Stochastic Modelling and Computational Sciences
Vol. 1 No. 2 (December, 2021)

Received: 21st June 2021 *Revised:19th September 2021* *Accepted: 09th October 2021*

Modified Diffrential Evolution for Multi Objective Optimization

Din Sung Lee, Chungbuk National University, Korea

Abstract

Most of the real world optimization problems have more than one objective to be optimized, and hence they are called Multi-Objective Optimization Problems (MOOPs). Evolutionary algorithms are gaining popularity for solving MOOPs due to their inherent advantages. In this paper, MDE-Modified Differential Evolution (Angira and Babu, 2005b) is extended for solving MOOPs and we call this extended algorithm as Modified Non-dominated Sorting Differential Evolution (MNSDE). The proposed algorithm is tested on two different benchmark test problems. Also, the effect of various key parameters on the performance of MNSDE is studied and compared with Non-dominated Sorting Differential Evolution (NSDE)-an extension of Differential Evolution for solving MOOPs.

1. INTRODUCTION

The field of search and optimization has changed over the last few years by the introduction of a number of non-classical, unorthodox and stochastic search and optimization algorithms. Ideally, multi-objective optimization problems require multiple trade-off solutions (a set of Pareto optimal solutions) to be found. The presence of multiple conflicting objectives makes the problem interesting to solve. Due to multiple conflicting objectives, no single solution can be termed as an optimum solution. Therefore, the resulting multi-objective optimization problem resorts to a number of trade–off optimal solutions. Classical optimization methods can at best find one solution in a single run; on the other hand evolutionary algorithms can find multiple optimal solutions in a single run due to their population based search approach.

Many engineering applications involve multiple criteria, and recently, the exploration of Evolutionary Multi-objective Optimization (EMO) techniques has

Key words: *Optimization; Multi-Objective Optimization; Evolutionary Algorithm; Differential Evolution; Modified Differential Evolution (MDE); Modified Non-dominated Sorting Differential Evolution (MNSDE); Systems Engineering; Design.*

increased (Coello, 1999). The ideal approach for a multi-objective problem is the one that optimizes all conflicting objectives simultaneously. Evolutionary algorithms inherently explore a set of possible solutions simultaneously. This characteristic enables the search for an entire set of Pareto optimal solutions in a single run. Additionally, evolutionary algorithms are less susceptible to problem dependent characteristics, such as the shape of the Pareto front (convex, concave, or even discontinuous), and the mathematical properties of the search space, whereas these issues are of concerns for mathematical programming techniques for mathematical tractability.

Schaffer (1985) proposed first practical approach to multi-criteria optimization using EAs, Vector Evaluated Genetic Algorithm (VEGA). After that there have been several other versions of evolutionary algorithms that attempt to generate multiple non-dominated solutions such as (Kursawe, 1991; Hajela & Lin, 1992). The concept of pareto-based fitness assignment was first proposed by Goldberg (1989), as a means of assigning equal probability of reproduction to all non-dominated individuals in the population. Fonseca & Fleming (1993) have proposed a multi-objective genetic algorithm (MOGA). Srinivas and Deb (1995) proposed NSGA, where a sorting and fitness assignment procedure based on Goldberg's version of Pareto ranking is implemented. Horn et al. (1994) proposed Niched Pareto Genetic Algorithm (NPGA) using a tournament selection method based on Pareto dominance. Knowles and Corne (2000) proposed a simple evolution strategy (ES), (1+1)-ES, known as the Pareto Archived Evolution Strategy (PAES) that keeps a record of limited non-dominated individuals. The more recent algorithms include the (Strength Pareto Evolutionary Algorithm) SPEA (Zitzler & Thiele, 1999), NSGA-II (Deb et al., 2002), Pareto-frontier Differential Evolution (Abbass et al., 2001), and Multi-Objective Differential Evolution (Xue et al., 2003; Babu & Jehan, 2003; Babu et al., 2005a, 2005b; Babu & Anbarasu, 2005; Angira & Babu, 2005a).

Previously, a few researchers (Abbass et al., 2001; Xue et al., 2003; Babu et al., 2005a, 2005b; Angira & Babu, 2005a) studied the extension of differential evolution to multi-objective optimization problem in continuous domain, but using different approach from that described in this chapter. In the present study an extension of Modified Differential Evolution (MDE) for solving MOOPs is proposed and the proposed algorithm (where same mutation & crossover scheme is used as in MDE, however the selection criterion is modified as it is being used for solving MOOPs) is tested on the two test problems. One test problem is

Schaffer's function and the other is cantilever design problem. Also, the effect of various key parameters on the performance of MNSDE is studied and compared with Non-dominated Sorting Differential Evolution (NSDE) - an extension of Differential Evolution for solving MOOPs.

2. MODIFIED NONDOMINATED SORTING DIFFERENTIAL EVOLUTION

In the previous studies it has been found that MDE took less computational time due to the use of single array of population (Angira & Babu, 2005b; Babu & Angira, 2006; Angira & Babu, 2006). In this study, MDE (Angira & Babu, 2005b; Babu & Angira, 2006; Angira & Babu, 2006) is extended for solving multi-objective optimization problems and the extended algorithm is called as MNSDE (Modified Non-dominated Sorting Differential Evolution). MNSDE is similar to NSDE (Angira & Babu, 2005a) except for the selection criterion. Also, MNSDE maintains only one set of population as against two sets in NSDE. The selection criterion used in MNSDE is different from that of NSDE and is as follows:

After mutation & crossover the trial solution is generated. Selection is made between this trial solution and target solution. If trial solution dominates the target solution, then the target solution is replaced by the trial solution in the population of current generation itself otherwise the target solution is kept as it is. The remaining procedure is same as that of NSDE. The use of single array of population in MNSDE as against two in NSDE may lead to reduction in memory and computational efforts required as is found for MDE. The pseudo code of MNSDE algorithm used in the present study is given below:

Set the values of NSDE parameters D, NP, CR and Max_gen (maximum generations).

