IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Application of the Differential Evolution (DE) Algorithm to Solve Engineering Optimization Problems

[1]Charvee M. Gurav, [2]Bhargavi A. Ketkar, [3]Sahil P. Patil, [4]Prashant A. Giri

[1, 2, 3]Undergraduate Students [4]Assistant Professor

[1, 2, 3, 4]Department of Chemical Engineering, Finolex Academy of Management and Technology, Ratnagiri, India

Abstract: Differential Evolution (DE) is a simple, rapid, and numerically robust evolutionary optimization approach. The DE algorithm is a population-based algorithm, similar to a genetic algorithm that employs comparable operators, such as crossover, mutation, and selection. The DE method is simple to apply to a wide range of real-valued situations. DE utilized in this work to tackle two engineering optimization problems in the literature. The approach presented in this article compared to other well-known conventional optimization method. According to the results obtained, the convergence speed of DE is much faster than that of the other approaches. As a result, the DE algorithm appears to be a potential method for solving engineering optimization issues.

***Index Terms* - Evolutionary Algorithms, Differential Evolution, Optimization, Genetic Algorithms**

I. INTRODUCTION

Optimization has various applications in Engineering field, with varied concerns like Production, Design, cost accounting and analysis. Optimization refers to finding one or more feasible solutions, which corresponds to extreme values of one or more objectives. The need for finding such optimal solutions in a problem comes mostly from the extreme purpose of either designing a solution for minimum cost of fabrication, or for maximum possible reliability, or others. Because of such extreme properties of optimal solutions, optimization methods are of great importance, particularly in engineering design, scientific experiment and business decision making. A wide variety of heuristic optimization techniques such as genetic algorithm (GA) (Lai, Ma, Yokoyama, & Zhao, 1997; Osman, Abo-Sinna, & Mousa, 2004), simulated annealing (SA) (Miranda, Srinivasan, & Proenca, 1998), Tabu search (M. Abido, 2002), and particle swarm optimization (PSO) (M. A. Abido, 2002; Das, Abraham, & Konar, 2008) are available in the literature for optimization (Vesterstrom & Thomsen, 2004).

In 1995, Price and Storn (Storn, 1995) proposed a new floating point encoded evolutionary algorithm for global optimization and named it Differential Evolution owing to a special differential operator, which they invoked to create new offspring from parent chromosomes instead of classical crossover or mutation. Easy methods of implementation and negligible parameter tuning made the algorithm quite popular soon. Biological and sociological motivations inspire the algorithm and can take care of optimality on rough, discontinuous and multi-modal surfaces (Lampinen, 2001; Storn, 1996a, 1996b, 1999; Storn & Price, 1997). The DE has three major advantages: it can find near optimal solution regardless of the initial parameter values, its convergence is fast and it uses a few number of control parameters. In addition, DE is simple in coding, easy to use and it can handle integer and discrete optimization (Babu, 2004; Babu & Angira, 2001, 2002; Babu & Jehan, 2003; Zaharie, 2007). Differential evolution (DE) is a method that optimizes a problem by iteratively trying to improve a candidate solution regarding a measure of quality. Differential Evolution optimizes a problem by maintaining a population of candidate solutions and creating new candidate solutions by combining existing ones according to its simple formulae, and then keeping whichever candidate solution has the best score or fitness on the optimization problem at hand (Das, Mullick, & Suganthan, 2016; Das & Suganthan, 2010).

Originally, Price and Storn (1995) proposed a single strategy for differential evolution, which they later extended to ten different strategies. Differential evolution has been successfully applied to a wide range of problems. It was observed that the convergence speed of DE is significantly better than that of GA and other heuristic techniques (Vesterstrom & Thomsen, 2004). The performance of DE was compared to PSO and evolutionary algorithms (EAs) and it was found that DE is the best performing algorithm, as it finds the lowest fitness value for most of the problems. Also, DE is robust; it can reproduce the same results consistently in many trials, whereas the performance of PSO is far more dependent on the randomized initialization of the individuals (Eltaeib & Mahmood, 2018; Fleetwood, 2004; Gämperle, Müller, & Koumoutsakos, 2002). In addition, the DE algorithm has been used to solve high-dimensional function optimization (up to 1000 dimensions) (Yang, Tang, & Yao, 2007). It is found that it has superior performance on a set of widely used benchmark functions. Therefore, DE algorithm seems to be a promising approach for engineering optimization problems. It has successfully been applied and studied to many artificial and real optimization problems (Giri, Apankar, & Gawas, 2018; Gurav et al.).

