Genetic Annealing Optimization: Design and Real World Applications

Mostafa A. El-Hosseini[1], Aboul Ella, Hassanien[2], Ajith Abraham[3], Hameed Al-Qaheri[2]

[1]*IRI - Mubarak city for Science and Technology, University and Research District,*
P.O. 21934 New Borg Alarab, Alex, Egypt
Email: melhosseini@mucsat.sci.eg & melhosseini@gmail.com
[2]*Department of Quantitative Methods and Information Systems, CBA, Kuwait University*
[3]*Norwegian University of Science and Technology,*
O.S. Bragstads plass 2E, NO-7491 Trondheim, Norway
Email: ajith.abraham@ieee.org, abraham.ajith@acm.org

Abstract

Both simulated annealing (SA) and the genetic algorithms (GA) are stochastic and derivative-free optimization technique. SA operates on one solution at a time, while the GA maintains a large population of solutions, which are optimized simultaneously. Thus, the genetic algorithm takes advantage of the experience gained in the past exploration of the solution space. Since SA operates on one solution at a time, it has very little history to use in learning from past trials. SA has the ability to escape from any local point; even it is a global optimization technique. On the other hand, there is no guarantee that the GA algorithm will succeeded in escaping from any local minima, thus it makes sense to hybridize the genetic algorithm and the simulated annealing technique. In this paper, a novel genetically annealed algorithm is proposed and is tested against multidimensional and highly nonlinear problems: Fed-batch fermentor for Penicillin production, and isothermal continuous stirred tank reactor CSTR. It is evident from the results that the proposed algorithm gives good performance.

1. Introduction

The dynamic optimization of fed-batch bioreactors is a very challenging problem due to several reasons. First, the control variable (feed rate) appears linearly in the system differential equations, so the problem is singular, creating additional difficulties for its solution (especially using indirect methods). For this type of problems, the optimal operating policy will be either bang-bang, or singular, or a combination of both. Second, most bioprocesses have highly nonlinear dynamics, and constraints are also frequently present on both the state and the control variables. These characteristics introduce new challenges to the existing solution techniques, as it will be discussed below. Therefore, efficient and robust methods are needed in order to obtain the optimal operating policies.

The well-known numerical optimization methods [2], [3], [4], [5], [6], [7], [8], [9], [10] of nonlinear programming do not always lead to acceptable solutions in practical problems, often becoming entrapped in local minima instead of yielding global solutions.

In contrast, many stochastic methods can locate the vicinity of global solutions with relative efficiency [1], [11], [19], but the cost to pay is that global optimality can not be guaranteed. However, in practice we can be satisfied if these methods provide us with a very good (often, the best available) solution in modest computation times. Furthermore, stochastic methods are usually quite simple to implement and use, and they do not require transformation of the original problem, which can be treated as a black box [12], [13], [14], [15], [16]. It has been widely confirmed [1], [11], [19] that real-number encoding performs better than binary or Gray encoding for function optimizations and constrained optimizations. Real-coded GA is faster, require low memory, has high precision, however there is no guarantee that it could succeeded to escape from local optimum.

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The heat causes the atoms to become unstuck from their initial positions (a local minimum of the internal energy) and wander randomly through states of higher energy; the slow cooling gives them more chances of finding configurations with lower internal energy than the initial one.

In the genetic algorithm, each new individual is constructed from two previous individuals, which means that in a few iterations, all the individuals in the population have a chance of contributing their good features to form one super-individual.

So, GA adds the experience learned while SA has the high probability to escape from the local optimum, thus it makes sense to hybridize the two algorithms. The new devised algorithm is presented in Section 2. Section 3 covers results and computer simulations followed by conclusion towards the end.

