Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 9 (20): 3652-3661, 2009
ISSN 1812-5654

Multi-Objective Differential Evolution Algorithm for Solving Engineering Problems

J.A. Adeyemo and F.A.O. Otieno
Department of Civil Engineering, Tshwane University of Technology,
Private Bag X680, Pretoria, 0001, South Africa

Abstract: Many engineering problems are characterized with multi-objectives. Differential Evolution (DE), an Evolutionary Algorithm (EA), known to be fast and robust in numerical optimization is extended to multi-objective problems in this study. The new algorithm named Multi-objective Differential Evolution Algorithm (MDEA) adjusts the selection scheme of traditional DE to solve multi-objective problems. The algorithm also modifies the domination criteria for the population. The offspring generated in subsequent generations are improved before domination check is performed on the population in the final generation. Moreover, trial solution replaces the target solution if it is better or equal in all the objectives. The proposed algorithm is coded in MATLAB 7.0 and has been successfully applied to five common test problems and an engineering cantilever design problem. Good spread of quality Pareto optimal solutions are achieved. The algorithm produces more Pareto optimal solutions than the previous algorithms and retains the fast convergence and diversity exhibited by DE in global optimization. The algorithm is a good choice for solving many practical engineering problems with ease.

Key words: Differential evolution, multi-objective, optimization, evolutionary algorithm, Pareto front, diversity, convergence

INTRODUCTION

Evolutionary Algorithms (EAs) are some of the most widely used algorithm for solving Multi-objective Optimization Problems (MOPs). EAs are applied to a wide range of problems from science to engineering design problems. They are powerful optimization algorithms for solving complex real-world MOPs in a single run. The primary reason why a problem has a multi-objective formulation is that it is not possible to have a single solution that optimizes all objectives. Therefore, an algorithm that gives a large number of alternative solutions lying on or near the Pareto-optimal front is of great practical value. In multi-objective optimization, there may not exist a solution that is best with respect to all objectives, instead, there are equally good solutions that are known as Pareto optimal solutions (Deb, 2001). A Pareto optimal set of solutions is such that when we go from any one point to another in the set, at least one objective function improves and at least one other worsens (Babu and Jehan, 2003). Neither of the solutions dominates over each other and all the sets of decision variables on the Pareto are equally good. Evolutionary algorithms have the ability to find multiple Pareto optimal solutions in one single simulation run because of their population-approach. There are two goals in multi-objective optimization. The goals are to discover solutions as close to the Pareto-front as possible and to find solutions as diverse as possible in the obtained non-dominated front.

Several studies have extended evolutionary algorithm to solve multi-objective numerical optimization problems (Deb, 2001; Deb *et al.*, 2002; Madavan, 2002; Babu and Jehan, 2003; Xue *et al.*, 2003; Rakesh and Babu, 2005; Babu *et al.*, 2005; Robic and Filipic, 2005; Santana-Quintero and Coello, 2005; Fan *et al.*, 2006; Reddy and Kumar, 2007). Over the past decade, a number of Multi-objectives Evolutionary Algorithms (MOEAs) have been suggested (Deb, 2001). Recently, researchers have extended Differential Evolution (DE) which is a family of EA to solve MOPs. It has been successfully used in solving single objective problems. Differential evolution is a very simple population-based, stochastic function minimizer and very powerful at the same time. DE handles continuous, discrete and integer variables. In terms of constraints, DE can handle multiple constraints. It has the

Corresponding Author: J.A. Adeyemo, Department of Civil Engineering, Tshwane University of Technology, Private Bag X680, Pretoria, 0001, South Africa Tel: +27123824434 Fax: +27123825226

advantages of simple structure, ease of use, speed and robustness. It does not make use of some probability distribution function in order to introduce variations into the population but instead, it uses the difference between randomly selected factors (individuals) as source of random variation for a third vector (individual), referred to as the target vector. Hence, the trial solutions which will compete with the parent solution are generated by adding the weighted difference vectors to the target vector.

Pareto Differential Evolution (PDE) proposed by (Abbass and Sarker, 2002) was the first algorithm to extend DE to MOPs. The PDE was compared to SPEA (Zitzler and Thiele, 1999) on two test problems and PDE was found to outperform it. Other studies by Madavan (2002), Xue *et al.* (2003) and Robic and Filipic (2005) have proposed Pareto Differential Evolution Approach (PDEA), Multi-Objective Differential Evolution (MODE) and Differential Evolution for Multi-Objective Optimization (DEMO), respectively. All these algorithms perform well on test problems.

Multi-Objective Differential Evolution (MODE) was proposed by Babu and Jehan (2003) and Babu *et al.* (2005). In the algorithm, the dominated solutions are removed from the population in each generation. Only the non-dominated solutions are allowed to undergo DE operations of mutation, crossover and selection. So, at each generation, the population size reduces. The offspring are placed into the population if they dominate the parents. The algorithm worked well on the test problems but the number of non-dominated solutions generated was few. The dominated solutions that are removed from the population initially can still be improved using the DE operations. During mutation stage, they may be combined with other vectors to produce offspring that may replace them instead of just removing them. In this way, at the end of the generation, the number of non-dominated solutions is increased with quality members giving wider Pareto optimal fronts. In the original DE with single objective, most of the initially generated solutions are infeasible. They undergo mutation and crossover before they are improved to be feasible. In the same way, the solutions in each generation should be allowed to mutate, recombine and undergo crossover. The solutions in the final generation are the only ones to be checked for domination. Therefore, domination check is done once instead of doing it in all the generations thereby reducing the number of function evaluation.