In this paper, we selected two optimization problems to find the global optimum solution for which the parameters are to be optimized. The problem is formulated as a linear and non-linear optimization problem with equality and inequality constraints. These problems have considered have their source in (Kumbhojkar G. V., "Applied Mathematics-III", Chemical Engineering 2015-16.) and have solved analytically. The optimization is carried out by evolutionary differential evolution algorithm. The results are compared to those reported in the literature and with other conventional method.

II. DE COMPUTATIONAL FLOW

The salient features of the DE algorithm can be stated as follows and also represented in fig. 1 (Dutta, 2016).

a. Population will be random between the upper and lower bounds for all the parameters.
b. Determine the objective function value for the initial population.
c. The next step is mutation and crossover.

Take i as population counter i = (0, 1, 2… 19)

i. *Randomly choose 3 population points a, b, and c such that* $i \neq a \neq b \neq c$
ii. *Select randomly a parameter j for mutation (j=0, 1)*
iii. *Generate a random number[0,1]*

If random number <CR,

$$Trial\ [j] = x_1\ [c]\ [j] + F\ (x_1\ [a_1]\ [j] - x_1\ [b]\ [j])$$

If random number > CR,

$$Trial\ [j] = x_1\ [i]\ [j]$$

Check for bounds:

If bounds are violated, then randomly generate the parameter as shown below:

Trial [j] = lower limit + rand.no. [0, 1] (upper limit - lower limit);

d. Repeat c until all parameters are mutated.
e. Calculate the objective function value for the vector obtained after mutation and crossover.
f. Select the least cost vector for next generation, if the problem is of minimization.
g. Repeat step 3 to 5 for a specified number of generations, or till some termination criterion is met.

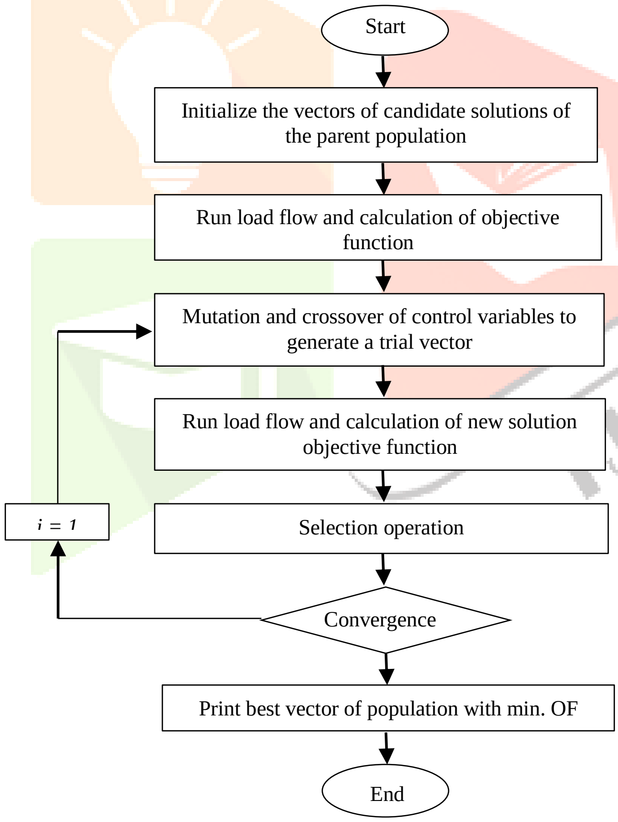

Figure 1. Flowchart for Differential Algorithm

III. CASE STUDIES:

Objective:

The objective of the present work was aimed at finding the global optimum solution for Mathematical problems. The optimization is carried out by evolutionary differential evolution algorithm and the results obtained are compared with other conventional and non-conventional techniques. The analytical solution of differential evolution optimization problem involves several iterations and they are time-consuming, use of software like MATLAB helps convergence to global optimum faster.