Initialize all the vectors of the population randomly within the bounds.

for i = 1 to NP

for j = 1 to D

$X_{i,j}$ *=Lower bound+ random number* (upper bound-lower bound);*

End for

End for

Perform mutation, crossover, selection and evaluation of the objective function for trial and target vector for a specified number of generations.

While (gen < Max_gen)

{*for i = 1 to NP* /** *first for loop****/

{ *For each vector* $\mathbf{X}_i$ *(target vector), select three distinct vectors* $\mathbf{X}_a$, $\mathbf{X}_b$ *and* $\mathbf{X}_c$ *randomly from the current population other than the vector* $\mathbf{X}_i$

do

{ *r1 = random number *NP*

*r*2 = random number *NP

*r*3 = random number *NP

} *While* ($r1=i$) OR ($r2=i$) OR ($r3=i$) OR ($r1=r2$) OR ($r2=r3$) OR ($r1=r3$)

Perform mutation and crossover for each target vector $\mathbf{X}_i$ and create a trial vector, $\mathbf{X}_{t,i}$. For binomial crossover:

{ *p = random number*

$j_{rand} = int\ (rand[0,1]* \mathrm{D})+1$

for $n = 1$ to D

{ *if* ($p<CR$ or $n = j_{rand}$)

$\mathbf{X}_{t,i} = \mathbf{X}_{a,i} + F\,(\mathbf{X}_{b,i} - \mathbf{X}_{c,i})$

} *else* $\mathbf{X}_{t,i} = \mathbf{X}_{i,j}$

}

Perform selection for each target vector, $\mathbf{X}_i$ *by comparing its function value with that of the trial vector,* $\mathbf{X}_{t,i}$. *If* $\mathbf{X}_{t,i}$ *dominates* $\mathbf{X}_i$ *then replace* $\mathbf{X}_{,i}$ *with* $\mathbf{X}_{t,i}$ *otherwise discard* $\mathbf{X}_{t,i}$.

If ($\mathbf{X}_{t,i}$ *dominates* $\mathbf{X}_i$)

Replace $\mathbf{X}_{,i}$ *with* $\mathbf{X}_{t,i}$ *into current generation population*

else *discard* $\mathbf{X}_{t,i}$

} /** *End of first for loop****/

} /** End of while loop***/

Evaluate the objective functions for each vector.

for i = 1 to NP

$\mathbf{C}_{i,j} = funct_j(\ )$. $j = 1,.., $ *no of objectives*

Remove all the dominated solutions using any one of the approaches proposed by Deb (2001).

Print the results (after the stopping criteria is met).

Nondominated Sorting can be done using any of the standard approaches reported in Deb (2001). In the present study, naïve and slow approach is used. In this approach, each solution *i* is compared with every other solution in the population to check if it is dominated by any solution in the population. If no solution is found to dominate solution *i*, it is member of the non-dominated set otherwise it does not belong the non-dominated set. This is how any other solution in the population can be checked to see if it belongs to the non-dominated set.

The stopping criteria for the algorithm can be any one of the following conditions:

(a) There is no new solution added to the non-dominated front for a specified number of generations.

(b) Till the specified number of generations.

However, in this study, the second condition is used as termination criterion.

3. TEST PROBLEMS

The algorithm is tested on the following two test problems (Deb, 2001). The first problem is of one dimension while the other is of two dimensions.

Schaffer's function

Minimize $f(x) = x^2$

Minimize $g(x) = (x\text{-}2)^2$

where $-1000 < x < 1000$

Cantilever Design Problem

A cantilever design problem with two decision variables is considered, i.e., diameter (d) and length (l). The beam has to carry an end weight load P. the objectives are minimization of weight (f_1) and minimization of end deflection (f_2). The first objective will resort to an optimum solution having the smaller dimensions of d and l, so that the overall weight of the beam is minimum. Since the dimensions are small, the beam will not be adequately rigid and the end deflection of the beam will be large. On the other hand, if the beam is minimized for end deflection, the dimensions of the beam are expected to be large, thereby making the weight of the beam large.

Minimize f_1 and f_2

where $f_1(d,l) = \frac{\rho d^2 l}{4}$, and $f_2(d,l) = \delta = \frac{64Pl^3}{3E\pi d^4}$

Subject to $\sigma_{\max} \leq S_y$

$\delta \leq \delta_{\max}$

$\sigma_{\max}$ is calculated using $\sigma_{\max} = \frac{32Pl}{\pi d^3}$ and the following parameters are used

ρ = 7800 kg/m^3, P = 1 kN, E = 207 GPa, S = 300 Mpa, $\delta_{\max}$ = 5 mm;

4. RESULTS & DISCUSSION

4.1. Schaffer's Function

Many experiments have been carried out in order to test the proposed algorithm by studying the effect of *Max_gen* and *F* & *CR* for Schaffer's function. Fig. 1 shows the Pareto front using the two techniques, i.e., NSDE, and MNSDE.

