OPTIMIZATION OF NON-LINEAR FUNCTIONS USING EVOLUTIONARY COMPUTATION

B.V. Babu* and Rakesh Angira

Department of Chemical Engineering
Birla Institute of Technology & Science
Pilani 333 031

Abstract:- Global optimization is important if the function has a number of optima of which some are local and some global. Since many real world problems contain multiple optima, traditional methods may not be adequate to solve such problems due to the possibility of getting trapped at local optimum. Hence they do not ensure global optima. Differential Evolution (DE) is an evolutionary optimization technique which is exceptionally simple, significantly faster & robust at numerical optimization and is more likely to find a function's true global optimum. In this paper, a non-linear function with three local optima and one global optima has been solved by using DE. The results obtained from DE are compared with that of Genetic Algorithms (GA). The results indicate that performance of DE is better than GA.

Key Words: Optimization, Himmelblau function, Evolutionary Computation, Genetic Algorithms, Differential Evolution.

1. INTRODUCTION

The goal of optimization is to find the values of the variables in the process that yield the best value of the performance criterion. What is usually involved is a trade-off between capital and operating costs. Typical real world problems have many, and possibly infinite number of, solutions. Optimization is concerned with selecting the best among the entire set by efficient quantitative methods. Unfortunately no single method or algorithm of optimization exists that can be applied efficiently to all problems. The method chosen for any particular case will depend primarily on: (1) the character of the objective function and whether it is known explicitly, (2) the nature of the constraints, and (3) the number of independent and dependent variables.

Most of the traditional optimization techniques based on gradient methods have the possibility of getting trapped at local optimum depending upon the degree of non-linearity and initial guess. Hence, these traditional optimization techniques do not ensure global optimum and also have limited applications.

During the last two decades there has been a growing interest in algorithms which are based on the principle of evolution (survival of the fittest). They are referred as Evolutionary Algorithms (EA) or Evolutionary Computation methods (EC methods)[1]. EAs (genetic algorithms [2], evolution strategies, evolution programming and genetic programming) are increasingly used to great advantage in applications as diverse as computer - aided design [3], optimal design of non-linear chemical engineering processes [4], parameter estimation [5], controller design [6] digital filter design [7] etc. In the present paper Differential evolution (DE) has been used to find the optimal solution of a non-linear function (Himmelblau function), given by:

$$f(x) = \left(x_1^2 + x_2 - 11\right)^2 + \left(x_1 + x_2^2 - 7\right)^2.$$

The results thus obtained are compared with that of GA [8].

***Corresponding Author: Group Leader, ET Group, ESD (Workshop), BITS, Pilani. Email: bvbabu@bits-pilani.ac.in Fax: (01596) 44183, Phone: (01596) 45073 Ext. 205 (off.).**

2. GENETIC ALGORITHM (GA)

GAs are computerized search and optimization algorithms based on the mechanics of natural genetics and natural selection. They mimic the 'survival of the fittest' principle of nature to make search process. The key parameters of control are: N - the population size, p_c – the crossover probability and p_m – the mutation probability [2]. Since their inception, GAs have been applied in many fields. The various applications of GAs are: process design and optimization [9], computer aided molecular design [3], optimal design of ammonia synthesis reactor [4], estimation of heat transfer parameters in trickle bed reactors [10], automated design of heat exchangers using artificial intelligence based optimization [11], optimal design of heat exchangers [12] etc.

