International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 12 (2018) pp. 10845-10854

An Efficient Differential Evolution for Engineering Design Problems

Raghav Prasad Parouha

Assistant Professor, Department of Mathematics,
Indira Gandhi National Tribal University Amarkantak
Madhya Pradesh – 484887 India.

Abstract

Many variants of Differential Evolution (DE) algorithms exist in literature to solve Engineering Design Problems. However, the performance of DE is highly affected by the inappropriate choice of its operators like mutation and crossover. Moreover, in general practice, DE does not employ any strategy of memorizing the so-far-best results obtained in the initial part of the previous cycle. In this paper a 'Memory based DE (MBDE)' propose which having two 'swarm operators'. These operators are based on the personal best and global best mechanism of particle swarm optimization (PSO). The feasibility of the proposed algorithm is tested on constrained and unconstrained engineering design problems and it is compared with well settled state-of-the-art approaches. Numerical, statistical and graphical analyses reveal the competence of proposed algorithm.

Keywords: Differential Evolution, Mutation, Crossover, Engineering Design Problem.

INTRODUCTION

An optimization problem, especially 'Engineering Design Problems' (EDPs) is a challenging area of research. There are many studies on solving EDPs using evolutionary algorithms (EAs). Among them, Differential Evolution (DE) [1] is an efficient and popular ingredient of EAs. The DE has many advantages like easy implementation, reasonably faster, robust and exhibits effective global search ability [2-5]. Therefore, it has been successfully applied to solve many real world problems such as fuzzy clustering of image pixel [6], economic load dispatch [7], mechanical engineering design problem [8], etc. However, DE has its individual shortcomings like sometimes the solution gets stacked in local optimum which leads to premature convergence [9, 10]. Also, DE does not guarantee to reach at global optimal solution in a finite time interval [11, 12]. Therefore, a number of attempts are made in the literature [13-20] to improve the strength of DE. A detailed survey on the variants of DE can be found in [13, 14].

In order to improve the robustness of DE, many varieties of mutation strategies and crossover operators are being proposed in [2, 9, 13, 14, 15, 21, 22]. Basically, DE is much sensitive to the choice of the mutation and crossover operator. It is very difficult to recommend a fixed set of parameters used in mutation and crossover for different problems [2, 13-15, 21, 22]. On the other hand, inappropriate choice of mutation and crossover operator may lead to premature convergence, stagnation and/or wastage of computational time [2, 23-26].

In fact, DE has no mechanism to memorize the previous obtained results but it uses only the global information about the search space [27-29]. Inspired by above fact a memory based differential evolution proposed in this paper where two novel operators (swarm mutation and swarm crossover) were introduced under the PSO environment. The reminder of this paper is organized as follows: Section 2 presents overview of basic DE. Section 3 presents a detailed description of the proposed algorithm, Section 4 presents result and discussion. Section 5 draws the conclusion with some future scope.

OVERVIEW OF BASIC DIFFERENTIAL EVOLUTION

Differential Evolution (DE) is an Evolutionary Algorithm (EA) proposed by Storn and Price in 1995 [1]. The outline of the classical DE may be given as follows.

Initialization

Initialize a population of NP target vectors (parents) $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD}); i = 1, 2, \ldots, NP$, is randomly generated within user-defined bounds, where D is the dimension of the optimization problem.

Mutation

Let $x_i(t) = x_{i1}(t), x_{i2}(t), \ldots, x_{iD}(t)$ be the 'i^{th}' individuals at 't^{th}' generation. A mutant vector $v_i(t+1) = (v_{i1}(t+1), v_{i2}(t+1), \ldots, v_{iD}(t+1))$ is generated as follows.

$$v_i(t+1) = x_{r_1} + F \times \left(x_{r_2}(t) - x_{r_3}(t)\right)$$

where, $r_1 \neq r_2 \neq r_3 \neq i$ and $F \in [0, 1]$ is the mutation factor.

Crossover

The target vector $x_i(t)$ and the mutant vector $v_i(t+1)$ create a new trial vector $u_i(t+1) = (u(t+1), u_{i2}(t+1), \ldots, u_{iD}(t+1))$ as follows.

$$u_{ij}(t+1) = \begin{cases} v_i(t+1); & \text{if } rand(0,1) \le CR \text{ or } j = rand(i) \\ x_i(t) \quad ; & \text{if } rand(0,1) > CR \text{ or } j \neq rand(i) \end{cases}$$

where j and $rand(i) \in (1, 2, \ldots, D)$, $CR \in [0, 1]$ is the crossover constant.

Selection

Selection operates by comparing the individuals' fitness to generate the next generation population.

$$x_{ij}(t+1) = \begin{cases} u_i(t+1); & \text{if } f\big(u_i(t+1)\big) \le f\big(x_i(t)\big) \\ x_i(t) & ; \text{ otherwise} \end{cases}$$

The cyclic implementation of mutation, crossover and selection is continued till it meets with the pre-defined stopping criterion.

PROPOSED METHOD

Motivated by the advantages and disadvantages of DE, and above observations, in this present study 'Memory Based Differential Evolution (MBDE)' proposed. where the mutation and crossover are termed as '*swarm mutation*' and '*swarm crossover*' because of these operator based on p^{best} and g^{best} mechanism of PSO [30]. The proposed operators are explained in the following section.