The algorithm proposed by Fan *et al.* (2006) is implemented by modifying the selection scheme of DE. In their selection scheme, the trial population is compared with its counterpart in the current population. If the trial candidate dominates the current population member it will survive from the tournament selection to the population of the next generation and replaces the current population otherwise the current population is retained. They suggest that if the trial solution is worse than the target solution in any one of the objectives, it should be discarded.

A new DE algorithm for Multi-Objective Optimization Problems (MOOPs) is proposed in this study and it is called multi-objective differential evolution algorithm (MDEA). The DE algorithm proposed is based on the existing DE algorithm proposed by Price and Storn (1997). The only difference is its implementation of multi-objectives.

MULTI-OBJECTIVE DIFFERENTIAL EVOLUTION ALGORITHM (MDEA)

MDEA combines the advantages of algorithm by Fan *et al.* (2006) and Babu and Jehan (2003) while overcoming the shortcomings of the algorithms. In this way, MDEA runs faster with more quality Pareto optimal solutions. The description of the MDEA is as follows. The vectors are randomly generated to create initial vectors and solutions to the problem. The generated solutions are allowed to undergo mutation, crossover and selection for the number of generations chosen. The solutions that evolve in the final generation are checked for domination and the dominated solutions are removed. The selection procedure of MDEA is similar to (Fan *et al.*, 2006). The trial solution survives to the next generation if its objective function is better or equal in all the objectives to the target solution. MDEA will produce many non-dominated solutions on the Pareto front than the algorithm by Fan *et al.* (2006) and Babu and Jehan (2003). Moreover, the algorithm by Fan *et al.* (2006) has not been modified to handle constraints except bound constraints. MDEA can handle multiple constraints. If any of the constraints is violated, a high value (10^8) is added to the objective function to make the solution infeasible (Deb, 2001). In this way, the solution will not be selected when compared with other solutions because of high value. MDEA uses a constant population structure like traditional DE. The final population is the same like the initial population. So the number of non dominated solutions are less than or equal to population size. This is an excellent advantage of MDEA over algorithms by Fan *et al.* (2006) and Babu and Jehan (2003). Though the proposed MDEA can be used on any strategy, the strategy used in this study is DE/rand/1/bin which is the most widely used of all the ten strategies of DE.

The proposed MDEA methodology can be summarized in the following pseudo code steps:

(1) Input the required DE parameters like population size (NP), crossover constant (CR), scaling factor (F), maximum generation, number of objectives, bound constraints etc.
(2) Initialize all the vectors randomly in the limit of bound constraints.
(3) Set the generation counter, G = 0
(4) Perform mutation and crossover operations as explained in the DE algorithm (Price and Storn, 1997) on all the population members.

- For each parent, select three distinct vectors randomly from the current population. The selected vectors must not be the parent vector. These vectors combine to produce an offspring. So in DE, there are 3 parents that mutate to produce one offspring
- Calculate new mutation vector using the expression, $V_i(g) = X_{i3}(g) + F^*(X_{i1}(g) - X_{i2}(g))$
- Perform crossover using any of the two crossover method. Binary crossover method is used because of the strategy DE/rand/1/bin used in MDEA in this study

(5) Evaluate each member of the population and check if it is better or equal to the parents. Replace the parents with offspring in the next generation if the offspring is better or equal to the parents otherwise, the parents proceed to the next generation.
(6) Increase the generation counter, G, by 1. i.e., G = G+1. If G < GMAX, then go to step 4 above and repeat mutation, crossover and selection. If G = GMAX, then goto step 7.
(7) Remove the dominated solutions in the last generation. A solution is dominated if there is another solution which is better than it in all the objectives. The method used here is naïve and slow suggested by Deb (2001).
(8) Output the non-dominated solutions.

The steps above are followed, using the pseudo code below, to code the algorithm in MATLAB 7.0 (The MathWorks Inc., USA) and is executed on a 1.7GHz, 2GB RAM PC and tested on some test problems below to demonstrate its ability to solve MOPs.

The Pseudo Code for Multi-Objective Differential Evolution Algorithm (MDEA):

```
*Initialize the values of D, NP, Cr, F, k (number of objective functions) and maximum generation (MAXGEN)
*input the boundary constraints of the problem
*Initialize all the vectors of the population randomly
    For i = 1 to NP
        For j = 1 to D
            $X_{i,j}$ = $Dlower_j$+random number[0,1] ×[$Dupper_j$-$Dlower_j$]
        Next j
    Next i
*Initialize gen = 1
for gen = 1 to MAXGEN
    for i = 1 to NP
        For m = 1 to k
            Evaluate the objective function values
        Next m
        Check the constraints violation
        If violated objective function value = objective function value + $8*10^8$ Deb (2001)
        For j = 1 to D
            (Mutation stage)
            Select 3 different vectors for perturbation different from i such that $i \neq r1 \neq r2 \neq r3$
            $V_{i,gen} = X_{r3,gen} + F(X_{r1,gen} - X_{r2,gen})$
            (Crossover Stage)
            $u_{j,i,gen} = \begin{cases} v_{j,i,gen} & \text{if radn } j \leq Cr \text{ or } j = n \\ x_{j,i,gen} & \text{otherwise} \end{cases}$
            and $n \in \{1, \ldots, D\}$
            next j
            (Selection stage)
            Start comparing vectors $U_{i,gen}$ and $X_{i,gen}$
            For m = 1 to k
                evaluate kth objective function, $f_k(U_{i,gen})$
                If $f_k(U_{i,gen}) = f_k(X_{i,gen})$
                    Select vector $U_{i,gen}$, the trial vector
                    $(X_{i,gen+1}) = (U_{i,gen})$
                    goto next step
            Else
                    Next m
            endif
            select $X_{i,gen}$ the current population member for the next generation
            $(X_{i,gen+1}) = (X_{i,gen})$
        next m
    next i
next gen
*remove the dominated solutions from the last generation using naïve and slow method proposed by Deb (2001)
print the results
```