Parameters used are same in both the techniques except for the *Max_gen*. For MNSDE, *Max_gen* is 100, while it is 200 for NSDE. This is done in order to

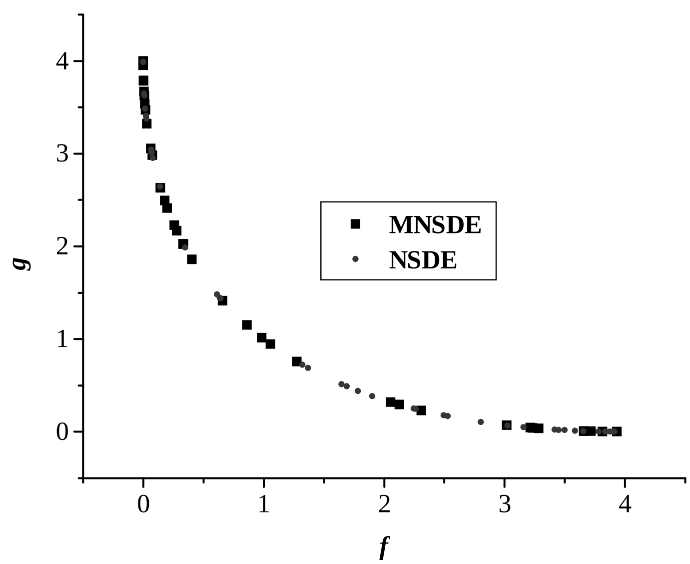


Fig. 1: Pareto front using MNSDE & NSDE

examine the effect of use of single array instead of double in NSDE. It is clear that both MNSDE & NSDE are able locate global Pareto optimal front but with different spread. It is to be noted that even though the *Max_gen* used in MNSDE is half of that used in NSDE, it is able to locate the global Pareto front.

4.1.1. Effect of Max_gen on MNSDE & its comparison with NSDE

The Fig. 2 shows the effect of Max_gen on MNSDE and its comparison with that of NSDE. From Fig. 2, it is clear that effect of *Max_gen* is same on the two techniques. Key parameters used are *F* = 0.5, *CR* = 0.5, *NP* = 100.

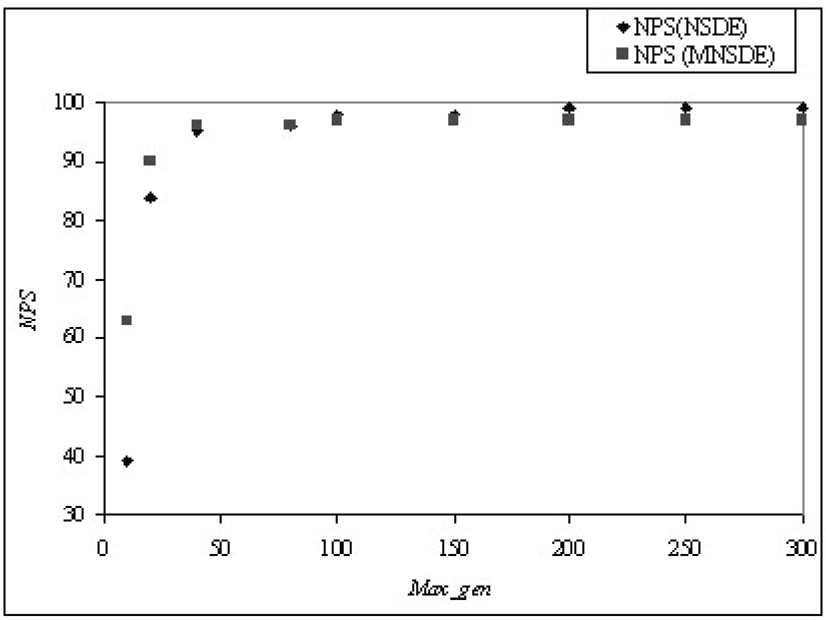


Fig. 2: Effect of *Max_gen* on *NPS* using MNSDE & NSDE

However, there is no significant change in *NPS* after *Max_gen* = 100 for MNSDE while same trend is observed for NSDE but after *Max_gen* = 200. Also maximum *NPS* obtained using NSDE is 99 as compared to 96 using MNSDE.

4.1.2. Effect of CR on MNSDE & its Comparison with NSDE

Fig. 3 shows the effect of *CR* on the performance of MNSDE. Fig. 4 shows the effect of seed and its comparison with NSDE. Keeping *F* = 0.5, *NP* = 100, and seed = 10, the *CR* value is changed from 0.1 to 1.0 in steps of 0.1. The parameters used are same for MNSDE & NSDE except *Max_gen* = 200 & 300 respectively. The effect of change in *CR* value on MNSDE is similar to that found for NSDE, i.e., there is no effect on *NPS* and Pareto front obtained (Fig. 3). Also, different seed values give different spread of Pareto front (Fig. 4a & 4b). However as compared to NSDE, the spread is different.

