Simulation optimization: A comprehensive review on theory and applications

EYLEM TEKIN & IHSAN SABUNCUOGLU

To cite this article: EYLEM TEKIN & IHSAN SABUNCUOGLU (2004) Simulation optimization: A comprehensive review on theory and applications, IIE Transactions, 36:11, 1067-1081, DOI: 10.1080/07408170490500654

To link to this article: http://dx.doi.org/10.1080/07408170490500654

Published online: 17 Aug 2010.

Submit your article to this journal

Article views: 906

View related articles

Citing articles: 102 View citing articles
Simulation optimization: A comprehensive review on theory and applications

EYLEM TEKIN¹ and IHSAN SABUNCUOGLU²

¹Department of Statistics and Operations Research, University of North Carolina—Chapel Hill, Chapel Hill, NC 27599-3180, USA
E-mail: eylem@email.unc.edu
²Department of Industrial Engineering, Bilkent University, 06533, Ankara, Turkey
E-mail: sabun@bilkent.edu.tr

Received January 2001 and accepted March 2004

For several decades, simulation has been used as a descriptive tool by the operations research community in the modeling and analysis of a wide variety of complex real systems. With recent developments in simulation optimization and advances in computing technology, it now becomes feasible to use simulation as a prescriptive tool in decision support systems. In this paper, we present a comprehensive survey on techniques for simulation optimization with emphasis given on recent developments. We classify the existing techniques according to problem characteristics such as shape of the response surface (global as compared to local optimization), objective functions (single or multiple objectives) and parameter spaces (discrete or continuous parameters). We discuss the major advantages and possible drawbacks of the different techniques. A comprehensive bibliography and future research directions are also provided in the paper.

1. Introduction

With the continuing developments in computer technology, simulation is receiving increasing attention as a decision-making tool. Most real-world systems are so complex that computing values of performance measures and finding optimal decision variables analytically is very hard and sometimes impossible. Therefore, computer simulation is frequently used in evaluating complex systems and optimizing responses.

The problem under consideration is the maximization or minimization of the expected value of the objective function with respect to its constraint set as given below:

$$\max \min_{\mathbf{X} \in \Theta} H(\mathbf{X})$$

where $H(\mathbf{X}) = E[L(\mathbf{X}, \epsilon)]$ is the performance measure of the problem. The quantity $L(\mathbf{X}, \epsilon)$ will be called the sample performance, $\epsilon$ represents the stochastic effects in the system, $\mathbf{X}$ is a $p$-vector of controllable factors and $\Theta$ is the constraint set on $\mathbf{X}$. If $H(\mathbf{X})$ is a one-dimensional vector, the problem is single objective optimization, whereas if its dimension is more than one, the problem becomes multiobjective. The optimum is denoted by $\mathbf{X}^*$. Without loss of generality, we will consider the minimization problem throughout the paper.

A variety of techniques and approaches have been proposed to solve the above optimization problem. In the literature, there are also several survey papers that discuss foundations, theoretical developments and applications of these techniques (Meketon, 1987; Jacobson and Schruben, 1989; Safizadeh, 1990; Azadivar, 1992; Fu, 1994a; Andradóttir, 1998; Swisher et al., 2000). The simulation optimization techniques covered in these papers are listed in Table 1. The objective of this paper is to provide a more comprehensive coverage of these techniques and their applications. Besides reviewing the “traditional” methods, we give emphasis to recent developments in this area (i.e., global optimization methods such as evolutionary algorithms, simulated annealing, and tabu search).

We classify the existing studies under two main headings: (i) local optimization; and (ii) global optimization. Local optimization techniques are further classified in terms of discrete and continuous parameter spaces. Figure 1 shows such a classification scheme. In this paper, in addition to reviewing the theoretical aspects of these techniques, we also discuss their applications to various production systems.

The rest of the paper is organized as follows. In Section 2.1, we focus on techniques for the discrete-parameter case such as ranking and selection, multiple comparison procedures, random search, Nelder-Mead simplex/complex search, Hooke-Jeeves pattern search and the single-factor method. For the continuous-parameter space case, we
Table 1. Techniques discussed in various survey papers

<table>
<thead>
<tr>
<th>Publication</th>
<th>Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meketon (1987)</td>
<td>StApp, PA, LRE, adaptive control, nonlinear programming</td>
</tr>
<tr>
<td>Jacobson and Schruben (1989)</td>
<td>FDA, StApp, PA, RSM, LRE, H-J, simplex method, random search, integral methods</td>
</tr>
<tr>
<td>Safizadeh (1990)</td>
<td>RSM, FDA, PA</td>
</tr>
<tr>
<td>Azadivar (1992, 1999)</td>
<td>RSM, FDA, StApp, LRE, IPA, FDE, complex search, SA</td>
</tr>
<tr>
<td>Fu (1994a)</td>
<td>MCP, R&amp;S, FDE, PA, FDA, LRE, StApp, RSM</td>
</tr>
<tr>
<td>Andradottir (1998)</td>
<td>MCP, R&amp;S, FDE, PA, FDA, LRE, StApp, sample path optimization</td>
</tr>
<tr>
<td>Swisher et al. (2000)</td>
<td>PA, StApp, N-M, H-J, MCP, R&amp;S, OO, SA, GA</td>
</tr>
</tbody>
</table>

consider response surface methodology, gradient-based methods (finite difference estimates, perturbation analysis, frequency-domain analysis, likelihood ratio estimators) and stochastic approximation methods in Section 2.2. In Section 3, we discuss global search methods such as evolutionary algorithms, simulated annealing, tabu search, Bayesian/sampling algorithms, and the gradient surface method. In Section 4, we review the studies for multiple objective problems. The paper ends with concluding remarks and suggestions for further research in Section 5.

