Proceedings of the 1991 Winter Simulation Conference
Barry L. Nelson, W. David Kelton, Gordon M. Clark (eds.)

COMPARISON OF GLOBAL SEARCH METHODS FOR DESIGN OPTIMIZATION USING SIMULATION

B. Stuckman
Dept. of Electrical Engineering
University of Louisville
Louisville, Kentucky

G. Evans
Dept. of Industrial Engineering
University of Louisville
Louisville, Kentucky

M. Mollaghasemi
Dept. of Industrial Engineering
University of Central Florida
Orlando, Florida

ABSTRACT

Simulation is a commonly used design tool in engineering. Once a prototype design is modeled and simulated however, many designers resort to either trial and error, univariate perturbation or local optimization methods to find a design which yields the optimum simulated results. The performance of the simulation as a function of the design parameters is complicated (that is why a simulation is used). There is generally no guarantee that this function is convex. A global optimization algorithm -- one which looks beyond any local optima is necessary.

This paper presents a methodology for the application of global search methods to optimizing the results of a computer simulation. Specific global optimization methods including simulated annealing, genetic algorithms, and Bayesian techniques will be discussed in terms of their strengths and weaknesses as applied to this methodology. In particular, the effects of simulation time, constraints, dimensionality, and computational complexity will be discussed as they relate to the choice of algorithms.

1 INTRODUCTION

Many engineers rely upon simulation to guide the process of system design. The system is analyzed and modeled, and a computer program is written which (one would hope) accurately describes the performance of the system as a function of one or more free parameters of the design. Frequently, the first step in the design process is to simulate the system performance at nominal values of the design parameters derived from some combination of approximation, theory, past-knowledge and pure guesswork.

If the design performance is sufficiently quantitative such that this performance can be represented by some real-valued function of the results of the simulation, then the design process reduces to the problem of refining the values of the design parameters to yield the best simulated performance of the system. Unfortunately, once an initial design has taken place, many designers will resort to a trial and error approach of varying the design parameters in hopes of finding the set which will yield the best performance.

A second group of designers will go a step further and systematically perturb the parameters one-by-one, determining the effect of each perturbation and attempting new solutions by varying the parameters according to the information gained. In effect, this second group is making a rudimentary attempt at implementing a local optimization algorithm. This method has the advantage that human intuition is being factored into the optimization problem. However, human patience will usually limit the duration of this optimization and thus the accuracy of the solution. Further, this *ad hoc* method of optimization does not incorporate the wisdom of the vast multitude of existing optimization algorithms.

This leads to a third group of designers who employ an direct search method of optimization to the problem. The search method makes a guess at the values of the design parameters. The system is simulated and the performance of the design is determined. This performance is fed to the search method which uses this information to derive a new guess. The procedure continues iteratively until the method stops, the designer decides to stop, or the designer runs out of resources.

The methodology employed by this third group of designers is preferable to the methods of the other groups in all but the most trivial of designs. The advantages are:

1) A systematic approach is taken to the optimization of the design parameters;

2) Since a computer algorithm will not be bored and quit, it can continue to run the simulation as long as there are resources to pay for the search;

3) A computer algorithm generally runs more efficiently in terms of both time and resources.

This third method has been discussed by many authors including Meketon (1987), and Safizadeh (1990). However, these discussions have been limited to the use of *local* search algorithms which tacitly assume the function to be optimized is convex. The performance of a design (as given by a simulation) is a fairly complicated function of the design parameters -- if not, why would a simulation be needed in the first place. Therefore, one would assume that the performance function would not be necessarily convex over the design parameter space. This means that a *global* search algorithm is needed to find the optimum solution.

Stuckman (1988a), (1990) has discussed the use of a particular Bayesian global search algorithm for optimizing a design via simulation. Specific design applications using Bayesian global search methods include control systems; Stuckman and Laursen (1989), Stuckman and Stuckman (1989), Stuckman, Stuckman & Lilly (1989), a communication system; Stuckman (1988b), a vibrometer, LSI integrated circuit, and nonstationary queuing systems; Mockus (1989). Simulated annealing global search algorithms have also been applied to design optimization problems such as computer design; Rose (1986), Leong (1986), Vecchi and Kirkpatrick (1983) and circuit design; Siarry and Dreyfus (1983).