Problem definitions:

Problem Statement 1

Minimize $Z=2x_1+3x_2-x_1^2-2x_2^2$

Subject to $2x_1+3x_2 \leq 6$

$5x_1+2x_2 \leq 10$

$x_1, x_2 \geq 0$

(Answer by Conventional Method: x_1= 1.648, x_2= 0.880 and Z_{min}=1.6713) (Kumbhojkar G. V., "Applied Mathematics-III", Chemical Engineering 2015-16.)

Problem Statement 2

Maximize $Z=3x_1^2 + x_2^2+ 2x_1x_2 + 6x_1+2x_2$

Subject to $2x_1+x_2 = 4$

$x_1, x_2 \geq 0$

(Answer by Conventional Method: x_1= 1, x_2= 2 and Z_{max}=28) (Kumbhojkar G. V., "Applied Mathematics-III", Chemical Engineering 2015-16.)

IV. METHODOLOGY:

By implementing DE algorithm, we have solved this problem analytically. This is an optimization problem in which objective function, along with the constraints, is given.

1. First, we have to choose DE key parameters, i.e. NP, CR and F.
2. Randomly choosing the value of x_1 and x_2 between the upper and lower bounds.
3. Third step is evaluation, i.e., we have to put the value of x_1 and x_2 in the objective function and calculate the function value.
4. To develop individual 1 for the next generation, the first member of the population is set as the target vector.
5. In order to generate the noisy random vector, 3 individuals (2, 4, and 6) from the population size are selected randomly. The weighted difference between individual 2 and individual 4 is added to the third randomly chosen vector individual 6.

We use :-

Trial vector = Target Vector + F× (Random Value 1 – Random Value 2)

6. Generate random number (0 to 1)

If random number > CR Target vector is used as Trial vector

If random number < CR Noisy random vector is used as Trial vector

i. Trial vector compared with target vector and vector with lowest value of the two becomes individual 1 for next generation.
ii. To develop individual 2 for next generation, the second member of the population is set as target vector and the above process is repeated.
iii. This process is repeated NP times until the new population set array is filled, which completes one generation.

Choice of DE key parameters:

The strategy used here is *DE/rand/1/bin.* Population size (*NP*) should be 5 to 10 times the value of *D*, i.e., the dimension of the problem. Choose mutation factor (*F*) as 0.5 initially. If this leads to premature convergence, then increase the value of *F*. The range of values of *F* is 0<*F*<1.2, but the optimal range is 0.4<*F*<1.0. Values of *F*<0.4 and *F*>1.0 are seldom effective. As a good first guess, Crossover Ratio (*CR*) shall be taken as 0.9. Try *CR* as 0.9 at first and then try *CR* as 0.1. Judging by the speed of convergence, choose a value of *CR* between 0 and 1 (Onwubolu & Babu, 2013).

V. RESULTS AND DISCUSSION

The performance of differential evolution algorithm is tested by applying it to above problems. The key parameter of DE- Crossover Ratio (*CR*), Number of population size (*NP*), Scaling Factor (*F*), and Number of iterations are varied over a wide range of their potential values. The above two optimization problems are solved by using differential evolution and conventional techniques and the results are obtained as shown in table 1 and table 2. The results obtained by differential evolution are compared with the conventional techniques; it is found that differential evolution is more suitable as compared to conventional techniques.

Implementation:

Initial trial runs were done with different values of DE key parameters such as differentiation (or mutation) constant *F*, crossover constant *CR*, size of population *NP*, and maximum number of generations *GEN*, which is used here as a stopping criterion. In this paper, the following values are selected as:

For problem statement 1: *F* = 0.8; *CR* = 0.5; *NP* = 10; *GEN* = 10

For problem statement 2: *F* = 0.8; *CR* = 0.5; *NP* = 10; *GEN* = 10.