2. Local optimization

Local optimization problems are discussed in terms of discrete and continuous decision spaces. In a discrete space, decision variables take a discrete set of values such as the number of machines in the system, alternative locations of depots, different scheduling rules or policies, etc. On the other hand, in a continuous space, the feasible region consists of real-valued decision variables such as order quantity and reorder quantity in inventory problems, release time of factory orders, etc.

2.1. Discrete decision space

The discrete case can further be classified into: finite parameter space and infinite parameter space. In the finite case, \( X \in \{\lambda_1, \ldots, \lambda_k\} \) where \( \lambda_i \) is one of the \( k \) points in the feasible region and the aim is to find \( X^* = \lambda_i \). The two most popular methodologies for the class of problems are: (i) ranking and selection; and (ii) multiple comparison procedures. For excellent reviews of these two classes of techniques, one can refer to Bechhofer et al. (1995) and Goldsman and Nelson (1998). Other methods (e.g., random search, Nelder-Mead simplex/complex search, single-factor method, Hooke-Jeeves pattern search) can operate in the infinite parameter space.

There are two main approaches with respect to ranking and selection. The first is the indifference-zone approach
This method guarantees that the performance measure value of the selected \( \lambda_i \) differs from the optimal solution value by at most a small amount \( \delta \), with a probability of at least \( P^* \). The second approach is called subset selection. It aims to select a subset of at most \( m \leq k \lambda_i \)'s and guarantees that this subset contains at least the best \( \lambda_i \), with probability of at least \( P^* \). The latter approach is more useful when the number of choices is quite large. Sullivan and Wilson (1989) propose two extensions to the subset selection procedure for normal populations with unknown moments. Butler et al. (2001) combine multivariate utility theory with the indifference-zone approach to make comparisons of systems that have multiple performance measures.

Nelson et al. (2001) present a general theory and procedures for reducing sampling efforts in a two-stage indifference-zone approach when the number of alternatives is large. Boesel (2000) and Boesel et al. (2003) study the problem of finding the best system when the number of systems is large and initial samples from each system have already been taken. These articles develop statistical procedures that identify the best system encountered in the search by using a variety of approaches including subset selection and indifference-zone. Kim and Nelson (2001, 2004) and Goldsman et al. (2002) present efficient fully-sequential indifference-zone techniques that eliminate systems deemed inferior as sampling progresses.

A disadvantage of some ranking and selection procedures is the requirement of independence between competing designs; but see the subsequent discussion. The assumption of normality is handled by the appropriate batching techniques (Lund et al., 1992).

In multiple comparison procedures, the idea is to run a number of replications and make conclusions on a performance measure by constructing confidence intervals. In general, for all \( \lambda_i \) and \( \lambda_j \) pairs, the differences between the estimates of the performance measures are computed and \((1 - \alpha)100\%\) confidence intervals are formed for each interval. We select the system corresponding to the confidence interval for which the differences with all other pairs are strictly negative (Hochberg and Tamhane, 1987).

Fu (1994a) and Hsu (1996) summarize the main ideas for three different multiple comparison procedures (i.e., all pairwise multiple comparisons, multiple comparisons with the best, and multiple comparisons with a control) and discuss the differences between these three approaches. Yang and Nelson (1991) propose refinements to the classical multiple comparison procedures by using variance reduction techniques such as common random numbers and control variates. Matejcik and Nelson (1995) and Damerdji and Nakayama (1999) develop two-stage sampling procedures for multiple comparisons with the best. Nelson and Matejcik (1995) study the use of common random numbers in a combined indifference-zone selection and multiple comparison procedure. Nakayama (1997) derives simultaneous confidence intervals based on standardized time-series methods for comparing systems with unknown steady-state means and unknown asymptotic variances.

Generally, real systems have a large number of decision variables and it is often infeasible (because of time constraints and computational budget) to find the best design for discrete-event system simulations. For this reason, Ordinal Optimization (OO) concentrates on finding a subset of “good enough” designs by sampling from a large set of solutions and evaluating a smaller set of designs (Ho, Sreenivas and Vakili, 1992). Lee et al. (1999) explain the goal-softening issue in OO. Ho (1999) presents a tutorial on the fundamentals of ordinal optimization, including the subset selection and alignment probability (i.e., probability of having the selected subset in the set of “good enough” subsets). Chen (1996) quantifies the satisfaction level of a selected subset by using a confidence probability and proposes two lower bounds. Lau and Ho (1997) give subset selection rules and calculate the alignment probabilities depending on these rules. Sullivan and Jacobson (2000) apply OO within a hill climbing algorithm framework. There are also some studies that enhance the efficiency of OO by several different approaches such as Chen, Chen and Yücesan (2000), Chen, Lin, Yücesan and Chick (2000) and Shi and Chen (2000).

Random search can work on an infinite parameter space. Inputs of upper and lower bounds on each of the controllable factors define an overall search region. The technique selects points at random from the overall search region (Smith, 1971). Since the search region contains a large number of combinations of \( p \)-dimensional points, the procedure stops when a specified number of computer runs has been completed. The point that gives the best response is selected to be optimal. Very little has been done with random search in trying to solve simulation optimization problems. The major drawback is that it converges slowly to an optimum because previous information is not used at each iteration.

Nelder-Mead simplex/complex search (N-M) (Nelder and Mead, 1965) constructs a \( p \)-dimensional simplex by choosing \( p + 1 \) extreme points for a response function of \( p \) parameters, and simulates the response at each extreme point (vertex). At each iteration, a new point is added to the simplex by projecting that point to the centroid of the remaining points of the simplex and dropping the worst point from the set. The procedure continues until the size of the simplex becomes sufficiently small (i.e., all the points in the simplex are located at nearly the same place). The complex search is the constrained version of the simplex search. This search procedure is similar to the simplex method except that a special effort is made to prevent the simplex from leaving the feasible region (Box, 1965). Azadivar and Lee (1988) develop a heuristic simplex method and compare the vertices statistically, i.e., the worst point that will be deleted is the one whose lower confidence limit is greater than the upper confidence limits of the other vertices of the simplex. Barton and Ivey (1996) propose a modified version of the
simplex method in order to avoid the risk of inappropriate termination.