EVALUATION AND RESULTS

Test problems: Deb (1999) suggested systematic ways of developing test problems for multi-objective optimization. Zitzler *et al.* (2000) followed these steps and suggested well known 6 test problems. Five of the test problems (ZDT1, ZDT2, ZDT3, ZDT4 and ZDT5) which are common benchmark domains used in literatures (Deb *et al.*, 2002; Madavan, 2002; Xue *et al.*, 2003), are chosen for evaluating the performance of our methods. These test problems have no constraints. Each of them illustrates a different class of problems. All the problems have two objectives, $f_1(x)$ and $f_2(x)$ which are minimized. The description of the test functions are shown in Table 1.

Performance measures: The common trend in the development of many successful solution methodologies, including multi-objective evolutionary algorithms is to compare these in terms of their performance on various test problems. In a multi-objective optimization, there are two goals unlike single-objective optimization. The two goals are; discover solutions as close to the Pareto-optimal solutions as possible and find solutions as diverse as possible in the obtained non-dominated front. A good multi-objective evolutionary algorithm will be known if both goals are satisfied (Deb, 2001). Therefore, there is a need to have at least two performance metrics for adequately evaluating both goals of multi-objective optimization. One performance metric evaluates the progress towards the Pareto-optimal front and the other evaluates the spread of solutions. Many performance metrics have been proposed in the literatures. The descriptions of three common metrics used are given below though in this study, two performance metrics will be used to evaluate MDEA. They are convergence metric Υ and diversity metric Δ.

Convergence metric Υ: This metric measures the distance between the obtained non-dominated front, Q and the set, P* of the Pareto-optimal solutions. It is defined as:

$$\Upsilon = \frac{\sum_{i=1}^{|Q|} d_i}{|Q|} \tag{1}$$

where, Q is the number of non-dominated vectors found by the algorithm being analyzed and d_i is the Euclidean distance (in the objective space) between the obtained non dominated front Q and the nearest member in the true Pareto front P.

Table 1: Test problems used in this study

Problems	n	Variable bounds	Objective functions	Optimal solutions	Comments
ZDT1	30	[0,1]	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-\sqrt{\left(\frac{x_1}{g(x)}\right)}\right]$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$	$x\in[0,1]$ $x_i=0$ I=2,…,n	Convex
ZDT2	30	[0,1]	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-\left[\frac{x_1}{g(x)}\right]^2\right]$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$	$x\in[0,1]$ $x_i=0$ I=2,…,n	Nonconvex
ZDT3	30	[0,1]	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-\left[\sqrt{\frac{x_1}{g(x)}}\right]-\frac{x_1}{g(x)}\sin(10\pi x_1)\right]$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$	$x\in[0,1]$ $x_i=0$ I=2,…,n	Convex, disconnected
ZDT4	10	$x_1\in[0,1]$ $x_i\in[-5,5]=2,\ldots n$	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-\sqrt{\left(\frac{x_1}{g(x)}\right)}\right]$ $g(x)=1+10(n-1)+\sum_{i=2}^{n}\left(x_i^2-10\cos(4\pi x_i)\right)$	$x\in[0,1]$ $x_i=0$ I=2,…,n	Nonconvex
ZDT6	10	[0,1]	$f_1(x)=x_1$ $f_2(x)=g(x)\left[1-\left(\frac{f_1(x)}{g(x)}\right)^2\right]$ $g(x)=1+\frac{9}{n-1}\sum_{i=2}^{n}x_i$	$x\in[0,1]$ $x_i=0$ I=2,…,n	Nonconvex, nonuniformly spaced

Generational Distance (GD): This metric measures the average distance of the solutions of Q obtained from the set P* of the Pareto-optimal solutions. It is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^{|Q|} d_i^2}}{|Q|} \quad (2)$$

where, d_i is the Euclidean distance (measure in the objective space) between non-dominated front Q and the nearest member in the true Pareto-front P. It should be noted that an algorithm having a small value of GD is better.

Diversity metric Δ: This metric measures the extent of spread achieved using the non-dominated solutions. It is defined as:

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{|Q|-1}\left(d_i - \bar{d}\right)}{d_f + d_l + (|Q| - 1)\bar{d}} \quad (3)$$

where, d_i is the Euclidean distance (measured in the objective space) between consecutive solutions in the obtained non-dominated front Q and $\bar{d}$ is the average of these distances. The parameter d_f and d_l are the Euclidean distances between the extreme solutions of the Pareto front P* and the boundary solution of the obtained front Q. Also an algorithm having a smaller value of diversity metric Δ is better.

