem

Method	x_1	x_2	x_3	x_4	$f(\mathbf{x})$
mDE	1.000000	1.414214	1.414214	1.000000	**1400.000**
SM [20]	–	–	–	–	1400–3000[†]
HEM [23]	1.000003	1.414214	1.414214	1.000000	1400.001

– Not available [†]Range of values in the Pareto front

Ceramic Grinding Design
The design of a ceramic grinding wheel is a maximization problem. The objective is to maximize the material removal rate, subject to a set of constraints comprising surface roughness, number of flaws and input variables [10, 32]. We set $G_{\max} = 300$ and $nfe_{\max} = 9000$. We compared the obtained results from mDE with GA [10] and hybrid particle swarm optimizer, HPSO [32]. The comparative results are shown in Table 17. It is shown that mDE outperforms other two methods when solving this maximization problem.

Table 17. Comparative results of ceramic grinding design problem

Method	x_1	x_2	x_3	$f(\mathbf{x})$
mDE	8.4888	12.1953	500.0000	**103.5237**
GA [10]	8.22	12.05	494.12	99.05
HPSO [32]	8.4878	12.1946	500.0000	103.5048

From the above discussion it is shown that in almost all engineering design optimization problems our mDE is competitive with other solution methods.

5 Conclusions

In this paper, to make the DE more efficient to handle the constraints in engineering design optimization problems, a modified differential evolution (mDE) algorithm is proposed. The modifications focus on self-adaptive control parameters and modified mutation. Inversion operation is also implemented in the proposed mDE. To handle the constraints effectively, all target and trial points are ranked all together using the global competitive ranking based on the objective function and the average constraint violation for competing in selection operation to decide which point wins for next generation population. This ranking strikes the right balance between the objective function and the constraint violation for obtaining global optimization while satisfying the constraints. Handling of mixed variables are also presented.

To test the effectiveness of proposed mDE, 16 well-known design problems have been considered. A comparison of the obtained results by mDE based on the best objective function value with the results by other existing methods reported in literature is presented. It is shown that in almost all design problems our mDE is competitive with other solution methods. In future we will focus on exact mixed variables handling techniques to include in the proposed mDE.

Acknowledgments. This work is supported by FCT (Fundação para a Ciência e a Tecnologia) and Ciência 2007, Portugal. We thank three anonymous referees for their valuable comments to improve this paper.

References

1. Akhtar, S., Tai, K., Ray, T.: A socio-behavioural simulation model for engineering design optimization. Eng. Optim. 34, 341–354 (2002)
2. Bernardino, H.S., Barbosa, H.J.C., Lemonge, A.C.C.: A hybrid genetic algorithm for constrained optimization problems in mechanical engineering. In: IEEE Congress on Evolutionary Computation, pp. 646–653 (2007)
3. Brest, J., Greiner, S., Bošković, B., Mernik, M., Žumer, V.: Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems. IEEE Trans. Evol. Comput. 10, 646–657 (2006)
4. Cagnina, L.C., Esquivel, S.C., Coello Coello, C.A.: Solving engineering optimization problems with the simple constrained particle swarm optimizer. Inform. 32(3), 319–326 (2008)
5. Coello Coello, C.A.: Treating constraints as objectives for single-objective evolutionary optimization. Eng. Optim. 32(3), 275–308 (2000)
6. Coello Coello, C.A., Cortés, N.C.: Hybridizing a genetic algorithm with an artificial immune system for global optimization. Eng. Optim. 36(5), 607–634 (2004)
7. Deb, K., Goyal, M.: Optimizing engineering designs using a combined genetic search. In: Back, I.T. (ed.) 7th International Conference on Genetic Algorithms, pp. 512–528 (1997)
8. Deb, K.: An efficient constraint handling method for genetic algorithms. Comput. Methods Appl. Mech. Eng. 186, 311–338 (2000)
9. Fourer, R., Gay, D.M., Kernighan, B.W.: AMPL: A Modelling Language for Matematical Programming. Boyd & Fraser Publishing Co., Massachusets (1993)
10. Gopal, A.V., Rao, P.V.: The optimisation of the grinding of silicon carbide with diamond wheels using genetic algorithms. Int. J. Adv. Manuf. Technol. 22, 475–480 (2003)
11. He, S., Prempain, E., Wu, Q.H.: An improved particle swarm optimizer for mechanical design optimization problems. Eng. Optim. 36(5), 585–605 (2004)
12. Hedar, A.-R., Fukushima, M.: Derivative-free filter simulated annealing method for constrained continuous global optimization. J. Glob. Optim. 35, 521–549 (2006)