Single-Factor Method (SFM) (Friedman and Savage, 1947) and Hooke-Jeeves pattern search (H-J) (Hooke and Jeeves, 1961) are also direct search methods that can work on an infinite parameter space. SFM involves coordinated movement of one factor while all other factors are held constant. On the other hand, H-J, starting from an initial base point, checks if an incremental change in a variable improves the response value and repeats it for all variables until a new setting is obtained. The procedure continues until the incremental values do not change the response value. H-J is generally used in conjunction with other methods (Chen and Tsai, 1996).

2.2. Continuous decision space

A great amount of work has been done with problems that have a continuous decision space. Below we discuss the most common methods from the literature.

2.2.1. Response surface methodology

Response Surface Methodology (RSM) is a class of procedures that: (i) fit a series of regression models to the responses of a simulation model evaluated at several points; and (ii) optimize the resulting regression function. The basic algorithm consists of two phases. In the first phase, a first-order model is fit to the response surface. Afterwards, a steepest descent direction is estimated from the model. The first phase is repeated until the computed slope becomes approximately zero, meaning that first-order design is no longer a good fit to the subregion on hand. In the second phase, a quadratic response surface is fitted using second-order experimental designs, and then the optimum is determined from this fit.

A survey of the RSM research from 1966 to 1988 is given in Myers et al. (1989). The book by Box and Draper (1986) has an extensive discussion on response surfaces and experimental designs. A study by Kleijnen (1998) discusses the use of statistical designs for what-if analysis in simulation and emphasizes how RSM combines regression analysis, statistical designs and the steepest descent (ascent) method to optimize a simulated system. Factorial and fractional factorial orthogonal designs are the best known first-order designs for RSM (Montgomery, 1991). Composite and rotatable designs are the most useful second-order designs (Montgomery and Evans, 1975). Ramberg et al. (1991) relate the orthogonal arrays advocated by Genichi Taguchi to classical experimental designs and use Taguchi’s techniques in the construction of mathematical metamodels for RSM. In order to get better fits from RSM, some researchers use it in conjunction with other methods such as gradient-based techniques, quasi-Newton methods, and simplex experimental designs (Safizadeh and Signorile, 1994; Joshi et al., 1998).

Since RSM is a widely accepted method in simulation optimization, there are a number of examples of its real-time implementations. Kleijn (1990, 1995) presents optimization of a decision support system of a Dutch company via RSM. In another study, Shang and Tadikamalla (1993) investigate a computer-integrated manufacturing system of an automated printed circuit board manufacturing plant and implement RSM to maximize output.

RSM provides a general methodology for optimization via simulation. Its biggest advantage is that it uses well-known statistical tools. Compared to many gradient methods, RSM is a relatively efficient method of simulation optimization in the number of simulations experiments needed. However, it has a drawback due to its computational requirements if applied blindly.

2.2.2. Gradient-based methods

There are four well-known methods in the simulation optimization literature that are used for estimating gradients of the response: (i) finite difference estimates; (ii) perturbation analysis; (iii) frequency domain analysis; and (iv) likelihood ratio estimates.

The Finite Difference Estimate (FDE) method is based on determining partial derivatives of \( H(X) \) by:

\[
\frac{\partial H(X)}{\partial X_i} = \frac{H(X_1, \ldots, X_i + \Delta X_i, \ldots, X_p) - H(X_1, \ldots, X_i, \ldots, X_p)}{\Delta X_i},
\]

In order to estimate the gradient at each point, at least \( p+1 \) evaluations of the simulation model are necessary. For a more reliable estimate, multiple observations for each derivative may be needed. An example of applying this method together with the H-J technique is discussed by Pegden and Gately (1977). It is also used in stochastic approximation applications as will be discussed in the next subsection.

Perturbation Analysis (PA) was introduced by Ho et al. (1979) in the context of a buffer allocation problem in serial production lines. PA, when applied properly to models that satisfy certain conditions, estimates all gradients of an objective function from a single simulation run. There are two classifications of PA: Finite Perturbation Analysis (FPA); and Infinitesimal Perturbation Analysis (IPA). FPA is designed for discrete parameters and is an heuristic that approximates the difference in a performance measure when a discrete parameter is perturbed by one unit. IPA is used to obtain derivatives of continuous parameters and estimates all partial derivatives from a single run by keeping track of related statistics of certain events during a run by computing:

\[
\frac{\partial L}{\partial X} = \sum_{i,k} \frac{\partial L}{\partial T_k} \frac{\partial T_k}{\partial T_i} \frac{\partial T_i}{\partial X}
\]
where \( \partial T_i / \partial X \) indicates how the change in the value of a system parameter, \( X \), changes the timing of events, \( \partial T_k / \partial T_i \) indicates how changes in the timing of some events \( T_i \) change the timing of other events \( T_k \), \( \partial L / \partial T_k \) indicates how the changes in the timing of some events \( T_k \) change the system performance (Ho and Cao, 1991).

Some restrictive conditions have to be satisfied for PA to be applicable. For instance, if \( \partial T_i / \partial X \) or \( \partial T_k / \partial T_i \) are not small, the accuracy of the PA estimates would be questionable. However, PA has been one of the most attractive research areas in simulation optimization because of its efficiency. Glasserman (1991) gives sufficient conditions for unbiased estimates of IPA. L’Ecuyer and Perron (1994) study the convergence rates of PA estimators. Dai (2000) investigates PA via coupling (i.e., generating multiple random samples), which is useful for variance reduction and efficient implementation.