3. DIFFERENTIAL EVOLUTION (DE)

DE is an improved version of GA for faster optimization. Unlike simple GA that uses binary coding for representing problem parameters, DE uses real coding of floating point numbers. Among the DE's advantages are its simple structure, ease of use, speed and robustness. In 1997, Price & Storn [13] gave the working principle of DE with single strategy. Later on, they suggested ten different strategies of DE. A strategy that works out to be the best for a given problem may not work well when applied for a different problem. Also, the strategy to be adopted for a problem is to be determined by trial & error. The key parameters of control are: NP - the population size, CR - the crossover constant, F - the weight applied to random differential (scaling factor). The following are the ten different working strategies proposed by Price & Storn [14]:

1. DE/best/1/exp
2. DE/rand/1/exp
3. DE/rand-to-best/1/exp
4. DE/best/2/exp
5. DE/rand/2/exp
6. DE/best/1/bin
7. DE/rand/1/bin
8. DE/rand-to-best/1/bin
9. DE/best/2/bin
10. DE/rand/2/bin

The general convention used is DE/x/y/z. DE stands for Differential Evolution, x represents a string denoting the vector to be perturbed, y is the number of difference vectors considered for perturbation of x, and z stands for the type of crossover being used (exp: exponential; bin: binomial).

The crucial idea behind DE is a scheme for generating trial parameter vectors. Basically, DE adds the weighted difference between two population vectors to a third vector. Choosing the values of NP, F, CR depends upon the specific problem applied, and is often difficult. But some general guidelines are available. Normally, NP should be about 5 to 10 times the number of parameters in a vector. As for F, it lies in the range 0.4 to 1.0. Initially F = 0.5 can be tried then F and/or NP is increased if the population converges prematurely. A good first choice for CR is 0.1, but in general CR should be as large as possible [13]. Price & Storn [14] have given some simple rules for choosing key parameters of DE for any given application.

DE has been successfully applied in various fields. The various applications of DE are: estimation of heat transfer parameters in trickle bed reactor [5], Design of fuzzy logic controller [6], digital filter design [7], dynamic optimization of continuous polymer reactor [15], optimal design of heat exchangers [16], optimal design of shell & tube heat exchanger [17], synthesis & optimization of heat integrated distillation system [18] etc.

The pseudo code of Differential Evolution is given below:

- Choose a strategy
- Initialize the value of D, NP, CR, F & gen_max.
- Initialize all the vector population randomly in the given upper & lower bound.
  For I=1 to NP
  { For j=1 to D
  x_{ij} = random Number}
- Evaluate the cost of each vector.
- Find out the vector with the lowest cost.
- Repeat
- Perform mutation, crossover, selection & evaluation of the objective function for a specified number of generations.

  (a). For each vector x_t (target vector), select three distinct vectors x_a, x_b & x_c (select five, if two vector differences are to be used) randomly from the current population (primary array) other than vector x_t.
  (b). Perform crossover for each target vector with its noisy vector to create a trial vector.

(c). Perform selection for each target vector, x_t by comparing its cost with that of the trial vector. Vector with lower cost is selected for next generation.

- Till termination criteria do not meet.
- Print results.

4. RESULTS AND DISCUSSION

Earlier, GA was applied by Deb [8] to the same function and following results were obtained:
Table I shows the number of times (out of 100 runs) that GAs have been able to converge to a solution within 1% of global minimum and the average number of function evaluations required to achieve that solution.

Table I Results of GA

p_c	p_m	N	G_{max}	$(NRC)_{GA}$	$F_{A, GA}$
0.8	0.05	20	625	77	426
0.8	0.05	30	416	79	484
0.8	0.05	40	312	76	650

In the present paper, DE (with all ten strategies) is applied to find the optimum solution of Himmelblau function and the results are shown in Table II (key parameters used are CR = 0.9, F = 0.51, NP = 20, seed = 5, accuracy = 0.0001%):

Table II Results of DE with all ten strategies

S.No.	Strategy	G_{min}	F_{eval}
1	DE/rand/1/bin	42	860
2	DE/best/1/bin	22	460
3	DE/best/2/bin	34	700
4	DE/rand/2/bin	54	1100
5	DE/rand-to-best/1/bin	31	640
6	DE/rand/1/exp	40	820
7	DE/best/1/exp	18	380
8	DE/best/2/exp	34	700
9	DE/rand/2/exp	49	1000
10.	DE/rand-to-best/1/exp	29	600

- Results of Table II indicate that the strategy no. 2 & 7 are better than other strategies because of less number of function evaluations.

