APPLICATIONS OF GENETIC ALGORITHMS IN CHEMICAL ENGINEERING

Vijaykumar Hanagandi[1] and Michael Nikolaou[2],
Texas A&M University
College Station, TX 77843-3122
INTERNET: m0n2431@venus.tamu.edu

Abstract

Genetic Algorithms (GAs) are emerging as powerful alternatives to traditional optimization methods which are too restrictive and CPU intensive. The field of chemical engineering offers challenging optimization and search problems. We show that, apart from being able to efficiently handle highly nonlinear, multimodal and nonlinear objective functions, GAs can also handle cases where the objective function is not clearly defined. Three case studies are presented to illustrate the effectiveness of GAs in solving complex optimization problems. The FORTRAN code along with the templates for the case studies can be obtained from the authors.

Introduction

Genetic Algorithms (GAs) are emerging as powerful alternatives to traditional optimization methods which are too restrictive and CPU intensive. Genetic Algorithms (GAs) accomplish the task of optimization by starting with a random "population" of values for the parameters of an optimization problem, and thereafter producing new "generations" of improved values that combine the best "parts" of values from previous populations.

The field of chemical engineering offers challenging optimization and search problems. In this paper we use GAs to solve representative problems in design and control of chemical processes and to solve a complex problem in

[1]present address: MS C312, Los Alamos National Laboratory, Los Alamos, NM 87545.

[2]Author to whom all correspondence should be addressed. For software enquiries please send e-mail to: vmh@killdeer.lanl.gov or vmh0648@sigma.tamu.edu

transport phenomena. In the light of some unique problems encountered in chemical engineering applications the following points summarize the advantages of using GAs to solve them:

- As a GA proceeds randomly (yet systematically) in its search, it does not require smoothness, derivability, continuity etc. in the objective function. The only requirement is that for a set of values for the optimization parameters, one must assign a fitness value. This feature makes it possible to solve complex optimization/search problems where the concept of an objective function is itself fuzzy and is more qualitative than quantitative.

- Through computer experimentation and some heuristic analysis GAs have been found to arrive at or close to the global optimum of an objective function.

- GAs can efficiently handle highly nonlinear and noisy objective functions as encountered in stochastic processes where traditional gradient based methods are inefficient.

- GAs are amenable to parallel processing. Unlike in gradient search algorithms, in GAs the objective function evaluation for one parameter set is independent of that for all others in the same generation. This facilitates the use of parallel computers for the search procedure.

We refer the reader to standard literature (Goldberg, 1989, Davis, 1991) for details of the theory and implementation of GAs. Here we discuss three case studies which represent a cross section of chemical engineering applications. The FORTRAN software and manual can be obtained from the authors. Parameters of GAs employed in the three case studies are grouped at the end of the third case study in Table. 4.

Case Study 1: Best Controller Synthesis using Qualitative Criteria

In this section we demonstrate the use GAs to accomplish controller design task with quantitative goals to be accomplished. The goals are defined by production engineers and operators. First, we illustrate our methodology on a text book example and then solve the linear controller design problem for a more complex nonlinear multi-input multi-output (MIMO) plant.

A Text Book Example

The following example illustrates the use of GAs for qualitative optimization:

Plant:

$$P(s) = \frac{1}{(s+1)(s+2)}$$

Controller:

$$C(s) = K_c(1 + \frac{1}{\tau_I s} + \tau_D s)$$

Problem:

Find the values of K_c, τ_I and τ_D so that the response of the closed loop for a step change in the set-point has the following qualitative features:

- it has *fast* rise;
- it has *smooth* rise;
- it has a *stable* response.

The optimization procedure is initialized by ten randomly generated sets of values for K_c, τ_I and τ_D in the intervals $[0, 10]$, $[0, 10]$ and $[0, 10]$ respectively. This set of parameters is the first generation. Using these values, closed loop simulations are conducted and the responses are ranked by the designer/operator, who decides the ranking by assigning a numerical value to each response. These fitness values do not correspond to integral square error (ISE) or any quantitative characteristic of the responses but they reflect the quality of the responses as seen by the expert with the abovementioned qualitative goals in mind. In the absence of such a fitness value only a mere ranking is enough for the GA to proceed, but convergence might be slow. Once the rankings and/or fitness values are assigned, the GA produces the parameter set values corresponding to the second generation and the above evaluation procedure is repeated. In this example problem runs up to ten generations were performed. Figs. 1.1 through 10.10 give the responses of the closed loop to step changes in the set-point. These figures are arranged in their decreasing order of fitness in each generation. As we note from the figures, the GA has managed to recognize our qualitative objective and is giving

more weight to those candidates (in the parameter set) that have desirable features thereby successively improving the responses.