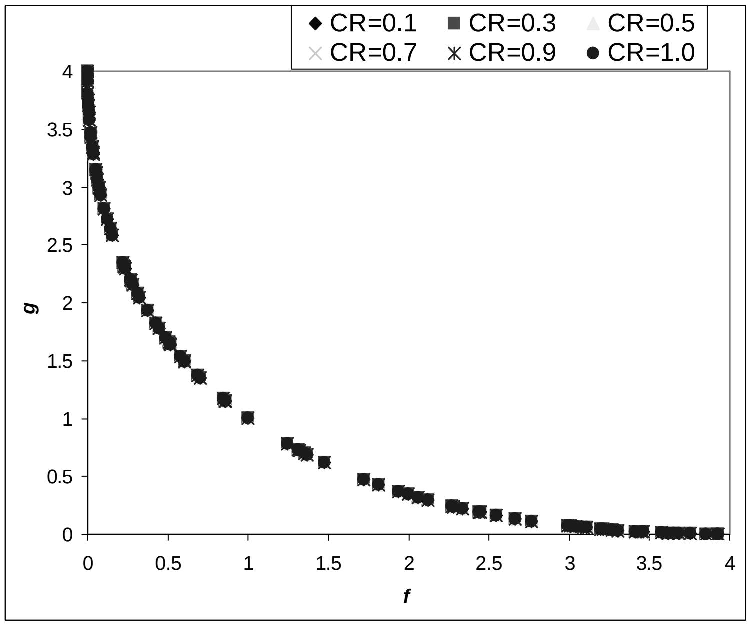
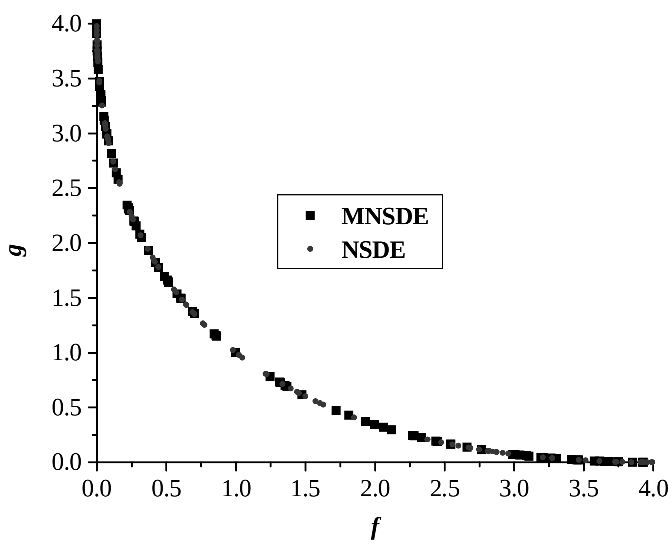
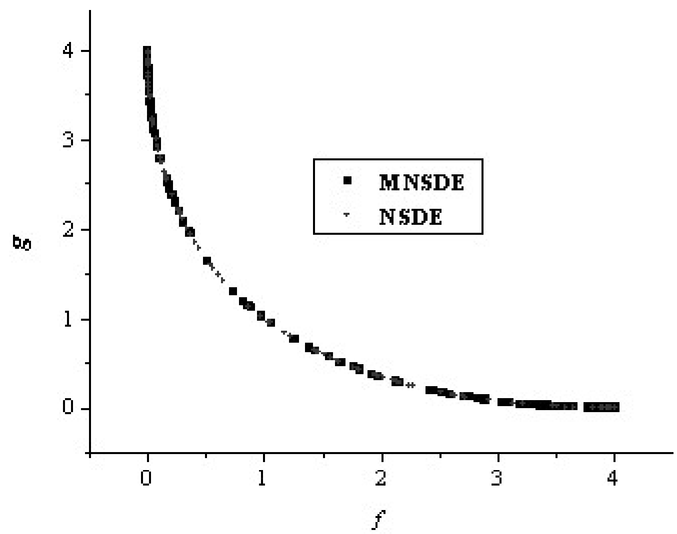


Fig. 3: Effect of *CR* using MNSDE



(a) Seed =10



(b). Seed =23

Fig. 4: Effect of seed for Schaffer's function using MNSDE and NSDE

Table-1 shows the effect of seed value of on maximum objective function value. It is observed that for the two seed value maximum objective function values are almost same as was expected. It is also seen that *NPS* is slightly affected (95 & 98) in case of MNSDE but it is nearly same (98 & 99) for NSDE.

Table 1
Comparison of Maximum Function Values for two Different Seeds

Seed	*NPS*	*f (max)*		*g (max)*	
	(NSDE/MNSDE	*MNSDE*	*NSDE*	*MNSDE*	*NSDE*
10	99/98	3.9290	3.9949	3.9972	4.0137
23	98/95	4.0016	4.0086	3.9928	3.9981

4.1.3. Effect of F on MNSDE & its Comparison with NSDE

It is found that *F* not only affects *NPS*, but also the spread as found in NSDE too. The variation of *NPS* with *F* is shown in Fig. 5 for both MNSDE & NSDE. It is clear that the effect of *F* on MNSDE is more significant than that on NSDE for the same seed value. In case of MNSDE, *NPS* varies from 88 to 99, while for NSDE it varies from 95 to 99. Fig. 6 shows the comparison of MNSDE & NSDE for *F* = 0.2. Both the techniques are able to find out global Pareto front but spread is different.

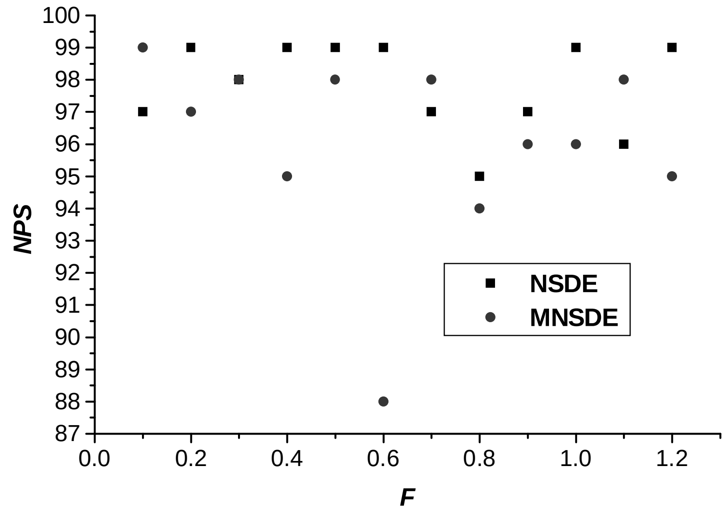


Fig. 5: Effect of *F* on *NPS* for Schaffer's function (MNSDE and NSDE)