Since PA performs well in simple discrete-event dynamic systems which can be modeled as queueing networks, there are a number of papers on the applications of PA to queueing systems which can be modeled as queueing networks, there are a number of papers on the applications of PA to queueing systems; i.e., Ho et al. (1984), Ho (1985), Ho and Hu (1990), Wardi et al. (1991), Chong and Ramadge (1993) and also Fu and Hu (1994). In addition, there is literature on the application of PA to \((s, S)\) inventory systems. Fu (1994b) considers a periodic review system with continuous demands and full backordering, and derives sample path derivatives of performance measures. Bashyam and Fu (1994) apply PA to obtain efficient derivative estimators of the expected cost per period with respect to \( s \) and \( S \), for a class of \((s, S)\) inventory systems.

PA has been widely used for optimizing manufacturing systems of interest. Donohue and Spearman (1993) determine the most profitable capacity configuration for a production line by using PA. Yan et al. (1994) use PA to develop algorithms to approximate the optimal threshold values in a manufacturing system with two tandem machines. Liberopoulos and Caramanis (1994) use IPA to obtain the first and second derivative estimates for manufacturing flow controllers of an unreliable flexible manufacturing system. Cheng (1994) considers a multistage make-to-stock system and establishes sample-path derivatives by using IPA to find an appropriate trade-off between reduced order-waiting time and increased process speeds. Brooks and Varaiya (1997) use IPA to determine asymptotically unbiased gradient estimates for computing the minimum average network delay in input ATM networks. Heidgott (1995) uses smoothed PA to optimize threshold values of repair times in a maintenance model.

Frequency-Domain Analysis (FDA) was introduced by Schruben and Cogliano (1981) for the purpose of factor screening in simulation experiments. The intuitive idea behind FDA is to oscillate the value of a parameter according to a sinusoidal function during simulation. The magnitude of the performance measure variation gives an indication of the relative sensitivity of the performance measure to the parameter. A vector of parameters is modulated as follows:

\[
X(t) = X_0 + \alpha \sin(\omega t)
\]

where \( X_0 \) is the parameter vector of interest, \( \alpha \) is the vector of oscillation amplitudes, and \( \omega \) is the vector of oscillation frequencies. Generally, the time variable \( t \) is not the simulation time but, instead, it is a variable of the model, which keeps track of certain statistics during each run. Jacobson et al. (1988, 1991) propose some ways to select \( t, \omega, \alpha \). Once these values are determined, one can obtain a quadratic response metamodel by approximating \( H \) through \( X_0 \) using a second-order Taylor series expansion:

\[
\nabla_i \hat{H}(X_0) = \lim_{T \to \infty} \lim_{\omega \to 0} \frac{2}{\alpha i T} \sum_{i=1}^{T} \hat{H}(X(t)) \sin(\omega_i t),
\]

where \( \nabla_i \) denotes the partial derivative with respect to \( X_i \).

Although FDA is seen to have a strong potential for enhancing the efficiency of simulation optimization, it has some drawbacks. First, it requires careful indexing of simulation observations together with sinusoidal variation of input variables according to a time index. Hazra et al. (1997) address the indexing problem by providing some general guidelines. Heidgott (1995) compares the performance of FDA and PA in estimating the sensitivity of the steady-state throughput of a manufacturing system with respect to various parameters (i.e., service times, buffer sizes). Furthermore, FDA has one added restriction that the region under investigation should be small in order to avoid large variances.

Likelihood Ratio Estimators (LRE) differentiate the underlying probability measure of the system. LRE assumes that the performance measure function is \( L(Y) \) where \( Y \) is a random vector with joint cumulative distribution function \( F(X, \cdot) \) and density \( f(X, \cdot) \), and dependence on \( X \) enters only through the random vector \( Y \). Thus:

\[
E[L(Y)] = \int L(y) dF(X, y).
\]

By differentiating the above equation with respect to \( X \), one can estimate the derivative of the performance measure together with the performance measure itself (Glynn, 1990; Rubinstein, 1991; Rubinstein and Shapiro, 1993). Implementation of the LRE technique to a \( GI/G/1 \) queue is given in Fu (1994a) by using both natural and regenerative estimators. Nakayama et al. (1994) discuss the application of LRE to the simulation of large Markovian models of highly dependable systems. Nakayama and Shahabuddin (1998) investigate the LRE method for estimating derivatives of finite performance measures in
generalized semi-Markov processes under some conditions on the transition probabilities.

LRE estimates the derivative of the performance measure and the performance measure itself in a single run. But the method is not applicable to structural parameters such as $s$ and $S$ in $(s, S)$ inventory systems. One other major drawback is that the variance of the estimator increases as the run length increases (Rubinstein, 1999). LRE usually finds application areas when it is used in conjunction with stochastic approximation. Andradottir (1996a) shows how to evaluate the performance measure of interest and its gradient for a general state space Markov chain by using LRE in conjunction with stochastic approximation. Fu et al. (1995) apply variants of PA and LRE to derive and compare estimates of the derivative of the steady-state mean time for a single queue system with nonidentical multiple servers. They report that neither method universally outperforms the other. Fu and Hu (1999) apply PA and LRE to derive estimators that can be used in gradient-based optimization algorithms and in sensitivity analysis when Monte Carlo simulation is used.

2.2.2. Stochastic approximation

Stochastic approximation (StApp) was first introduced by Robbins and Monro (1951) and Kiefer and Wolfowitz (1952). The basic assumption underlying StApp is that the original problem given by the formula $\min_{x} H(x)$ can be solved by $\nabla H(x) = 0$. This method uses a recursive formula:

$$X_{n+1} = \Pi_{\Theta}(X_n - a_n \hat{\nabla} H_n),$$

where $a_n$ is a series of real-valued step sizes that satisfy $\sum a_n < \infty$, $\sum a_n^2 < \infty$. The quantity $X_n$ is the estimated value at the beginning of iteration $n$, $\nabla H_n$ is an estimate of the gradient $\nabla H(X_n)$ from iteration $n$, and $\Pi_{\Theta}$ is a projection onto $\Theta$. The Robbins-Monro algorithm uses an unbiased estimator for $\nabla H(X_n)$ whereas the Kiefer-Wolfowitz algorithm uses finite-difference estimates.