The performance of differential evolution algorithm is tested by applying it to above problems. The key parameter of DE- Crossover Ratio (*CR*), Number of population size (*NP*), Scaling Factor (*F*), and Number of iterations are varied over a wide range of their potential values.

The results obtained by differential evolution are compared with the conventional techniques; it is found that differential evolution is more robust and faster as compared to conventional techniques. The proposed DE algorithm has been developed and implemented using the MATLAB software. Initial trial runs were done with different values of DE key parameters such as differentiation (or mutation) constant *F*, crossover constant *CR*, size of population *NP*, and maximum number of generations *GEN*, which is used here as a stopping criterion. In this paper, the following values are selected as:

Solution to problem 1:

Problem statement 1 is the maximization problem where answers by conventional method are 1.648 for x_1, 0.88 for x_2 and value of function is 1.6713. Whereas by using DE, the answers are 1.256 for x_1, 0.858 for x_2 and value of function is 2.086.

Table 5.1: Solution for Problem Statement – 1

GEN	x_1	x_2	*f(x)*
Ind. 1	0.68	0.89	1.9834
Ind. 2	0.536	0.92	1.8519
Ind. 3	0.22	0.14	0.7724
Ind. 4	0.12	0.09	0.4794
Ind. 5	0.136	0.81	1.371
Ind. 6	0.768	0.63	2.042
Ind. 7	0.99	0.69	2.1177
Ind. 8	1.10	0.84	2.0988
Ind. 9	1.256	0.858	2.086
Ind. 10	1.64	0.754	1.7153

In this problem, the value of mutation factor (*F*) is 0.8, crossover ratio (CR) is 0.5, and population size (*NP*) is 10. Therefore, for the next generation, we generate 10 individuals, as shown in table 5.1.

Solution for problem statement 2:

In this problem, *F* = 0.8; *CR* = 0.5; *NP* = 10; *GEN* = 10. Problem statement 2 is the maximization problem where answers by conventional methods are 1 for x_1, 2 for x_2 and value of function is 28. Whereas by using DE, the answers are 1.117 for x_1, 2.506 for x_2 and value of function are 27.35, as shown in table 5.1.

Table 5.2: Solution for Problem Statement – 2

`GEN	x_1	x_2	*f(x)*
Ind. 1	0.56	1.29	9.99
Ind. 2	1.117	2.506	27.35
Ind. 3	0.984	1.431	16.53
Ind. 4	0.633	0.533	7.02
Ind. 5	1.436	0.298	16.34
Ind. 6	1.345	0.583	16.57
Ind. 7	1.023	1.982	21.225
Ind. 8	0.986	1.843	19.549
Ind. 9	0.769	2.00	17.464
Ind. 10	1.403	1.952	17.498

VI. Conclusion

Differential Evolution optimization algorithm has been successfully proposed and applied to solve simple mathematical problems. Two problems have been solved using DE in this present work. DE successfully converges 20% more to global optimum for Problem 1 and 10% more to global optimum for Problem 2, respectively. The evolutionary algorithm gives a varied choice of parameters, which help to achieve better result with minimum effort. Results show DE is more reliable, efficient and hence a better approach to the optimization of non-linear problem.

Differential Evolution technique is much faster, has less computational burden when compared to non-traditional techniques and the estimation is much more accurate and efficient. Differential evolution requires fewer efforts for function evaluations and assures convergence from any starting point. Differential evolution has been proved efficient for solving Mathematical and Engineering problems. Based on results of above solutions, we conclude that differential evolution explores the decision space more efficiently than conventional and non-conventional techniques. Differential Evolution is more effective in obtaining better quality solutions.

References

1) Abido, M. (2002). Optimal power flow using tabu search algorithm. *Electric power components and systems, 30*(5), 469-483.
2) Abido, M. A. (2002). Optimal power flow using particle swarm optimization. *International Journal of Electrical Power & Energy Systems, 24*(7), 563-571.
3) Babu, B. (2004). *Process plant simulation*: Oxford University Press, USA.
4) Babu, B., & Angira, R. (2001). *Optimization of non-linear functions using evolutionary computation*. Paper presented at the Proceedings of the 12th ISME International Conference on Mechanical Engineering, India.
5) Babu, B., & Angira, R. (2002). *A differential evolution approach for global optimization of MINLP problems.* Paper presented at the Proceedings of the Fourth Asia Pacific Conference on Simulated Evolution and Learning (SEAL 2002), Singapore.
6) Babu, B., & Jehan, M. M. L. (2003). *Differential evolution for multi-objective optimization.* Paper presented at the The 2003 Congress on Evolutionary Computation, 2003. CEC'03.