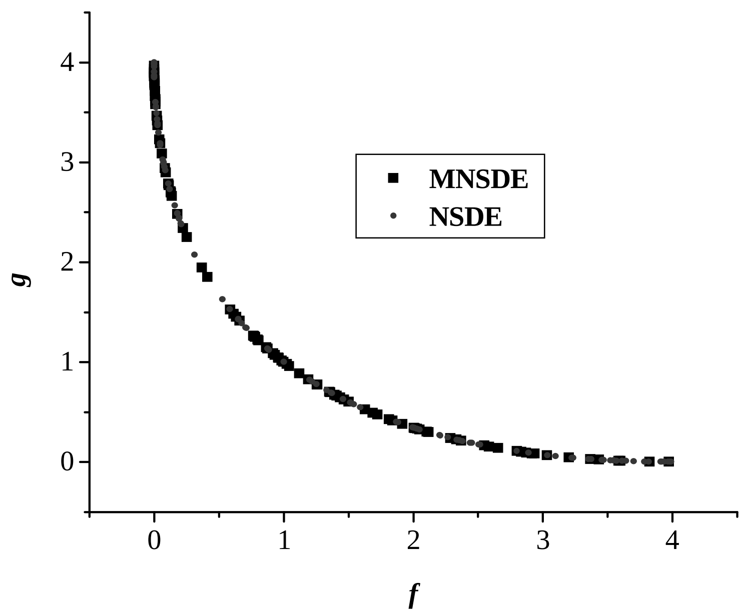


Fig. 6: Comparison of MNSDE and NSDE at seed =10 and *F* = 0.2

4.2. Cantilever Design Problem

Various experiments have been carried out in order to test the proposed algorithm by studying the effect of *Max_gen* and *F* & *CR* for Schaffer's function.

4.2.1. Effect of Max_gen on MNSDE & its Comparison with NSDE

Fig. 7 shows the effect of *Max_gen* on the performance of MNSDE and its comparison with that of NSDE. The key parameters used are $F = 0.5$, $CR = 0.5$, $NP = 100$. It is seen from Fig. 7 that there is not significant difference in *NPS* till *Max_gen* = 500 for the two techniques. And after *Max_gen* = 500, *NPS* in case of NSDE becomes almost constant with further increase in *Max_gen*. But in the case of MNSDE, it increases even after *Max_gen* = 500 and becomes nearly constant after *Max_gen* = 900. *NPS* is more in the case of MNSDE than for NSDE for higher values of *Max_gen* (>500). This is different from what is observed in the result with Schaffer' function.

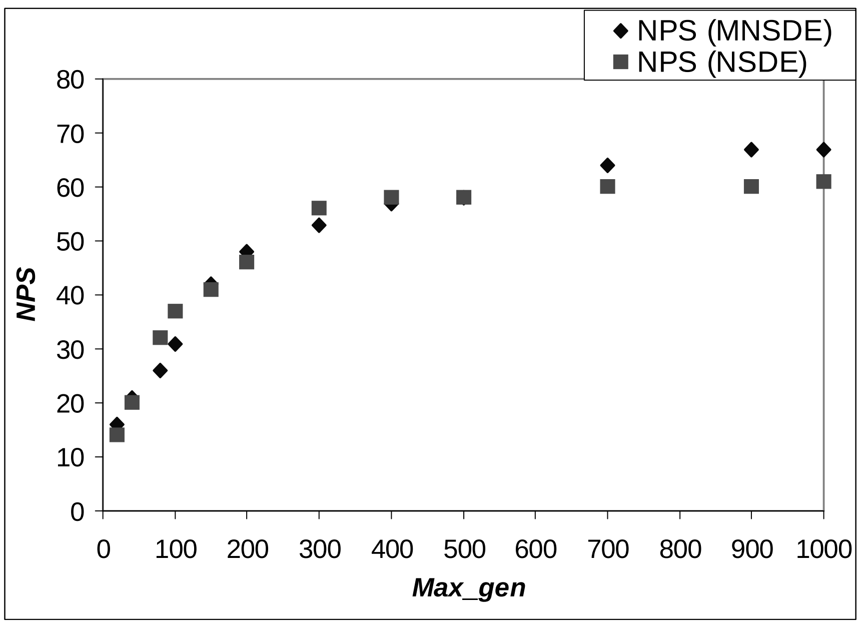


Fig. 7: Effect of *Max_gen* on *NPS* (MNSDE and NSDE) for cantilever design

4.2.2. Effect of F on MNSDE & its Comparison with NSDE

Fig. 8 shows the effect of *F* on *NPS* using MNSDE and its comparison with that using NSDE. From Fig. 8 it is seen that for MNSDE, *NPS* is high for lower value of *F* while a value of $F = 0.5$ gives highest *NPS* for NSDE. This is in agreement with what is observed in Schaffer's function. It is interesting to note that at $F = 0.5$, *NPS* is same for both the techniques. Hence this value can be used for further comparison of Pareto front obtained in the two algorithms. It is found that *F* not only affects the *NPS* but also the spread of Pareto front. Also, the spread of Pareto front is different for different seed value. Fig. 9 shows the Pareto front obtained using the two algorithms for F = 0.5. Although the spread is different yet both the algorithms are able to locate the global Pareto front.