2. Algorithm

The proposed algorithm starts with Initial random generation, then mark the best individual, start the annealing process, and then replace the best one, with the output of the annealed process, then do crossover, and continue until the stopping criteria is met. The notion of annealing the best chromosome is pretty good by logic, since we try to eliminate the effect of the best chromosome to direct the search toward that solution "usually it is the local one", and start to anneal it to find the stable solution "global". Also, annealing the best solution helps in exploring new areas of search space. The proposed algorithm is as follows:

step [1] Create Initial population that contains the candidate solutions in Phenotype form.

step [2] Evaluate the fitness function for all individuals

step [3] if < stopping criteria not met > do

i. Selection

ii. Choose suitable Crossover

iii. Choose the best individual, mark it, Then start annealing, (replace the best one with the annealed)

iv. Generate new population

v. Go to step 2, otherwise Exit

The Glauber algorithm [18] accepts all moves with the following probability:

$$\frac{\exp(-move\_value/T)}{1+\exp(-move\_value/T)} \quad (1)$$

So here an improving move may be rejected. This leads to a search that is well diversified, so it will come closer to a global optimum, but may take longer than a Metropolis-based search, which is more likely to find a good solution quickly.

The core of the proposed algorithm is the real-coded GA. So, the algorithm requires low memory to run, has high precision, and it is faster than binary coded GA. Also the algorithm does not have mutation operator. Annealing the genetic algorithm helps reducing the number of chromosomes and the number of generations required to converge.

3. Experimental Results

To judge the performance and efficiency of the proposed technique, two real world problems are selected.

3.1. Optimal Control of a Fed-batch fermentor for Penicillin Production

A model of a fed-batch fermentor for the production of Penicillin [20] is illustrated in Figure 1. The objective function is to maximize the amount of Penicillin produced using the feed rate as the control variable.

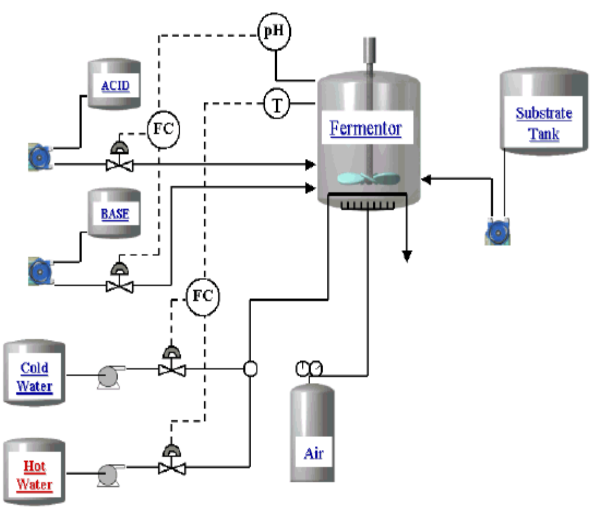

Figure 1. Fed-batch penicillin fermentor

Find $u(t)$ and t_f over $t \in [t_0, t_f]$ to maximize $J = x_2(t)x_4(t)$

Subject to

$$\frac{dx_1}{dt} = h_1 x_1 - u\left(\frac{x_1}{500x_4}\right) \tag{2}$$

$$\frac{dx_2}{dt} = h_2 x_1 - 0.001x_2 - u\left(\frac{x_2}{500x_4}\right) \tag{3}$$

$$\frac{dx_3}{dt} = -\frac{h_1 x_1}{0.47} - h_2\frac{x_1}{1.2} - x_1\left(\frac{0.029x_3}{0.0001+x_3}\right) + \frac{u}{x_4}\left(1-\frac{x_3}{500}\right) \tag{4}$$

$$\frac{dx_4}{dt} = \frac{u}{500} \tag{5}$$

$$h_1 = 0.11\left(\frac{x_3}{0.006x_1 + x_3}\right) \tag{6}$$

$$h_2 = 0.0055\left(\frac{x_3}{0.0001 + x_3(1+10x_3)}\right) \tag{7}$$

where x_1, x_2, and x_3 are the biomass, penicillin and substrate concentration, and x_4 is the volume. The initial conditions are: $x(t_0) = [1.5 \quad 0 \quad 0 \quad 7]^T$. The upper and lower bounds on the state variables are

$$\begin{aligned} 0 \le x_1 \le 40 \\ 0 \le x_3 \le 25 \\ 0 \le x_4 \le 10 \end{aligned} \tag{8}$$