For comparison purposes, the optimal solution for this problem (according to the internal model control (IMC) principle) is

$$K_c = 10 \quad (ideally \quad \infty) \qquad \tau_I = 1.5 \qquad \tau_D = 0.333$$

and the corresponding step-change response is shown in Fig. 11.

Best linear controller synthesis for a nonlinear plant

Chien and Ogunnaike (1992) give details of the high-purity distillation column we attempt to control. For control purposes, this is a two-input (y_1, y_2) two-output (u_1, u_2) dynamical system. The open loop responses to various step changes in the inputs indicate that the system is highly nonlinear. The controller structure comprises of two proportional-integral (PI) loops as indicated in Fig. 12. kc_1, $\tau_{I,1}$, kc_2 and $\tau_{I,2}$ are the parameters to be tuned. We follow the steps exactly as outlined in the previous example to arrive at the best PI settings. The set-point changes in the simulations are $[0, 0] \longrightarrow [0.05, 0]$. In many plant operations, set point changes are known beforehand (eg. in startup, shutdown, etc.) so the optimal linear controller can be tuned using all those set-point changes in the simulations. The criteria used to evaluate controller fitness were:

- *fast* rise;
- *smooth* rise;
- *stable* responses with *less* oscillations.
- *realizable* control actions (both move sizes and move velocities).

Figs. 13.1 through 13.10 show the simulations for the first generation controller performances. As one can imagine, this set has a wide range of random controller settings and performances. Figs. 14.1 through 14.10 give the performances of the controller settings in the tenth generation. The best setting of this set is shown in Fig. 14.1 which shows an improvement in performance over the best setting of the first generation shown by Fig. 13.1. Also, unlike the first generation performances the tenth generation settings give more

stable and good responses which satisfy the qualitative criteria showing that the GA is converging towards better and better settings.

This example illustrates that GAs can successfully converge to controller settings to meet qualitative goals. An advantage of this technique is that in deciding which is the best response the experience of the operator or the engineer is used. This allows us to incorporate qualitative criteria which can not be captured by quantitative measures like ISE, percent overshoot, etc. Also, in our study we found that this technique is robust in the sense that some (but inevitable) lack of consistency in fitness assignments is tolerable.

Case Study 2: Optimization of back mix reactors in series

In this section we examine the classic problem of optimally designing a continuously stirred tank reactor (CSTR) train. In the past this problem has been studied by various researchers (Szépe and Levenspiel (1964), Wood and Stevens (1964), Luss (1965), Crooks (1966) and Edgar and Himmelblau (1988)). In a train of four CSTRs the problem is to design CSTR volumes to achieve maximum conversion. The sum of the individual CSTR volumes is constrained by a prespecified value. This problem is successfully solved for an arbitrary order, irreversible, single reaction power law kinetics by previous researchers. For comparison purposes, we solve the same problem by a GA. The reaction

$$A \longrightarrow B \qquad r = kA^n$$

takes place under isothermal conditions in a series of four CSTRs whose dynamics are given by

$$\frac{d(V_i c_i)}{dt} = F c_{i-1} - F c_i - r_i V_i, \qquad i = 1, ..., 4 \tag{1}$$

The L.H.S of the above equation is set to zero and the exit concentrations $c_i, i = 1, ..., 4$, of the four CSTRs are solely determined by inlet flow rate F, reaction constraint k and feed concentration c_0. Parameter values and variable nomenclature are given in Table 1. Since GAs are insensitive to the analytic properties of the objective function they can handle any general kinetic expression. Here a GA is used as a function optimizer to solve the following:

$$\min_{V_1, ..., V_4} \quad [c_4]$$

subject to

$$V_1 + V_2 + V_3 + V_4 = 20m^3$$

In each fitness evaluation, the routine FZERO (Kahaner, Moler and Nash, 1988) is used to solve for the steady state algebraic equation yielding c_4 and the fitness is set equal to $-c_4$. When the constraints are violated the fitness is set equal to that of the minimum fitness encountered in that generation. Fig. 15 shows the evolution of the solution. The fact that the average population minima approaches that of the best member in each population indicates that the minimum is indeed global. Table 2 shows a comparison between this study and that of Edgar and Himmelblau (1988).