This paper will compare three classes of global search algorithms in the context of on this problem of design optimization; specifically, simulated annealing algorithms, genetic algorithms and Bayesian/sampling algorithms. Section II will discuss the overall methodology of optimizing a design via simulation. Section III will present the three different classes of algorithms and discuss the important features and properties. Section IV will discuss these properties from the context of the design problem and give conditions where one class of algorithm will be preferred over the other classes. Section V will provide a summary of results and conclusions.

2 DESIGN BY SIMULATION

Assume a system exists which is represented by some linear or non-linear model. This design is dependent on a set of n bounded design parameters $\mathbf{X} = (x_1, x_2, \ldots, x_n)$ where $\mathbf{X} \in \mathbf{S}$. Some particular set of values for the design parameters, $\mathbf{X}_i$, represents a possible design. Further, assume that the design performance is sufficiently quantitative such that this performance can be derived from the results of the simulation. Then, the performance of a given design can be represented as a function $f(\mathbf{X})$ where the design $\mathbf{X}_i$, is preferred to a second design $\mathbf{X}_j$, if and only if $f(\mathbf{X}_i) > f(\mathbf{X}_j)$.

The problem of optimizing the performance of the system then becomes the problem of determining the values of design variables $\mathbf{X}^o$, that maximize the performance function. By definition, $f(\mathbf{X}^o) \geq f(\mathbf{X})$, for all $\mathbf{X} \in \mathbf{S}$. Due to the overall generality afforded to this function, a global search technique is needed to consider the entire solution space rather than a local technique which might not find the optimal solution. The overall methodology can thus be described as follows:

1) Define the performance of a design $\mathbf{X}_i$, in terms of some function $f(\mathbf{X}_i)$ of the results of the simulation.

2) Utilize a global search to find the optimal set of values of the design parameters as follows: The global search algorithm will pick trial design values for $\mathbf{X}_i$ to be input to the simulation. The results of the trial design will be found by simulation and the performance of the trial design, $f(\mathbf{X}_i)$, will be determined. This performance will be fed back to the search which will pick the next trial design $\mathbf{X}_{i+1}$, (see Figure 1). The search will continue iteratively until the optimal values are found within some numerical accuracy or the resources allocated to this procedure are expended.

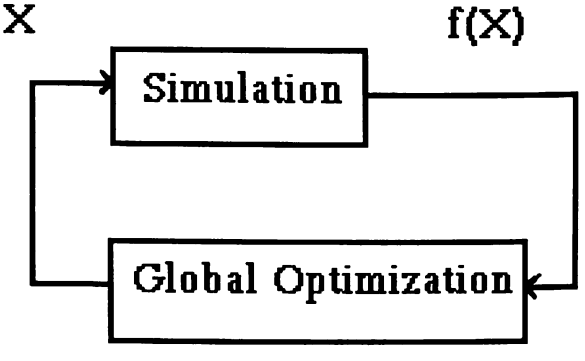

Figure 1: Global Search Applied to Simulation

There are no restrictions on the performance

function, other than the fact that it must be real-valued and that $f(\mathbf{X}_i) > f(\mathbf{X}_j)$ implies that $\mathbf{X}_i \succ \mathbf{X}_j$ (i.e. $\mathbf{X}_i$ is preferred to $\mathbf{X}_j$). Therefore, a designer is free to choose any relationship that is meaningful to the application with little regard to complexity (assuming the complexity is small with respect to the complexity of the simulation itself) Thus, the performance of a data communications system might be determined by the number bits correctly received and the time of broadcast for a simulated message. If one were interested in designing the controller for a transmission system, the performance could be a function of acceleration, slip, and maximum torque. The performance of the design of a fire control system for a tank might be measured by the number of "kills" in a simulated battle. In short, any attributes which are important to the design can be built into the performance as long as they can be measured in some quantitative way and as long as two competing designs can be compared and distinguished.

3 METHODS OF GLOBAL OPTIMIZATION

3.1 Bayesian/Sampling Algorithms

Most Bayesian/sampling algorithms are variations on a method developed by H. J. Kushner (1963). Kushner's one-dimensional method was based on the assumption that an unknown function could be modeled heuristically as a sample function of a Gaussian random process, specifically a Wiener or Brownian motion stochastic process. Kushner was able to show the following properties of the model:

a) the expected value of the unknown function f(x) conditioned on all of the measurements taken is a piecewise linear approximation of f(x) itself; and

b) the conditional variance of the approximation is quadratic between observation points.