The upper and lower bounds on the feed rate are

$$0 \le u \le 50 \tag{9}$$

The parameters setting of the proposed algorithm is *random function*, *Glauber Accepting function* , number of cycles equal to 2, *Maximum Annealing Temperature* $T_{\max} = 100$, *Frozen Annealing Temperature* = 94, *Annealing Schedule* $\lambda = 0.95$. The number of chromosomes used is 20, number of generation is 30. When applying the suggested technique with penalty to the fermentor problem at hand, it gives performance index equal to $J = 83.1827$, and the optimal control vector $u = 11.3190$, for $t_f = 132h$. The biomass, penicillin and substrate concentrations, and the volume are depicted in Figure 2.

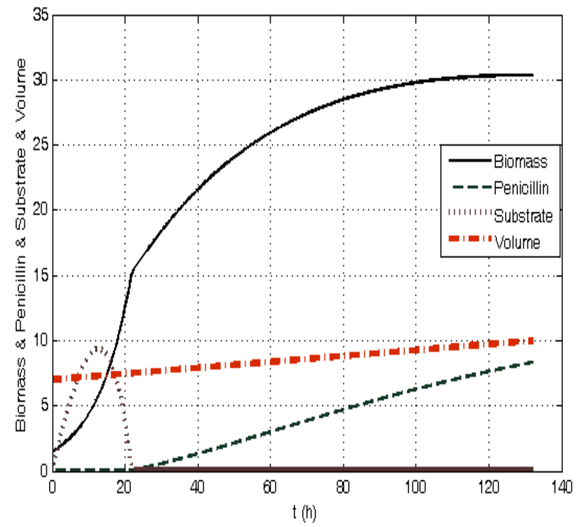

Figure 2. Penicillin production fermentor states

The effect of annealing can be shown by plotting the best feed rate before and after annealing in all the 30 generations, from the Figure 3, it is evident that the annealing tries to reduce the effect of the best point and try to reduce the speed of convergence “may be “ to local optimum.

It is evident that the suggested technique never losses the best point and it is always seeks the best point until it reaches it and never forget. The convergence of the proposed algorithm is observed though plotting the first generation with generation No. 15, and Generation No. 15 against generation No. 25, please see Figures 4 and 5.