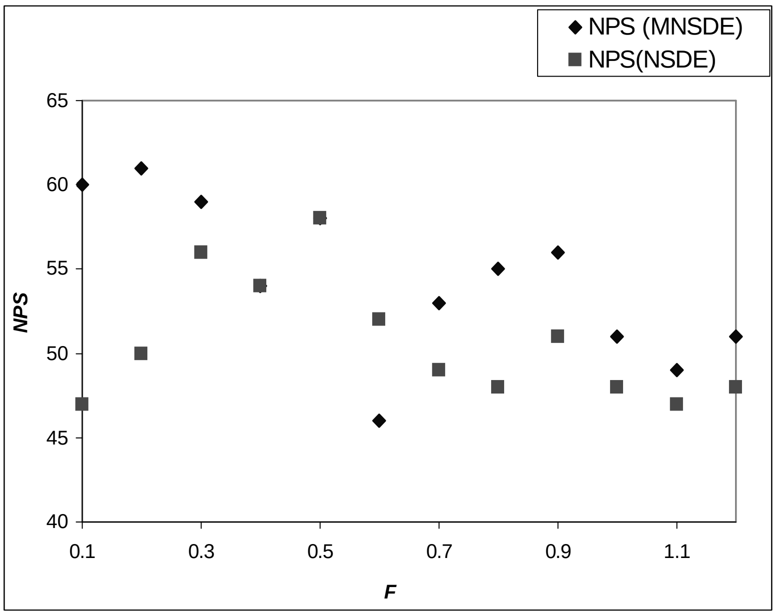


Fig. 8: Effect of *F* on *NPS* for cantilever design problem using MNSDE and NSDE (CR = 0.5)

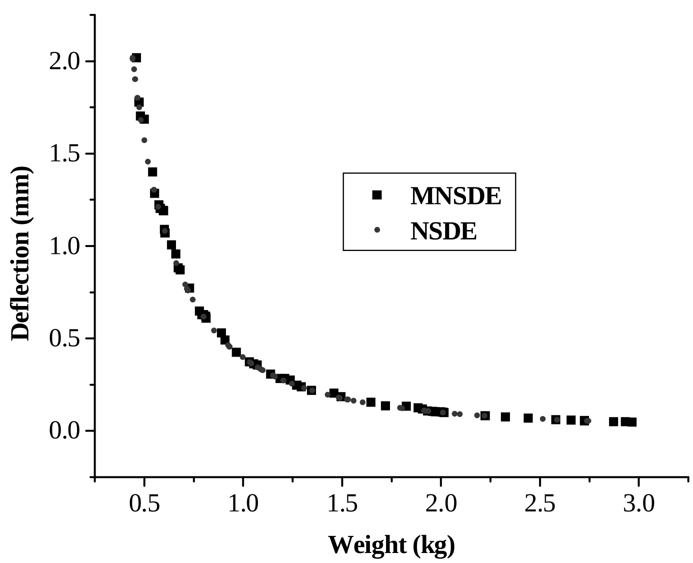


Fig. 9: Pareto front for Cantilever Design Problem using MNSDE and NSDE

4.2.3. Effect of CR on MNSDE & its Comparison with NSDE

The effect of *CR* on *NPS* (for both MNSDE & NSDE) for cantilever design problem is significant as compared to Schaffer's function. The variation of *NPS* with *CR* is shown in Fig. 10. Best value of *CR* seems to be 0.7 for MNSDE and 1.0 for NSDE. However, the value of *NPS* is nearly same for *CR* = 0.9 & 1.0 for both the algorithms.

