Proceedings of the 1994 Winter Simulation Conference
ed. J. D. Tew, S. Manivannan, D. A. Sadowski, and A. F. Seila

A TUTORIAL REVIEW OF TECHNIQUES FOR SIMULATION OPTIMIZATION

Michael C. Fu
University of Maryland at College Park
College Park, Maryland 20742

ABSTRACT

We review techniques for optimizing stochastic discrete-event systems via simulation, for both discrete and continuous parameters. For discrete parameters, we focus on the techniques for optimization from a finite set: multiple-comparison procedures and ranking-and-selection procedures. For continuous parameters, we discuss sequential response surface methodology procedures and stochastic approximation gradient-based procedures, and describe gradient estimation based on perturbation analysis, likelihood ratio and frequency domain experimentation. We then discuss two applications: an inventory control problem with a"noisy" constraint and a call option pricing problem in finance.

1 INTRODUCTION

The problem under consideration is the following parametric optimization problem:

$$\min_{\theta \in \Theta} J(\theta), \tag{1}$$

where $J(\theta) = E[L(\theta,\omega)]$ is the performance measure of interest, $L(\theta,\omega)$ is the *sample* performance, ω represents the stochastic effects of the system, θ is a controllable vector of p parameters, and Θ is the constraint set on θ. Define the optimum by $\theta_* = \arg\min_{\theta \in \Theta} J(\theta)$. In the experimental design literature, the performance measure is usually referred to as the *response* and the parameters as *factors*. In this paper, we consider only the single response problem. For expository purposes, we will often discuss application of the various techniques to the following two discrete-event system simulation models (cf. Law and Kelton 1991, Cassandras 1993).

Example 1. For a $GI/G/1$ queue, find the mean service time of the server that minimizes a cost function which trades off the expected mean time in system and the server speed:

$$\min_{\theta \in \Theta} c_0 E\left[\frac{1}{N}\sum_{i=1}^{N} T_i\right] + c_1/\theta, \tag{2}$$

where T_i is the ith system time, N is the number of customers served, c_0 and c_1 are given costs, θ is the mean service time, λ is the arrival rate, and $\Theta = [\delta, 1/\lambda - \delta]$, for some $\delta \approx 0$. For the $M/M/1$ queue in steady state, the optimum can be determined analytically as $\theta_* = (\lambda + \sqrt{c_0/c_1})^{-1}$.

Example 2. For an (s,S) inventory control system, find the values of s and $q = S - s$ to minimize a cost function on holding, ordering, and backlogging.

$$\min_{\theta \in \Theta} \frac{1}{N} E\left[\sum_{i=1}^{N} C(X_i)\right], \tag{3}$$

where X_i is the inventory level (and inventory position, as well, under zero lead time) at review epoch i, N is the number of periods in the horizon, $C(x) = hx^+ + px^- + I\{x < s\}[K + c(S-x)]$, h, p, K, and c are the holding, backlogging, order set-up, and order per-unit costs, respectively, $\theta = (s,q)$, and $\Theta = \mathcal{R} \times \mathcal{R}^+$.

The best possible convergence rate with "pure" stochastic optimization algorithms is generally $O(n^{-1/2})$, where n represents (roughly) the computational effort. However, we point out that this is an *asymptotic* convergence rate, and that $O(n^{-1/2})$ is also the best convergence rate obtainable for simulation *estimation* (vs. optimization) of any (non-trivial) output random variable.

2 OPTIMIZATION OVER A FINITE SET

Oftentimes, the number of choices in the parameter set is finite. This may be due to the nature of the problem itself, or it may be due to a reduction through other analyses, or it may be a simplifying step due to practical considerations. If the number of choices is not too large, then statistical procedures

based on ranking and selection or multiple comparisons can be applied. Roughly speaking, ranking-and-selection procedures specify some criterion, such as choosing the best with some pre-specified confidence level, and then derive a statistical procedure, usually sequential, that meets the criterion. Multiple-comparisons procedures, on the other hand, specify the use of certain pairwise comparisons to make inferences in the form of confidence intervals; they are not inherently sequential procedures. A simple example comparing and contrasting different techniques can be found in Goldsman et al. (1991).

Again, we wish to solve the parametric optimization problem (1), where now the parameter set is finite: $\Theta = \{\lambda_1, \lambda_2, ..., \lambda_K\}$, i.e., we wish to find λ_i s.t. $\lambda_i = \theta_*$. Let us denote the estimate of performance from the jth sample path (replication) at λ_i by L_{ij}. Thus, our estimate of $J(\lambda_i)$ over n sample paths (replications) is simply the sample mean: $\widehat{J}_i = \bar{L}_i = \frac{1}{n}\sum_{j=1}^{n} L_{ij}$.

