International Journal of Research in Advent Technology, Vol.6, No.5, May 2018
E-ISSN: 2321-9637
Available online at www.ijrat.org

A novel differential evolution for model order reduction

Raghav Prasad Parouha
Department of Mathematics, Indira Gandhi National Tribal University Amarkantak M.P. India
Email id: rparouha@gmail.com

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 and global best mechanism of particle swarm optimization (PSO). Proposed MBDE is validate on 13 typical benchmark functions. Further to test its efficacy five different model order reduction (MOR) problems for single-input and single-output system are solved by MBDE. The results of MBDE are compared with state-of-the-art algorithms that also solved those problems. Numerical, statistical and graphical analysis reveal the competency of the proposed algorithm.

Index Terms- Differential Evolution, Mutation, Crossover, Engineering design problems.

1. INTRODUCTION

Nowadays, optimization problems are very important and frequently appear in the real world. Solving such problems is a challenging area of research in the science and engineering disciplines. In spite of many Evolutionary Algorithms (EAs), Differential Evolution (DE) [1] is an efficient, formidable and popular ingredient. The DE has many advantages like easy implementation, reasonably faster, robust and exhibits effective global search ability [2-8]. In general, it also has efficient global search ability and hence considered as global optimization algorithm [9]. Therefore, it has been applied to solve many engineering optimization problems such as mechanical engineering design problem [10], fuzzy clustering of image pixel [11], economic load dispatch [12], nuclear reactor core design [13] and many others [9]. However, most of the time, the solution gets stacked in some local optima. As a result, it leads to a premature convergence. It is because of DE have some individual shortcomings like sometimes the solution gets stacked in local optimum which leads to premature convergence [8, 14, 15]. Also, same as other EAs, DE does not guarantee to reach at global optimal solution in a finite time interval [8]. Therefore, in order to improve the performance of basic DE, a number of attempts are made in the literature [2-23]. A detailed survey on the variants of DE can be found in [20, 21]. Moreover, in order to improve the robustness of DE, a number of mutation strategies of DE have been proposed in [9, 16, 17, 19, 20, 21, 22]. Basically, DE is much sensitive to choice of the mutation strategy. Also, it is very difficult to recommend a fixed set of parameters for different problems [9, 16, 17, 19, 20, 21, 22]. On the other hand, inappropriate choice of mutation strategy may lead to premature convergence, stagnation or wastage of computational time [9].

Similarly, researchers mainly used two types of crossover schemes in DE namely binomial crossover and exponential crossover [1]. In [24], Price recommended the use of binomial crossover is better.

But later, it is observed that there are no significant differences between these crossovers [25].

Unfortunately, according to 'No Free Lunch Theorem [26])', no single optimization method exist which is able to solve consistently to all global optimization problems. In spite of quite a good number of DE variants exist in the literature; DE further yields improved results while hybridizing with Particle Swarm Optimization (PSO) [27]. Each of them is capable of dominating the shortcoming of the other to add the robustness in the resultant hybrid algorithm. The magical synergy of DE and PSO has been well established and has crossed many success milestones in recent past. Yet, many hybrid methods of DE and PSO have been proposed [28-46]. These approach moves around the enhancement of capabilities of DE and PSO in various aspects and successfully applied to wide range of optimization problems such as medical image processing problem [30], engineering design optimization [37, 38], well placement optimization [40], power systems optimization [41], digital FIR filter design [42] and many others.

Though many variants of DE and its hybrid algorithms have been suggested in the literature to solve optimization problems, they are unable to provide satisfactory result. The reason behind this is DE has no mechanism to memorize the so-far best solution but it uses only the global information about the search space [36]. Therefore, in spite of the increased convergence rate of DE, the algorithm mostly loses its computing power and eventually leads to premature convergence as reported in [4, 9, 15].

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 the basic DE. Section 3 presents a detailed description of the proposed algorithm. The efficiency of the proposed algorithm is validated through unconstrained benchmark functions in Section 4. The proposed algorithm is applied to solve MOR problems in Section 5. Section 6 draws the conclusion of the paper along with some highlights on future scopes.

2. 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) $x_i = (x_{i1}, x_{i2}, \dots, x_{iD}); i = 1, 2, \dots, NP,$ is randomly generated within user-defined bounds, where *D* is the dimension of the optimization problem.

Mutation

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

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

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

Selection

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

$$x_{ij}(t+1) = \begin{cases} u_i(t+1); & \text{if } f\left(u_i(t+1)\right) \leq f\left(x_i(t)\right) \\ x_i(t) \quad ; & \text{otherwise} \end{cases}$$

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

3. 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 [27]. The proposed operators are explained in the following section.

3.1. *Swarm mutation*

Let $x_i(t) = x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t)$ 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) $v_i(t) = v_{i1}(t), v_{i2}(t), \dots, v_{iD}(t)$ 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.

3.2. *Swarm crossover*

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

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

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

3.3. *Steps of proposed algorithm*

Steps of the proposed MBDE presented below.

Step 1: Randomly generate all vectors i.e. $x_i(t) = (x_1, x_2, \dots, x_{NP})$ 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

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

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

4. VALIDATION OF PROPOSED ALGORITHM

In this section before solving the MOR problem, proposed MBDE applied to solve 13 unconstrained benchmark functions (reported in Table 1 and taken from [23]).

Table 1 Benchmark Functions

f	Function Name	Formulation	S	C	f_{min}
f_1	Sphere	$f(x) = \sum_{i=1}^{D} x_i^2$	[-100, 100]	US	0
f_2	Schwefel 2.22	$f(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	[-10,10]	UN	0
f_3	Schwefel 1.2	$f(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_i^2 \right)$	[-100, 100]	UN	0
f_4	Schwefel2.21	$f(x) = \max_i \{ \lvert x_i \rvert, 1 \le i \le D \}$	[-100, 100]	UN	0
f_5	Step	$f(x) = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	[-100, 100]	US	0
f_6	Quartic	$f(x) = \sum_{i=1}^{D} i x_i^4 + \text{random}[0,1)$	[-1.28, 1.28]	US	0
f_7	Rosenbrock	$f(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[-30, 30]	UN	0
f_8	Schwefel	$f(x) = 418.9829 \times D - \sum_{i=1}^{D} \left(x_i \sin \sqrt{\lvert x_i \rvert} \right)$	[-500, 500]	MS	0
f_9	Rastrigin	$f(x) = 10D + \sum_{i=1}^{D} \left(x_i^2 - 10\cos(2\pi x_i) \right)$	[-5.12, 5.12]	MS	0
f_{10}	Ackley	$f(x) = 20 + e - 20e^{-\left(\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2} \right)} - e^{\left(\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i) \right)}$	[-32, 32]	MN	0
f_{11}	Griewank	$f(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1$	[-600, 600]	MN	0
f_{12}	Penalized	$f(x) = \frac{\pi}{D} \left\{ 10\sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2 \right\} + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$ and $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & ; \ x_i > a \\ 0 & ; \ -a \le x_i \le a \\ k(-x_i - a)^m & ; \ x_i < -a \end{cases}$	[-50, 50]	MN	0
f_{13}	Penalized2	$f(x) = 0.1 \left\{ \sin^2(\pi x_1) + \sum_{i=1}^{D-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_D - 1)^2 [1 + \sin^2(2\pi x_D)] \right\} + \sum_{i=1}^{D} u(x_i, 5, 100, 4)$ where $y_i = 1 + \frac{1}{4}(x_i + 1)$ and $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & ; \ x_i > a \\ 0 & ; \ -a \le x_i \le a \\ k(-x_i - a)^m & ; \ x_i < -a \end{cases}$	[-50, 50]	MN	0

