EVOLUTIONARY COMPUTATION FOR GLOBAL OPTIMIZATION OF NON-LINEAR CHEMICAL ENGINEERING PROCESSES

Rakesh Angira[1] & B. V. Babu[2]*

angira@bits-pilani.ac.in[1] *bvbabu@bits-pilani.ac.in*[2]*
Department of Chemical Engineering
Birla Institute of Technology & Science
Pilani - 333 031 (Rajasthan) India

Abstract: 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 the present study, DE has been used to solve the two non-linear chemical engineering problems. Comparison is made with αBB algorithm (which is based on a branch-and-bound approach). The results indicate that performance of DE is better than the αBB algorithm.

Key words: Evolutionary Computation Method, Optimization, Differential Evolution, αBB algorithm, Non-linear Programming Problems.

1. INTRODUCTION

Realistic treatments of physical and engineering systems frequently involve nonlinear models. Non-linearities are introduced by process equipment design relations, by equilibrium relations and by combined heat and mass balances. The design variables may be continuous [non-linear programming (NLP) problems] or some may be integer [mixed integer non-linear programming (MINLP) problems]. Model nonlinearities give rise to nonconvexities, which, in turn, lead to multiple local optima. Gradient optimization techniques have only been able to tackle special formulations, where continuity or convexity had to be imposed, or by exploiting special mathematical structures. Stochastic algorithms, also known as adaptive random search, have successfully tackled nonconvex problems, mostly in area of Chemical Engineering.

The optimization of non-linear constrained problems is relevant to chemical engineering practice (Salcedo, 1992; Floudas, 1995). In recent years, evolutionary algorithms (EAs) have been applied to the solution of NLP in many engineering applications. The best-known algorithms in this class include Genetic Algorithms (GA), Evolutionary Programming (EP), Evolution Strategies (ES) and Genetic Programming (GP). There are many hybrid systems, which incorporate various features of the above paradigms and consequently are hard to classify, which can be referred just as EC methods (Dasgupta and Michalewicz, 1997). They differ from the conventional algorithms since, in general, only the information regarding the objective function is required. EC methods have been applied to a broad range of activities in process system engineering including modeling, optimization and control. Differential Evolution (DE), developed by Price & Storn (1997), is one of the best EC methods. This method provides one of the best genetic algorithms for solving the real-valued test function. The convergence speed of DE is very high.

In the present study, DE - a hybrid evolutionary computation method, has been used to solve the two non-linear chemical engineering problems viz., (1) Heat exchanger network design and (2) Reactor network design. These problems arise from the area of chemical engineering, and represent difficult non-linear optimization problems, with equality & inequality constraints. Comparison is made with αBB algorithm (Adjiman et al., 1998a; 1998b), which can be used to solve problems belonging to the broad class of twice-differentiable constrained NLPs. The αBB algorithm is based on a branch-and-bound approach, where a lower bound on the optimal solution is obtained at each node through the automatic generation of a valid convex underestimating problem. It is found that DE, an exceptionally simple evolutionary computation method, is significantly faster and yields the global optimum for a wide range of the key parameters.

2. DIFFERENTIAL EVOLUTION

DE (Price & Storn, 1997) is an improved version of GA (Goldberg, 1989) 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. Price & Storn (1997) gave the working principle of DE with single strategy. Later on, they suggested ten different strategies of DE (Price & Storn, 2002). 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 and key parameters to be adopted for a problem are 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 detailed Differential Evolution algorithm used in the present study is given below:

- Choose a strategy
- Initialize the value of D (Number of independent parameters), 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 & recombination.
  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.
- After the mutation & recombination, if the bound (i.e. lower & upper limit of a variable) is violated then it can be brought in the bound range (i.e. between lower & upper limit) either by forcing it to lower/upper limit (forced bound) or by randomly assigning a value in the bound range (without forcing).
- 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.