Procedures based on multiple comparisons are of very basic importance in statistical inference, since applications inevitably require comparisons. Like most statistical techniques, the two major assumptions underlying the procedures are *independence* and *normality*. The former directly conflicts with some of the advantages of discrete-event simulation, e.g., the implementation of powerful variance reduction techniques such as common random numbers (CRN) and control variates. We will sketch the main ideas of three multiple-comparisons procedures: (1) a "brute force" paired-t, Bonferroni, all-pairwise comparisons approach that works particularly well when CRN apply; (2) an all-pairwise multiple comparisons (MCA) approach; (3) a multiple comparisons with the best (MCB) approach more tailored to optimization purposes than the previous two approaches, and requiring far fewer comparisons.

The idea of the "brute force" approach is simple:
1. Calculate a difference estimate for each possible pair of replications.
2. Form the usual $(1-\alpha)100\%$ confidence intervals for each difference.
3. Apply the Bonferroni inequality to arrive at a lower bound on the *overall* confidence level.

After forming all $K(K-1)/2$ confidence intervals, one would simply look to see if there is a "clear winner," i.e., a λ_i such that the confidence interval for the difference with all other pairs is strictly negative. If not, one can crudely eliminate some candidates, estimate the number of additional replications needed to make conclusive inference, and repeat the process with the smaller set.

MCA works in principle similar to the above, except that instead of constructing separate confidence intervals and using Bonferroni to determine an overall confidence bound, a *simultaneous* set of confidence intervals at an overall $(1-\alpha)100\%$ level is formed by calculating a single confidence half-width via the pooled sample variance. Although the original version of MCA requires independence between the replications, a control-variate variation allowing the use of CRN was derived in Yang and Nelson (1991).

The intent of multiple comparisons with the best (MCB) procedures is to reduce the number of comparisons, since we are interested in the optimization goal of picking only the best. The procedure is as follows (Hsu and Nelson 1988):
1. Form K confidence intervals for each choice with the *best of the rest.*
2. If only one of the confidence intervals falls on the negative side of 0, then the λ_i corresponding to that interval would be declared the optimum; otherwise, all of the λ_i with intervals having some part on the negative side of 0 could potentially be the optimum.

In practice, the pooled variance could be used to estimate the additional number of replications needed to make a more conclusive determination. Hsu and Nelson (1988) demonstrate the procedure for Example 2, the (s, S) inventory system; see also Yang and Nelson (1991).

In terms of ranking-and-selection procedures, two approaches have been taken: indifference zone and subset selection. The method of Dudewicz and Dalal (cf., Law and Kelton 1991) falls into the indifference-zone approach. It has two strong points that make it particularly suitable for optimization of discrete-event simulations: the variances do not have to be equal and they do not have to be known. However, independence must be maintained, thus precluding the use of CRN. The procedure guarantees that with user-specified probability at least P^* the selected λ_i will guarantee that $J(\lambda_i)$ is within δ of the optimal value $J(\theta_*)$, where δ represents the "indifference zone," i.e., $P\{J(\lambda_i) - J(\theta_*) < \delta\} \geq P^*$, including the possibility that $\lambda_i = \theta_*$.

The basic idea of the procedure is the following:
1) Take a first-stage set of replications for each of the different parameter settings to calculate first-stage sample means and sample variances. 2) Use the first-stage sample variances to determine the number of second-stage replications needed for each of the different parameter settings. 3) Use the second-stage replications to get second-stage sample means. 4) Take a weighted average of the first-stage and second-stage sample means. 5) Choose the λ_i with the smallest weighted average estimate of $J(\lambda_i)$.

A subset-selection procedure would be algorithmi-

cally similar, with the notable exception being the last step, where instead of selecting a single λ_i, a *subset* of all λ_i having the weighted average estimate of $J(\lambda_i)$ within some preselected distance is selected, up to some maximum number. Although very powerful tools, at present the major disadvantage of ranking-and-selection procedures for simulation optimization is the requirement of independence over competing designs, which precludes the use of most variance reduction techniques such as CRN.

3 RESPONSE SURFACE METHODS

Broadly speaking, response surface methodology (RSM) attempts to fit a polynomial (possibly after some initial transformation on the variables) of appropriate degree to the response of the system of interest. The application of RSM to simulation optimization falls into two main categories: metamodels and sequential procedures. In the context of optimization, it usually takes the form of the latter, whereby through successive experimental stages, one attempts to "home in" on the optimal region where a "final" (usually quadratic) polynomial is fitted and the optimum determined through the usual deterministic means. We will briefly outline the general approach in the context of discrete-event simulation.