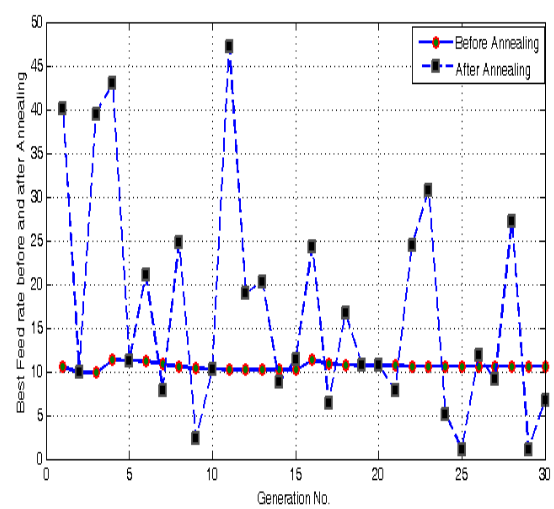

Figure 3. Annealing effect

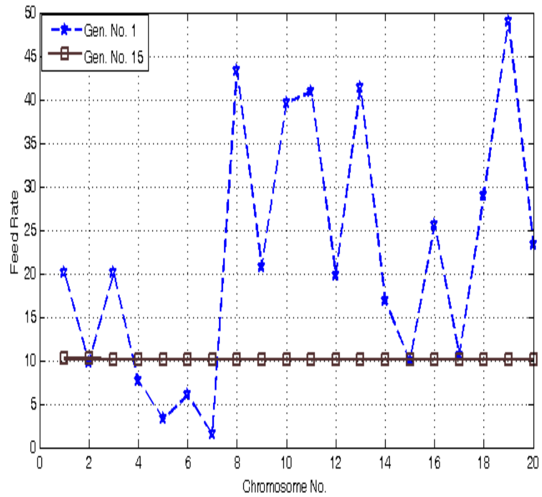

Figure 4. Generation no. 1 against generation no.15

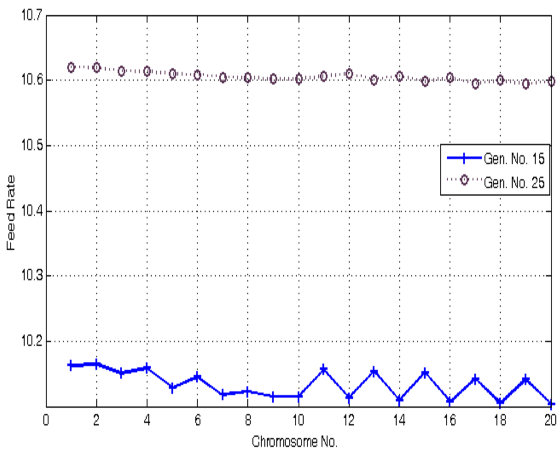

Figure 5. Generation no. 15 against generation no. 25

The Zoom in view of the all chromosomes values in the last generation show that most of the chromosomes in the last generation are succeeded to find the optimal value of the control vector, see Figure 6. The best performance index for each generation is illustrated in Figure 7.

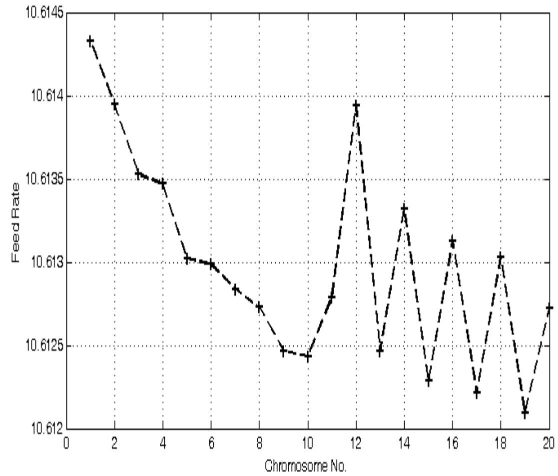

Figure 6. Zoom in picture of the last generation

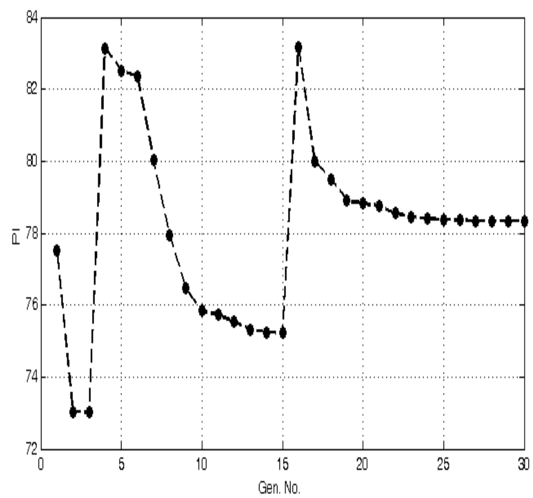

Figure 7. Best PI in each generation

3.2. Isothermal Continuous Stirred Tank Reactor (CSTR)

The system equations are [17]:

$$\dot{x}_1 = 6.0 - qx_1 - 17.6x_1x_2 - 23.0x_1x_6u_3 \tag{10}$$

$$\dot{x}_2 = u_1 - qx_2 - 17.6x_1x_2 - 146.0x_2x_3 \tag{11}$$

$$\dot{x}_3 = u_2 - qx_3 - 73.0x_2x_3 \tag{12}$$

$$\dot{x}_4 = -qx_4 + 35.2x_1x_2 - 51.3x_4x_5 \tag{13}$$

$$\dot{x}_5 = -qx_5 + 219.0x_2x_3 - 51.3x_4x_5 \tag{14}$$

$$\dot{x}_6 = -qx_6 + 1026x_4x_5 - 23.0x_1x_6u_3 \tag{15}$$

$$\dot{x}_7 = -qx_7 + 46.0x_1x_6u_3 \tag{16}$$

Where

$$q = 6.0 + u_1 + u_2 \tag{17}$$

and the performance index to be maximized is

$$I = \int_0^{0.2} \begin{pmatrix} 5.8(qx_1 - 6) - 3.7u_1 - 4.1u_2 - 5u_3^2 + \\ q(23x_4 + 11x_5 + 28x_6 + 35x_7) - 0.099 \end{pmatrix} dt \tag{18}$$