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. Price & Storn (2002) 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: digital filter design (Storn, 1995), batch fermentation process (Chiou and Wang, 1999; Wang and Cheng, 1999), estimation of heat transfer parameters in trickle bed reactor (Babu and Sastry, 1999), optimal design of heat exchangers (Babu and Munawar, 2000; 2001), synthesis & optimization of heat integrated distillation system (Babu and Singh, 2000), optimization of an alkylation reaction (Babu and Gaurav, 2000), scenario-integrated optimization of dynamic systems (Babu and Gautam, 2001), optimization of non-linear functions (Babu and Angira, 2001a), optimization of thermal cracker operation (Babu and Angira, 2001b), a differential evolution approach for global optimization of MINLP problems (Babu and Angira, 2002) etc.

3. CASE STUDIES

3.1 *Problem-1. Heat Exchanger Network Design*

This problem addresses the design of a heat exchanger network as shown in Fig. 1. One cold stream must be heated from 100 ^{0}F to 500 ^{0}F using three hot streams with different inlet temperatures. The goal is to minimize the overall heat exchange area. It has been taken from Floudas and Pardalos (1990). It is also solved by Adjiman *et al.* (1998) using αBB algorithm.

Min $f = x_1 + x_2 + x_3$

Subject to
$0.0025(x_4 + x_6) - 1 = 0$
$0.0025(-x_4 + x_5 + x_7) - 1 = 0$
$0.01(-x_5 + x_8) - 1 = 0$
$100x_1 - x_1x_6 + 833.33252x_4 - 83333.333 \le 0$
$x_2x_4 - x_2x_7 - 1250x_4 + 1250x_5 \le 0$
$x_3x_5 - x_3x_8 - 2500x_5 + 1250000 \le 0$
$100 \le x_1 \le 10000$
$1000 \le x_2, x_3 \le 10000$
$10 \le x_4, x_5, x_6, x_7, x_8 \le 1000$
Where
x_1, x_2, and x_3 are areas of heat exchangers and
x_4, x_5, x_6, x_7, and x_8 are temperatures of streams as shown in Fig. 1.

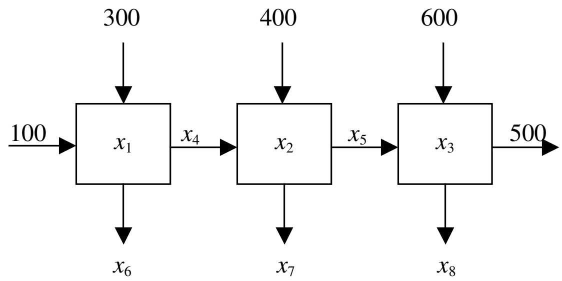

Fig. 1 Heat exchanger network design problem

The global optimum is: (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8; f) = (579.19, 1360.13, 5109.92, 182.01, 295.60, 217.9, 286.40, 395.60; 7049.25).

The above problem can be reformulated by eliminating equality constraint as given below:

Min $f = x_1 + x_2 + x_3$

Subject to

$$100x_1 - x_1(400 - x_4) + 833.33252x_4 - 83333.333 \le 0$$
$$x_2x_4 - x_2(400 - x_5 + x_4) - 1250x_4 + 1250x_5 \le 0$$
$$x_3x_5 - x_3(100 + x_5) - 2500x_5 + 1250000 \le 0$$
$$100 \le x_1 \le 10000$$
$$1000 \le x_2, x_3 \le 10000$$
$$10 \le x_4, x_5 \le 1000$$

3.2 *Problem-2. Reactor Network Design*

This example, taken from Ryoo and Sahinidis (1995), is a reactor network design problem, describing the system shown in Fig. 2. It involves the design of a sequence of two CSTR reactors where the consecutive reaction $A \rightarrow B \rightarrow C$ takes place. The goal is to maximize the concentration of product B in the exit stream. This problem is known to have caused difficulties for other global optimization methods.