Instead of exploring the entire feasible region, which may be impractical or computationally prohibitive, small subregions are explored in succession, where successive subregions are selected for their potential improvement. A point, e.g., the center of the subregion currently being explored, "represents" the current "best" θ value. The basic algorithm consists of two phases:

- **Phase I**

In this phase, first-order experimental designs are used to get a least-squares fit. Then, a steepest descent direction is estimated from the model, and a new subregion chosen to explore via

$$\theta_{n+1} = \theta_n - a_n \nabla J_n, \tag{4}$$

where θ_n is the representative point of the nth explored subregion, ∇J_n is the estimated (from the fitted linear response) gradient direction, and a_n is a step size determined by a line search or some other means. This is repeated until the linear response surface becomes inadequate, which is indicated when the slope is "approximately" zero, when the interaction effects become larger than the main effects.

- **Phase II**

A quadratic response surface is fitted using more detailed second-order experimental designs; the optimum is then determined analytically from this fit.

From the algorithm, one can see that Phase II is done just once, whereas Phase I is iterated a number of times. Thus, for each iteration of Phase I, one should strive to expend fewer replications, whereas in Phase II, the region should be explored quite thoroughly by using a large number of replications.

RSM sequential procedures provide a very general methodology for optimization via simulation. RSM's biggest advantage is its generality, but its biggest drawback if applied blindly is its computational requirements. Other techniques or analyses based on the nature of the system of interest which can be used to improve the efficiency of RSM are crucial. For example, efficient gradient estimation techniques may be used to *complement* the sequential aspects of RSM by reducing the number of simulation points.

4 STOCHASTIC APPROXIMATION

In this section, we consider gradient-based stochastic optimization algorithms, where the "best guess" of the optimal parameter is updated iteratively based on an estimate of the gradient of the performance measure with respect to the parameter. Actually, the sequential RSM procedure also implements a gradient-based algorithm in Phase I, where the gradient is found from the regression model.

The basic underlying assumption of stochastic approximation is that the original problem given by (1) can be solved by finding the zero of the gradient, i.e., by solving $\nabla J(\theta) = 0$. Of course, in practice, this may lead only to local optimality. The general form of the stochastic algorithm takes the following form:

$$\theta_{n+1} = \Pi_\Theta \left(\theta_n - a_n \widehat{\nabla} J_n \right), \quad \widehat{\nabla} J_n = [\widehat{\nabla}_1 J_n \cdots \widehat{\nabla}_p J_n]^T, \tag{5}$$

where θ_n is the parameter value at the beginning of iteration n, $\widehat{\nabla} J_n$ is an estimate of $\nabla J(\theta_n)$ from iteration n, a_n is a (positive) sequence of step sizes, and Π_Θ is a projection onto Θ. When finite differences are used to estimate $\nabla J(\theta_n)$, (5) is called a **Kiefer-Wolfowitz** algorithm; when a direct (possibly unbiased) estimator is used for $\nabla J(\theta_n)$, (5) is called a **Robbins-Monro**-like algorithm (cf. Kushner and Clark 1978). The usual requirements needed for the convergence of (5) to the optimum are that (i) the step size go to zero at a rate not too fast to lead to convergence to the wrong value and not too slow to avoid convergence to a value at all, and (ii) that the bias of the gradient estimate go to zero. One set of common assumptions on the step sizes is $\sum_n a_n = \infty, \sum_n a_n^2 < \infty$, which for example the harmonic series $a_n = a/n$ (for some constant a) satisfies. In terms of practical implementation for discrete-

event simulation, one must select various parameters in the algorithm such as the initial step size a and the observation horizon, as well as a projection rule and a stopping rule for the algorithm.

We discuss four gradient estimation techniques: finite differences (FD), perturbation analysis (PA), the likelihood ratio (LR) method, and frequency domain experimentation (FDE). FD and FDE techniques alter the input and analyze the resulting output, whereas PA and LR involve an "add-on" to the simulator itself, which involves *additional* accumulations and calculations. However, the underlying simulator (by which we mean the event-generation scheme) is *not* altered, and as a result both LR and PA can also be implemented for on-line gradient estimation and optimization.