S: domain of the variables, f_{min}: Global minima, C: Function characteristics, U: Unimodal, M: multimodal, S: separable, N: non-separable

Table 2 Comparison of MBDE with others in [23] for 13 (10D) benchmark functions, in terms of mean and S. D. of the best objective function value

f	Max. NFEs	Algorithm				
		MBDE	GDE	DE/rand/1	DE/best/1	DE/target-to-best/1
f_1	100000	**1.429e-295** **(0.000e+00)**	4.821e-46 (1.13e-45)	1.382e-36 (1.19e-36)	1.602e-39 (1.39e-39)	2.954e-41 (2.694e-41)
f_2	100000	**2.642e-293** **(0.000e+00)**	2.875e-21 (4.99e-21)	7.475e-19 (3.58e-19)	4.291e-20 (3.63e-20)	9.352e-21 (3.646e-21)
f_3	100000	**4.623e-291** **(0.000e+00)**	1.338e-24 (2.77e-24)	1.168e-20 (9.57e-21)	1.84e-22 (1.35e-22)	4.685e-24 (3.772e-24)
f_4	100000	**1.182e-296** **(0.000e+00)**	1.042e-14 (2.91e-14)	3.048e-13 (2.35e-13)	2.683e-14 (2.80e-14)	3.943e-15 (1.695e-15)
f_5	100000	**0.000e+000** **(0.000e+00)**	1.325e-15 (3.37e-15)	2.325e-12 (3.35e-12)	2.278e-21 (3.28e-21)	5.986e-22 (1.260e-22)
f_6	100000	**0.000e+00** **(0.00e+00)**	**0.000e+00** **(0.00e+00)**	**0.000e+00** **(0.00e+00)**	**0.000e+00** **(0.00e+00)**	**0.000e+00** **(0.00e+00)**
f_7	100000	**1.037e-006** **(2.420e-08)**	1.329e-03 (6.047e-4)	1.78e-03 (6.776e-4)	2.011e-03 (8.390e-4)	1.694e-03 (7.761e-04)
f_8	100000	**0.000e+000** **(0.000e+00)**	7.787e+02 (1.871e+2)	2.460e+02 (3.611e+2)	6.664e+02 (3.734e+2)	5.012e+02 (1.347e+2)
f_9	100000	**0.000e+000** **(0.000e+00)**	5.597e+00 (1.570e+0)	1.882e+01 (3.235e+0)	6.467e+00 (1.641e+0)	2.192e+01 (3.391e+0)
f_{10}	100000	**1.284e-015** **(0.000e+00)**	7.993e-15 (2.90e-15)	4.440e-15 (0.000e+0)	5.151e-15 (1.49e-15)	4.440e-15 (0.000e+0)

f_{11}	100000	**0.000e+000** **(0.000e+00)**	8.883e-02 (4.691e-2)	1.819e-02 (9.154e-2)	1.219e-02 (1.021e-1)	3.127e-02 (8.615e-02)
f_{12}	100000	**1.511e-032** **(1.258e-48)**	4.7116e-32 (1.15e-47)	4.7116e-32 (1.15e-37)	1.711e-32 (1.15e-47)	4.711e-32 (1.53e-47)
f_{13}	100000	**1.346e-032** **(2.042e-48)**	1.349e-32 (2.884e-48)	1.349e-32 (2.884e-48)	1.349e-32 (2.884e-48)	1.349e-32 (2.884e-48)

Table 3 Comparison of MBDE with others in [23] for 13 (30D) benchmark functions, in terms of mean and S. D. of the best objective function value

f	Max. NFEs	Algorithm				
		MBDE	GDE	DE/rand/1	DE/best/1	DE/target-to-best/1
f_1	300000	**4.162e-290** **(0.000e+00)**	6.074e-24 (8.53e-24)	1.355e-03 (5.304e-4)	4.968e-04 (3.323e-4)	1.096e-04 (4.7257e-05)
f_2	300000	**1.495e-292** **(0.000e+00)**	1.759e-07 (4.185e-7)	2.130e-01 (7.311e-2)	2.882e-02 (7.528e-3)	2.040e-02 (8.340e-03)
f_3	300000	**4.528e-289** **(0.000e+00)**	1.764e-02 (2.105e-2)	1.314e+04 (3.752e+2)	4.742e+02 (1.814e+2)	2.692e+02 (7.346e+1)
f_4	300000	**1.620e-291** **(0.000e+00)**	3.256e-01 (2.675e-1)	2.813e+00 (3.646e+1)	1.000e+00 (3.224e-1)	8.337e-01 (1.919e-01)
f_5	300000	**0.000e+000** **(0.000e+00)**	5.217e+00 (5.189e+0)	2.722e+01 (6.322e-1)	3.348e+01 (2.827e+1)	2.856e+01 (2.035e+1)
f_6	300000	1.584e-012 (6.519e-18)	**1.557e-23** **(2.65e-23)**	1.363e-03 (3.836e-4)	6.735e-04 (3.156e-4)	1.032e-04 (3.624e-05)
f_7	300000	**2.402e-006** **(1.501e-08)**	1.899e-02 (6.103e-3)	2.483e-02 (6.148e-3)	2.759e-02 (6.852e-3)	2.029e-02 (5.103e-03)
f_8	300000	**0.000e+000** **(0.000e+00)**	2.897e+03 (8.860e+2)	7.00e+03 (2.866e+2)	3.097e+03 (7.15e+12)	4.377e+03 (1.338e+3)
f_9	300000	**0.000e+000** **(0.000e+00)**	4.745e+01 (1.201e+1)	1.964e+02 (7.629e+1)	1.106e+02 (1.898e+1)	2.019e+02 (6.946e+0)
f_{10}	300000	**1.654e-015** **(1. 462e-16)**	2.129e-10 (1.12e-10)	1.796e-02 (3.406e-3)	8.160e-03 (2.819e-3)	3.603e-03 (9.845e-04)
f_{11}	300000	**0.000e+000** **(0.000e+00)**	8.127e-03 (9.785e-3)	7.260e-03 (2.931e-3)	5.785e-03 (5.361e-3)	4.030e-03 (3.991e-03)
f_{12}	300000	**1.674e-021** **(2.183e-30)**	6.133e-21 (7.05e-22)	5.678e-04 (2.638e-4)	1.191e-04 (7.364e-5)	3.317e-05 (3.053e-05)
f_{13}	300000	**1.642e-023** **(1.038e-30)**	5.541e-23 (9.19e-23)	2.508e-03 (9.607e-4)	1.401e-03 (3.463e-3)	1.024e-04 (7.882e-05)