13. Himmelblau, D.M.: Applied Nonlinear Programming. McGraw-Hill, New York (1972)
14. Kaelo, P., Ali, M.M.: A numerical study of some modified differencial evolution algorithms. Eur. J. Oper. Res. 169, 1176–1184 (2006)
15. Lampinen, J., Zelinka, I.: Mixed integer-discrete-continuous optimization by diffrential evolution. In: Proceedings of the 5th International Conference on Soft Computing, pp. 71–76 (1999)
16. Lee, K.S., Geem, Z.W.: A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Comput. Methods Appl. Mech. Eng. 194, 3902–3933 (2005)
17. Liu, T.-C.: Developing a fuzzy proportional-derivative controller optimization engine for engineering optimization problems. PhD Thesis, Chapter 6 (2006), http://grc.yzu.edu.tw/OptimalWeb/Content.aspx?CatSubID=129
18. Parsopoulos, K.E., Vrahatis, M.N.: Unified particle swarm optimization for solving constrained engineering optimization problems. In: Wang, L. Chen, K. Ong, Y.S. (eds.) ICNC 2005, LNCS, vol. 3612 pp. 582–591, Springer-Verlag (2005)
19. Ray, T., Tai, K.: An evolutionary algorithm with a multilevel pairing strategy for single and multiobjective optimization. Found. Comput. Decis. Sci. 26(1), 75–98 (2001)
20. Ray, T., Liew, K.M.: A swarm metaphor for multiobjective design optimization. Eng. Optim. 34(2), 141–153 (2002)
21. Ray, T., Liew, K.M.: Society and civilization: An optimization algorithm based on the simulation of social behavior. IEEE Trans. Evol. Comput. 7(4), 386–396 (2003)
22. Reddy, M.J., Kumar, D.N.: An efficient multi-objective optimization algorithm based on swarm intelligence for engineering design. Eng. Optim. 39(1), 49–68 (2007)
23. Rocha, A.M.A.C., Fernandes, E.M.G.P.: Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems. Int. J. Comput. Math. 86(10), 1932–1946 (2009)
24. Runarsson, T.P., Yao, X.: Stochastic ranking for constrained evolutionary optimization. IEEE Tran. Evol. Comput. 4(3), 284–294 (2000)
25. Runarsson, T.P., Yao, X.: Constrained evolutionary optimization – the penalty function approach. In: Sarker, R., Mohammadian, M., Yao, X. (eds.) Evolutionary Optimization: International Series in Operations Research and Management Science, pp. 87–113 (2003)
26. Sandgren, E.: Nonlinear integer and discrete programming in mechanical design optimization. J. Mech. Des. (ASME) 112, 223–229 (1990)
27. Storn R., Price, K.: Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. J. Glob. Optim. 11, 341–359 (1997)
28. Tomassetti, G.: A cost-effective algorithm for the solution of engineering problems with particale swarm optimization. Eng. Optim. 42(5), 471–495 (2010)
29. Wang, J., Yin, Z.: A ranking selection-based particle swarm optimizer for engineering design optimization problems. Struct.Multidisc. Optim. 37(2), 131–147 (2007)
30. Wang, Y., Cai, Z., Zhou, Y., Fan, Z.: Constrained optimization based on hybrid evolutionary algorithm and adaptive constraint-handling technique. Struct. Multidisc. Optim. 37, 395–413 (2009)
31. Yildiz, A.R.: A novel hybrid immune algorithm for global optimization in design and manufacturing. Robotics and Computer-Integrated Manuf. 25, 261–270 (2009)
32. Zahara, E., Hu, C.-H.: Solving constrained optimization problems with hybrid particle swarm optimization. Eng. Optim. 40(11), 1031–1049 (2008)