

EVOLUTIONARY COMPUTATION - AT A GLANCE

Prof. (Dr.) B.V.Babu*
Group Leader
Engineering Technology Group
BITS – Pilani (Rajasthan)

Introduction

The manufacturing and chemical industries have undergone significant changes during the past 15 years due to the increased cost of energy and increasingly stringent environmental regulations. Modification of both plant design procedures and plant operating conditions have been implemented in order to reduce costs and meet the constraints. Most industry observers believe that the emphasis in the near future will be on improving efficiency and increasing profitability of existing plants rather than on plant expansion. One of the most important engineering tools that can be employed in such activities is ‘optimization’. As computers have become more powerful, the size and complexity of problems which can be simulated and solved by optimization techniques, have correspondingly expanded. 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. In the recent past, non traditional search and optimization methods based on natural phenomenon (Evolutionary Computation) such as Simulated Annealing (SA) (Kirkpatrick et al., 1983), Genetic Algorithms (GA) (Goldberg, 1989), and Differential Evolution (DE) (Price & Storn, 1997), to name a few, have been developed to overcome this problem.

Differential Evolution (DE), a recent optimization technique, is an exceptionally simple evolution strategy, which is significantly faster & robust at numerical optimization and is more likely to find a function’s true global optimum (Price & Storn, 1997). Simple GA uses a binary coding for representing problem parameters whereas DE uses real coding of floating point numbers. Among the DE’s advantages are its simple structure, ease of use, speed and robustness. It can be used for optimizing functions with real variables and many local optima.

But application of DE to chemical processes is scarce (Babu & Sastry, 1999; Lee et al., 1999; Babu & Singh 2000; Babu & Munawar, 2000; Babu & Munawar, 2001; Munawar, 2000; Babu & Gaurav, 2001; Babu & Rakesh, 2001). Babu & Sastry (1999) concluded that DE takes less computational time to converge as compared to existing techniques without compromising on the accuracy of the parameter estimates. Earlier DE dealt with a single strategy (Price & Storn, 1997). Later on ten different strategies have been suggested by Price & Storn (2001). A strategy that works out to be the best for a given problem may not work well when applied for a different problem. The strategy to be adopted for each problem is to be determined separately by trial & error.

* E-mail: bvbabu@bits-pilani.ac.in; Phone: +91-1596-45073 Ext. 205; Fax: +91-1596-44183

Optimization

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 problems in engineering design or plant operation 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. The general objective in optimization is to choose a set of values of the variables subject to the various constraints that will produce the desired optimum response for chosen objective function.

There are two distinct types of optimization algorithms that are in use today. Firstly, there are algorithms which are deterministic, with specific rules for moving from one solution to the other. These algorithms (also known as traditional methods) have been successfully applied to some of the engineering design problems. Secondly, there are algorithms, which are stochastic in nature with probabilistic transition rules. These are comparatively new and are gaining popularity due to certain properties, which the deterministic algorithms do not have. As stated earlier, most of the traditional optimization algorithms based on gradient methods have the possibility of getting trapped at local optimum depending upon the degree of non-linearity and initial guess. Unfortunately, none of the traditional algorithms guarantee the global optimal solution, but GAs & SA algorithms are found to have a better global perspective than the traditional methods (Deb, 1996). Moreover, when an optimal design problem contains multiple global solutions, designers are not only interested in finding just one global optimum solution, but as many as possible for various reasons. Firstly, a design suitable in one situation may not be valid in another situation. Secondly, designers may not be interested in finding the absolute global solution. Instead they are interested in a solution, which corresponds to a marginally inferior objective function value but is more amenable to fabrication. Thus, it is always prudent to know about other equally good solutions for later use. However, if the traditional methods are used to find multiple optimal solutions, they need to be applied a number of times, each time starting from a different initial guess and hoping to achieve a different optimal solution.

Evolutionary Computation

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). A common term, accepted recently, refers to such techniques as Evolutionary Algorithms (EA) or Evolutionary Computation methods (EC methods). The best-known algorithms in this class include Genetic Algorithms, Evolutionary Programming, Evolution Strategies and Genetic Programming. 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 & Michalewicz, 1997).

Simulated Annealing (SA) is a probabilistic nontraditional optimization technique, which mimics the cooling phenomenon of molten metals to constitute a search procedure. Rutenbar (1989) gave a detailed discussion of the working principle of SA and its applications. Since its introduction SA has diffused widely into many diverse applications. The various applications of SA include: traveling salesman problem, scheduling of serial multi-component batch processes, heat exchanger network synthesis, preliminary design of multi-product non continuous plants, separation sequence synthesis, synthesis of utility systems, reactor networks synthesis, optimal design of heat exchangers, etc.

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 a search process. The key parameters of control in GA are: N , the population size; p_c , the crossover probability; and p_m , the mutation probability (Goldberg, 1989; Deb, 1996). Since their inception, GAs have evolved like the species they try to mimic and have been applied successfully in many diverse fields. The various applications of GAs are: process design and optimization, computer-aided molecular design, heat integrated processes, synthesis & optimization of non-ideal distillation systems, design of ammonia synthesis reactor, online optimization of culture temperature for ethanol fermentation, generating initial parameter estimations for kinetic models of catalytic processes, molecular scale catalyst design, estimation of heat transfer parameters in trickle bed reactors, automated design of heat exchangers using artificial intelligence based optimization, optimal design of heat exchangers, etc.