Kushner's search strategy can be described at each iteration; the next guess is chosen to be the point in the search domain which maximizes the probability that the function exceeds the largest value by some positive constant conditioned upon all past evaluation of the function, $\text{Prob}[\, f(X) \geq f_n^{max} + \epsilon_n \mid X_1, X_2, \ldots, X_n]$. Kushner's method takes guesses in areas where the mean is high, i.e. where the maximum is likely to occur. Yet, the probabilistic search strategy dictates guesses at regions of the search space which are relatively unexplored once local solutions become "overpopulated".

Kushner's method is strictly a one-dimensional method, however, there are many n-dimensional Bayesian methods which are either extensions of Kushner's original method or similar variations (see Easom (1990)). The common factors are that the unknown function is modeled as a stochastic process, and that the next guess is derived based upon a probabilistic search strategy conditioned upon all past evaluations of the function.

Proposition 1 -- Bayesian Algorithms must have a computational complexity of at least $O(i^2)$, where i is the number of iterations.

Discussion: a true Bayesian algorithm chooses its next guess based upon the information achieved from all previous guesses. In a worst-case situation, it must review all previous guesses at each iteration. Thus, the computation necessary at the ith iteration is proportional to $(i - 1)$. Therefore the total computation for i iterations must be:

$$c = \alpha(i\text{-}1)i + g(i) \tag{1}$$

or

$$c = \alpha(i^2 - i) + g(i) \tag{2}$$

where c is the computation time and $g(i)$ is any additional computation time needed. This implies that the computational complexity is at least $O(i^2)$.

Bayesian methods require a bounded search space. Therefore the search space is defined as

$$\mathbf{S} \triangleq \text{all } \mathbf{X}_i \text{ satisfying } b_{j1} \leq x_{ij} \leq b_{j2} \tag{3}$$

where x_{ij} is the jth component of $\mathbf{X}_i$, j=1,2, ... n and where b_{j1} and b_{j2} are, respectively, the lower and upper bounds on the jth component of any X_i. There is some variation on how particular algorithms are initialized. The most successful methods (in terms of rate of convergence on sets of standard test functions) require 2^n initial points in an n-dimensional space. This is not a general requirement of Bayesian methods and some methods rely upon a fixed set of random starting points. Notice however, the methods with 2^n initialization have the advantage that they are completely deterministic. Therefore, for a given function they will always chose the same progression of search points.

3.2 Simulated Annealing Algorithms

There are many different simulated annealing algorithms. Collins (1990) gives annotated bibliography of many of these methods and applications. The general method can be explained in two parts.

Step 1) A random starting point is chosen by the user. Next, a new point is selected randomly from the neighborhood of the previous point. This neighborhood may be described by a uniform distribution of fixed circular or square radius or by virtually any other distribution centered around the previous point. If the new point yields a function evaluation, f(X), which is less than the current minimum function value (in the case of minimization, greater in terms of maximization), f_{min}, initially picked to be $+\infty$, the process is repeated, otherwise step 2 is performed.

Step 2) If the new function evaluation, f(X), is greater than the current function minimum, f_{min}, then an acceptance expression is used as the decision function to determine if the new point will be accepted. The acceptance expression is the failsafe which allows for the algorithm to exit the region of a local solution by deliberately selecting a point which is not an improvement over the previous point. This acceptance expression can vary as the search progresses and may be given as a Boltzmann distribution or a Cauchy distribution.

Proposition 2 -- Simulated annealing algorithms can have a computational complexity of $O(i)$, where i is the number of iterations.

Discussion: A simulated annealing algorithm choses its next point randomly based upon the results of only the last iteration. Even in a worst-case situation, the algorithm will require a constant amount of computation to derive the next guess. Therefore the total computation for i iterations must be:

$$c \leq \alpha i \tag{4}$$

where c is the computation time and α is some arbitrary constant. This implies that the computational complexity is $O(i)$.