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 size (*NP*) is taken as 10D and 30D (where D is the dimensions of the problems) and after fine-tuned crossover rate are recommended as CR = 0.9 for further study.

For each problem, 50 independent runs are performed and numerical results are reported in respective tables. The boldface values in each table represents the better value achieved by that corresponding algorithm and 'Na' shows the non-availability of the results. For a fair comparison the stopping criterion in this study remains the same as in GDE [23]. The results produced by MBDE on 13 typical unconstrained benchmark functions have been compared with state-of-the-art algorithms.

4.1. *Numerical analysis*

The mean and standard deviation (S.D.) of the best objective function value for 50 runs is reported in Table 2 and Table 3 for 10D and 30D benchmark functions, respectively. The results produced by proposed algorithm are compared with DE/rand/1, DE/best/1, DE/target-to-best/1 and group based differential evolution (GDE) [23]. It is observed that from Table 2 and Table 3 that MBDE outperforms its competitors in all benchmark functions both for 10D and 30D. Only for f_6, MBDE performs equally to others for 10D, but better than others in 30D except GDE. It is worth noting that in almost all cases MBDE impacts very less S. D.

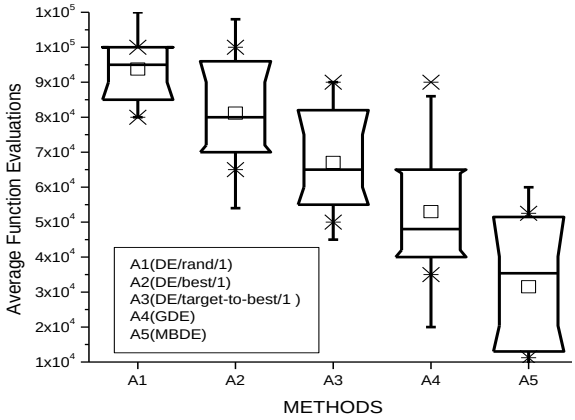

Fig. 1(a). Average NFEs of MBDE with others in [23] for 13 (10D) benchmark Functions

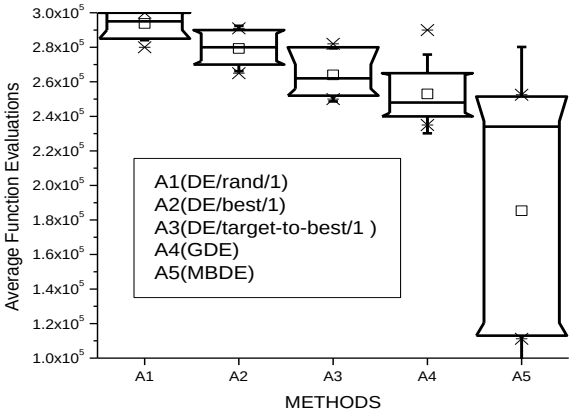

Fig. 1(b). Average NFEs of MBDE with others in [23] for 13 (30D) benchmark Functions

The average numbers of function evaluations (NFEs) for all the algorithms under consideration are compared in Fig. 1(a) and 1(b) for 10D and 30D, respectively. From these figure it can be concluded that MBDE uses fewer number of NFEs compared to rest algorithms.

4.2. *Convergence analysis*

The convergence speed of MBDE is compared with the other algorithms reported in [23] over a set of 5 higher dimensional (30D) typical test functions (Sphere, Rosenbrock, Schwefel, Rastrigin and Ackley) are picked. The convergence graphs are presented in Fig. 2(a-e). Form these figures it can be concluded that in all the functions MBDE converge much faster than other algorithms. It can be concluded that, MBDE is greatly stable to minimize the unconstrained optimization problems.

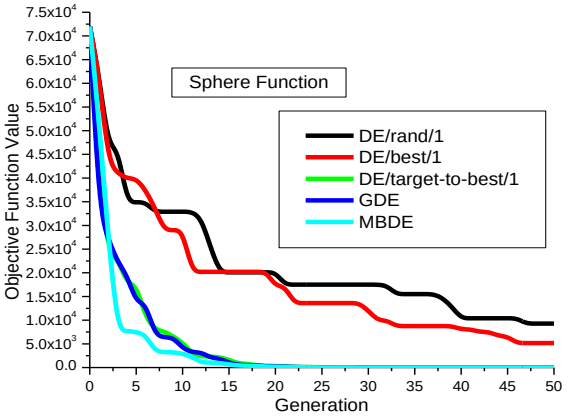

Fig. 2(a). Convergence of MBDE with others in [23] for Sphere Function (30D)

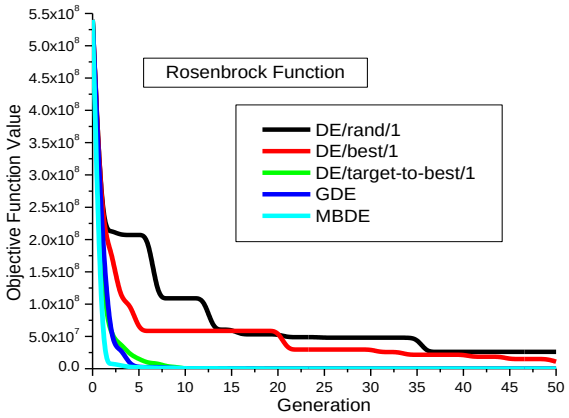

Fig. 2(b). Convergence of MBDE with others in [23] for Rosenbrock Function (30D)