Cantilever design problem: MDEA is further tested on an engineering design problem of cantilever design (Deb, 2001). Figure 1 shows a schematic representation of a cantilever beam. Figure 1 presents the end load P, the length l and the diameter d. The problem has 2 decision variables of diameter (d) and length (*l*). The beam will carry an end load P. There are 2 conflicting objectives that should be minimized; the weight f_1 and end deflection f_2. Minimizing the weight, f_1, will result in an optimum solution that will have small dimensions of d and l. If the dimensions are small, the beam will not be adequately rigid and the end deflection of the beam will be large. If on the other hand, the beam is minimized for end deflection, the dimensions of the beam will be large, thereby making the weight of the beam to be large. There are 2 constraints in this design problem. They are; the maximum stress, σ_{max} is less than the allowable strength S_y and the end deflection, δ is smaller than a specified limit of δ_{max}. The constrained optimization problem is formulated as follows:

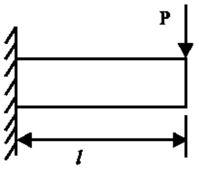

Fig. 1: A schematic representation of a cantilever beam

Objective function 1: Minimization of weight

$$\text{Minimize } f_1 = \frac{\rho\pi d^2 l}{4} \quad (4)$$

Objective function 2: Minimization of end deflection

$$\text{Minimize } f_2 = = \frac{64Pl^3}{3E\pi d^4} \quad (5)$$

The objective functions 2 are subjected to the following constraints:

Constraint 1: Maximum stress

The maximum stress should be less than the allowable strength, S_y:

$$\sigma_{max} \leq S_y \quad (6)$$

Constraint 2: End deflection

The end deflection, δ should be smaller than a specified limit of δ_{max}

$$\delta \leq \delta_{max} \quad (7)$$

Bound constraints:

$$10 \leq d \leq 50 \text{ mm} \quad (8)$$

$$200 \leq l \leq 1000 \text{ mm} \quad (9)$$

where, ρ, P, d and *l* are the density, force, diameter and length respectively. The maximum stress is calculated as follows:

$$\sigma_{max} = \frac{32Pl}{\pi d^3} \quad (10)$$

The following parameter values are used:
$\rho = 7800$ kg m^{-3}, P = 1000 N, E = 207 GPa, S_y= 300 MPa and δ_{max}= 5 mm

RESULTS

MDEA was investigated for lower values of CR and F when the recommended values of 0.8 and 0.9 failed to achieve convergence in the high dimensional functions. MDEA performs better with lower values of CR and F than the values suggested in the practical advice on DE that F is to be a little lower or higher than 0.8 and CR to be 0.9 (Price and Storn, 2008). In test problems ZDT1, ZDT2 and ZDT3, the values of F and CR of 0.6 and 0.5, respectively give faster convergence and better diversity. The value of NP used in all the test problems is 100. The values of CR and F go down to as low as 0.3 and 0.35, respectively to make the algorithm robust enough for the test problems ZDT4 and ZDT6. The number of non-dominated solutions generated for all the test problems is 100. Therefore, all the solutions in the final generation (100) are non-dominated. This is an advantage of this algorithm over other EAs.

The statistics of the test problems are presented in Table 2-6. The convergence and diversity metric for algorithms NSGA-II (real coded), NSGA-II (binary coded), SPEA, PAES, PDEA, MODE, SDE and DEMO are taken from the literatures (Deb *et al.*, 2002; Madavan, 2002; Robic and Filipic, 2005; Xue *et al.*, 2003). They are used to compare the effectiveness of MDEA.

In Fig. 2a-c, the dominated solutions for NSGA, DEMO and MDEA are presented. It is found that the non dominated solutions for MDEA are of more quality than those of NSGA and DEMO. Though the generated Pareto optimal front is similar to NSGA and DEMO, MDEA outperforms them. The convergence metric results in Table 2 show that MDEA has lower values in the metric than all the other algorithms except NSGA-II (binary coded). Also the diversity metric for MDEA is lower than all the other algorithms. The value is close to PDEA. This shows that MDEA is better in diversity than other algorithms.

Figure 3 shows the Pareto optimal fronts for NSGA, DEMO and MDEA for test problem ZDT2. It is found from

Table 2: Statistics of the results on test problem ZDT1

Algorithm	Convergence metric	Diversity metric
NSGA-II (real coded)	0.033482±0.004750	0.390307±0.001876
NSGA-II (binary coded)	0.000894±0.000000	0.463292±0.041622
SPEA	0.001799±0.000001	0.784525±0.004440
PAES	0.082085±0.008679	1.229794±0.000742
PDEA	N/A	0.298567±0.000742
MODE	0.005800±0.000000	N/A
SDE	0.002741±0.000385	0.382890±0.001435
DEMO	0.001132±0.000136	0.319230±0.031350
MDEA	0.000921±0.000005	0.283708±0.002938

N/A: Not available

Table 3: Statistics of the results on test problem ZDT2

Algorithm	Convergence metric	Diversity metric
NSGA-II (real coded)	0.072391±0.031689	0.430776±0.004721
NSGA-II (binary coded)	0.000824±0.000000	0.435112±0.024607
SPEA	0.001339±0.000000	0.755184±0.004521
PAES	0.126276±0.036877	1.165942±0.007682
PDEA	N/A	0.317958±0.001389
MODE	0.005500±0.000000	N/A
SDE	0.002203±0.000297	0.345780±0.003900
DEMO	0.000780±0.000035	0.326821±0.021083
MDEA	0.000640±0.000000	0.450482±0.004211