Simulated annealing algorithms do not require bounded variables, thus, they attempt to searching a space from $\pm\infty$ in each dimension. As indicated above, the search is initialized by a single point, chosen either randomly or by the user. These algorithms are, by nature, nondeterministic since the progression of the search is dictated by a series of random numbers.

3.3 Genetic Algorithms

Genetic algorithms are heuristic search methods which attempt to mimic the process of natural selection(see Goldberg (1989). These methods start with a fixed initial population of randomly selected points and chose new points in an evolutionary process in attempts at finding the global optimum point. This class of algorithms is characterized by three common factors.

1) Selection -- The current points in the space are ranked in terms of their fitness by their respective function values. A probability is assigned to each point which is proportional to its fitness and parents (a mating pair) are randomly selected.

2) Crossover -- The new point or offspring is chosen based upon some combination of the "genetics of the two parents. In terms of a distance criteria, the offspring is located somewhere between the parents in the search space.

3) Mutation -- The location of the offspring is also susceptible to mutation, a process which occurs with some probability, p, where a random offset is given offspring location.

A general genetic algorithm can be described by the following steps.

Step 1) The search is initialized by selecting a random population of k "parents".

Step 2) These points are evaluated.

Step 3) Perform selection, crossover, and mutation on a new offspring and insert it into the population. Delete the worst member of the popolation to maintain a fixed population size.

Step 4) Repeat until some stopping criteria is reached.

A common modification to the general algorithm is the generational genetic algorithm which generates k new offspring at once and "kills off" all of the parents. This modification allows the algorithm to search the same points multiple times; a feature which allows the method to perform well on nondeterministic functions (problems where function

evaluations are perturbed by noise).

Proposition 3 -- Genetic Algorithms can have a computational complexity of $O(i)$, where i is the number of iterations.

Discussion: Mating pairs are chosen for a fixed population size. In a worst-case situation, the algorithm must review a constant number of previous guesses at each iteration. Thus, the computation necessary at the ith iteration is constant . The total computation for i iterations must be:

$$c \leq \alpha i \tag{5}$$

where c is the computation time and α is some arbitrary constant. This implies that the computational complexity is $O(i)$.

Since these algorithms allow mutation of offspring and a random selection of mating pairs, they are nondeterministic. Typically, these algorithms choose the initial points from some bounded region similar to the Bayesian methods. However, they have the capability of operating on an unbounded space.

3.4 Comparison of Algorithm Properties

Table 1 presents a comparison of various properties of Bayesian, simulated annealing and genetic global search algorithms. Notice that simulated annealing and genetic algorithms, while much different conceptually, have generally the same properties. Thus a comparison of these three methods, to some extent becomes a comparison of Bayesian methods on one hand and simulated annealing and genetic algorithms on the other.

Table 1: Comparison of Global Search Algorithms

Algorithm	Deterministic	Complex.	Init.	Var.
Bayesian	yes	$O(n^2)$	2^n	bounded
Genetic	no	O(n)	rand.	unbounded
Sim. Anneal.	no	O(n)	1	unbounded

A complete comparison of these methods on the basis of rate of convergence has not been performed. However, several Bayesian methods have been compared with a simulated annealing algorithm on the basis of a large set of standard test functions by Easom[15]. Results show that the Bayesian methods converge more quickly. These results are not unexpected. Bayesian methods model the unknown function by considering all past observations and thus are more "intelligent" in the selection of their next guess. This intelligence comes at the expense of increased computational complexity. One would expect that the same would be true for a comparison of Bayesian vs. genetic algorithms -- that Bayesian methods would converge more quickly. This fact will be assumed for the basis of this work. Thus, If rate of convergence were the only criteria, one would always select a Bayesian algorithm. However, there are many other important, and often, overriding properties to be considered. The next section will discuss these properties and their relationship to the classes of global search algorithms.

4 COMPARISON OF METHODS ON THE DESIGN PROBLEM

The suitability of the algorithms discussed in the previous section for the optimization of a design by simulation is dependent, largely upon the properties of the simulation to be performed. This section will discuss the various properties of a simulation which will have an impact upon the choice of algorithm. The specific properties to be discussed: dimensionality, simulation length, deterministicity, and constraints, are presented in decreasing order of importance.