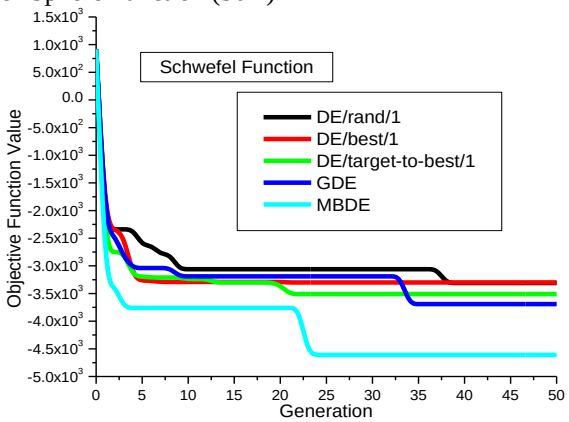

Fig. 2(c). Convergence of MBDE with others in [23]

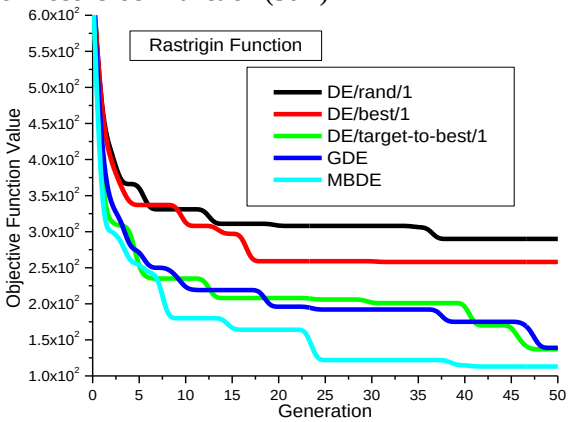

Fig. 2(d). Convergence of MBDE with others in [23]

for Schwefel Function (30D) for Rastrigin Function (30D)

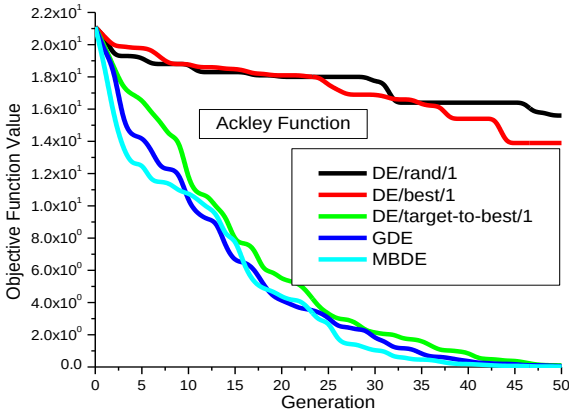

Fig. 2(e). Convergence of MBDE with others in [23] for Ackley Function (30D)

4.3. *Statistical analysis*

To compare the performance of multiple algorithms on the test suite, a well-known ranking based test namely Friedman test [47] is selected. In Table 4 the average ranking of MBDE, GDE, DE/rand/1, DE/best/1, DE/target-to-best/1 on unconstrained benchmark functions are presented.

Table 4 Average rankings achieved by Friedman test

Algorithms	Ranking
MBDE	5.18
GDE	4.27
DE/rand/1	2.26
DE/best/1	3.05
DE/target-to-best/1	3.57

The performance of the compared algorithms can be sorted by average ranking into the following order: MBDE, GDE, DE/target-to-best/1, DE/best/1, DE/rand/1. The best average ranking was obtained by the proposed MBDE, which outperforms the other algorithms.

5. APPLICATION OF PROPOSED ALGORITHM

In this section proposed MBDE is applied to solve the model order reduction (MOR) problem. A brief description of the MOR problem is presented below.

5.1. *Problem statement*

Let's consider a n^{th} order linear time invariant dynamic single-input and single-output system (SISO) system given as follows.

$$G(s) = \frac{N(s)}{D(s)} = \frac{\sum_{i=0}^{n-1} a_i s^i}{\sum_{i=0}^{n} b_i s^i} \quad (3)$$

Where a_i and b_i are known constants. The problem is to find out rth order reduced model in the transfer function form $R(s)$, where $r < n$ represented by Eq. (4) such that the reduced model retains the important characteristics of the original system and approximates its step responses as closely as possible for the same type of inputs with minimum Integral square error (ISE) as well as Impulse response energy(IRE).

$$R(s) = \frac{N_r(s)}{D_r(s)} = \frac{\sum_{i=0}^{r-1} a_i' s^i}{\sum_{i=0}^{r} b_i' s^i} \quad (4)$$

where $a_i'$and $b_i'$are unknown constants.

Mathematically, the ISE of step responses of the original and the reduced system can be expressed by the following error index given by Eq. 5).

$$J = \int_0^\infty [y(t) - y_r(t)]^2 dt \quad (5)$$

where $y(t)$ is the unit step response of the original system and $y_r(t)$ is the unit step response of the reduced system. The error index is the function of unknown coefficients of reduced order model so that the error index is minimized.

The IRE for the original and the various reduced models is given by as following Eq. (6).

$$IRE = \int_0^\infty g(t)^2 dt \quad (6)$$

where, $g(t)$ is the Impulse response of the system. In this paper the objective is to minimize the objective function based on both ISE and IRE.

5.2. *Procedure to reduced second order model for the given higher order system*

Let's consider the original high order system transfer function is given by Eq. (7).

$$G(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + \cdots + b_1 s + b_0}{a_m s^m + a_{m-1} s^{m-1} + \cdots + a_1 s + a_0} \quad (7)$$

where m is the order of the system and $m > n$.

$$G(s) = \frac{b_n s^n + b_{n-1} s^{n-1} + \cdots + b_1 s + b_0}{(s+\lambda_1)(s+\lambda_2)(s+\lambda_3)\ldots(s+\lambda_m)} \quad (8)$$

where $-\lambda_1 - \lambda_2 < -\lambda_3 < \cdots < -\lambda_{m-1} < -\lambda_m$ are distinct real Eigen values of the system. The unit step responses of (8) can be determined as follows.

$$T(s) = \frac{G(s)}{s} = \frac{k_0}{s} + \frac{k_1}{s+\lambda_1} + \frac{k_2}{s+\lambda_2} + \cdots + \frac{k_{m-1}}{s+\lambda_{m-1}} + \frac{k_m}{s+\lambda_m} \quad (9)$$

where $k_i's$ are real constants. Taking inverse Laplace transformation of Eq. (9),

$$y(t) = k_0 + k_1 e^{-\lambda_1 t} + k_2 e^{-\lambda_2 t} + \cdots + k_m e^{-\lambda_m t} \quad 10)$$

where $k_0$is steady state response and the rest of the terms are transient response of the system given in equation (8). Let the proposed reduced order system constructed is of 2nd order, where

$$R(s) = \frac{a_0 s + a_1}{a_2 s^2 + a_3 s + a_4} = \frac{a_0 s + a_1}{(s+\mu_1)(s+\mu_2)} \quad (11)$$

where μ_1 and μ_2 are distinct and real Eigen values and $-\mu_1 < -\mu_2$.