The initial starting point $x(0)$ is the vector (0.1883, 0.2507, 0.0467, 0.0899, 0.1804, 0.1394, 0.1046) and the controls are bounded as follows $0 \le u_1 \le 20$, $0 \le u_2 \le 6$ and $0 \le u_3 \le 4$. This is a 7-dimensional system with 3 control parameters and would appear to be a severe test for the suggested algorithm.

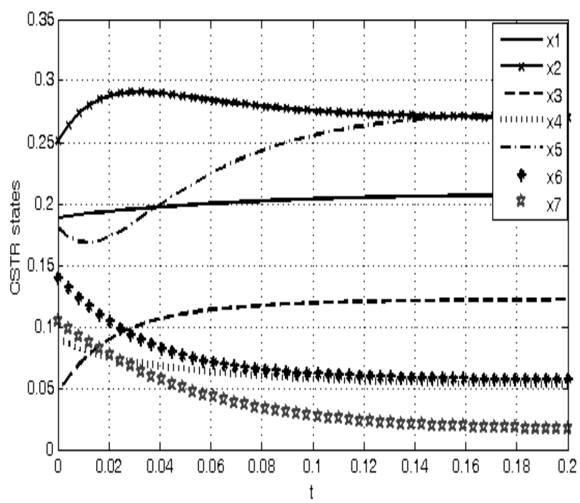

Figure 8. States of CSTR (30 generations)

The parameters setting of the proposed algorithm is *random function*, *Glauber Accepting function* , number of cycles equal to 5, *Maximum Annealing Temperature* $T_{\max} = 100$, *Frozen Annealing Temperature* = 60, *Annealing Schedule* $\lambda = 0.95$. The performance index equal to 19.0524, u1 equal to 12.0493, u2 equal to 5.2467, and u3 equal to 0.6674. The 7 states are depicted in Figure 8.The best Performance index in each generation is illustrated in Figure 9 and the annealing effect is depicted in Figure 10.