Swarm mutation

Let $x_i(t) = x_{i1}(t), x_{i2}(t), \ldots, x_{iD}(t)$ is the target vector and $p_i(t)$ is the personal best position vector, in the current generation 't'. Then a mutant vector (i.e. perturbed vector) $v_i(t) = v_{i1}(t), v_{i2}(t), \ldots, v_{iD}(t)$ is generated by 'Swarm Mutation' as follows.

$$v_i(t) = x_i(t) + \left|\frac{f\left(p_i^{best}(t)\right)}{f\left(x_i^{worst}(t)\right)}\right| \times \left(p_i^{best}(t) - x_i(t)\right) + \left|\frac{f\left(g^{best}(t)\right)}{f\left(x_i^{worst}(t)\right)}\right| \times \left(g^{best}(t) - x_i(t)\right) \quad (1)$$

where $p_i^{best}(t)$ is the personal and $g^{best}(t)$ is the global best position of the vector $p_i(t)$ respectively, $f\left(p_i^{best}(t)\right)$ is the personal best and $f\left(g^{best}(t)\right)$ is the global best function value of the vector $p_i(t)$ respectively, $f\left(x_i^{worst}(t)\right)$ is the worst function value of vector $x_i(t)$, in the current generation 't'. Whenever $f\left(x_i^{worst}(t)\right) = 0$ it will be replaced by a large positive constant 'r', in order to impact a small perturbation to $x_i(t)$. Clearly, each of the ratio-coefficient of the terms $\left(p_i^{best}(t) - x_i(t)\right)$ and $\left(g^{best}(t) - x_i(t)\right)$ generates a real constant factor between $[0, 1]$ that controls the amplification of the differential variation.

Swarm crossover

To generate a new trial vector $\boldsymbol{u_i(t)} = \boldsymbol{u_{i1}(t)}, \boldsymbol{u_{i2}(t)}, \ldots, \boldsymbol{u_{iD}(t)}$, 'Swarm Crossover' works as follows.

$$u_{ij}(t) = \begin{cases} v_i(t) + rand(0,1) + \left(g^{best}(t) - p_i^{best}(t)\right); \\ \qquad \text{if } rand(0,1) \le CR \text{ or } j = rand(i) \\ x_i(t) + rand(0,1) + \left(g^{best}(t) - p_i^{best}(t)\right); \\ \qquad \text{if } rand(0,1) > CR \text{ or } j \ne rand(i) \end{cases} \quad (2)$$

where $v_i(t)$ is the mutant vector, $x_i(t)$ is the target vector, j and $rand(i) \in (1, 2, \ldots, D)$, $CR \in [0, 1]$ is the crossover constant.

Steps of proposed algorithm

Steps of the proposed MBDE presented below.

Step 1: Randomly generate all vectors i.e. $x_i(t) = (x_1, x_2, \ldots, x_{NP})$ in the prescribed search range

Step 2: Evaluate $x_i(t)$; i. e. find the function values of $x_i(t)$ for i = 1, 2, …, NP

Step 3: Set t = 0

Step 4: Construct a matrix $p(t) = x_i(t), p_i^{best}(t) = p(t)$, go to step 6

Step 5: Update $p(t) = p_i^{best}(t)$ using

$$p(t) = \begin{cases} x_i(t) & ; \text{ if } f(x_i(t)) \le f\big(p_i(t-1)\big) \\ p_i(t-1); & \text{otherwise} \end{cases}$$

Step 6: Find $g^{best}(t)$

Step 7: Swarm Mutation using the Eq. (1)

Step 8: Swarm Crossover using the Eq. (2)

Step 9: Apply Elitism

Step 10: Stop if the termination criterion is met, else set t = t + 1 and go to Step-5

Moreover, the 'bracket operator penalty' [31, 32] approach is used in this paper to handle the equality and inequality constraints.

RESULT AND DISCUSSION

The simulations were conducted on Intel(R) Core-i3, 2.20 GHz, 2GB RAM, computer in the C-Free Standard 4.0 Environment. To execute the performance of MBDE, population sizes is taken as NP =50. After large experiment fine tuning the crossover rate (CR) and penalty parameter (R) are recommended as CR = 0.9 and R = 1000 to use in MBDE for further study.

Numerical analysis

In order to evaluate the effeciency of MBDE, 5 well-known constrained (picked from [33]) and 6 popular unconstrained (taken from [34]) engineering design problems has been considered. To verify the robustness of MBDE, the stopping criteria and the number of independent runs are kept same as in compred algorithms. The boldface values in each tables represents the better value achieved by the corresponding algorithm and 'NaN' shows the non-availability of the results.

On constrained engineering design problems

To verify the performance of MBDE, 5 well-known constrained engineering design problems (CEDPs) having different characteristics presented in Table 1 are picked from [33]. These are given below (CEDP-1 to CEDP-5).

CEDP-1: Welded Beam design problem

Minimize

$$f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

Subject to:

$$g_1(x) = \tau(x) - \tau_{max} \le 0$$
$$g_2(x) = \sigma(x) - \sigma_{max} \le 0$$
$$g_3(x) = x_1 - x_4 \le 0$$
$$g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0$$
$$g_5(x) = 0.125 - x_1 \le 0$$
$$g_6(x) = \delta(x) - \delta_{max} \le 0$$
$$g_7(x) = P - P_c(x) \le 0$$
$$0.1 \le x_i \le 2;\ i = 1, 4 \text{ and } 0.1 \le x_i \le 10;\ i = 2, 3$$

where

$$\tau(x) = \sqrt{(\tau')^2 + 2\tau'\tau''\frac{x_2}{2R} + (\tau'')^2},$$
$$\tau' = \frac{p}{\sqrt{2}x_1x_2},\quad \tau'' = \frac{MR}{J}$$
$$M = P\left(L + \frac{x_2}{2}\right),\quad R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2}$$
$$J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$
$$\sigma(x) = \frac{6PL}{x_4x_3^2},\quad \delta(x) = \frac{4PL^3}{Ex_3^3x_4},$$
$$p_c(x) = \frac{4.013E\sqrt{(x_3^2x_4^6/36)}}{L^2} \times \left(1 - \frac{x_3}{2L}\right)\sqrt{\frac{E}{4G}}$$
$$P = 6\,\text{lb},\ L = 14\text{in},\ E = 30\times10^6\ \text{psi},\ G = 12\times10^6\ \text{psi}$$
$$\tau_{max} = 13{,}600\,\text{psi},\ \sigma_{max} = 30{,}600\,\text{psi},\ \delta_{max} = 0.25\,\text{in}$$