Min $-x_4$

Subject to

$$x_1 + k_1x_2x_5 = 1$$
$$x_2 - x_1 + k_2x_2x_6 = 0$$
$$x_3 + x_1 + k_3x_3x_5 = 1$$
$$x_4 - x_3 + x_2 - x_1 + k_4x_4x_6 = 0$$
$$x_5^{0.5} + x_6^{0.5} \le 4$$

$(0, 0, 0, 0, 10^{-5}, 10^{-5}) \le (x_1, x_2, x_3, x_4, x_5, x_6) \le (1, 1, 1, 1, 16, 16)$.

Where $k_1 = 0.09755988$
$k_2 = 0.99k_1$
$k_3 = 0.0391908$
$k_4 = 0.9k_3$

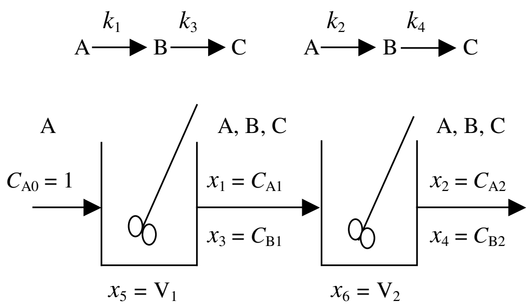

Fig. 2 Reactor network design

The global optimum is: $(x_1, x_2, x_3, x_4, x_5, x_6; f)$ = (0.771462, 0.516997, 0.204234, 0.388812, 3.036504, 5.096052; -0.388812).

This example constitute a very difficult test problem as it possesses a local minimum with an objective function value that is very close to that of the global solution. The local solutions are with f = -0.375 and f = -0.3881. Interestingly enough, the two local solutions utilize only one of the two reactors whereas the global solution makes use of both reactors. This problem can be reformulated by eliminating equality constraint as follows:

$$\text{Max } f = \frac{k_2x_6(1 + k_3) + k_1(1 + k_2x_6)}{(1 + k_1x_5)(1 + k_2x_6)(1 + k_3x_5)(1 + k_4x_6)}$$

Subject to

$$x_5^{0.5} + x_6^{0.5} \le 4$$

$(10^{-5}, 10^{-5}) \le (x_5, x_6) \le (16, 16)$.
Global optimum is same after reformulation.

4. RESULTS AND DISCUSSION

Table-1 & 2 show the results obtained using DE with/without forcing the bound on variables, and the comparison of DE with αBB algorithm respectively. The stopping criteria adopted for DE is to terminate the search process when one of the following conditions is satisfied: (1) the maximum number of generations is reached (assumed 2000 generations). (2) $| f_{\max}^k - f_{\min}^k | < 10^{-5}$ where f is the value of objective function for k-th generation. After the mutation & recombination, if the bound (i.e. lower & upper limit of a variable) is violated then it can be brought in the bound range (i.e. between lower & upper limit) either by forcing it to lower/upper limit (forced bound) or by randomly assigning a value in the bound range (without forcing). In Table-1, NFE & NRC represent respectively, the mean number of objective function evaluations and the percentage of runs converged to the global optimum in all the 10 executions (with different seed values). The key parameters of DE[1] (NP/CR/F) used for problem-1 and problem-2 are 50/0.8/0.5 and 20/0.8/0.5 respectively.

Table 1 Results of DE[1]

Problem No.	DE[1] (NFE/NRC) (Forced Bound)	DE[1] (NFE/NRC) (Without Forcing)
1.	40130/80	36620/100
2.	996/10	1210/50

DE[1] Strategy used is DE/rand/1/bin (Price and Storn, 2002)

In Problem-1, NFE without forcing is 8.75% less than NFE with forced bound (Table-1). However, for Problem-2 NFE with forced bound is 17.69% less than NFE without forcing. Also, the NRC with forced bound is not 100% in both the problems while NRC without forcing is 100% for the Problem-1. It is important to note that in Problem-2, NRC with forced bound is just 10%. It is because when upper limit of bound is violated, the value of variable is forced to the upper limit that resulted in convergence to non-optimal solution. However, for Problem-1, the NRC is 80% for forced bound. Tolerance in the present study is 10^{-5} as compared to 10^{-3} in Adjiman et al. (1998b).