The unit step response of (10) can be determined as follows.

$$T_1(s) = \frac{R(s)}{s} = \frac{k_0'}{s} + \frac{k_1'}{s+\mu_1} + \frac{k_2'}{s+\mu_2} \quad (12)$$

where k_0', k_1', k_2', μ_1 and μ_2 are real constants. Let the inverse Laplace transformation of Eq. (12) is as follows.

$$y_r(t) = k_0' + k_1' e^{-\mu_1 t} + k_2' e^{-\mu_2 t} \quad (13)$$

In order that steady state part of the responses of the original high order system (8) and the reduced order system (11) are matched exactly, the following condition should be fulfilled.

$$k_0 = k_0' \quad (14)$$

The ISE of the transient responses of the system of (8) and (11) is given as follows.

$$J = \int_0^\infty [(y(t) - k_0) - (y_r(t) - k_0')]^2 dt$$
$$= \int_0^\infty [y(t) - y_r(t)]^2 dt \text{ since } k_0 = k_0'$$
$$= \int_0^\infty \left\{\sum_{i=1}^n k_i e^{-\lambda_i t} - \sum_{j=1}^2 k_j' e^{-\mu_j t}\right\}^2 dt \quad (15)$$

where $k_i's$ and $\lambda_i's$ are known and k_1', k_2', μ_1, μ_2 are all unknown constants, which are randomly generated in MBDE and the value of J are calculated. The reduced model $R(s)$ is obtained over 30 independent runs so that the integral square error J is minimized. In the next step that reduced model $R(s)$ is taken where IRE given by Eq. (6) is also minimized.

5.3. *Simulations results and discussion*

In order to assess the efficiency of the proposed MBDE, it has been applied on five numerical examples of MOR problems. These numerical examples are reported in Table 5, having real and distinct Eigen values. Proposed MBDE aims at solving these numerical examples with two objectives. They are (i) to minimize the Integral Square Error (ISE) and (ii) Impulse Response Energy (IRE); between original higher order system and reduced lower order system.

The result produced by MBDE for considered five test system is compared with FBDE [18], LICLDE [48] and ABC [49]. For the fair comparison, the population
size, independent runs and stopping criterion are same as LICLDE [48]. Remaining parameters of MBDE are same as section 4. The best reported ISE and IRE are boldfaced in the respective tables. The original and the reduced systems for numerical examples 1, 2, 3, 4 and 5 are presented in Tables 6, 7, 8, 9 and 10, respectively.

Table 5 List of 5 numerical example of Model Order Reduction (MOR) problem

Numerical example	Original Source	Transfer function of original model
1. Fourth-order system	Lucas [50]	$G_1(s) = \frac{8169.13s^3+50664.97s^2+9984.32s+500}{100s^4+10520s^3+52101s^2+10105s+500}$
2. Fourth-order system	Pal [51]	$G_2(s) = \frac{s+4}{s^4+19s^3+113s^2+245s+150}$
3. Fourth-order system	Aguirre [52]	$G_3(s) = \frac{4.269s^3+5.10s^2+3.9672s+0.9567}{4.3992s^4+9.0635s^3+8.021s^2+5.362s+1}$
4. Eighth-order system	Shamash [53]	$G_4(s) = \frac{18s^7+514s^6+5982s^5+36380s^4+122664s^3+222088s^2+185760s+40320}{s^8+36s^7+546s^6+4536s^5+22449s^4+67284s^3+118124s^2+109584s+40320}$
5. Ninth-order system	Eydgahi et al. [54]	$G_5(s) = \frac{s^4+35s^3+291s^2+1093s+1700}{s^9+9s^8+66s^7+294s^6+1029s^5+2541s^4+4684s^3+5856s^2+4620s+1700}$

Table 6 Comparison of proposed method with existing methods for numerical example 1

Method of reduction	Reduced Models $R_1(s)$	ISE	IRE
Original	$G_1(s)$	----------	**34.069**
MBDE	$\frac{72.523558s + 252.508258}{s^2 + 89.582564s + 252.508258}$	**0.00125892**	34.068896
FBDE [18]	$\frac{85.33529245s + 462.3004006}{s^2 + 113.6582937s + 462.3004006}$	0.0017826566	**34.06884**

LICLDE [48]	$\frac{101.3218182s + 867.893179}{s^2 + 169.4059231s + 867.893179}$	0.0036228741	34.069918
ABC [49]	$\frac{485s+50000}{s^2+4187s+50000}$	0.011624954	34.065841

Table 7 Comparison of proposed method with existing methods for numerical example 2

Method of reduction	**Reduced Models** $R_2(s)$	**ISE**	**IRE**
Original	$G_2(s)$	----------	**0.00026938**
MBDE	$\frac{-0.0045958s + 0.08260527}{s^2 + 4.021015s + 3.0988975}$	**4.001259e-006**	**0.0002699**
LICLDE [48]	$\frac{-0.0195s + 0.2884}{s^2 + 14.9813s + 10.82}$	4.3168e-006	**0.00027**
ABC [49]	$\frac{0.0318s+4.0074}{s^2+13.7409s+150.2775}$	0.0005	0.0039

Table 8 Comparison of proposed method with existing methods for numerical example 3

Method of reduction	**Reduced Models** $R_3(s)$	**ISE**	**IRE**
Original	$G_3(s)$	----------	**0.54536**
MBDE	$\frac{0.75984s + 2.94328}{s^2 + 3.15286s + 3.079858}$	**0.21501e-01**	**0.545388**
LICLDE [48]	$\frac{0.7853s+2.949}{s^2+3.1515s+3.0823}$	0.338 e−01	0.54538
ABC [49]	$\frac{0.2034s+8.994}{s^2+7.9249s+9.4008}$	0.03501	0.54552

Table 9 Comparison of proposed method with existing methods for numerical example 4