CEDP-2: Pressure Vessel design problem

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

Subject to

$$g_1(x) = -x_1 + 0.0193x_3 \le 0$$
$$g_2(x) = -x_2 + 0.00954x_3 \le 0$$
$$g_3(x) = -\pi x_3^2x_4 - (4/3)\pi x_3^3 + 1296000 \le 0$$
$$g_4(x) = x_4 - 240 \le 0$$
$$\textit{where}\ \ 0 \le x_i \le 100,\ i = 1, 2;\quad 10 \le x_i \le 200,\ i = 3, 4$$

CEDP-3: Speed reducer design problem

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

Subject to

$$g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \le 0,\ g_2(x) = \frac{397}{x_1x_2^2x_3} - 1 \le 0$$
$$g_3(x) = \frac{1.93x_4^2}{x_2x_6^4x_3} - 1 \le 0,\ g_4(x) = \frac{1.93x_5^2}{x_2x_7^4x_3} - 1 \le 0$$
$$g_5(x) = \frac{\left[(745(x_4/x_2x_3))^2 + 16.9\times10^6\right]^{1/2}}{110x_6^3} - 1 \le 0$$
$$g_6(x) = \frac{\left[(745(x_5/x_2x_3))^2 + 157.9\times10^6\right]^{1/2}}{85x_7^3} - 1 \le 0$$
$$g_7(x) = \frac{x_2x_3}{40} - 1 \le 0,\ g_8(x) = \frac{5x_2}{x_1} - 1 \le 0,$$
$$g_9(x) = \frac{x_1}{12x_2} - 1 \le 0,\ g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0,$$
$$g_{11}(x) = \frac{1.5x_7 + 1.9}{x_5} - 1 \le 0$$

where

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

CEDP-4: Three-bar truss design problem

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

Subject to

$$g_1(x) = \frac{(\sqrt{2}x_1 + x_2)}{\sqrt{2}x_1^2 + 2x_1x_2} p - \sigma \le 0,$$

$$g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} p - \sigma \le 0,$$

$$g_3(x) = \frac{1}{\sqrt{2}x_2 + x_1} p - \sigma \le 0$$

$where\ \ 0 \le x_i \le 1,\ \ i = 1,2;\ l = 100\text{cm},$

$P = 2\ \text{kN/cm}^2,\ \sigma = 2\ \text{kN/cm}^2$

CEDP-5: Tension/Compression Spring design problem

$$Minimize\ f(x) = (x_3 + 2)x_2x_1^2$$

Subject to

$$g_1(x) = 1 - (x_2^3x_3/71785x_1^4) \le 0,$$

$$g_2(x) = \left(4x_2^2 - x_1x_2/12566(x_2x_1^3 - x_1^4)\right) + (1/5108x_1^2) - 1 \le 0$$

$$g_3(x) = \left(1 - (40.45x_1/x_2^2x_3)\right) \le 0,$$

$$g_4(x) = (x_2 + x_1)/1.5 - 1 \le 0$$

where

$0.05 \le x_1 \le 2.00,\ 0.25 \le x_2 \le 1.30,\ 2.00 \le x_3 \le 15.00$

Table 1. Characteristics of 5 constrained engineering design problems

Problem	**D**	**LI**	**NI**
Welded beam	4	2	5
Pressure vessel	4	3	1
Speed reducer	7	4	7
Three-bar truss	2	0	3
Tension/compression spring	3	1	3

D:Dimension, **LI**: Number of linear inequalities, **NI**: Number of nonlinear inequalities

Table 2. Simulation results produced by MBDE and compared with others for 5 constrained engineering design problems (CEDPs)

CEDPs	Algorithm	Best	Mean	Worst	S. D.	NFEs
CEDP-1	MBDE	**1.724846**	**1.724846**	**1.724846**	**0.00E+00**	**5500**
	PSO-DE	1.724852309	1.724852309	1.724852309	6.7E-16	66600
	DSS-MDE	2.3809	2.38095	2.38095	2.1E-10	24000
	DETPS	1.724852	1.724852	1.724853	2.1E-7	10000
	$(\mu + \lambda)$-ES	1.724852	1.777692	2.074562	8.8E-2	30000
	UPSO	1.92199	2.83721	4.88360	6.8E-1	100000
	CPSO	1.728024	1.748831	1.782143	1.3E-2	200000
	CoDE	1.733462	1.768158	1.824105	2.2E-2	240000
	ABCA	1.724852	1.741913	NaN	3.1E-2	30000
	TLBO	1.724852	1.728447	NaN	NaN	10000
CEDP-2	MBDE	**5884.689986**	**5884.689986**	**5884.689986**	**0.00E+00**	**10000**
	PSO-DE	6059.714335	6059.714335	6059.714335	1.0E-10	42100
	DETPS	5885.3336	5887.3161	5942.3234	1.0E+01	**10000**
	$(\mu + \lambda)$-ES	6059.7016	6379.9380	6820.3975	2.1E+2	30000
	UPSO	6544.27	9032.55	11,638.20	9.9E+2	100000
	CPSO	6061.0777	6147.1332	6363.8041	8.6E+1	200000
	CoDE	6059.7340	6085.2303	6371.0455	4.3E+1	240000
	ABCA	6059.7147	6245.3081	NaN	2.1E+2	30000
	TLBO	6059.7143	6059.7143	NaN	NaN	**10000**