As $n$ approaches infinity, $X_n$ approaches a value such that the theoretical regression function of the stochastic response is minimized. Kouritzin (1996) and Kulkarni and Horn (1996) give alternate proofs for the convergence of the stochastic approximation method. The difficulty with StApp is that a large number of iterations of the recursive formula are needed to come up with the optimum. L’Ecuyer and Yin (1998) identify how to allocate a total available computational budget to StApp iterations for different gradient estimators. In addition, the choice of the observation length for each iteration is an important problem with respect to preventing conditional bias caused by the information known at the beginning of the iteration. Arising from the facts that the classical StApp algorithm can converge extremely slowly when applied to flat functions and can sometimes diverge when applied to functions with superlinear growth, Andradottir (1995) proposes a variant of StApp defined over a growing sequence of compact sets. Kleinman et al. (1999) describe the use of common random numbers for reducing the variance in stochastic approximation estimates.

A number of different gradient estimators can be used in StApp algorithms. L’Ecuyer and Glynn (1994) study the minimization of the mean system time in a GI/G/1 queue and prove the convergence of finite difference estimates, likelihood ratios and infinitesimal perturbation analysis methods in StApp algorithms under mild conditions. In a companion paper, they report numerical examples with an M/M/1 queue (L’Ecuyer et al., 1994). Fu and Hill (1997) propose a simultaneous perturbation method as an alternative to classical finite-difference estimates in StApp. The method is a finite-difference-like technique but requires only two simulations per gradient estimate regardless of the number of parameters in the problem. The authors illustrate the method’s performance on a single-server queue, a queuing network and a bus transit network. They report that the method results in substantial computational savings for large dimensional systems. In another study, Chong and Ramadge (1994) consider a load-sharing problem for a multiprocessor system in which jobs have real-time constraints. If the waiting time of a job exceeds a random amount, this job is lost. In order to minimize the steady-state probability of loss, a StApp algorithm with LRE is used and the results are reported. Andradottir (1996b) presents a scaled StApp algorithm and reports that it has some practical advantages over the Robbins-Monro algorithm.


3. Global search methods

All of the techniques discussed in Section 2 assume a unimodal surface. The methods that we will discuss in this section, such as evolutionary algorithms, simulated annealing, tabu search, Bayesian/sampling algorithms, gradient surface method, are designed for problems with multi-modal response surfaces. Moreover, these methods have some other advantages when compared to “traditional” methods. The traditional methods are adapted to cases where the domains of the decision variables are real intervals. Several
of them are iterative, and some (i.e., perturbation analysis, finite difference estimates, etc.) require gradient information. Thus, when the response surface is high-dimensional, discontinuous, and nondifferentiable, or when the decision variables are qualitative (e.g., queueing strategies), these methods often fail to find the optimal solution. On the other hand, methods such as evolutionary algorithms can be applied to these type of problems (Azadivar and Tomkins, 1999; Pierreval and Paris, 2000). Below we review the literature on global optimization methods.

3.1. Evolutionary algorithms

Evolutionary Algorithms (EAs) are heuristic search methods that implement ideas from the evolution process. As opposed to a single solution used in traditional methods, EAs work on a population of solutions in such a way that poor solutions become extinct, whereas the good solutions evolve to reach for the optimum. Recently, there has been an increasing interest in using EAs in simulation optimization because they require no restrictive assumptions or prior knowledge about the shape of the response surface (Bäck and Schwefel, 1993). Biethahn and Nissen (1994) identify alternative combinations of EAs in simulation optimization and discuss how they differ from traditional optimization methods. In general, an EA for simulation optimization can be described as follows: (i) generate a population of solutions; (ii) evaluate these solutions through a simulation model; (iii) perform selection, apply genetic operators to produce a new offspring (or solution), and insert it into the population; and (iv) repeat until some stopping criterion is reached.

In the literature, the most popular EAs are Genetic Algorithms (GAs) (Goldberg, 1989), Evolutionary Programming (EP) (Fogel, 1992), and Evolution Strategies (ES) (Schwefel, 1981). These algorithms differ in the representation of individuals, the design of variation operators, and the selection of their reproduction mechanisms. Bäck et al. (1997) describe the purpose, structure, and working principles of these three well-known EAs. In general, each point in the solution space is represented by a string of values for the decision variables (i.e., each position in the string represents the decision alternatives regarding a parameter in the system). The use of appropriate crossover and mutation operators reduces the probability of trapping to a local optimum. The crossover operator breaks the strings representing two members of the population and exchanges certain portions of the strings to produce two new strings, where the mutation operator selects a random position in a string and changes the value of that variable with a specified probability. Liepins and Hilliard (1989), Davis (1991) and Muhlebein (1997) give detailed overviews on the different techniques used in applying GAs.