N/A: Not Available

Table 4: Statistics of the results on test problem ZDT3

Algorithm	Convergence metric	Diversity metric
NSGA-II (real coded)	0.114500±0.004940	0.738540±0.019706
NSGA-II (binary coded)	0.043411±0.000042	0.575606±0.005078
SPEA	0.047517±0.000047	0.672938±0.003587
PAES	0.023872±0.000010	0.789920±0.001653
PDEA	N/A	0.623812±0.000225
MODE	0.021560±0.000000	N/A
SDE	0.002741±0.000120	0.525770±0.043030
DEMO	0.001236±0.000091	0.328873±0.019142
MDEA	0.001139±0.000024	0.299354±0.023309

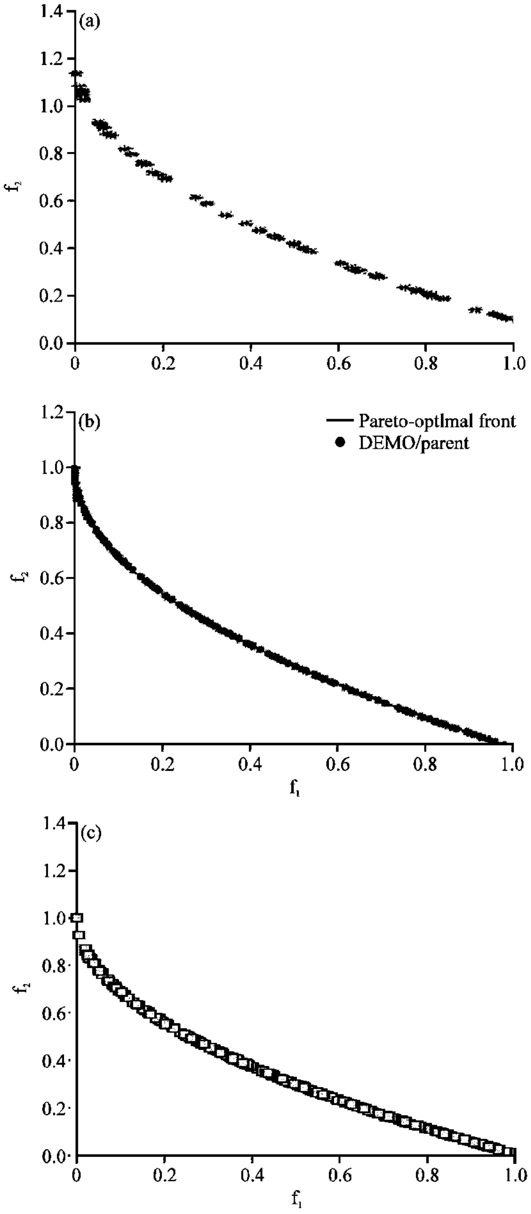

Fig. 2: The results of test problem ZDT1 using (a) NSGAII, (b) DEMO and (c) MDEA

Fig. 3a-c that MDEA is better in the uniform spread of non dominated solutions than NSGA. Though they all have the same shape, MDEA is better in diversity than NSGA but comparable to DEMO. Table 3 shows the results of the convergence and diversity metrics. It is found that MDEA is better in convergence than all the other algorithms but very close to DEMO. From the results of diversity metrics, MDEA is worse in diversity than NSGA-II (real coded), NSGA-II (binary coded), PDEA, SDE and DEMO but better than SPEA and PAES. The reason MDEA's worse diversity in this test problem is that MDEA converged faster than the other algorithms. So the diversity is sacrificed for convergence in the test problem ZDT2.

It is found in the Fig. 4a-c that MDEA outperforms both NSGA and DEMO. From the convergence and diversity results in Table 4, it is found that the value of convergence metric for MDEA is 0.001139. This value is lower than the ones obtained for other algorithms. DEMO's convergence metric value is the closest to MDEA's. Also the diversity metric gives 0.299354 which is the lowest for all the algorithms. DEMO obtains the closest value of 0.328873. Therefore, MDEA is the best algorithm for test problem ZDT3.

From the Fig. 5a-c, MDEA outperforms NSGA but comparable to DEMO. This is also confirmed with the results of convergence and diversity metrics in Table 5. Comparison with other algorithms shows that MDEA