CEDP-3	MBDE	**2993.758872**	**2993.758872**	**2993.758872**	**0.00E+00**	**6500**
	PSO-DE	2996.348165	2996.348165	2996.348166	1.0E-07	70100
	DSS-MDE	2994.4710	2994. 47	2994.4710	3.5E-12	30000
	DETPS	2996.348	2996.348	2996.348	5.2E-05	**10000**
	$(\mu + \lambda)$-ES	2996.348	2996.348	2996.348	**0.00E+00**	30000
	ABCA	2997.058	2997.058	NaN	**0.00E+00**	30000
	TLBO	2996.348	2996.348	NaN	NaN	**10000**
CEDP-4	MBDE	**263.891782**	**263.891782**	**263.891782**	**1.08567E-19**	**10000**
	PSO-DE	263.89584338	263.89584338	263.89584338	1.2E-10	17600
	DSS-MDE	263.89584	263.89584	263.89584	9.2E-7	15000
CEDP-5	MBDE	**0.012638**	**0.012638**	**0.012638**	**4.350272E-15**	**5500**
	PSO-DE	0.012665233	0.012665233	0.012665233	4.9E-12	42100
	DSS-MDE	0.012665233	0.012669366	0.012738262	1.25E-05	24000
	DETPS	0.012665	0.012680	0.012769	2.7E-5	10000
	$(\mu + \lambda)$-ES	0.012689	0.013165	0.014078	3.9E-4	30000
	UPSO	0.013120	0.022948	0.050365	7.2E-3	100000
	CPSO	0.012675	0.012730	0.012924	5.2E-5	200000
	CoDE	0.012670	0.012703	0.012790	2.7E-5	240000
	ABCA	0.012665	0.012709	NaN	1.3E-2	30000
	TLBO	0.012665	0.012666	NaN	NaN	10000

The statistical results provided by MBDE were compared with DSS-MDE [8], PSO-DE [33], DETPS [35], $(\mu + \lambda)$ -ES, [36], UPSO [37], CPSO [28], CoDE [39], ABCA [40] and TLBO [41]. The results provided by these approaches were directly taken from the original references. To verify the robustness of MBDE, 30 trial runs and the Number of Function Evaluations (NFEs) for each problem are listed in Table 2 are made for the sake of comparison.

The best optimal solution obtained by MBDE for CEDP-1, CEDP-2, CEDP-3, CEDP-4 and CEDP-5 is $f(0.205729, 3.470488, 9.036623, 0.205729) = 1.724846$, $f(0.7781, 0.38464, 0.3196, 200.00) = 5884.689986$, $f(3.5, 0.7, 17.0, 7.3, 7.8, 3.343364, 5.285351) = 2993.758872$, $f(0.788663, 0.408242) = 263.891782$ and $f(0.051432, 0.351062, 11.609791) = 0.012638$ respectively. The MBDE minimizes all 5 CEDPs without violating any constraints. It is interesting to observe that Table 2, MBDE outperforms than all methods, in terms of best, mean, worst and S. D. (standard deviation). Moreover, it is also observed that the NFEs by MBDE is very less compared to other methods, it confirms the high efficiency.

In comparison between non MBDE (i.e. PSO-DE, DSS-MDE, DETPS, $(\mu + \lambda)$-ES, UPSO, CPSO, CoDE, ABCA and TLBO), PSO-DE is perform marginally better/equal and solved all 5 CEDPs. So it is necessary to check the effect of MBDE and PSO-DE over a random generation for visualizing the rate of convergence. For 5 CEDPs the effects of DPD and PSO-DE are shown in Fig. 3(a-e). Undoubtedly from Fig. 3(a-e), it is clear that for each CEDP, MBDE converges faster than PSO-DE.

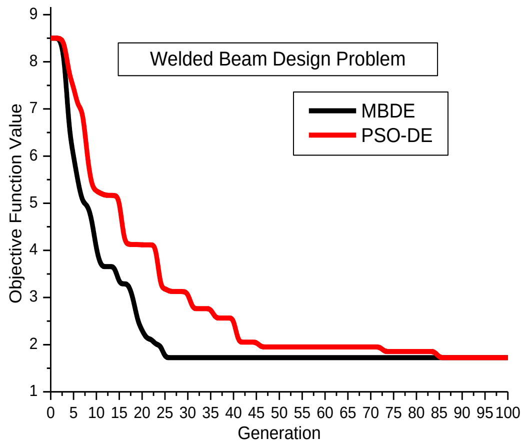

Figure 3(a). Convergence for Welded Beam Design Problem (CEDP-1)

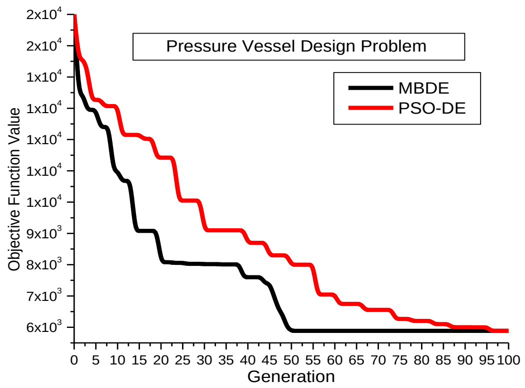

Figure 3(b). Convergence for Pressure Vessel Design Problem (CEDP-2)

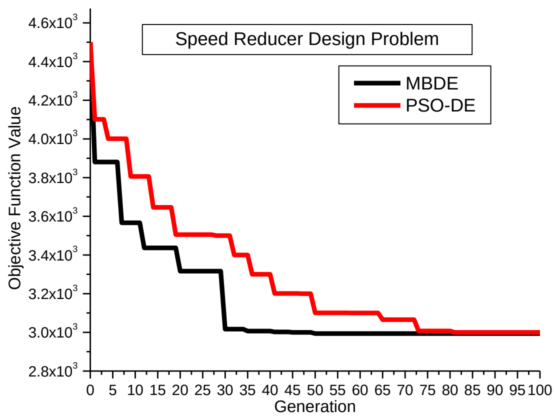

Figure 3(c). Convergence for Speed Reducer Design Problem (CEDP-3)