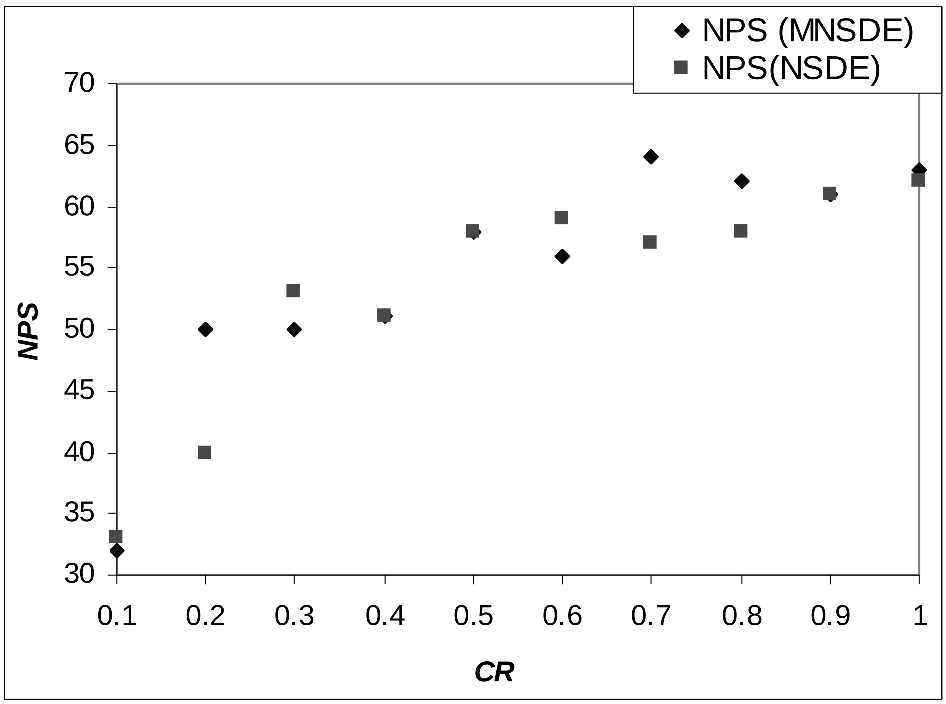
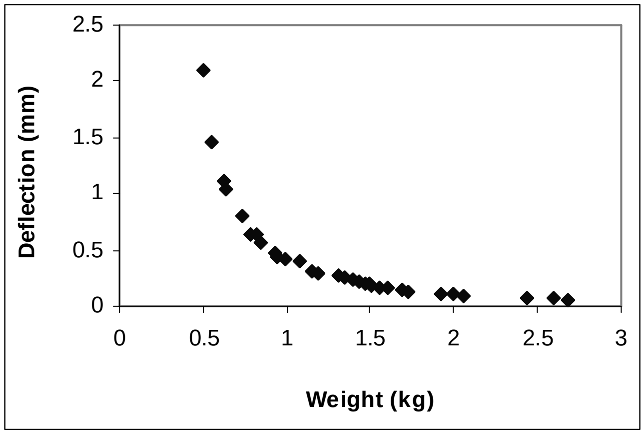
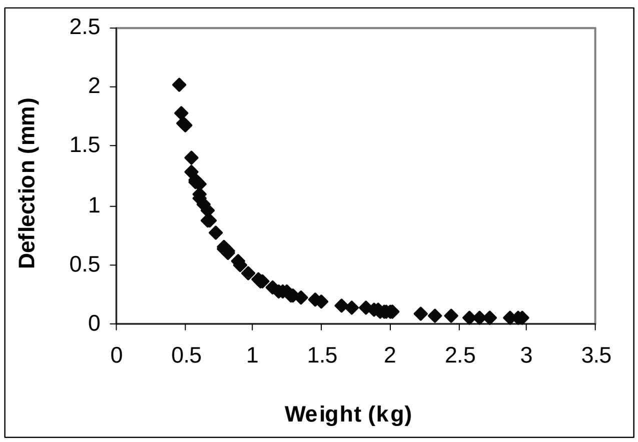


Fig. 10: Effect of *CR* on *NPS* for Cantilever Design problem (MNSDE and NSDE)

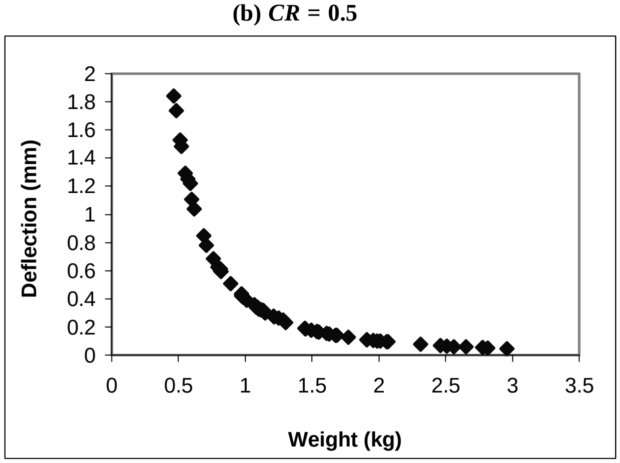
Fig. 11 shows the Pareto fronts for different *CR* values using MNSDE. It is evident that as *CR* value is increased for given *F* & *Max_gen*, the shape of the Pareto front gets improved. In other words, it is closest to global Pareto front for *CR* ≅ 1.0 (Fig. 11c) rather than for lower *CR* values (Fig. 11a & 11b). This is similar to what is observed with the results using NSDE. The effect of *CR* value on the maximum value of objective function and *NPS* is shown in Table-2. Also a comparison is made with the results obtained using NSDE.



(a) *CR* = 0.1



(b) *CR* = 0.5



(c) *CR* = 1.0

Fig. 11. Effect of CR on shape of Pareto front for cantilever design problem (MNSDE)

Table 2
Effect of *CR* on Maximum value* of Objective Functions

CR	*NPS (NSDE/MNSDE)*	*Weight (max)*		*Deflection (max)*	
		MNSDE	*NSDE*	*MNSDE*	*NSDE*
0.1	33/32	2.6789	2.9703	2.0929	2.0723
0.5	58/58	2.9683	2.7960	2.0173	2.0095
1.0	62/63	2.9591	2.9103	1.8434	1.9982

*Literature values for maximum Deflection = 2.04, and maximum Weight = 3.06

It is clear from Table 2 that for both MNSDE and NSDE, the *NPS* increases with *CR* and also the *NPS* value is same. At *CR* value of 0.5, *NPS* is almost double the value at *CR* = 0.1. Further increase in *CR* dose not increases *NPS* significantly. At *CR* = 0.5, the MNSDE is closer to literature value for maximum deflection and weight rather than NSDE.

5. CONCLUSIONS

The two test problems are solved using MNSDE and results are also compared with those obtained using NSDE. The effect of various parameters (*Max_gen*, *CR*, and *F*) is discussed and analyzed. It is observed that *NPS* increases with *Max_gen* up to a certain value which is problem dependent. For Schaffer's function, it is about 100 while for cantilever design problem it is about 900. The effect of *CR* is significant in case of cantilever design problem. A high value of *CR* (≥ 0.7) is found suitable. It is important to note that scaling factor not only affects the *NPS* but also the shape and spread of Pareto optimal front.