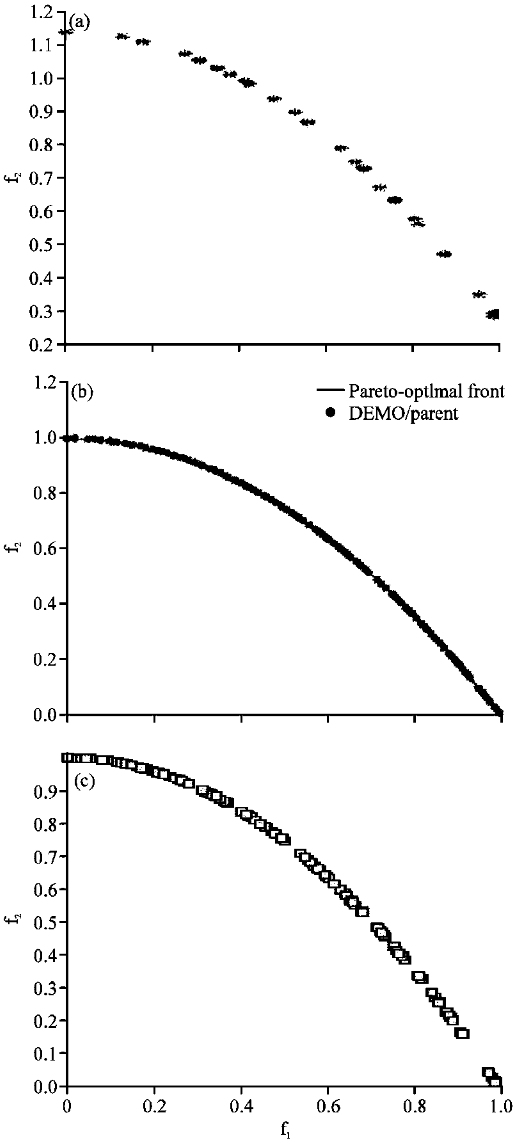

Fig. 3: The results of test problem ZDT2 using (a) NSGAII, (b) DEMO and (c) MDEA

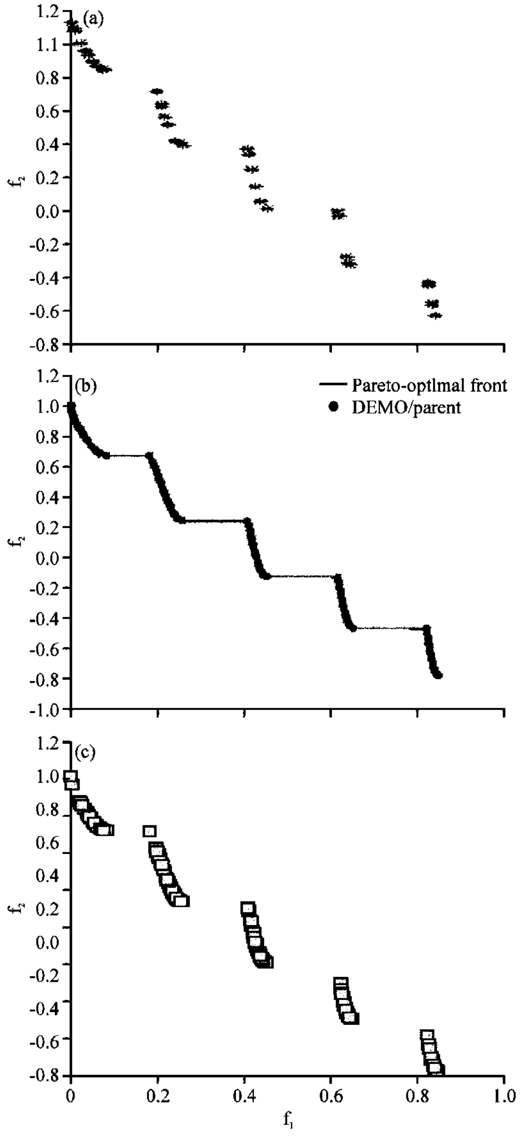

Fig. 4: The results of test problem ZDT3 using (a) NSGAII, (b) DEMO and (c) MDEA

outperforms all of them except DEMO even in hard optimization (ZDT4) in both metrics.

Test problem ZDT6 Pareto optimal front is presented in Fig. 6a-c. MDEA is comparable to DEMO but better than NSGA in the figure. Results in Table 6 shows that MDEA outperforms all other algorithms in test problem ZDT6 in closer to those of SDE than other algorithms in both metrics. Therefore, MDEA is comparable to SDE in test problem ZDT6.

In Fig. 2-6, MDEA has achieved the two goals of multi-objective optimization which are convergence to the true Pareto-front and uniform spread of solutions along the front (Deb, 2001). From the figures, MDEA's results are comparable to NSGA II and DEMO/parent. In all the Fig. 2-6, MDEA curve is comparable to that of DEMO but better than NSGA II in terms of convergence and diversity. It can be concluded from these results that MDEA's performance is encouraging. The results have shown that MDEA is a good optimizer as proposed in this study. It is able to solve multi-objective optimization problems. The performance is comparable to other multi-objective Eas.

Comparison of the MDEA to MODE and NSGA on engineering test problem of cantilever design is shown in Fig. 7a and b. It is found that the non-dominated solutions generated by MDEA are comparable to those of MODE and NSGA. MDEA produces quality non dominated solutions along the Pareto front. This shows that MDEA can perform well on real-world engineering problems.

Table 5: Statistics of the results on test problem ZDT4

Algorithm	Convergence metric	Diversity metric
NSGA-II (real coded)	0.513053±0.118460	0.702612±0.064648
NSGA-II (binary coded)	3.227636±7.307630	0.479475±0.009841
SPEA	7.340299±6.572516	0.798463±0.014616
PAES	0.854816±0.527238	0.870458±0.101399
PDEA	N/A	0.840852±0.035741
MODE	0.638950±0.500200	N/A
SDE	0.100100±0.446200	0.436300±0.110000
DEMO	0.041012±0.063920	0.407225±0.094851
MDEA	0.048962±0.536358	0.406382±0.062308

N/A: Not available

Table 6: Statistics of the results on test problem ZDT6

Algorithm	Convergence metric	Diversity metric
NSGA-II (real coded)	0.296564±0.013135	0.668025±0.009923
NSGA-II (binary coded)	7.806798±0.001667	0.644477±0.035042
SPEA	0.221138±0.000449	0.849389±0.002713
PAES	0.085469±0.006664	1.153052±0.003916
PDEA	N/A	0.473074±0.021721
MODE	0.026230±0.000861	N/A
SDE	0.000624±0.000060	0.361100±0.036100
DEMO	0.000642±0.000029	0.458641±0.031362
MDEA	0.000436±0.000055	0.305245±0.019407