  Also the accuracy has been increased from 1% to 0.0001% still the numbers of function evaluations are less in case of strategy no. 7 than GA with N=20.

Tables III & IV show the number of times (out of 100 runs) that strategy no.2 & 7 have been able to converge to a solution within 1% of global minimum and the average number of function evaluations required to achieve that solution (with key parameters) respectively.

Table III Results of Strategy No.2 (DE/best/1/bin)

CR	F	NP	G_{max}	$(NRC)_{DE}$	$F_{A, DE}$	$F_{A, GA}$
0.9	0.51	20	625	100	257	426
0.9	0.51	30	416	100	400	484
0.9	0.51	40	312	100	515	650

Table IV Results of Strategy No.7 (DE/best/1/exp)

CR	F	NP	G_{max}	$(NRC)_{DE}$	$F_{A, DE}$	$F_{A, GA}$
0.9	0.51	20	625	100	243	426
0.9	0.51	30	416	99	350	484
0.9	0.51	40	312	98	496	650

- From Tables III & IV it is evident that number of function evaluations are less in strategy no. 2 & 7 than that of GA.
- The results of the Table III show that $F_{A, DE}$ is 39.67%, 17.35%, 20.77% less than $F_{A, GA}$ for NP = 20, 30, 40 respectively.
- Also, number of runs converged are 100 (out of 100 runs) i.e. 100% in DE/best/1/bin. But in case of GA it is between 76-79 (out of 100 runs) for different population size.
- And, from Table IV it is clear that $F_{A, DE}$ is 42.96%, 27.69%, 23.69% less than $F_{A, GA}$ for NP = 20, 30, 40 respectively.
- As the results of Table III & IV show that strategy no. 7 (DE/best/1/exp) is best because of least number of function evaluations and number of runs converged to be 100, 99, 98 for NP = 20, 30, 40 respectively.

Figure 1 shows the $F_{A, DE}/F_{A, GA}$ vs N/NP for GA and DE (strategy no. 2 (DE2) & strategy no. 7 (DE7))

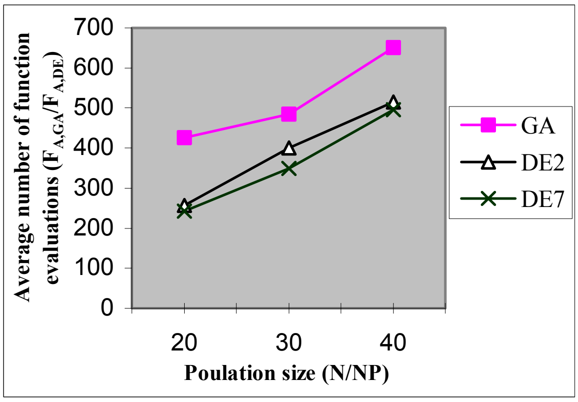

Figure 1.

5. CONCLUSIONS

In this paper the optimization of a non-linear function (Himmelblau function) using Differential Evolution (an evolutionary computation method) has been presented. The strategy no. 7 (that takes minimum number of function evaluations) is found to be the best for the present problem. The best key parameters for the present problem are: NP =20, CR = 0.9, F=0.51. From results, it is clear that $F_{A, DE} < F_{A, GA}$ and $(NRC)_{DE} > (NRC)_{GA}$. Hence it can be concluded that the performance of DE is better than that of GA.

This successful application of DE over GA for the Himmelblau function indicates that DE has great potential and can be applied to advantage in all the highly non-linear & complex engineering problems.