Method of reduction	**Reduced Models** $R_4(s)$	**ISE**	**IRE**
Original	$G_4(s)$	----------	**21.740**
MBDE	$\frac{14.886587s + 4.78524}{s^2 + 5.98827s + 4.78524}$	**0.08e-03**	**21.74**
FBDE [18]	$\frac{17.32178s + 5.3660}{s^2 + 7.0240s + 5.3660}$	**0.80e-03**	**21.74**
LICLDE [48]	$\frac{17.203s + 5.3633}{s^2 + 6.9298s + 5.3633}$	**0.80e-03**	**21.74**
ABC [49]	$\frac{17.387s+5.3743}{s^2+7.091s+5.3743}$	0.00085	21.696

Table 10 Comparison of proposed method with existing methods for numerical example 5

Method of reduction	**Reduced Models** $R_5(s)$	**ISE**	**IRE**
Original	$G_5(s)$	----------	0.47021
MBDE	$\frac{-0.598856818s + 0.9965860}{s^2 + 1.49989469s + 0.9965860}$	**0.20595e-01**	0.471842
LICLDE [48]	$\frac{-0.6372s+1.0885}{s^2+1.5839s+1.0885}$	0.209e−01	**0.4718**

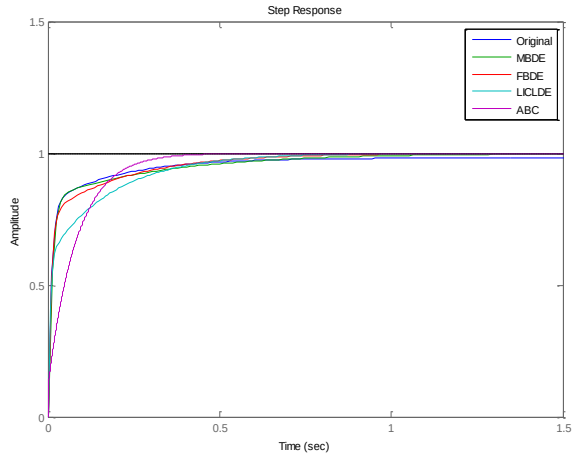

Fig. 3(a). Comparison of step response for numerical example 1

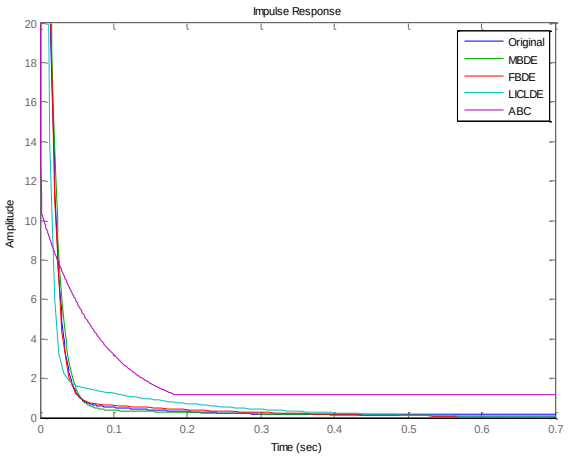

Fig. 3(b). Comparison of impulse response for numerical example 1

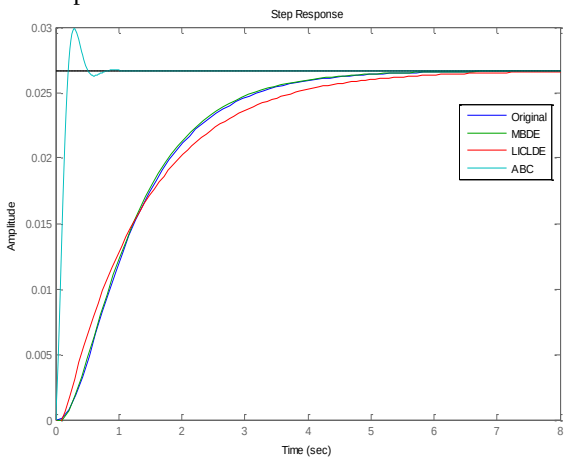

Fig. 4(a). Comparison of step responses for numerical example 2

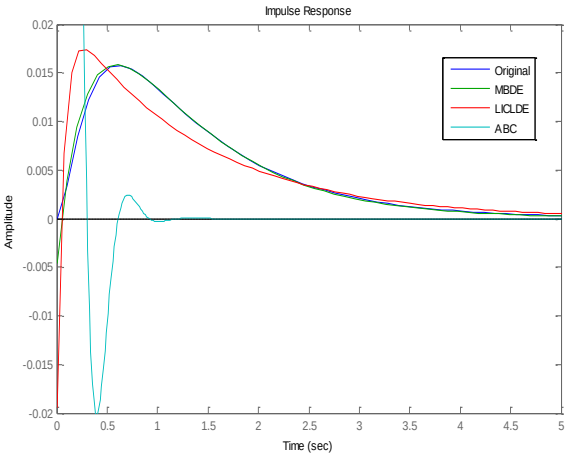

Fig. 4(b). Comparison of impulse responses for numerical example 2

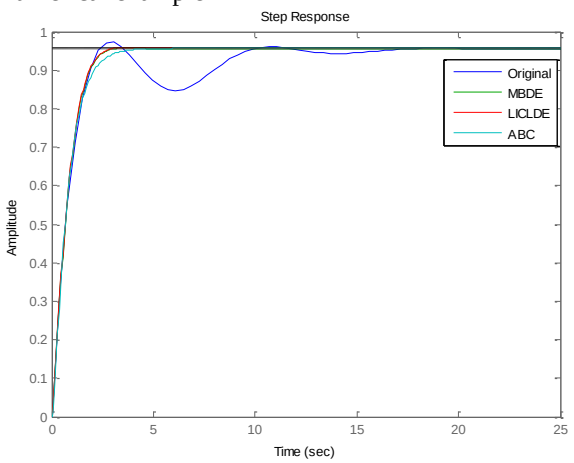

Fig. 5(a). Comparison of step responses for numerical example 3

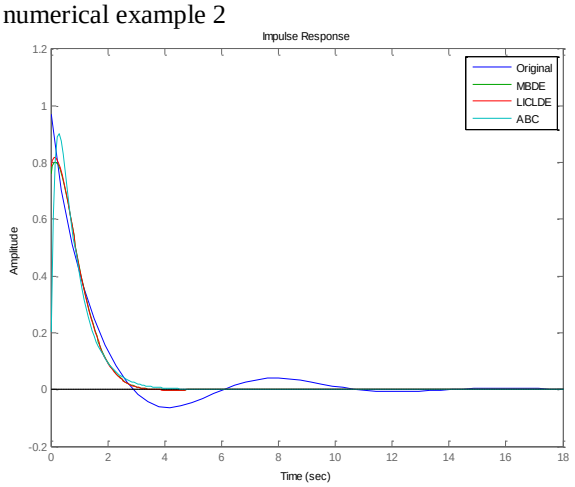

Fig. 5(b). Comparison of impulse responses for numerical example 3

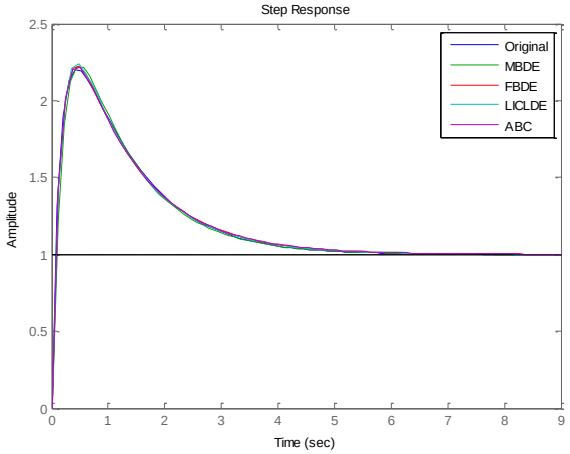

Fig. 6(a). Comparison of step responses for numerical example 4

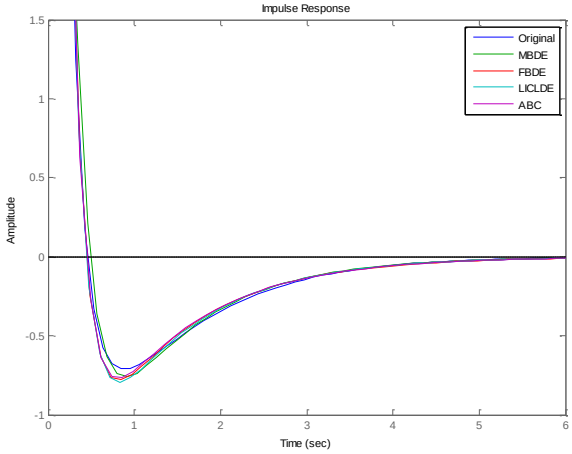

Fig. 6(b). Comparison of impulse responses for numerical example 4

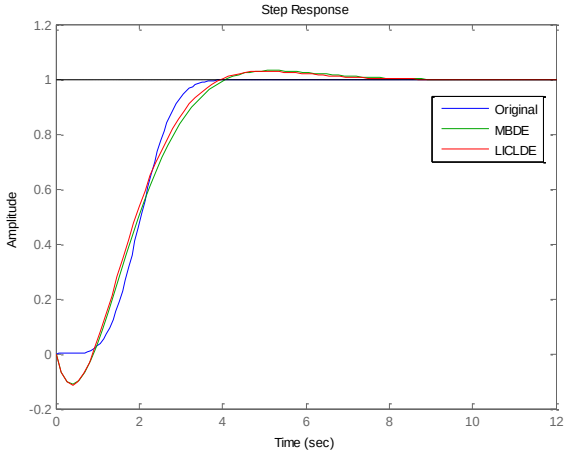

Fig. 7(a). Comparison of step responses for numerical example 5

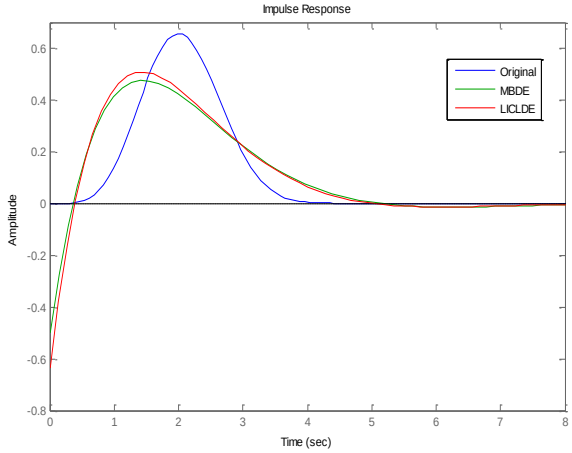

Fig. 7(b). Comparison of impulse responses for numerical example 5

The unit step responses of the original, the reduced systems using MBDE and best reported method in [18, 48, 49] are shown in Figs. 3(a), 4(a), 5(a), 6(a) and 7(a), respectively for corresponding numerical examples. Also the impulse responses of the original, the reduced systems using MBDE and best reported method in [18, 48, 49] are shown in Figs. 3(b), 4(b), 5(b), 6(b) and 7(b), respectively for corresponding numerical examples. It can be observed that for numerical examples 1, 2, 3 and 5, ISEs obtained by MBDE are significantly less than that of other methods. However, the ISE for numerical example 4 obtained by MBDE significantly equal to LICLDE and FBDE but significantly less than of other methods.