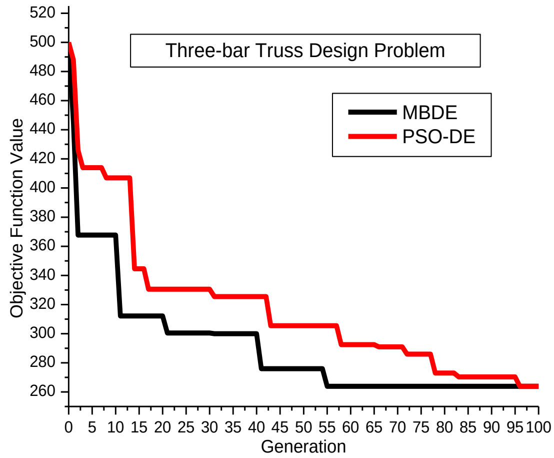

Figure 3(d). Convergence for Three-Bar Truss Design Problem (CEDP-4)

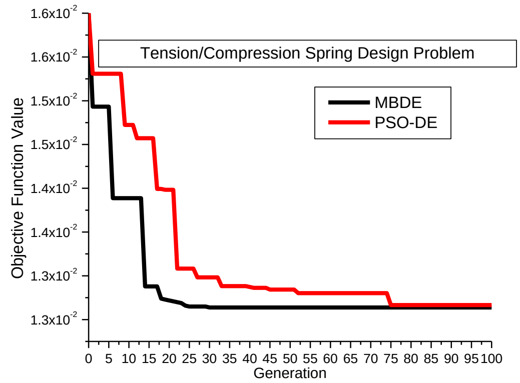

Figure 3(e). Convergence for Tension/compression Spring Design Problem (CEDP-5)

On unconstrained engineering design problems

In order to further verify the performance of MBDE, it is applied to solve 6 well-known unconstrained engineering design problems (UEDPs) taken from [34], which are listed below (UEDP-1 to UEDP-6).

UEDP-1: Gas transmission design

$$Minimize\ f(x) =$$

$$8.61*10^5 x_1^{1/2} x_2 x_3^{-2/3} \left(x_2^2 - 1\right)^{-1/2} + 3.69*10^4 x_3$$

$$+7.72*10^8 x_1^{-1} x_2^{0.219} - 765.43*10^6 x_1^{-1}$$

Subject to

$$10 \le x_1 \le 55, \quad 1.1 \le x_2 \le 2, \quad 10 \le x_3 \le 40$$

UEDP-2: Optimal capacity of gas production facilities

$$Minimize\ f(x) =$$

$$61.8 + 5.72 x_1 + 0.2623[(40 - x_1)\ ln\ (x_2 / 200)]$$

$$+0.087(40 - x_1)\ ln\ (x_2 / 200) + 700.23 x_2^{-0.75}$$

Subject to

$$17.5 \le x_1 \le 40, \quad 300 \le x_2 \le 600$$

UEDP-3: Design of a gear train

$$Minimize\ f(x) = \left\{\frac{1}{6.931} - \frac{T_d T_b}{T_a T_f}\right\} = \left\{\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right\}$$

Subject to

$12 \le x_i \le 60$, x_i should be integers and i = 1, 2, 3, 4.

UEDP-4: Optimal thermo-hydraulic performance of an artificially roughened air heater

$$Maximize\ \mathrm{L} = 2.51\ ln\ e^{+} + 5.5 - 0.1 R_M - G_H$$

$$where:\ R_M = 0.95 x_2^{0.53},\ G_H = 4.5\left(e^{+}\right)^{0.28}(0.7)^{0.57},$$

$$e^{+} = x_1 x_3 \left(\bar{f}/2\right)^{1/2},\ \bar{f} = (f_s + f_r)/2,$$

$$f_s = 0.079 x_3^{-0.25},\ f_r = 2\left(0.95 x_3^{0.53} + 2.5\ ln(1/2x_1)^2 - 3.75\right)^{-2}$$

Subject to

$$0.02 \le x_1 \le 0.8,\quad 10 \le x_2 \le 40,\quad 3000 \le x_3 \le 20000$$

UEDP-5: Frequency modulation sounds parameter identification problem

The frequency modulation sound model is represented as follows.

$$y(t) = a_1 \sin(w_1 t\theta + a_2 \sin(w_2 t\theta + a_3 \sin(w_3 t\theta)))$$

where a_1, w_1, a_2, w_2, a_3 , w_3 are six parameters in the bounds -6.4 to 6.35, $\theta = 2\pi/100$. The fitness function is defined as the sum of square error between the evolved data and the model data, as follows.

$$f(a_1, w_1, a_2, w_2, a_3, w_3) = \sum_{t=0}^{100} (y(t) - y_0(t))$$

The model data are given by the following equation:

$$y_0(t) = 1.0 \times \sin(5.0 \times t\theta - 1.5 \times \sin(4.5 \times t\theta + 2.0 \times \sin(4.9 \times t\theta)))$$

UEDP-6: The spread spectrum radar poly-phase code design problem

$$Minimize\ f(x) = max\left\{f_1(X), ..., f_{2m}(X)\right\}$$

where

$$X = \{(x_1, ..., x_D) \in R^D \mid 0 \le x_j \le 2\pi,\ j = 1, 2, ..., D\}$$

and $m = 2D - 1$

$$\text{with}\ f_{2i-1}(x) = \sum_{j=i}^{D} \cos \sum_{k=|2i-j-1|+1}^{j} x_k \quad i = 1, 2, ...D$$

$$f_{2i}(x) = 0.5 + \sum_{j=i+1}^{D} \cos \sum_{k=|2i-j|+1}^{j} x_k \quad i = 1, 2, ...D-1$$

$$f_{m+i}(X) = -f_i(X), \quad i = 1, 2, ...m$$

The mean and standard deviation (S. D.) of the best objective function value over 30 runs is reported in Table 3 for 6 UEDPs.The performance of MBDE is compared with DEPSO-2S, PSO-2S, SPSO-2007 and DE. The result of these algorithms have been taken directly from [34]. From Table 3, it is observed that, MBDE provides better values for all problems except UEDP-2, where it performs equally with others. It is worth noting that in almost all cases MBDE impacts very less S. D.