7) Das, S., Abraham, A., & Konar, A. (2008). Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. In *Advances of computational intelligence in industrial systems* (pp. 1-38): Springer.
8) Das, S., Mullick, S. S., & Suganthan, P. N. (2016). Recent advances in differential evolution–an updated survey. *Swarm and evolutionary computation, 27*, 1-30.
9) Das, S., & Suganthan, P. N. (2010). Differential evolution: A survey of the state-of-the-art. *IEEE transactions on evolutionary computation, 15*(1), 4-31.
10) Dutta, S. (2016). *Optimization in chemical engineering*: Cambridge University Press.
11) Eltaeib, T., & Mahmood, A. (2018). Differential evolution: A survey and analysis. *Applied Sciences, 8*(10), 1945.
12) Fleetwood, K. (2004). *An introduction to differential evolution.* Paper presented at the Proceedings of Mathematics and Statistics of Complex Systems (MASCOS) One Day Symposium, 26th November, Brisbane, Australia.
13) Giri, P. A., Apankar, D. D., & Gawas, S. S. (2018). Optimization of some Chemical Processes using Differential Evolution (De) Algorithm. *International Journal of Creative Research Thoughts, 6*(2).
14) Gurav, C. M., Ketkar, B. A., Patil, S. P., Giri, P. A., Apankar, D. D., & Gawas, S. S. Differential Evolution (DE) Algorithm: Population Based Metaheuristic Search Algorithm for Optimization of Chemical Processes.
15) Gämperle, R., Müller, S. D., & Koumoutsakos, P. (2002). A parameter study for differential evolution. *Advances in intelligent systems, fuzzy systems, evolutionary computation, 10*(10), 293-298.
16) Lai, L. L., Ma, J., Yokoyama, R., & Zhao, M. (1997). Improved genetic algorithms for optimal power flow under both normal and contingent operation states. *International Journal of Electrical Power & Energy Systems, 19*(5), 287-292.
17) Lampinen, J. (2001). A bibliography of differential evolution algorithm. *Lappeenranta University of Technology, Finland*.
18) Miranda, V., Srinivasan, D., & Proenca, L. M. (1998). Evolutionary computation in power systems. *International Journal of Electrical Power & Energy Systems, 20*(2), 89-98.
19) Onwubolu, G. C., & Babu, B. (2013). *New optimization techniques in engineering* (Vol. 141): Springer.
20) Osman, M., Abo-Sinna, M. A., & Mousa, A. (2004). A solution to the optimal power flow using genetic algorithm. *Applied mathematics and computation, 155*(2), 391-405.
21) Storn, R. (1995). Differrential evolution-a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical report. *International Computer Science Institute, 11*.
22) Storn, R. (1996a). *Differential evolution design of an IIR-filter*. Paper presented at the Proceedings of IEEE international conference on evolutionary computation.
23) Storn, R. (1996b). *On the usage of differential evolution for function optimization.* Paper presented at the Proceedings of North American fuzzy information processing.
24) Storn, R. (1999). System design by constraint adaptation and differential evolution. *IEEE Transactions on Evolutionary Computation, 3*(1), 22-34.
25) Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization, 11*(4), 341-359.
26) Vesterstrom, J., & Thomsen, R. (2004). *A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems.* Paper presented at the Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753).
27) Yang, Z., Tang, K., & Yao, X. (2007). *Differential evolution for high-dimensional function optimization.* Paper presented at the 2007 IEEE congress on evolutionary computation.
28) Zaharie, D. (2007). *A comparative analysis of crossover variants in differential evolution.* Paper presented at the Proceedings of IMCSIT.