Also for all numerical examples, IREs of the reduced models obtained by MBDE are closest to that of the originals. Moreover, the step response (Figs. 3(a), 4(a), 5(a), 6(a) and 7(a)) as well as impulse response curve (Figs. 3(b), 4(b), 5(b), 6(b) and 7(b)) seem to be closely lying on the original curve. It may also be seen that the steady state responses of the original and the reduced order models by MBDE are exactly matching while the transient response matching is also very close as compared to other methods. Thus these numerical examples establish the superiority of MBDE over other methods. Finally, MBDE performance is superior to state-of-the-art algorithms. Thus, MBDE may be treated as a robust method to solve MOR problem.

6. CONCLUSION AND FUTURE WORKS

In this paper a 'Memory Based Differential Evolution (MBDE)' is proposed for solving model order reduction 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 large real world problem and engineering and for solving multi-objective optimization problems.

REFERENCES

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

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

[3] O. Hrstka and 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.

[4] S. Das, A. Konar and 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.

[5] J. Liang, A. Qin, P. Suganthan and 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.

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

[7] S. Rahnamayan, H. R. Tizhoosh and M. M. A. Salama, "Opposition-Based Differential Evolution", IEEE Transactions on Evolutionary Computation, vol. 12, no. 1, pp. 64–79, 2008.

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

[9] K. V. Price, R. M. Storn and J. A. Lampinen, "Differential Evolution: A Practical Approach to Global Optimization", Berlin: Springer, 2005.

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

[11] S. Das and 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.

[12] Y. Wang, B. Li and 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.

[13] W.F. Sacco and N. Henderson, "Differential evolution with topographical mutation applied to nuclear reactor core design", Progress in Nuclear Energy, vol. 70, pp. 140–148, 2014.

[14] N. Noman and H. Iba, "Accelerating differential evolution using an adaptive local search", IEEE Transactions on Evolutionary Computation, vol. 12, no. 1, pp. 107–125, 2008.

[15] J. Lampinen and 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.