Table 3. Comparison of MBDE with others for 6 unconstrained engineering design problems (UEDPs), in terms of Mean and Standard Deviations of best objective function values

UEDPs	D	NFEs	Algorithm				
			MBDE	DEPSO-2S	PSO-2S	SPSO-2007	DE
UEDP-1	3	24000	**2.9624e+6 (0.00e+00)**	2.964e+6 (1.40e-009)	7.432e+6 (2.28e-009)	2.9644e+6 (4.66e-010)	2.964e+6 (0.264829)
UEDP-2	2	16000	**1.6984e+2 (1.14e-013)**	**1.698e+2 (1.14e-013)**	**1.698e+2 (1.14e-013)**	1.6984e+2 (1.14e-013)	1.698e+2 (0.000021)
UEDP-3	4	32000	**1.206e-10 (4.65e-016)**	1.397e-10 (2.65e-010)	1.401e-10 (3.35e-010)	1.4362e-9 (5.05e-009)	1.7638e-8 (3.5157e-8)
UEDP-4*	3	24000	**4.2146e+0 (1.16e-010)**	3.1712e-5 (7.54e-005)	2.3198e-6 (1.25e-005)	7.267e-16 (5.69e-016)	4.21422 (5.0847e-7)
UEDP-5	6	144000	**2.0695e+0 (1.28e-005)**	2.0743e+0 (3.07e+000)	2.5853e+0 (3.30e+000)	9.7517e+0 (6.65e+000)	3.01253 (0.367899)
UEDP-6	10	240000	**1.0415e-1 (1.25e-004)**	3.0049e-1 (7.07e-002)	3.5080e-1 (7.19e-002)	5.0075e-1 (1.61e-001)	0.626379 (0.0821391)
	20	480000	**1.0826e-1 (1.08e-002)**	5.3799e-1 (8.25e-002)	5.3979e-1 (1.25e-001)	8.6597e-1 (2.52e-001)	1.07813 (0.0812955)

D: Dimension; *: Maximization Problem

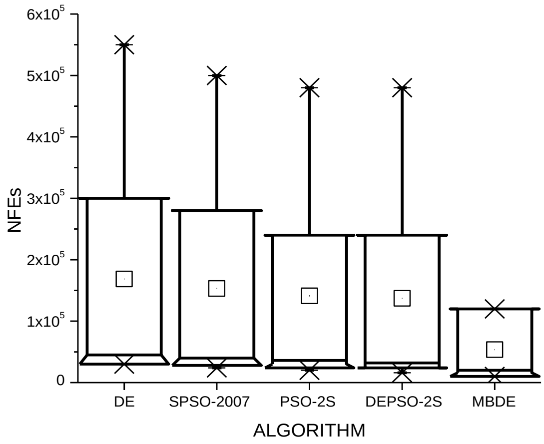

Figure 4. Average NFEs of MBDE with others for unconstrained engineering design Problems

The average NFEs for all algorithms under consideration are compared in Fig. 4. Fro this figures clearly seen that, MBDE uses fewer number of NFEs compared to rest algorithms, which add more value to its efficiency and robustness.

Statistical Analysis

In order to compare the performance of multiple algorithms on the test suite, a well-known ranking based test namely Friedman test [42] is selected. In Table 4 the average ranking of MBDE, PSO-DE, DSS-MDE, DETPS, $(\mu + \lambda)$-ES, UPSO, CPSO, CoDE, ABCA, TLBO on CEDPs and MBDE, DEPSO-2S, PSO-2S, SPSO-2007, DE on UEDPs are presented.

Table 4. Average rankings achieved by Friedman test

Algorithms	Ranking
on constrained engineering design problems	
MBDE	5.26
PSO-DE	4.68
DSS-MDE	3.26
DETPS	4.05
$(\mu + \lambda)$-ES	3.51
UPSO	2.57
CPSO	2.52
CoDE	4.24
ABCA	1.58
TLBO	3.02
on unconstrained engineering design problems	
MBDE	5.58
DEPSO-2S	4.96
PSO-2S	3.24
SPSO-2007	2.57
DE	2.61

The performance of the compare algorithms can be sorted by average ranking into the following order: MBDE, PSO-DE, CoDE, DETPS, $(\mu + \lambda)$-ES, DSS-MDE, TLBO, UPSO, CPSO, ABCA on CEDPs and MBDE, DEPSO-2S, PSO-2S, DE, SPSO-2007 on UEDPs. The best average ranking was obtained by the proposed MBDE, which outperforms the other four algorithms.

Performance analysis

The evaluations of 'performance' (as reported in [43]) are being carried out under - (i) 5 CEDPs [33] and (ii) 6 UEDPs [34]. For this, total of 30 independent runs with 10000 function evaluations for a run are fixed. The '**performance**' of MBDE with (i) PSO-DE, DSS-MDE, DETPS, $(\mu + \lambda)$-ES, UPSO, CPSO, CoDE, ABCA and TLBO for CEDPs, and with (ii) DEPSO-2S, PSO-2S, SPSO2007 and DE for UEDPs, are reported in Fig. 5(a) and Fig. 5(b) respectively. From these figures it is concluded that the 'performance' of MBDE is much better than its individual competitors, as it occupies maximum area in each pi-charts.