Although EP and ES have not been widely used in simulation optimization, GAs have found considerable interest in optimizing problems that arise in complex manufacturing systems. Bowden and Bullington (1996) simulate a manufacturing system to discover opportunistic control rules for job routing and process sequence flexibility; and then they use GAs to develop cooperative strategies for selecting the best route through the manufacturing system based on simulated real-time information. Dengiz et al. (1997) use GAs for optimization of an (s, S) periodic review inventory control system with stochastic lead times, and for optimization of buffer sizes of an asynchronous automatic assembly system. They compare the results of GA with the exhaustive search and random search methods. Results indicate that GA performs better than random search. Azadivar and Tomkins (1999) develop an object-oriented method that automatically generates a simulation model and computes responses for a given set of decision factors. These responses are in turn provided to the GA to generate the next population. The authors apply this method to a particular manufacturing system where the decision variables are the choice of a machine in each station, the dispatching rule, and other characteristics of the components of the system. They report that GA outperforms random search on three sample problems. They also note that GA consistently achieves a larger fraction of the possible improvement at each iteration. Dümmler (1999) considers the problem of sequencing n lots, where each lot can be processed by any of m available cluster tools to minimize the cycle time. In this study, GA is used to generate the lot sequences. Numerical results on three problem instances suggest that optimal or close-to-optimal sequences can be found in a short time. Wellman and Gemmill (1995) apply GA to optimize the performance of asynchronous automatic assembly systems. Their results indicate that GA performs as well as stochastic quasi-gradient methods. There are also other studies in the literature in which the authors report successful applications of GAs to various manufacturing problems (an AGV network problem by McHaney (2000), assembly line optimization by Lee et al. (2000), optimization of management parameters in flowlines by Fontanilli et al. (2000) and finding the best facility layout by Suresh et al. (1995)).

Stuckman et al. (1991) compare GAs, simulated annealing and Bayesian/sampling algorithms, and conclude that GAs and simulated annealing are suitable for problems with high dimensionality. Lacksonen and Anusornnitisarn (1995) compare GAs with simulated annealing, H-J and N-M on 20 test problems and report that GAs give the best results. Yunker and Tew (1994) compare a GA, H-J and response surface method search for accuracy and stability on an example problem. They report that the GA is superior to the other two in finding the optimum. The variance also results in a narrower confidence interval with GA.

Hall et al. (1996) use ES to solve the kanban sizing problem for a manufacturing system with 39 decision variables. They provide some insights on the effects of the number of decision variables on the population size and the fitness of the solutions. The study by Cassady et al. (2000) uses ES to find the optimal decision variables and to obtain the
cost performance of a combined control chart-preventive maintenance model.

Pierreval and Tautou (1997) propose a new EA to optimize both quantitative and qualitative variables. They focus on the general schema of the EAs given in Muhlebein (1997). The method is applied to a workshop producing yogurt pots. The near-optimal solutions are compared with the results of an exhaustive search. The results indicate that the algorithm achieves reasonably good solutions. Pierreval and Paris (2000) suggest the use of a distributed EA for optimization of simulation models. In the distributed approach, several computers do their own simulation experiments with their own populations of solutions, but also exchange the solutions by using a migration operator. The benefits of this new approach are demonstrated through an example involving a transport lot sizing and transporter allocation problem in a flow shop. The authors discuss how the new approach allows the use of qualitative variables.

3.2. Simulated annealing

As discussed in Kirkpatrick et al. (1983) and Cerny (1985), a Simulated Annealing (SA) algorithm starts with an initial solution, generally chosen at random. A neighbor of this solution is then generated by a suitable mechanism and a change in the objective value is calculated. If a reduction occurs, the generated neighbor replaces the current solution. If there is no reduction, the SA algorithm may accept this solution with some probability to avoid entrapment in a local optimum. Generally, the probability of accepting such points is less than the value of acceptance function which is set to $\exp(-I/T)$. In this expression, $I$ is the difference between the function value evaluated at the current solution and its neighborhood, and $T$ is a control parameter that is analogous to temperature in physical annealing. To achieve the optimum, SA algorithms begin with high values of $T$, and stay at the same temperature for a prespecified number of iterations, gradually decreasing it until a final temperature is reached. In order to implement SA algorithms, all the parameters (i.e., initial temperature, final temperature, and number of iterations at each temperature) must be determined a priori, a task which requires additional computation and cooling schedules. Collins et al. (1988) and Hajek (1988) have suggested a great variety of cooling schedules. Overviews on theory and applications of SA algorithms can be found in Van Laarhoven and Aarts (1987), Johnson et al. (1989), Eglese (1990) and Kouamias et al. (1994).

There is a substantial literature that aims to outline general working principles for SA algorithms. Haddock and Mittenthal (1992) apply a SA with a heuristic cooling function to a hypothetical model. Their results indicate that a lower final temperature, a slower rate of temperature decrease and a large number of iterations performed at each temperature yield better solutions. Catoni (1992) derives finite-time estimates for the cooling schedules. Alkhamis et al. (1999) extend the basic results for SA algorithms to a stochastic optimization problem where the objective function is evaluated through Monte Carlo simulation. They show that the modified SA algorithm converges in probability to a global optimum under suitable conditions on the random error. Alrefaei and Andradottir (1999) also propose a modified SA algorithm that uses a constant (instead of decreasing) temperature. They use two approaches to estimate the optimal solution and show that both are guaranteed to converge to the set of global optimal solutions.

Yucesan and Jacobson (1996) consider solving the accessibility problem by using a local search procedure and five different variants of SA algorithms. SA outperforms the local search, but must be customized (i.e., modification of the annealing schedule) to the problem at hand, a major undertaking.

SA algorithms are also widely used for optimization of manufacturing systems. Manz et al. (1989) apply an SA algorithm to an automated manufacturing system. Brady and McGarvey (1998) compare SA, tabu search, GA and a frequency-based heuristic in the optimization of staffing levels in a pharmaceutical manufacturing laboratory. Barretto et al. (1999) apply a variant of the Linear Move and Exchange Move (LEO) optimization algorithm (Barretto et al., 1998) based on SA to a steelworks simulation model.