[16] E. Mininno and F. Neri, "A memetic differential evolution approach in noisy optimization", Memetic Computing, vol. 2, no. 2, pp. 111–135, 2010.

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

[18] H. Sharma, J. Bansal and K. Arya, "Fitness based differential evolution", Memetic Computing, vol. 4, no. 4, pp. 1–14, 2012.

[19] W. Gong and Z. Cai, "Differential evolution with ranking based mutation operators", IEEE transaction on Systems Man and Cybernetics Part B- Cybernetics, vol. 43, no. 6, pp. 2066–2081, 2013.

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

[21] S. Das and 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.

[22] Qin, V. Huang and 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.

[23] M. F. Han, S. H. Liao, J. Y. Chang and C. T. Lin, "Dynamic group-based differential evolution using a self-adaptive strategy for global optimization problems", Applied Intelligence, vol. 39, no. 1, pp. 41-56, 2013.

[24] K. V. Price, "An Introduction to Differential Evolution", In David Corne, Marco Dorigo,

and Fred Glover, editors, New Ideas in Optimization, McGraw-Hill, London, pp. 79–108, 1999.

[25] K. Price and R. Storn, "http://www.icsi.berkeley.edu/~storn/code.html", website as in August 2008.

[26] D. H. Wolpert and W. G. Macready, "No Free Lunch Theorems for Optimization", IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, pp. 67-82, 1997.

[27] J. Kennedy and R. C. Eberhart, "Particle Swarm Optimization", In: Proceeding of IEEE International Conference on Neural Networks, pp.1942–1948, 1995.

[28] T. Hendtlass, "A combined swarm differential evolution algorithm for optimization problems", Lecture Notes Computer Science, Springer Verlag, vol. 2070, pp. 11-18, 2001.

[29] W. J. Zhang and X. F. Xie, "DEPSO: Hybrid particle swarm with differential evolution operator", in: proceedings IEEE International Conference Systems Man Cybernetics, vol. 4, pp. 3816-3821, 2003.

[30] H. Talbi and M. Batouche, "Hybrid particle swarm with differential evolution for multimodal image registration", in: proceedings of the IEEE International Conference on Industrial Technology, vol. 3, pp. 1567–1573, 2004.

[31] S. Das, A. Konar and U. K. Chakraborty, "Improving particle swarm optimization with differentially perturbed velocity", in: proceedings Genetic Evolutionary Computation Conference, pp. 177-184, 2005.

[32] P. W. Moore and G. K. Venayagamoorthy, "Evolving digital circuit using hybrid particle swarm optimization and differential evolution", International Journal of Neural Systems, vol. 16, no. 3, pp. 163-177, 2006.

[33] Z. F. Hao, G. H. Guo and H. Huang, "A particle swarm optimization algorithm with differential evolution", in: proceedings sixth International Conference Machine Learning and Cybernetics Hong Kong China, pp. 1031-1035, 2007.

[34] S. Das, A. Abraham and A. Konar, "Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives", Advances of Computational Intelligence in Industrial Systems, Studies in Computational Intelligence, Springer Verlag, Germany, pp. 1–38, 2008.

[35] C. Zhang, J. Ning, S. Lu, D. Ouyang and T. Ding, "A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization", Operations Research Letters, vol. 37, no. 2, pp. 117–122, 2009.

[36] X. Wang, Q. Yang and 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.

[37] H. Liu, Z. X. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization", Applied Soft Computing, vol. 10, no. 2, pp. 629-640, 2010.

[38] A. E. Dor, M. Clerc and 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.

[39] B. Xin, J. Chen, J. Zhang, H. Fang and Z. Peng, "Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and tax-onomy", IEEE Transactions on Systems Man and Cybernetics Part C: Applications and Reviews, vol. 42, no. 5, pp. 744–767, 2012.

[40] E. Nwankwor, A. Nagar and D. Reid, "Hybrid differential evolution and particle swarm optimization for optimal well placement", Computational Geosciences, vol. 17, no. 2, pp. 249-268, 2013.

[41] S. Sayah and A. Hamouda, "A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems", Applied Soft Computing, vol. 13, no. 4, pp. 1608–1619, 2013.

[42] Vasundhara, D. Mandal, R. Kar and S. P. Ghoshal, "Digital FIR filter design using fitness based hybrid adaptive differential evolution with particle swarm optimization", Natural Computing, vol. 13, no. 1, pp. 55-64, 2014.

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

[44] R. P. Parouha and K. N. Das, "Parallel hybridization of Differential Evolution and Particle Swarm Optimization for constrained optimization with its application", International Journal of Systems Assurance Engineering and Management, vol. 7, no. 1, pp. 143–162, 2016.

[45] K. N. Das and R. P.Parouha, "Optimization with a novel hybrid algorithm and applications", OPSEARCH, vol. 53, no. 3, pp. 443–473, 2016.

[46] K. N. Das, R. P. Parouha and K. Deep, "Design and applications of a new DE-PSO-DE algorithm for unconstrained optimization

problems", International Journal of Swarm Intelligence, vol. 3, no. 1, pp. 23-57, 2017.

[47] S. Garcia, A, Fernandez, J, Luengo and 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.

[48] J. C. Bansal and H. Sharma, "Cognitive learning in differential evolution and its application to model order reduction problem for single-input single-output systems", Memetic Computing., vol. 4, no. 3, pp. 209-229, 2012.

[49] J. C. Bansal, H. Sharma and K. V. Arya, "Model Order Reduction of Single Input Single Output Systems Using Artificial Bee Colony Optimization Algorithm", Nature Inspired Cooperative Strategies for Optimization (NICSO 2011) Studies in Computational Intelligence, vol. 387, pp. 85-100, 2011.

[50] T. Lucas, "Continued-fraction expansion about two or more points: a flexible approach to linear system reduction", Journal of the Franklin Institute, vol. 321, no. 1, pp. 49–60, 1986.

[51] J. Pal, "An algorithmic method for the simplification of linear dynamic scalar systems", International Journal of Control, vol. 43, no. 1, pp. 257–269, 1986.

[52] L. Aguirre, "The least squares padé method for model reduction", International Journal of Systems Science, vol. 23, no. 10, pp. 1559–1570, 1992.

[53] Y. Shamash, "Linear system reduction using pade approximation to allow retention of dominant modes", International Journal of Control, vol. 21, no. 2, pp. 257–272, 1975.

[54] A. Eydgahi, E. Shore, P. Anne, J. Habibi and B. Moshiri, "A matlab toolbox for teaching model order reduction techniques", In: International conference on engineering education, Valencia, Spain, pp. 1–7, 2003.