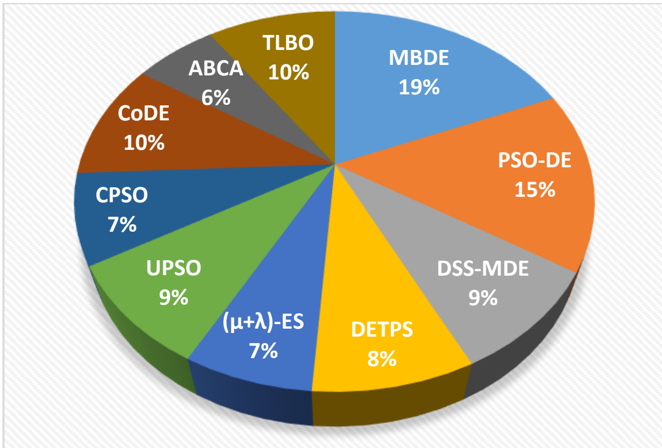

Figure 5(a). Performance evaluation of MBDE with others under 5 constrained engineering design problems

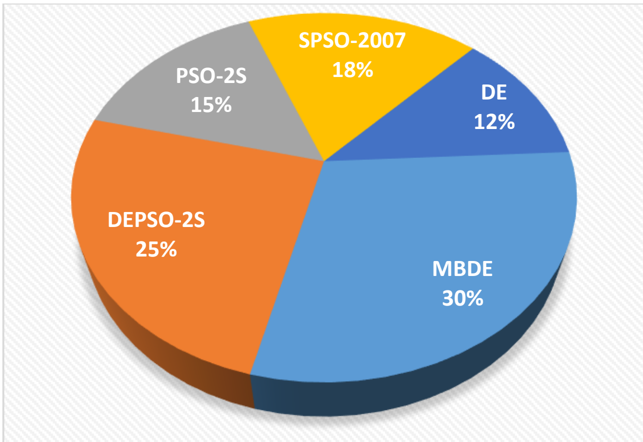

Figure 5(b). Performance evaluation of MBDE with others under 6 unconstrained engineering design problems

CONCLUSION AND FUTURE SCOPES

In this paper a 'Memory Based Differential Evolution (MBDE)' is proposed for solving engineering design problems. It employs two new operators (swarm mutation and swarm crossover) based on PSO environment.

The performance of MBDE has been compared with state-of-the-art variants of DE and other recent algorithms. The experimental and graphical comparisons conclude the proposed MBDE (i) have few parameters to fine tune, it is easy to use for solving optimization problems, (ii) have better solution quality, rate of convergence, efficiency and efficacy as compared to its competitors, (iii) have well balanced diversity (iv) probably avoids the stagnation and helps to get rid of stacking in local minima and (v) with respect to minimize the considered engineering optimization problem, achieves a marginal improvement over others.

As a future works, MBDE can be applied in typical real world and engineering problems and for solving multi-objective optimization problems.

REFERENCES

[1] R. Storn, K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces", Journal of Global Optimization, Vol. 11, No. 4, pp. 341–359, 1997.

[2] K. Price, R. Storn, J. Lampinen, "Differential evolution – a practical approach to global optimization", Berlin, Springer, 2005.

[3] S. Das, A. Abraham, U.K. Chakraborty, A. Konar, "Differential evolution using a neighborhood-based mutation operator", IEEE Transactions on Evolutionary Computation, Vol. 13, No. 3, pp. 526–553, 2009.

[4] H.Y. Fan, J. Lampinen, "A trigonometric mutation operation to differential evolution", Journal of Global Optimization, Vol. 27, No. 1, pp. 105–129, 2003.

[5] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, "Opposition-based differential evolution", IEEE Transactions on Evolutionary Computation, Vol. 12, No. 1, pp. 64–79, 2008.

[6] S. Das, S. Sil, "Kernel-induced fuzzy clustering of image pixels with an improved differential evolution algorithm", Information Sciences, Vol. 180, No. 8, pp. 1237–1256, 2010.

[7] Y. Wang, B. Li, T. Weise, "Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems", Information Sciences, Vol. 180, No. 12, pp. 2405–2420, 2010.

[8] M. Zhang, W. Luo, X. Wang, "Differential evolution with dynamic stochastic selection for constrained optimization", Information Sciences, Vol. 178, No. 15, pp. 3043–3074, 2008.

[9] A. Qin, V. Huang, P. Suganthan, "Differential evolution algorithm with strategy adaptation for global numerical optimization", IEEE Transactions on Evolutionary Computation, Vol. 13, No. 2, pp. 398–417, 2009.

[10] N. Noman, H. Iba, "Accelerating differential evolution using an adaptive local search", IEEE Transactions on Evolutionary Computation, Vol. 12, No. 1, pp. 107-125, 2008.

[11] O. Hrstka, A. Kucerova, "Improvement of real coded genetic algorithm based on differential operators preventing premature convergence", Advances in Engineering Software, Vol. 35, No. 3-4, pp. 237–246, 2004.

[12] J. Liang, A. Qin, P. Suganthan, S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions", IEEE Transactions on Evolutionary Computation, Vol. 10, No. 3, pp. 281–295, 2006.

[13] S. Das, P.N. Suganthan, "Differential evolution: a survey of the state-of-the-art", IEEE Transactions on Evolutionary Computation, Vol. 15, No. 1, pp. 4-31, 2011.

[14] F. Neri, V. Tirronen, "Recent advances in differential evolution: a survey and experimental analysis", Artificial Intelligence Review, Vol. 33, No. 1-2, pp. 61–106, 2010.

[15] R. Mallipeddi, P. Suganthan, Q. Pan, M. Tasgetiren, "Differential evolution algorithm with ensemble of parameters and mutation strategies", Applied Soft Computing, Vol. 11, No. 2, pp. 1679–1696, 2011.

[16] H. Sharma, J. Bansal, K. Arya, "Fitness based differential evolution", Memetic Computing, Vol. 4, No. 4, pp. 303-316, 2012.