4.1 Dimensionality

Currently the initialization of the most effective Bayesian optimization methods require 2^n evaluations of the function in an n - dimensional design. For this reason, these methods have been, thus far, limited to applications with ten or fewer dimensions. Notice that a design of a fifty-dimensional system would require an evaluation of approximately 10^{15} simulations just to initialize the search. Conversely, simulated annealing and genetic algorithms require a fixed number of initial points, independent of the dimensionality of the problem. Research is currently being performed by several groups to attempt to modify existing Bayesian methods for initialization with a fixed number of points. However, at this juncture, simulated annealing and genetic algorithms are the methods of choice for problems of high dimensionality -- systems with more than 10 design

variables.

4.2 Simulation Length

A fundamental premise of Bayesian optimization methods is that, in the process of finding a global solution, one wishes to minimize the number of evaluations of the function at the expense of increased computation time. This may not always be the case. A more exact formulation of this premise is that a designer wishes to minimize his or her computational resources, R, defined as

$$R = A(n) + nL \tag{6}$$

where $A(n)$ is the amount of computation time used by the search algorithm for n iterations and L is the amount of computation time for a single simulation. If L is small for a given system, the resources could be minimized by a minimizing $A(n)$. This implies that it might be more effective to take more iterations of a less complex algorithm than converge in the fewest iterations. Therefore, given the computational complexities presented in the previous section, simulated annealing and genetic algorithms would be the methods of choice if the simulation time L where so small such that the total computation resources would be dominated by a Bayesian algorithm if it were used. Alternately, If the L is large, one would be willing to use substantial computational resources to minimize the number of iterations. This implies that a Bayesian method would be preferred in this instance.

4.3 Deterministicity

If a simulation is deterministic, it will always yield the same performance for fixed values of the design parameters. A nondeterministic simulation can possibly yield different values for different runs with the same design parameter values. The results of a nondeterministic simulation violate a fundamental assumption of most Bayesian algorithms -- that the exact performance of a given guess is *known*. Alternatively, simulated annealing and genetic algorithms make no such assumption -- in fact, they are both capable of evaluating a function at the same location more than once.

The practical effects of using nondeterministic data in a Bayesian algorithm have not been well studied. It is possible that the effects would be slight since the stochastic model is merely an approximation of the function to be optimized. However, the optimum value could not be determined to a greater accuracy than the variation in the performance function due to the nondeterministic nature of the simulation. One solution would be to run the simulation k times in order to obtain performance value with less variation. The effect of this increased computation should be considered when weighing the methods in terms of simulation length.

4.4 Constraints

Bayesian algorithms require that the design variables be bounded as in Expression (3). Simulated annealing and genetic algorithms are free to operate on a bounded or unbounded search space. However, there is rarely the case where a design parameter is allowed to go to infinity, since infinity, while possible in theory, is generally unattainable in practice. If variables are allowed to approach infinity, the global search problem becomes ill-posed. Convergence would not be possible since there would always be an infinite portion of the search space which was not explored.

Bounding the search space is desirable regardless of the algorithm used. This can be accomplished based upon the transformation of variables to bounded quantities, bounding the variables based upon some practical limitation on their values due to implementational constraints, or bounding the variables by setting an "approximation" for infinity -- recall, a digital computer will not allow a variable to be infinity anyway.

5 SUMMARY

A methodology for the application of global search methods to optimizing the results of a computer simulation has been presented. Specific global optimization methods including simulated annealing, genetic algorithms, and Bayesian techniques have been discussed in terms of their strengths and weaknesses as applied to this methodology. In particular, the effects of simulation time, constraints, dimensionality, and computational complexity were considered as they relate to the choice of algorithms.

Simulated annealing and genetic algorithms perform similarly yet differ in many ways from the class of Bayesian algorithms. Bayesian algorithms spend additional computation time in modeling all past values of the unknown function in an effort to minimize the number of evaluations of the function. These methods would be the algorithms of choice for determining the optimal design via simulation given

the number of design variables is less than 10 and the time required to run a single simulation is large compared with the time it takes the algorithm to determine the next point. Using a Bayesian algorithm requires the design variables be bounded in some way, and that a nondeterministic simulation be run many times at each trial set of design parameter values to yield an accurate approximation of the global solution.