N/A: Not available

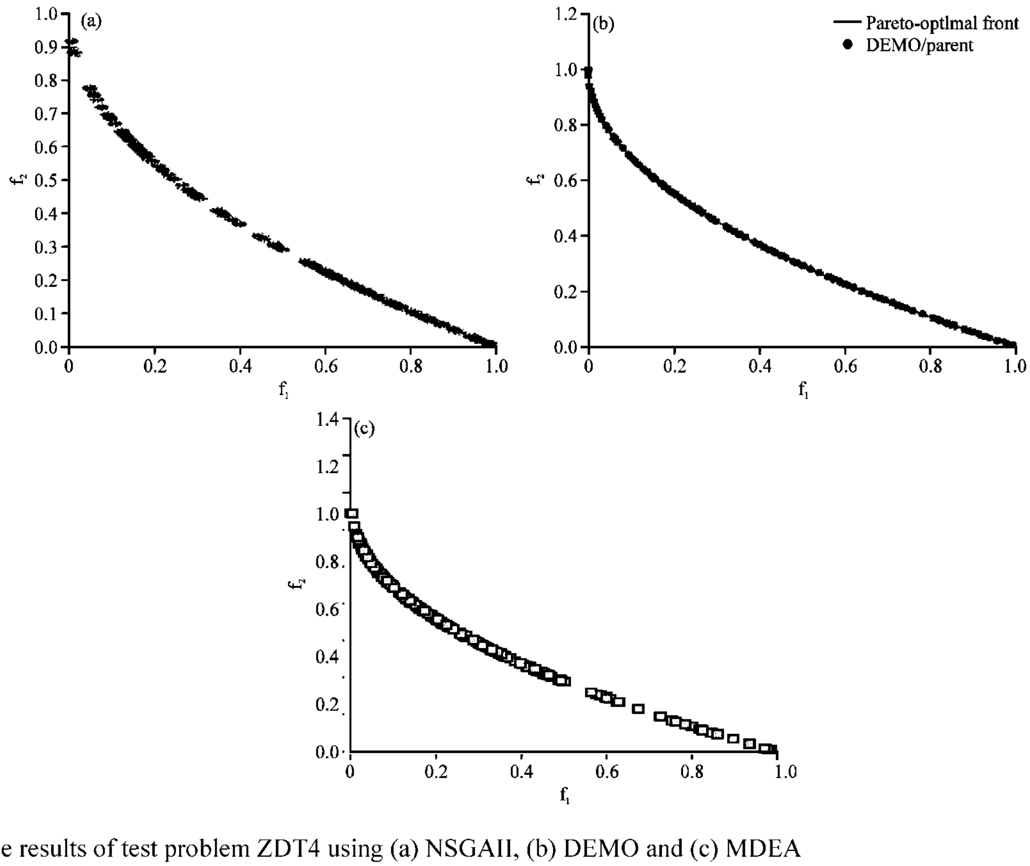

Fig. 5: The results of test problem ZDT4 using (a) NSGAII, (b) DEMO and (c) MDEA

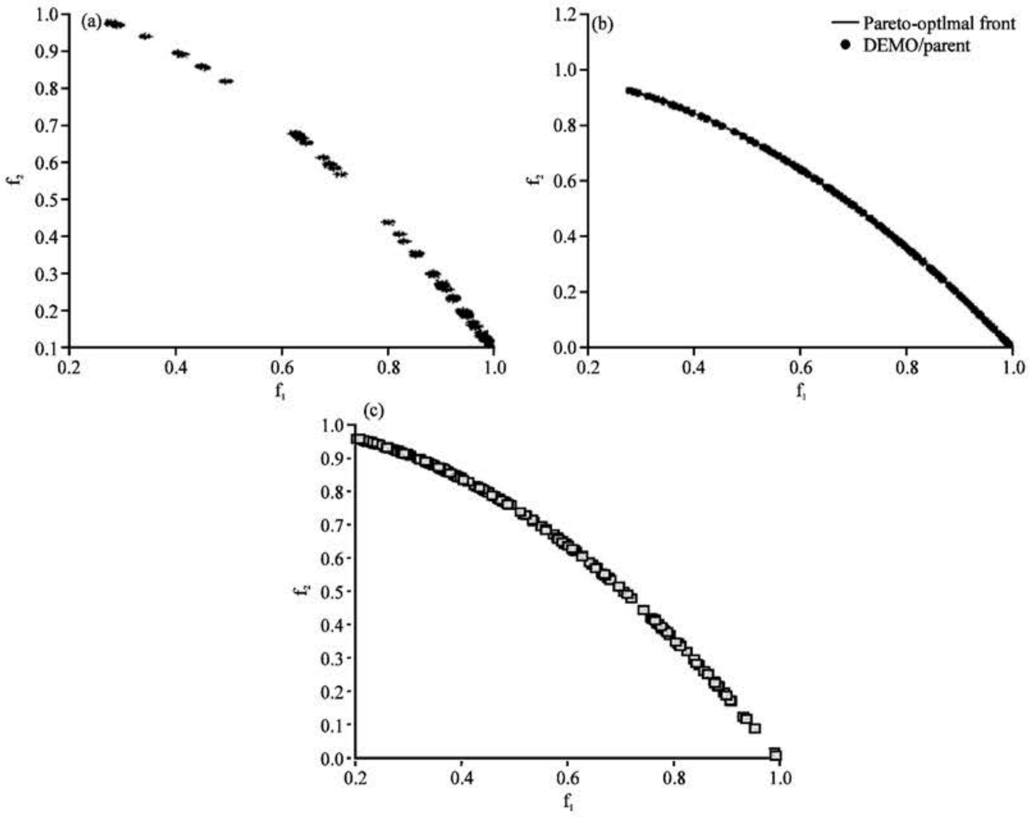

Fig. 6: The results of test problem ZDT6 using (a) NSGAII, (b) DEMO and (c) MDEA

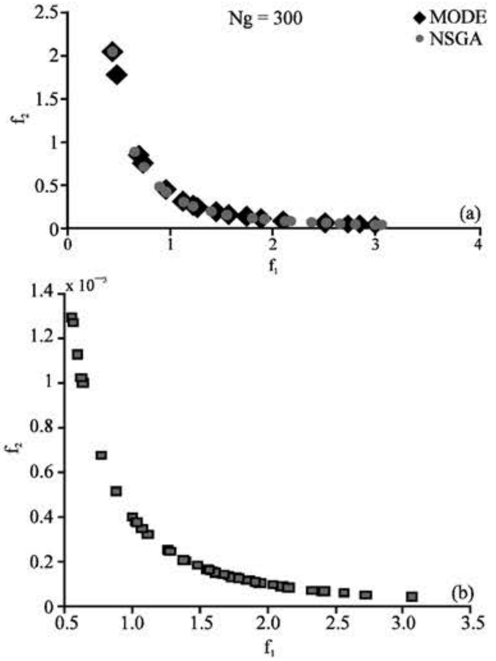

Fig. 7: The results of (a) MODE and (b) NSGA for cantilever problem

CONCLUSION

Multi-Objective Differential Evolution Algorithm (MDEA) is illustrated in this study. The selection scheme in the original DE is improved to accommodate multi-objective optimization. Also, dominated solutions are removed in the last generation only instead of removing them in all the generations thereby reducing the number of function evaluation. This allows the dominated solutions in every generation to be improved by mutation, crossover and selection operations of DE. It increases the number of non-dominated solutions giving us wider Pareto optimal fronts with better diversity. Moreover, DE has been shown to be capable of solving multi-objective high dimensional problem with few control parameters. The advantage of traditional DE which is ease of use is also applicable to MDEA. The test problems used to validate the algorithm show that MDEA is a good optimizer. The algorithm gives good spread of solutions maintaining diversity and convergence. At higher iterations, we have uniform distribution of solutions. Therefore, MDEA is a good alternative for solving engineering problems and other multi-objective problems. MDEA as a family of DE can also handle continuous, discrete, integer variables and multiple constraints.

REFERENCES

Abbass, H.A. and R. Sarker, 2002. The Pareto differential evolution algorithm. Int. J. Art Int. Tools, 11: 531-552.

Babu, B.V. and M.M. Jehan, 2003. Differential evolution for multi-objective optimization. Proceedings of IEEE Congress on Evolutionary Computation, Dec. 8-12, IEEE Press, Canberra, Australia, pp: 2696-2703.

Babu, B.V., J.H.S. Mubeen and G.C. Pallavi, 2005. Multi-objective differential evolution (MODE) for optimization of adiabatic styrene reactor. Chem. Eng. Sci., 60: 4822-4837.

Deb, K., 1999. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. Evol. Comput., 7: 205-230.