[17] A. Semnani, I.T. Rekanos, M. Kamyab, T.G. Papadopoulos, "Two-dimensional micro wave imaging based on hybrid scatterer representation and differential evolution", IEEE transaction on Antennas and Propagation, Vol. 58, No. 10, pp. 3289–3298, 2010.

[18] C.T. Su, C.S. Lee, Network reconfiguration of distribution system susing Improved mixed-integer hybrid differential evolution, IEEE transaction on Power Delivery, Vol. 18, No. 3, pp. 1022–1027, 2003.

[19] Z.H. Cai, W.Y. Gong, C.X. Ling, H. Zhang, "A clustering-based differential evolution for global optimization", Applied Soft Computing, Vol. 11, No. 1, pp. 1363–1379, 2011.

[20] C.W. Chiang, W.P. Lee, J.S. Heh, "A2-Opt based differential evolution for global optimization", Applied Soft Computing, Vol. 10, No. 4, pp. 1200–1207, 2010.

[21] E. Mininno, F. Neri, "A memetic differential evolution approach in noisy optimization", Memetic Computing, Vol. 2, No. 2, pp. 111–135, 2010.

[22] W. Gong, Z. Cai, C.X. Ling, H. Li, "Enhanced differential evolution with adaptive strategies for numerical optimization", IEEE transaction on Systems Man and Cybernetics Part B- Cybernetics, Vol. 43, No. 6, pp. 2066–

2081, 2013.

[23] D. Zaharie, "Control of population diversity and adaptation in differential evolution algorithms", In: Proceedings of the 9th International Conference on Soft Computing, Brno, pp. 41–46, 2003.

[24] R. Gämperle, S.D. Müller, P. Koumoutsakos, "A parameter study for differential evolution", In: Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation, WSEAS Press, Interlaken, Switzerland, pp. 293–298, 2002.

[25] S. Das, A. Konar, U.K. Chakraborty, "Two improved differential evolution schemes for faster global search", In: Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, pp. 991–998, 2005.

[26] J. Lampinen, I. Zelinka, "On stagnation of the differential evolution algorithm", In: Proceedings of MENDEL 2000, 6th International Mendel Conference on Soft Computing, pp. 76–83, 2000.

[27] X. Wang, Q. Yang, Y. Zhao, "Research on hybrid PSODE with triple populations based on multiple differential evolutionary models", In: Proceedings International Conference Electrical Control Engineering Wuhan China, pp. 1692-1696, 2010.

[28] Y.C. Wu, W.P. Lee, C.W. Chien, "Modified the Performance of Differential Evolution Algorithm with Dual Evolution Strategy", In: 2009 International Conference on Machine Learning and Computing, Vol. 3, pp. 57-63, 2011.

[29] J.K. Kordestani, A. Rezvanian, M.R. Meybodi, "CDEPSO: a bi-population hybrid approach for dynamic optimization problems", Applied Intelligence, Vol. 40, No. 4, pp. 1-13, 2014.

[30] J. Kennedy, R. Eberhart, "Particle swarm optimization", IEEE International Conference on Neural Networks, IV (IEEE Press, Perth, Australia), pp. 1942-1948, 1995.

[31] Z. Michalewicz, "A survey of constraint handling techniques in evolutionary computation methods", In: Proceedings of the 4th Annual Conference on Evolutionary Programming, MIT Press, Cambridge, MA, pp. 135–155, 1995.

[32] K. Deb, "Optimization for Engineering Design: Algorithms and Examples", Prentice-Hall of India, New Delhi, 1995.

[33] H. Liu, Z. X., Cai, Y. Wang, Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization, Applied Soft Computing 10(2) (2010) 629-640.

[34] A. El Dor, M. Clerc, P. Siarry, "Hybridization of Differential Evolution and Particle Swarm Optimization in a new algorithm DEPSO-2S", Swarm and Evolutionary Computation, Vol. 7269, pp. 57-65, 2012.

[35] G. Zhang, J. Cheng, M. Gheorghe, Q. Meng, "A hybrid approach based on differential evolution and tissue membrane systems for solving constrained manufacturing parameter optimization problems", Applied Soft Computing, Vol. 13, No. 3, pp. 1528–1542, 2013.

[36] E. Mezura-Montes, C.A.C. Coello, "Useful infeasible solutions in engineering optimization with evolutionary algorithms", In: MICAI 2005, Advances in Artificial Intelligence, Lecture Notes in Computer Science, Vol. 3789, pp. 652-662, 2005.

[37] K.E. Parsopoulos, M.N. Vrahatis, "Unified particle swarm optimization for solving constrained engineering optimization problems", Advances in Natural Computation, Lecture Notes in Computer Science, Vol. 3612, pp. 582-591, 2005.

[38] Q. He, L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems", Engineering Applications of Artificial Intelligence, Vol. 20, No. 1, pp. 89–99, 2007.

[39] F.Z. Huang, L. Wang, Q. He, "An effective co-evolutionary differential evolution for constrained optimization", Applied Mathematics and Computation, Vol. 186, No. 1, pp. 340–356, 2007.

[40] B. Akay, D. Karaboga, "Artificial bee colony algorithm for large-scale problems and engineering design optimization", Journal of Intelligent Manufacturing, Vol. 23, No. 4, pp. 1001-1014, 2012.

[41] R.V. Rao, V.J. Savsani, D.P. Vakharia, "Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems", Computer-Aided Design, Vol. 43, No. 3, pp. 303–315, 2011.

[42] S. Garcia, A. Fernandez, J. Luengo, F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power", Information Sciences, Vol. 180, No. 10, pp. 2044–2064, 2010.

[43] K.N. Das, R.P. Parouha, "An ideal tri-population approach for unconstrained optimization and applications", Applied Mathematics and Computation, Vol. 256, pp. 666–701, 2015.