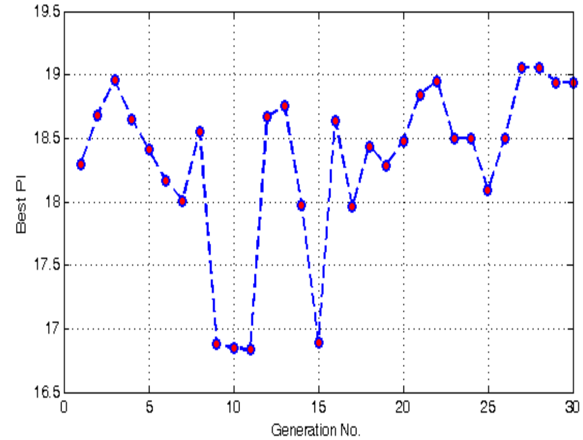

Figure 9. Best PI in each generation

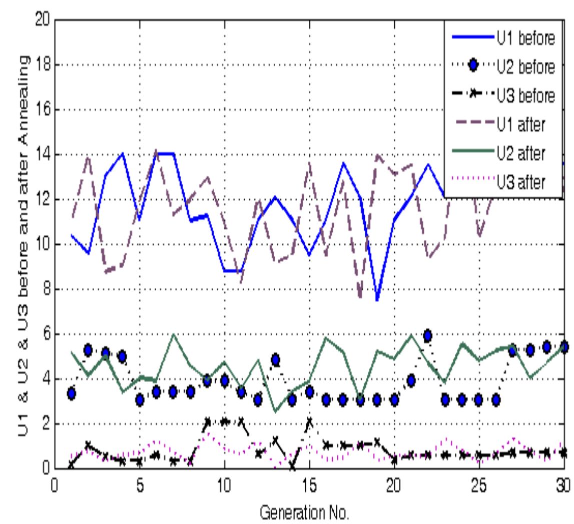

Figure 10. Annealing effect

4. Conclusions

The proposed genetic annealing algorithm succeeded in reducing the number of chromosomes, number of generations and is also able to reduce the effect of premature convergence to a certain extent. In addition, there is no mutation parameter used. The suggested algorithm takes the advantage of the real-coded GA; benefits from the experience, and the advantage of SA; the ability to escape from local optimum. The application of the proposed algorithm using the highly nonlinear and multidimensional case studies shows that the algorithm performed very well.

References

[1] A. E. Eiben, R. Hinterding, Z. Michalewicz, “Parameter Control in Evolutionary Algorithms”, IEEE Transactions on Evolutionary Computation, Volume 3, Issue 2, Page(s):124 – 141, July 1999

[2] C. G. Broyden, “The Convergence of a Class of Double-rank Minimization algorithms," J. Inst. Maths. Applies., Vol. 6, pp 76-90, 1970.

[3] D. F. Shanno, “Conditioning of Quasi-Newton Methods for Function nimization”, Mathematics *of Computing,* Vol. 24, pp 647-656, 1970.

[4] D. Goldfarb, “A Family of Variable Metric Updates Derived by Variational Means”, Mathematics *of Computing,* Vol. 24, pp 23-26, 1970.

[5] D. Marquardt, “An Algorithm for Least-Squares Estimation of Nonlinear Parameters” *SIAM J. Appl. Math.* Vol. 11, pp 431-441, 1963.

[6] J. A. Nelder and R. Mead, “ A Simplex method for function minimization”, Computer Journal , Vol. 7, pp 308-313, 1965

[7] J. Nocedal and S. J. Wright. “Numerical Optimization”, Springer series in operations Research, New York 1999

[8] JR. Leigh, "Control Theory", the institute of Electrical Engineering, London, Uinted Kingdom, 2004

[9] K. Levenberg, “A Method for the Solution of Certain Problems in Least Squares”, *Quart. Appl. Math.* Vol. 2, pp 164-168, 1944

[10] M. C. Biggs, “Constrained Minimization Using Recursive Quadratic Programming," Towards Global Optimization (L.C.W. Dixon and G.P. Szergo, eds.), North-Holland, pp 341-349, 1975

[11] M. Gen, R. Cheng, “Genetic Algorithms And Engineering Optimization”, A Wiley-Interscience Publication, 2000

[12] P. Mazumder, E. M. Rudnick, “Genetic Algorithms for VLSI Design, Layout & Test Automation”, Prentice Hall PTR, 1999

[13] R. E. King, “Computational Intelligence in Control Engineering”, Marcel Decker NY 1999, Revision 1/2005

[14] R. Fletcher, “Practical Methods of Optimization” John Wiley and Sons, 1987.

[15] R. L. Haupt, S. E. Haupt, "Practical Genetic Algorithms", second Edition, Wiley Interscience, 2004

[16] R. S. Burns, "Advanced Control Engineering", ButterWorth Heinemann, 2001

[17] S. Smith, “An evolutionary program for a class of continuous optimal control problems”, IEEE Int. Conf. on Evolutionary Computation, 29 Nov. 1995, Vol. 1, pp. 418 – 422

[18] T. F. Edgar, D. M. Himmelblau, and L. S. Lasdon, “Optimization of Chemical processes”, McGraw-Hill, 2001

[19] W. Chang, “An improved real-coded GA for Parameters estimation of nonlinear systems “, Mechanical Systems and Signal processing Journal, Science Direct 2006

[20] W. D. Seider, J. R. Banga, "Global Optimization of Chemical Processes using stochastic algorithms", in State of the Art in Global Optimization, Floudas C.A. and Pardalos PM (Eds), pp563-583 Kluwer Academic, 1996