Deb, K., 2001. Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley and Sons, Chichester, UK., ISBN: 047187339X.

Deb, K., A. Pratap, S. Agarwal and T. Meyarivan, 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Trans. Evol. Comput., 6: 182-197.

Fan, H.Y., J. Lampinen and L. Yeshayahou, 2006. An easy-to-implement differential evolution approach for multi-objective optimizations. Eng. Comput. Int. J. Computer-Aided Eng. Software, 23: 124-138.

Madavan, N.K., 2002. Multi-objective optimization using a pareto differential evolution approach. Proceedings of the Evolutionary Computation on 2002, May 12-17, IEEE Computer Society, Washington, DC, USA., pp: 1145-1150.

Price, K. and R. Storn, 1997. Differential evolution-a Sample and efficient heuristic optimization over continuous spaces. J. Global Optimization, 11: 341-359.

Price, K. and R. Storn, 2008. Differential evolution (DE) for continuous function optimization. http://www.ICSI.Berkeley.edu/~storn/code.html.

Rakesh, A. and B.V. Babu, 2005. Non-dominated sorting differential evolution (NSDE): An extension of differential evolution for multi-objective optimization. Proceedings of the 2nd Indian International Conference on Artificial Intelligence, Dec. 20-22, Pune, India, pp: 1428-1443.

Reddy, M.J. and D.N. Kumar, 2007. Multiobjective differential evolution with application to reservoir system optimization. J. Comput. Civil Eng., 21: 136-146.

Robic, T. and B. Filipic, 2005. DEMO: Differential evolution for multiobjective optimization. Proceedings of 3rd International Conference on Evolutionary Multi-Criterion Optimization, March 9-11, Guanajuato, Mexico, pp: 912-912.

Santana-Quintero, L.V. and C.A.C. Coello, 2005. An algorithm based on differential evolution for multi-objective problems. Int. J. Comput. Intel. Res., 1: 151-169.

Xue, F., A.C. Sanderson and R.J. Graves, 2003. Pareto-based multi-objective evolution. Proceedings of Congress on Evolutionary Computation (CEC), Dec. 8-12, IEEE Press, Canberra, Australia, pp: 862-869.

Zitzler, E. and L. Thiele, 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput., 3: 257-271.

Zitzler, E., K. Deb and L. Thiele, 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. Evol. Comput., 8: 173-195.