The time taken by DE is much less than that of αBB algorithm (Table-2). Of course the CPU-times cannot be compared directly because different computers are used.

Table 2 Comparison of DE[1] with αBB Algorithm

Problem No.	DE[1] (CPU-time) (Without Forcing)	DE[1] (CPU-time) (Forced bound)	αBB Algorithm (CPU-time)
1.	1.445** s	1.566** s	54.4* s
2.	0.044** s	0.033** s	5.5* s

* CPU-time obtained using HP9000/730 (66MHz) with convergence tolerance of 0.001 (Adjiman et al., 1998b).

** CPU-time obtained using Pentium-III (500MHz) with convergence tolerance of 0.00001 (present study).

However, a comparison can be made after considering a factor of 10 (high enough) i.e. if the same problems would have been solved on HP9000/730 (66MHz) using DE they might have taken ten times more of CPU-time than at Pentium-III, 500MHz. Even then the CPU-time in DE is 73.44% less for the first problem and 92% less for second problem than that in αBB algorithm respectively. Therefore, it is evident that DE took least CPU-times to achieve global optima in each of the above test problems. Hence the performance of DE proved to be better than that of αBB algorithm in optimizing the nonlinear chemical engineering problems considered in the present study.

5. CONCLUSIONS

Two chemical engineering case study problems have been solved using DE in the present work. Results indicate that the bound on variables, when violated, should not be forced to lower/upper limit. In such cases, assigning a random value between lower & upper limit found to give a better convergence to global optimum. Also there was difficulty in dealing with equality constraints. However, when reformulated by eliminating these equality constraints, the algorithm exhibited high convergence (100% in Problem-1). The performance of DE is found to be the best in the two problems studied. Results indicate that DE is more reliable, efficient and hence a better approach to the optimization of non-linear problems.

6. REFERENCES

Adjiman, C.S., I.P. Androulakis and C.A. Floudas. (1998b). A Global Optimization Method, αBB, for general twice differentiable constrained NLPs – II. Implementation and computational results, *Computers & Chemical Engineering*, **22** (9), 1159 – 1179.

Adjiman, C.S., S. Dallwig, Floudas, C.A. and A. Neumaier (1998a). A Global Optimization Method, αBB, for general twice-differentiable NLPs – I. Theoretical advances. *Computers & Chemical Engineering*, **22** (9), 1137 – 1158.

Babu, B.V. and R. Angira (2001a). Optimization of Non-linear functions using Evolutionary Computation. *Proceedings of 12th ISME Conference*, India, January 10−12, 153-157 (2001).
(Also available via Internet as .pdf file at http://bvbabu.50megs.com/custom.html/#34).

Babu, B.V. and R. Angira (2001b). Optimization of thermal cracker operation using Differential Evolution. *Proceedings of International Symposium & 54th Annual Session of IIChE (CHEMCON-2001)*, Chennai, December 19-22.
(Also available via Internet as .pdf file at http://bvbabu.50megs.com/custom.html/#38) & Application No. 20, Homepage of Differential Evolution, the URL of which is: http://www.icsi.berkeley.edu/~storn/code.html

Babu, B.V. and R. Angira (2002). A Differential Evolution Approach for Global Optimization of MINLP Problems. Presented at *4th Asia-Pacific Conference on Simulated Evolution And Learning (SEAL'02)*, Singapore, November 18 – 22.
(Also available via Internet as .pdf file at *http://bvbabu.50megs.com/custom.html/#46*).