3.3. Tabu search

Tabu search is a constrained search procedure, where each step consists of solving a secondary optimization problem. At each step, the search procedure omits a subset of the solution space to search. This subset changes as the algorithm proceeds and is usually defined by previously considered solutions which are called the reigning tabu conditions (Glover and Laguna, 1997). A number of papers employ tabu search procedures to simulation optimization. Hu (1992) investigates the reliability and efficiency of a tabu search algorithm by using some standard test functions and reports that it outperforms the random search and a composite GA on the example problems. Garcia and Bolivar (1999) develop a simulation system which they call the Stochastic Inventory System Simulator (SISS) and optimize five stochastic inventory models with a wide range of probability distributions for demand and lead time by using tabu search. Lutz et al. (1998) address the problem of buffer location and storage size in a manufacturing line and use tabu search to find the optimal output. Martin et al. (1998) implement four different variations of tabu search to determine the number of kanbans and lot sizes in a generic kanban system. They report that tabu search performs much better than local search, but SA provides the same results as tabu search with better computation times. They conclude that the algorithm can rapidly identify optimal or near-optimal schedules for a broad range of industrial settings. Dengiz and Alabas (2000) also apply a tabu search algorithm to a simulation...
model of a just-in-time system to find the optimal number of kanbans and report that it outperforms random search.

3.4. Bayesian/sampling algorithms

The Bayesian/Sampling (B/S) methodology is a search strategy where at each iteration, the next guess is chosen to be the point that maximizes the probability of not exceeding the previous value by some positive constant \( \psi_n \) as given by:

\[
P(H(X_{n+1}) \leq H(X_n) + \psi_n),
\]

For a minimization problem, this method takes guesses in areas where the mean performance evaluated by simulation is low. Initially, \( \psi_n \) is small but increases as the optimization search becomes more local in nature to speed up the convergence. In order to search for a global minimum, the search space is explored once in an area until local solutions become overpopulated. Lorenzen (1985) and Easom (1990) employ this method for multi-dimensional solution spaces. Stuckman and Easom (1992) give a survey of existing B/S methods (i.e., Stuckman's method, Mockus's method, Perttunen's method) and compare them with other methods such as a clustering algorithm, a SA algorithm and the Monte Carlo method on functions of continuous variables. From their results, it appears that B/S algorithms consistently outperform Monte Carlo and SA.

3.5. Gradient surface method

Ho, Shi, Dai and Gong (1992) proposed the Gradient Surface Method (GSM) for optimization of discrete-event dynamic systems. GSM differs from other global search techniques because it uses traditional search methods to globally explore a response surface. It combines the advantages of RSM and efficient derivative estimation techniques such as PA and LRE in conjunction with stochastic approximation algorithms. In GSM, the gradient estimation is obtained by PA or LRE, and the performance gradient surface is fit from these gradient estimates by using least squares methods as in RSM. Zero points of the fitted gradient surface are than taken as the estimates of the optimal solution. GSM is a global search algorithm because at each iteration, it uses the information from all data points rather than just the local gradient. The most attractive features of the algorithm are that it obtains the gradient estimates by a single run, and it quickly gets to the vicinity of the optimal solution because of its global orientation. Ho, Shi, Dai and Gong (1992) apply this algorithm to six examples of queuing networks, but do not provide any comparisons with other methods.

4. Multiple-objective problems

Real problems of interest may be characterized by several objectives that should be optimized simultaneously. While optimizing more than one objective via simulation, a number of difficulties arise due to the stochastic nature of the responses. In addition, some of the objectives may be in conflict, in the sense that improving one makes at least another one worse. Because of these complexities, only limited work has been done in multiobjective simulation optimization.

In the literature, a number of different approaches have been used. Kleijnen (1990) assigns one of the responses as the primary response to be optimized and tries to have certain levels of achievement on the other objective functions. Variations of the goal programming approach are used by Clayton et al. (1982) and Rees et al. (1984). Chen and Tsai (1996) develop a direct search algorithm expanded from the H-J for multiobjective manufacturing systems where user-specified goals can be precise and/or fuzzy.

Mollaghasemi et al. (1991) describe a procedure based on the gradient projection technique and the use-of-value function. Teleb and Azadivar (1994) propose a modified complex search method in which they assume that the objective functions and the stochastic constraints are distributed as normal random variables, and use the maximum likelihood concept.

Some interactive approaches have been developed for multiobjective simulation optimization. Mollaghasemi (1994) introduces an interactive approach based on the Geoffrion-Dyer-Feinberg (GDF) vector maximal algorithm. Boyle and Shin (1996) propose an interactive multicriteria method (i.e., the Pairwise Comparison Stochastic Cutting Plane (PCSCP) method) that combines features from interactive multiobjective mathematical programming and RSM. Lee et al. (1996) also present an interactive algorithm by using a cutting plane approach with trade-off weights and trade-off cuts. Baesler and Sepulveda (2000) integrate a stochastic GA heuristic with a goal programming model to solve multiple objective simulation optimization problems.

5. Discussion and further research directions

In the past, simulation has often been regarded as the method of last resort among OR researchers and users. However, over the years as a result of advances in computer technology and simulation languages, it has become one of the most valuable OR tools (it is usually ranked at the top of the list). The need for dynamic and stochastic models to analyze complex systems is also expected to increase the interest for simulation in the future. A simulation model is typically a descriptive model of the system, i.e., it describes the behavior of the system under consideration, and helps us to understand the dynamics and complex interactions among the elements of the system. In
the scientific community, simulation has often been criticized for a lack of optimization capability; simulation results have typically been runs with a set of observations, rather than an optimum solution to the problem as usually found by the prescriptive or normative models (e.g., linear programming, dynamic programming). By incorporating optimization features in simulation systems, we not only eliminate its major limitation, but also open a door for simulation in terms of new application areas and research possibilities. With the ability of performing optimization, simulation can also become an operational tool to solve short-term decision-making problems as well as strategic and tactical problems (i.e., design and long-term planning problems).

In this paper, we presented a comprehensive survey on techniques for simulation optimization. We classified the techniques according to the characteristics of the problems such as objective functions, parameter spaces and shape of the response surface, i.e., unimodal or multimodal. We also discussed major advantages, drawbacks and, comparisons of these techniques in the paper. We point out that there are several heuristics and approaches that integrate ideas from the well-known methods such as retrospective techniques (Healy and Schruben, 1991; Fu and Healy, 1997), and sample-path optimization (Gürkan, 2000) which are not discussed extensively here.