Schaffer's function: The effect of *Max_gen* is same for both the algorithms (NSDE and MNSDE). *CR* does not affect the *NPS* as well as shape & spread of Pareto front for given value of *F*, *Max_gen* and seed. This is found to be true for both the algorithms. Effect of *F* on *NPS* is more pronounced in MNSDE as compared to NSDE. However for both the algorithms, the spread of Pareto front is different for different values of *F* for same seed value.

Cantilever Design Problem: In both the algorithms, *NPS* increases with *Max_gen* till a certain value. But this 'certain value' is problem dependent. Also, in the two algorithms, *F* not only affects the spread but also the *NPS*. Lower values of *F* (0.1, 0.2, and 0.3) are found to give higher *NPS* for MNSDE while for NSDE best value is 0.5. It is found that the parameter *CR* not only affects the *NPS* but also the shape and spread of Pareto front, for the two algorithms. Use of high *CR* value (≅ 1.0) is found to beneficial in the two algorithms.

It is important to note that even though all controlling parameters are same still the spread of Pareto front can be different in the two algorithms.

Based on the results, it is recommended to use a high *CR* value for both NSDE & MNSDE and lower value of *F* for MNSDE and a value of 0.5 for NSDE. *Max_gen* is found to be problem dependent.

REFERENCES

[1] Abbass, H. A., Sarkar, R., and Newton, C. (2001), PDE: A Pareto-frontier differential evolution approach for multi-objective optimization problems. *Proceedings of IEEE congress on Evolutionary Computation*, 971-978.

[2] Angira, R. and Babu, B.V. (2005a), 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* (*IICAI-2005*), Pune, India, December 20-22, pp. 1428-1443.

[3] Angira, R. and Babu, B.V. (2005b), Optimization of Non-linear chemical processes using Modified Differential Evolution (MDE). *Proceedings of the 2nd Indian International Conference on Artificial Intelligence (IICAI-05)*, Pune, India, December 20-22, pp. 911-923.

[4] Angira, R. and Babu, B.V. (2006), Optimization of process synthesis and design problems: A Modified Differential Evolution Approach. Accepted (In Press). *Chemical Engineering Science*.

[5] Babu, B.V. and Anbarasu, B. (2005), Multi-Objective Differential Evolution (MODE): An Evolutionary Algorithm for Multi-Objective Optimization Problems (MOOPs). *Proceedings of the Third International Conference on Computational Intelligence, Robotics, and Autonomous Systems (CIRAS-2005)*, Singapore, December 13-16.

[6] Babu, B.V. and Angira, R. (2006), Modified Differential Evolution (MDE) for Optimization of Non-Linear Chemical Processes. Accepted (In Press). *Computers & Chemical Engineering*.

[7] Babu, B.V. and Jehan, M. M. L. (2003), Differential Evolution for Multi-Objective Optimization. *Proceedings of International Conference on Evolutionary Computation (CEC-2003)*, Canberra, Australia, December 8-12, 2003, 2696-2703.

[8] Babu, B.V., Chakole, P. G., and Mubeen, J. H. S. (2005a). Multiobjective Differential Evolution (MODE) for Optimization of Adiabatic Styrene Reactor. *Chemical Engineering Science*, **60** (17), 4824-4839.

[9] Babu, B.V., Mubeen, J. H. S., and Chakole, P. G. (2005b). Multiobjective Optimization Using Differential Evolution. *TechGenesis-The Journal of Information Technology*, *2* (2), 4-12.

[10] Coello, C. A. (1999), A Comprehensive Survey of Evolutionary- Based Multi objective Optimization Techniques. *International Journal of Knowledge and Information Systems*, **1 (3)**, 269-308.

[11] Deb, K. (2001), *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons.

[12] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002), A fast and elitist multi objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, **6 (2)**, 182-197.

[13] Fonseca C. M. and Fleming, P. J. (1993), Genetic algorithms for multi objective optimization: formulation, discussion, and generalization. *Genetic Algorithms: Proceeding of the Fifth International Conference*, Forrest S. ed., 416-423.

[14] Goldberg, D. E. (1989), *Genetic algorithms in search, optimization, and machine learning*, Reading, MA: Addison-Wesley.

[15] Hajela, P., and Lin, C. Y. (1992), Genetic search strategies in multicriterion optimal design. *Structural Optimization*, 4, 99-107.

[16] Horn, J., Nafpliotis, N., and Goldberg, D.E. (1994), A niched Pareto genetic algorithm for multi objective optimization. *Proceedings of the First IEEE Conference on Evolutionary Computation*, 82–87.

[17] Knowles, J., and Corne, D. (2000), Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, *8*, 142-172.

[18] Kursawe, F. (1991), A variant of evolution strategies for vector optimization. In Schwefel H.P. and Manner, R. eds., *Parallel problem solving from nature, First workshop proceedings*, **496** of Lecture notes in computer science, 193-197.

[19] Schaffer, J. D. (1985), Multiple objective optimization with vector evaluated genetic algorithm. *Proceeding of the First International Conference on Genetic Algorithms*, 93-100.

[20] Srinivas, N. and Deb, K. (1995), Multi-objective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, **2 (3)**, 221-248.

[21] Xue, F., Sanderson, A. C., Graves, R. J. (2003). Pareto-based multi-objective differential evolution. *Proceedings of the Congress on Evolutionary Computation (CEC'2003)*, **2**, Canberra, Australia, IEEE Press, 862–869.

[22] Zitzler, E., and Thiele, L. (1999), Multi-objective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, **3**, 257-271.