Babu, B.V. and C. Gaurav (2000). Evolutionary Computation strategy for Optimization of an Alkylation Reaction. *Proceedings of International Symposium & 53rd Annual Session of IIChE (CHEMCON-2000)*, Calcutta, December 18-21.
(Also available via Internet as .pdf file at http://bvbabu.50megs.com/custom.html/#31) & Application No. 19, Homepage of Differential Evolution, the URL of which is: http://www.icsi.berkeley.edu/~storn/code.html

Babu, B.V. and K. Gautam (2001). Evolutionary Computation for Scenario-Integrated optimization of Dynamic Systems. *Proceedings of International Symposium & 54th Annual Session of IIChE (CHEMCON-2001)*, Chennai, December 19-22.
(Also available via Internet as .pdf file at http://bvbabu.50megs.com/custom.html/#39) & Application No. 21, Homepage of Differential Evolution, the URL of which is: http://www.icsi.berkeley.edu/~storn/code.html

Babu, B.V. and S.A. Munawar (2000). 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).
(Also available via Internet as .pdf file at *http://bvbabu.50megs.com/custom.html/#28*).

Babu, B.V. and S.A. Munawar (2001). Optimal Design of Shell & Tube Heat Exchanger by Different strategies of Differential Evolution. *PreJournal.com - The Faculty Lounge, Article No. 003873, posted on website Journal* http://www.prejournal.com.
(Also available via Internet as .pdf file at http://bvbabu.50megs.com/custom.html/#35) & Application No. 18, Homepage of Differential Evolution, the URL of which is: http://www.icsi.berkeley.edu/~storn/code.html

Babu, B.V. and R.P. 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).

Babu, B.V. and K.K.N. Sastry (1999). Estimation of heat-transfer parameters in a trickle-bed reactor using differential evolution and orthogonal collocation. *Computers & Chemical Engineering,* **23**, 327–339. *(Also available via Internet as .pdf file at http://bvbabu.50megs.com/custom.html/#24)* & Application No. 13, Homepage of Differential Evolution, the URL of which is: http://www.icsi.berkeley.edu/~storn/code.html

Chiou, J.P. and F.S. Wang (1999). Hybrid method of evolutionary algorithms for static and dynamic optimization problems with application to a fed-batch fermentation process. *Computers & Chemical Engineering,* **23**, 1277-1291.

Dasgupta, D. and Z. Michalewicz (1997). *Evolutionary algorithms in Engineering Applications,* 3 - 23, Springer, Germany,.

Floudas, C.A. (1995). *Nonlinear and mixed-integer optimization*. Oxford University Press, New York.

Floudas, C.A. and P.M. Pardalos (1990). A Collection of Test Problems for Constrained Global Optimization Algorithms. *Lecture notes in computer Science*, **Vol. 455**. Springer, Germany.

Goldberg, D.E. (1989). *Genetic Algorithms in search, Optimization, and Machine learning,* Reading, MA, Addison-Wesley.

Price, K. and R. Storn (1997). Differential Evolution - A simple evolution strategy for fast optimization. *Dr. Dobb's Journal,* **22** (4)*,* 18 – 24 and 78.

Price, K. and R. Storn (2002). *Web site of DE as on November 2002,* the URL of which is: http://www.ICSI.Berkeley.edu/~storn/code.html

Ryoo, H.S., and B.P. Sahinidis (1995). Global Optimization of Nonconvex NLPs and MINLPs with Application in Process Design. *Computers & Chemical Engineering,* **19**, 551.

Salcedo, R. L. (1992). Solving Nonconvex Nonlinear Programming Problems with Adaptive Random Search. *Industrial & Engineering Chemistry Research,* **31**, 262.

Storn, R. (1995). Differential Evolution design of an IIR-filter with requirements for magnitude and group delay. *International Computer Science Institute,* TR-95-026.

Wang, F. S. and W.M. Cheng (1999). Simultaneous optimization of feeding rate and operation parameters for fed-batch fermentation processes. *Biotechnology Progress,* **15** (5), 949-952.