REFERENCES

N.E. Collins, R.W. Eglese, B.L. Golden. 1990. Simulated Annealing -- An Annotated Bibliography, *Am Journal of Math. & Management Sciences,* **8,** Nos. 3 & 4, 209-307.

E.E. Easom. 1990. A survey of global optimization techniques, M.Eng. Thesis, University of Louisville.

D. Goldberg. 1989. *Genetic Algorithms in Search, Optimization & Machine Learning,* Addison-Wesley.

H.J. Kushner. 1963. A new method of locating the maximum of an arbitrary multipeak curve in the presence of noise, *Proceedings of the Joint Automatic Control Conference.*

H.W. Leong. 1986. Routing Problems in the Physical Design of Integrated Circuits, Report UIUCDCS-R-86-1242, University of Illinois at Urbana-Champaign.

M.S. Meketon. 1987. Optimization in Simulation: A Survey of Recent Results, *Proceedings of the 1987 Winter Simulation Conference,* 58-67.

J. Mockus. 1989. *Bayesian Approach to Global Optimization,* Kluwer Academic Publishers.

J.S. Rose, D.R. Blythe, W.M. Snelgrove and Z.G. Vranesic. 1986. Fast, High Quality VLSI Placement on a MIMD Mulitprocessor, *Proceedings of the International Conference on CAD,* November, 42-45.

M.H. Safizadeh. 1990. Optimization in Simulation: Current Issues and the Future Outlook, *Naval Research Logistics,* **37**, 807-825.

P. Siarry & G. Dreyfus. 1983. An Application of Physical Methods to the Computer-aided Design of Electronic Circuits, Ph.D. Thesis, University de Paris.

B.E. Stuckman. 1988a. Design of General Systems by Methods of Global Search: a Computer-Aided Engineering Approach, *Proceedings of the 1988 AMSE International Istanbul Conference on Modelling and Simulation,* **1B**, 81 - 90.

B.E. Stuckman & P.L. Stuckman. 1990. Design Optimization Using Simulation and Stochastic Global Search: a Computer-Aided Engineering Approach, *Advances in Modelling and Simulation,* **18,** No. 4, 13-33.

B.E. Stuckman & N.W. Laursen. 1989. A Method of Control System Design Using Global Search, *IEEE Transactions on Automatic Controls,* **34,** No. 1, 111-115.

P.L. Stuckman & B.E. Stuckman. 1989. Iterative Global Design of PID Controllers for Systems without Models, *Proceedings of the 17th IASTED International Symposium -- Simulation & Modelling,* June, pp. 320 - 321.

B.E. Stuckman, P.L. Stuckman & J. Lilly. 1988. Finding the 'Best' Optimal Control using Multivariate Stochastic Global Search, *Proceedings of the 31st Midwest Symposium on Circuits and Systems,* August, 467 - 470.

B.E. Stuckman. 1988b. Computer-Aided Design Optimization of a Communication System Using Stochastic Global Search, *Proceedings of the 31st Midwest Symposium on Circuits and Systems,* August, 531 - 534.

M.P. Vecchi & S. Kirkpatrick. 1983. Global Wiring by Simulated Annealing, *IEEE Trans. on CAD of IC Circuits,* **2,** No. 4, 215-222.

AUTHOR BIOGRAPHIES

BRUCE E. STUCKMAN is an Assistant Professor of Electrical Engineering at the University of Louisville and Director of Electrical and Systems Engineering at the OASYS Research Corp. His research interests include numerical optimization, simulation and modeling, instrumentation and measurement error, telecommunication and communication systems, and technology education and policy. He is a member of Tau Beta Pi, Eta Kappa Nu, Sigma Xi, ASNE, a Senior member of IEEE and an editor of the Journal of Law and Education.

GERALD W. EVANS is an associate professor in the Department of Industrial Engineering at the University of Louisville. His research interests include multicriteria optimization and simulation modeling, especially applied to problems in manufacturing systems design and operation, engineering management, and the service industries. He is an active member of TIMS, ORSA, IIE, and DSI.

MANSOOREH MOLLAGHASEMI is an Assistant Professor in the Department of Industrial Engineering at the University of Central Florida. She recieved her Ph.D. in industrial engineering from the University of Louisville in 1991. Her research interests include the application of optimization and simulation modeling to manufacturing system design. She is a member of ORSA and IIE.