The methods proposed for simulation optimization each carry their own assumptions, and the performances depend on the characteristics of the problems to which they are applied. In the literature, there have been also some research efforts to compare the methods. A list of the comparative studies is presented as Table 2. The conclusions from these studies vary, i.e., one method can perform very well for one problem but can be the worst for another problem. In our opinion, one future research effort would be to generate benchmark problems or define some standard test functions to evaluate performances of different techniques. These test problems also provide important insights into which methods to integrate for solving a particular problem. As discussed in Law and McComas (2000), some commercial simulation companies have already integrated their simulation systems with some kind of global search methods (SA, GA, and tabu search). In this context, these benchmark problems can also serve as a platform to evaluate the effectiveness (solution quality and run times) of future simulation packages.

Since the problems that are to be solved by simulation optimization vary in terms of the number and structure (i.e., discrete or continuous, quantitative or qualitative) of decision variables, and shape of the response function, there is no single method to solve all of these problems. This forces researchers to develop more robust techniques that can handle a larger class of problems. For instance, EAs require no prior idea about the response surface and can handle both quantitative and qualitative variables. Therefore, they seem to be very promising in terms of future research potential in this area. At the same time, there is also a need for special methods that work effectively under certain circumstances (Fu et al., 2000). This is especially needed when simulation optimization has to be used on a continuous basis to solve day-to-day operational problems.

As pointed out by Andradóttir in Fu et al. (2000), simulation optimization still requires a considerable amount of computer time. This is one of the important barriers on future simulation optimization applications since a model has to be run under a large number of experimental conditions. One possible solution (or at least a way to alleviate the problem) would be to use parallel processors or the utilities of distributed computing. There are already some commercial simulation products that make use of the benefits of

<table>
<thead>
<tr>
<th>Publication</th>
<th>Techniques</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smith (1971)</td>
<td>RSM, SFM</td>
<td>Hypothetical surfaces</td>
</tr>
<tr>
<td>Azadivar and Talavage (1980)</td>
<td>StApp, RSM</td>
<td>Unimodal polynomials</td>
</tr>
<tr>
<td>Azadivar and Lee (1988)</td>
<td>Heuristic based on simplex search, integer gradient search, RSM</td>
<td>Hypothetical surfaces</td>
</tr>
<tr>
<td>Reiman and Weiss (1989)</td>
<td>LRE, PA</td>
<td>—</td>
</tr>
<tr>
<td>Stuckman et al. (1991)</td>
<td>B/S, SA, GA</td>
<td>—</td>
</tr>
<tr>
<td>Hu (1992)</td>
<td>Tabu search, RSM, GA</td>
<td>Standard test functions</td>
</tr>
<tr>
<td>Stuckman and Easom (1992)</td>
<td>B/S</td>
<td>Hypothetical surfaces</td>
</tr>
<tr>
<td>L’Ecuyer and Perron (1994)</td>
<td>IPA, FDE</td>
<td>GI/G/1 queue</td>
</tr>
<tr>
<td>L’Ecuyer et al. (1994)</td>
<td>FDE, LRE, PA</td>
<td>M/M/1 queue</td>
</tr>
<tr>
<td>Fu (1994b)</td>
<td>LRE, FDE, PA, FDA</td>
<td>—</td>
</tr>
<tr>
<td>Yunker and Tew (1994)</td>
<td>GA, H-J, RSM</td>
<td>—</td>
</tr>
<tr>
<td>Heidergott (1995)</td>
<td>FDE, PA</td>
<td>A manufacturing system</td>
</tr>
<tr>
<td>Fu and Healy (1997)</td>
<td>Retrospective methods, PA</td>
<td>(s, S) inventory systems</td>
</tr>
</tbody>
</table>
distributed computing in a multiprocessing environment. The other possibility to reduce the computational burden, as suggested by Harrell in Fu et al. (2000), would be to use statistical methods such as factor screening techniques to reduce the search space. More research also needs to be conducted in these areas to improve the efficiency of simulation optimization systems.

It appears that most of the existing simulation software packages with optimization features rely on meta-heuristics (SA, GA, tabu, etc.). However, there is already a great deal of work that has been done in RSM and gradient-based methods. In future studies, these traditional methods together with meta-heuristics can be jointly used to develop better hybrid approaches in dealing with simulation optimization problems.

Acknowledgement

The authors would like to thank David Goldsman for his many helpful comments and suggestions.

References


A review on the simulation optimization literature


Biographies

Eylem Tekin is an Assistant Professor in the Department of Statistics and Operations Research at the University of North Carolina—Chapel Hill. She received her B.S. and M.S. degrees in Industrial Engineering from Bilkent University, Turkey, and her Ph.D. degree from Northwestern University in Industrial Engineering and Management Sciences in 2002. Her research interests include stochastic modeling of discrete-event systems, design and control of production systems, and simulation.
has published papers in Management Science and European Journal of Operational Research. She is a member of IIE and INFORMS.

Ihsan Sabuncuoglu is Professor of Industrial Engineering at Bilkent University. He received B.S. and M.S. degrees in Industrial Engineering from the Middle East Technical University and a Ph.D. degree in Industrial Engineering from Wichita State University. He teaches and conducts research in the areas of simulation, scheduling, and manufacturing systems. He has published papers in many journals, including IIE Transactions, Decision Sciences, Simulation, International Journal of Production Research, International Journal of Computer Integrated Manufacturing, European Journal of Operational Research, International Journal of Production Economics, and Journal of Intelligent Manufacturing. He is on the Editorial Boards of International Journal of Operations and Quantitative Management, Journal of Operations Management, and International Journal of Computer Integrated Manufacturing, and is an Associate Editor of Transactions on Operational Research. He is an associate member of IIE and INFORMS.