Price & Storn (1997) have given the working principle of Differential Evolution (DE), which is an improved version of GA, along with its application to polynomial fitting problems. They have also suggested some simple rules for choosing the key parameters such as NP- the population size, CR- crossover constant and F- the weight applied to random differential (scaling factor) of DE for any given application. In their web site, Price & Storn (2001) have given the various recent applications of DE along with the program codes in various computer languages. They also suggested ten different working strategies of DE and some guidelines in applying these strategies for any given problem. Different strategies can be adopted in DE algorithm depending upon the type of problem for which DE is applied. The strategies can vary based on the vector to be perturbed, number of difference vectors considered for perturbation, and finally the type of crossover used. The following are the ten different working strategies proposed by Price & Storn (2001):

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 above 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). Thus, the working algorithm outlined by Price & Storn (1997) is the seventh strategy of DE i.e. DE/rand/1/bin. Hence the perturbation can be either in the best vector of the previous generation or in any randomly chosen vector. Similarly for perturbation either single or two vector differences can be used. For perturbation with a single vector difference, out of the three distinct randomly chosen vectors, the weighted vector differential of any two vectors is added to the third one. Similarly for perturbation with two vector differences, five distinct vectors, other than the target vector are chosen randomly from the current population. Out of these, the weighted vector difference of each pair of any four vectors is added to the fifth one for perturbation. In exponential crossover, the crossover is performed on the D (the dimension, i.e., number of variables to be optimized) variables in one loop until it is within the CR bound. For discrete optimization problems (such as Design of shell & tube heat exchanger), first time a randomly picked number between 0 and 1 goes beyond the CR value, no crossover is performed and the remaining D variables are left intact (Babu & Munawar, 2001). In binomial crossover, the crossover is performed on each of the D variables whenever a randomly picked number between 0 and 1 is within the CR value. So for high values of CR, the exponential and binomial crossovers yield similar results. The strategy to be adopted for each problem is to be determined separately by trial and error. A strategy that works out to be the best for a given problem may not work well when applied for a different problem.

Choosing the values of NP, F, and CR depends on the specific problem applied, and is often difficult. But some general guidelines are available. Generally, 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 and 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 (Price & Storn, 1997). The best combination of these key parameters of DE for each of the strategies mentioned earlier is again different. Price & Storn (2001) have mentioned some simple rules for choosing the best strategy as well as the corresponding key parameters. Among DE's advantages are its simple structure, ease of use, speed and robustness.

DE has been successfully applied in various fields: digital filter design, neural network learning, fuzzy decision making problems of fuel ethanol production, design of fuzzy logic controllers, batch fermentation process, multi sensor fusion, dynamic optimization of continuous polymer reactor, estimation of heat transfer parameters in trickle bed reactor, optimal design of heat exchangers, optimal design of shell & tube heat exchanger, synthesis & optimization of heat integrated distillation systems, expert systems for the optimal design of heat exchangers, etc.

There are many chemical processes, which are highly non-linear and complex with reference to optimal operating conditions with many equality and inequality constraints. Evolutionary Computation in general, and Differential Evolution Strategies in particular could be a solution for the above optimization problems.

References

- Babu, B. V. & Gaurav Chaturvedi (2000). Evolutionary Computation strategy for optimization of an Alkylolation Reaction. *Proceedings of 53rd Annual session of IChE (CHEMCON-2000)*, Science City, Calcutta, December 18 – 21, 2000.
- Babu, B. V. & Munawar, S. A. (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, March 11, 2000.
- Babu, B. V. & Munawar, S. A. (2001). Optimal Design of Shell & Tube Heat Exchanger by Different strategies of Differential Evolution. *PreJournal.com – The Faculty Lounge, Article No. 003873*, website Journal <http://www.prejournal.com>, March 03, 2001.
- Babu, B. V. & Rakesh Angira (2001). Optimization of Non-Linear Functions Using Evolutionary Computation. *Proceedings of 12th ISME Conference on Mechanical Engineering*, Chennai, January 10-12, 2001.
- 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, March 11, 2000.
- 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.
- Dasgupta, D. & Michalewicz, Z. (1997). *Evolutionary algorithms in Engineering Applications*. Germany, Springer.
- Deb, K. (1996). *Optimization for engineering design: Algorithms and examples*. New Delhi, Prentice-Hall.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*, Reading, MA: Addison-Wesley.
- Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4568), 671-680.
- 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.
- Munawar, S. A. (2000). Expert Systems for the optimal design of heat exchangers. *M. E. Dissertation*, B.I.T.S., Pilani.
- Price, K. & Storn, R. (1997). Differential evolution. *Dr. Dobb's Journal*, 18-24.
- Price, K. & Storn, R. (2001). Differential Evolution Homepage (Web site of Price and Storn) as on April, 2001. <http://www.ICSI.Berkeley.edu/~storn/code.html>
- Rutenbar, R. A. (1989). Simulated annealing algorithms: An overview. *IEEE Circuits & Devices Magazine*, 100, 19-26.