In gradient based solution of this problem, as c_4 can not be solved explicitly as a function of c_0, c_4 is held constant and c_0 is maximized w.r.t. $V_i, i = 1, ..., 4$ in each cycle. If the maximum c_0 does not match with $20\frac{kgmol}{m^3}$, c_4 is changed using linear interpolation and the optimization is done again. This procedure is continued until c_0 matches the given inlet concentration of $20\frac{kgmol}{m^3}$. To demonstrate the complexity of this technique a sequential quadratic programming (SQP) technique (Zhou and Tits, 1989) was used to arrive at optimal V_is for $c_4 = 0.3961\frac{kgmol}{m^3}$. 1190 function evaluations were needed to converge to the maximum inlet concentration of $20\frac{kgmol}{m^3}$ (see Fig. 16 for the convergence profile). Obviously, many more function evaluations are needed if we start from an arbitrary initial value of c_4 and change c_4 after each cycle. Clearly this is a very cumbersome method.

Table. 1. Parameter values and nomenclature for the CSTR train

symbol	*meaning*	*value*
F	feedrate	$71\frac{m^3}{hr}$
$V_i, i = 1, 2, 3, 4$	volume of i^{th} reactor	variable
$c_i, i = 1, 2, 3, 4$	concentration of A in i^{th} CSTR	variable
c_0	concentration of species A in the feed to the first reactor	$20\frac{kgmol}{m^3}$
$r_i, i = 1, 2, 3, 4$	rate of dissipation of A in CSTR i	$= kc_i^n$
n	reaction order	2.5
k	reaction constant	$0.00625[\frac{m^3}{kgmol}]^{1.5}s^{-1}$

Table. 2. Results of Case Study 2

Variable value at the minimum	*This Study*	*Edgar and Himmelblau (1988)*
c_4	$0.3962\frac{kgmol}{m^3}$	$0.3961\frac{kgmol}{m^3}$
V_1	$2.234m^3$	$2.242m^3$
V_2	$3.698m^3$	$3.884m^3$
V_3	$6.163m^3$	$5.849m^3$
V_4	$7.905m^3$	$8.025m^3$

Case Study 3: Solution of lattice model to predict the adsorption of polymer molecules[3]

In this section we discuss a method of solving the lattice model by GAs and compare it with a classical technique. Next, we state the lattice model and pose the optimization problem. We will solve the optimization problem using GAs. We will also suggest ways to improve the convergence properties of GAs and compare the results of these modifications with Levenberg-Marquardt technique.

The lattice model

The lattice model aims at computing the structure of adsorbed polymer molecules near the surface and into the bulk. It aims at computing probabilities of various polymer chain conformations by using physical properties. The structure is described by two parameters viz. ϕ_i the segment volume fraction in layer i and P_i the free segment probability in layer i. As we see below, the physics of the problem is such that computation of P_i and ϕ_i is not straightforward since both depend on one another implicitly. The probability of finding a free segment (monomer) in layer i is defined by

$$\ln P_i = \chi_s \delta_{1,i} + \chi(\langle \phi_i \rangle - \langle \phi_i^0 \rangle) + \ln \phi_i^0 \tag{2}$$

where

$$\langle \phi_i \rangle = \sum_{j=1}^{M} \lambda_{j-i} \phi_j \tag{3}$$

[3]The authors wish to thank Dr. Harry J. Ploehn of Texas A&M University for his assistance in this case study.

and

$$\langle \phi_i^0 \rangle = \sum_{j=1}^{M} \lambda_{j-i} \phi_j^0 \tag{4}$$

The free segment probability P_* for a segment in the bulk solution is defined by

$$\ln P_* = \chi(\phi_* - \phi_*^0) + \ln \phi_*^0. \tag{5}$$

Now the free segment probability p_i with respect to the bulk solution is

$$p_i = \frac{P_i}{P_*} \tag{6}$$

which, with

$$\phi_* = \frac{nr}{Lp(r)} \tag{7}$$

gives

$$\phi_i = \frac{\phi_*}{r} \frac{1}{p_i} \sum_{s=1}^{r} p(i,s) p(i, r-s+1) \tag{8}$$

where $p_{i,s}$ is end segment probability, i.e. the probability that the end segment of an s-mer is in layer i and

$$p(i,s) = \frac{P_{i,s}}{P_*^s} = p_i[\lambda_1 p(i-1, s-1) + \lambda_0 p(i, s-1) + \lambda_1 p(i+1, s-1)] \tag{9}$$

Table. 3 gives the parameter values, for more details see Scheutjens et al. (1979).