6. NOTATION

$F_{A, GA}$	average number of function evaluations using Genetic Algorithm
$F_{A, DE}$	average number of function evaluations using Differential Evolution
$(NRC)_{DE}$	number of runs converged (out of 100 runs) using DE
$(NRC)_{GA}$	number of runs converged (out of 100 runs) using GA
G_{min}	minimum number of generations required for convergence
G_{max}	maximum generation allowed
F_{eval}	number of function evaluations
D	numbers of parameter in each vector

7. REFERENCES

1. Dasgupta, D. & Michalewicz, Z. (1997). *Evolutionary algorithms in Engineering Applications.* Germany, Springer.
2. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Reading, MA: Addison-Wesley.
3. Venkatasubramanian, V., Chan, K., & Caruthers, J. M. (1994). Computer-aided molecular design using genetic algorithms. *Computers & Chemical Engineering, 18*, 833-844.
4. Upreti, S. R. & Deb, K. (1996). Optimal design of an ammonia synthesis reactor using genetic algorithms. *Computers & Chemical Engineering, 21*, 87-93.
5. Babu, B. V. & Sastry, K. K. N. (1999). Estimation of heat-transfer parameters in a trickle-bed reactor using differential evolution and orthogonal collocation. *Computers & Chemical Engineering, 23*, 327 - 339.
6. Sastry, K. K. N., Behra, L., & Nagrath, I. J. (1998). Differential evolution based fuzzy logic controller for nonlinear process control. *Fundamenta Informaticae*: *Special Issue on Soft Computation.*
7. Storn, R. (1995). Differential evolution design of an IIR-filter with requirements for magnitude and group delay. TR-95-018, *International Computer Science Institute.*
8. Deb, K. (1996). *Optimization for engineering design: Algorithms and examples.* New Delhi, Prentice-Hall.
9. Androulakis, I. P. & Venkatasubramanian, V. (1991). A genetic algorithm framework for process design and optimization. *Computers & Chemical Engineering, 15*(4), 217-228.
10. Babu, B. V. & Vivek, N. (1999). Genetic algorithms for estimating heat transfer parameters in trickle bed reactors. *Proceedings of 52nd Annual session of IIChE (CHEMCON-99)*, Chandigarh, India, December 20 – 23, 1999.
11. Babu, B. V. & Mohiddin, S. B. (1999). Automated Design of Heat Exchangers Using Artificial Intelligence based optimization. *Proceedings of 52nd Annual session of IIChE (CHEMCON-99)*, Chandigarh, India, December 20 – 23, 1999.
12. Manish, C. T., Yan Fu & Urmila, M. D. (1999). Optimal design of heat exchangers: A genetic algorithm framework. *Industrial*

& Engineering Chemistry Research, 38, 456- 467.

13. Price, K. & Storn, R. (1997). Differential evolution. *Dr. Dobb's Journal*, 18-24.
14. Price, K. & Storn, R. (2000). Web site of Price and Storn as on July, 2000. http://www.ICSI.Berkeley.edu/~storn/code.html
15. Lee, M. H., Han, C., & Chang, K. S. (1999). Dynamic optimization of a continuous polymer reactor using a modified differential evolution. *Industrial & Engineering Chemistry Research, 38*(12), 4825-4831.
16. Babu, B. V. & Munawar, S. A. (2000a). Differential Evolution for the optimal design of heat exchangers. *Proceedings of All-India seminar on Chemical Engineering Progress on Resource Development: A Vision 2010 and Beyond*, IE (I), Bhubaneswar, India, March 11, 2000.
17. Babu, B. V. & Munawar, S. A. (2000b). Optimal Design of Shell & Tube Heat Exchanger by Different strategies of Differential Evolution. *Communicated to Computers & Chemical Engineering.*
18. Babu, B. V. & Rishinder Pal Singh (2000). Synthesis & optimization of Heat Integrated Distillation Systems Using Differential Evolution. *Proceedings of All-India seminar on Chemical Engineering Progress on Resource Development: A Vision 2010 and Beyond*, IE (I), Bhubaneswar, India, March 11, 2000.