Table. 3. Lattice model parameters and variables.

symbol	*meaning*	*value*
M	number of lattice layers	20
r	number of chains per segment	1000
ϕ_*	segment volume fraction in bulk	0.01
λ_0, λ_1	fraction of nearest neighbors in the same layer, and in the adjacent layer	0.5,0.25
χ	Flory-Huggins polymer solvent interaction parameter	in Case 1.: 0.0 in Case 2.: 0.5
χ_s	differential adsorption energy parameter	1.0
ϕ_i	segment volume fraction	variable
ϕ_i^0	solvent volume fraction in layer i	$= 1 - \phi_i$
ϕ_*^0	solvent volume fraction in the bulk solution	$= 1 - \phi_*$
p_i	free segment probability w.r.t. the bulk solution	variable

A general framework to solve the lattice model

The lattice model can be solved in an iterative manner as follows:

1. Guess ϕ_i, $i = 1, 2, ...M$
2. Solve for p_i, $i = 1, 2, ...M$, using Eqs. 7, 8, 9, 10 and 11
3. Get $\phi_i^{'}$, $i = 1, 2, ...M$ using Eqs. 12, 13 and 14
4. Use the difference $J = \sum_{i=1}^{M} (\phi_i - \phi_i^{'})^2$ to correct ϕ_i, $i = 1, 2, ...M$
5. Stop if the difference is satisfactorily small, else goto step 2.

So the problem of solving the lattice model can be cast as a constrained optimization problem (the constraint is $0.0 < \phi_i < 1.0$)

Solution of the Lattice Model by GAs

We solve the lattice model by solving the optimization problem stated earlier. Observing results in the existing literature (Scheutjens, 1979) we

note that the profiles of P_i and ϕ_i are not varying everywhere so it will be easier if we approximate the solution by the following polynomial form to reduce dimensionality, i.e.

$$\phi_i = \sum_{k=1}^{N} a_k i^{k-1}, \qquad i = 1, 2, ..., M \qquad and \quad N < M \tag{10}$$

and estimate a_k, $k = 1, 2, ..., N$ that minimize the objective function $J = \sum_{i=1}^{M} (\phi_i - \phi_i^{'})^2$. When $N = M$ this approximation approaches to the original problem where no such approximation was made. N can be increased or decreased depending on problem complexity. We start with a set of several subsets, each subset containing the coefficients of the polynomial approximation and proceed as outlined in the previous section. In order to enhance the convergence properties we can augment the GA by a gradient search technique either at the end or between each generation. The tuning parameters for the GA are the population size, length of the vector representations, *mutation* and *crossover* rates and the number of *children* in each generation. Results of the optimization procedure are shown in Figs. 17 and 18 for $\chi = 0.0$ and $\chi = 0.5$ respectively. Figs. 19 and 20 show the plots of ϕ_i and p_i vs. i for the last two iteration cycles of the optimization procedure indicating convergence.

Comparison with other Techniques

For purposes of comparison we solved the lattice model using a constrained Levenberg-Marquardt (LM) method using finite difference gradients. Figs. 21, 22, 23 and 24 show the results when we did not use the polynomial approximation of the solution but optimized using the whole vector $[\phi_1, \phi_2, ..., \phi_M]^T$. Figs. 25, 26, 27 and 28 show the corresponding results when we used the LM method along with polynomial approximation of the solution. Figs. 29, 30, 31 and 32 show the results of the modified GA technique (GA+LM) with the LM method implemented after 10 generations. In the last method we let the GA run for the first ten generations and from the tenth generation we picked the best candidate and did gradient search by a constrained LM technique. For all the LM trials, we observed that the solution was very sensitive to the initial guesses and convergence was not always guaranteed. For the method employing a GA, we observed that convergence was guaranteed from any random initial population but was not

sharp. The GA+LM method not only ensured convergence but gave sharp convergence in relatively less number of total function evaluations. It should be noted that in this study, in order to develop a general solution strategy, Roe's approximation (Scheutjens, 1979) was not used as an initial guess for optimization. Instead, the initial solution was randomly picked in all cases. Contrary to the experience of Scheutjens (1979) constrained nonlinear optimization works and might work even better when Roe's approximation is used as a starting point. Polynomial approximation pays when we have extended and complex cases of the same problem. Our experience in this case shows that GA+LM is the safest approach for guaranteed convergence.

Since GAs are blind to the underlying nonlinearities and the analytic properties of the objective function they are suitable for solving complex lattice models. GAs can be effectively combined with traditional constrained gradient techniques to ensure guaranteed and sharp convergence in the solution with very little or no a priori knowledge of the physics of the system.

Table. 4. Parameters of GAs employed in the three case studies[4].

Case Study 1 (Best controller synthesis for a nonlinear plant)	population size	=	10
	mutation probability	=	0.0081
	crowding factor	=	0.3
	generation gap	=	0.4
	crossover rate	=	1
Case Study 2 (Optimization of back mix reactors in series)	population size	=	10
	mutation probability	=	0.0030
	crowding factor	=	0.5
	generation gap	=	0.4
	crossover rate	=	1
Case Study 3 (Solution of lattice model)	population size	=	10
	mutation probability	=	0.0080
	crowding factor	=	0.5
	generation gap	=	0.6
	crossover rate	=	1

[4] A generation gap of 0.4 implies that the number of offsprings in each generation equals $4(= 10 \times 0.4)$.

Conclusions

We demonstrated the ability of GAs to solve complex optimization problems in chemical engineering applications. The performance of GAs was compared with the traditional techniques where available[5]. This study shows that GAs can handle abstract and nonlinear objective functions efficiently. With a good front-end software (made available by the authors) the application of GAs to any optimization problem is very easy and requires minimal a priori knowledge about the physics of the problem or the mathematical theory behind the optimization technique.

References

[1] Chien, I. L., and B. A. Ogunnaike, "Modeling and Control of High-Purity Distillation Columns", *AIChE* annual meeting Miami, FL (1992)

[2] Crooks, W. M., "Denbig's 2-tank CSTR system", *British Chemical Engng.*, **11**, 7, 710-712 (1966)

[3] Davis, L, *Handbook of Genetic Algorithms*, Van Nostrand, New York (1991).

[4] Edgar, T. F., and D. N. Himmelblau, *Optimization of Chemical Processes*, McGraw-Hill (1988)

[5] Goldberg, D. E., *Genetic Algorithms in Search Optimization and Machine Learning*, Addison Wesley, Reading, MA (1989).

[6] Hansen, J. M., C. Lim, and J. Hong, "Optimization of Autocatalytic Reactions", *Chemical Engineering Science*, **48**, 13, 2375-2390 (1993)

[7] Kahaner, D., C. Moler, and S. Nash, *Numerical Methods and Software*, Prentice Hall (1988)

[5]It should be emphasized that the CPU times in case studies 2, 3 and 4 were in the order of seconds on a Sparc station 2. Comparison of CPU times between GAs and other techniques will only be fair if the former were implemented on a parallel machine.

[8] Luss, D.,"Optimum Volume Ratios for Minimum Residence Time in CSTR sequences", *Chemical Engng Sci.*, **20**, 171 (1965)

[9] Scheutjens, J. M. H. M., and G. J. Fleer, "Statistical Theory of the Adsorption of Interacting Chain Molecules. 1. Partion Function, Segment Density Distribution, and Adsorption Isotherms", *The Journal of Physical Chemistry*, **83**, 12, (1979).

[10] Szépe, S., and O. Levenspiel, "Optimization of Backmix Reactors in Series for a Single Reaction", *Ind. Engng Chem. Proc. Des. and Dev.*, **3**, 3, 214-217 (1964)

[11] Wood, R. K., and W. F. Stevens, "Optimum Volume Ratios for Minimum Residence Time in CSTR sequences", *Chemical Engng Sci.*, **19**, 426 (1964)

[12] Zhou, J. L., and A. L. Tits, "User's Guide for FSQP Version 3.3b: A FORTRAN Code for Solving Constrained Nonlinear (Minimax) Optimization Problems, Generating Iterates Satisfying All Inequality and Linear Constraints" (1989)