The most obvious way to estimate the gradient is to run multiple simulations to estimate some secant as an approximation to the tangent. We call this the finite difference (FD) estimate. The symmetric difference version is given by

$$\widehat{\nabla}_i J_n = \frac{\widehat{J}(\theta_n + c_n e_i) - \widehat{J}(\theta_n - c_n e_i)}{2c_n}, \qquad (6)$$

where e_i denotes the ith unit vector. Note that this estimate requires $2p$ simulations. The forward difference would simply replace $\widehat{J}(\theta_n - c_n e_i)$ with $\widehat{J}(\theta_n)$ and hence would require only $p+1$ simulations; however, the convergence rate when used in a stochastic approximation algorithm is worse.

A potentially more efficient version of finite differences is the simultaneous perturbation (SP) finite difference estimate proposed by Spall (1992). Let $\{\Delta_1, ..., \Delta_p\}$ be a set of i.i.d. perturbations satisfying the conditions given in Spall (1992), and define the vector $\Delta = [\Delta_1 ... \Delta_p]$. Then, the SP estimator is given by

$$\widehat{\nabla}_i J_n = \frac{\widehat{J}(\theta_n + c_n \Delta) - \widehat{J}(\theta_n - c_n \Delta)}{2c_n \Delta_i}. \qquad (7)$$

Note that whereas in the finite-difference estimators, there is a pair of numerators for each parameter, thus requiring $2p$ simulations, here the *same* pair is used in the numerator for all parameters, and the denominator changes; thus, only two simulations are required. This method was applied to a variation of Example 1, an M/U/1 queue, with a two-dimensional vector parameter in the service time distribution, in Hill and Fu (1994).

Kiefer-Wolfowitz algorithms require $c_n \to 0$ (at an appropriate rate) for convergence, and generally the best asymptotic convergence rate achievable is $O(n^{-1/3})$, versus $O(n^{-1/2})$ when an unbiased estimate is used. Although this procedure has the dual disadvantages of being computationally more intensive and having a slower convergence rate, it is straightforward to implement and the most generally applicable.

Perturbation analysis is one technique for obtaining unbiased gradient estimates efficiently. The books by Ho and Cao (1991) and Glasserman (1991) concentrate on the two most developed forms: infinitesimal perturbation analysis (IPA), which is simply the derivative of the sample performance $dL/d\theta$, and smoothed perturbation analysis (SPA), which is based on the "smoothing" property of conditional expectation and covers cases where IPA does not apply.

In terms of our two examples, for the $GI/G/1$ queue, an unbiased gradient estimator for mean steady-state system time T is given by the IPA estimator

$$\left(\frac{dT}{d\theta}\right)_{IPA} = \frac{1}{N} \sum_{m=1}^{M} \sum_{i=1}^{n_m} \sum_{j=1}^{i} \frac{dX_{(j,m)}}{d\theta}, \qquad (8)$$

where n_m is the number of customers served in the mth busy period, M is the number of busy periods, $N = \sum_{m=1}^{M} n_m$ is the total number of customers served, and $X_{(j,m)}$ is the service time of the jth customer in the mth busy period. An unbiased estimator for the second derivative can be easily derived via SPA. For the (s, S) inventory system example, SPA can be used to derive consistent estimators (Fu 1994). In Fu and Healy (1992), the estimators were applied to the optimization problem of Example 2.

Another technique for obtaining unbiased gradient estimates is the likelihood ratio (LR) method, also known as the score function (SF) method (cf. Rubinstein and Shapiro 1993). The basic idea of the method is to differentiate the underlying probability measure of the system, but it can more generally be viewed as a special case of importance sampling. Because the LR method requires the differentiation of a probability measure, the technique is not usually applicable to structural parameters such as s and S in the (s, S) inventory system. We present a brief informal overview of the LR technique, and derive estimators for Example 1. We assume that the dependence on θ enters only through a random vector X with joint cumulative distribution function $F(\theta, \cdot)$ and density $f(\theta, \cdot)$ depending on a parameter (or vector of parameters) θ: $E[L(X)] = \int L(x) dF(\theta, x)$. Differentiating, we have

$$\frac{\partial E[L]}{\partial \theta} = \frac{\partial}{\partial \theta} \int L(x) f(\theta, x) dx = \int L(x) \frac{\partial f(\theta, x)}{\partial \theta} dx$$

$$= \int L(x) \frac{\partial \ln f(\theta, x)}{\partial \theta} f(\theta, x) dx = E\left[L(X) \frac{\partial \ln f(\theta, X)}{\partial \theta}\right] \qquad (9)$$

Thus, in a single simulation, one can estimate the derivative of the performance measure along with the performance measure itself. Higher derivatives can be handled in a similar manner. However, the "naive" estimator for (9) leads to unbounded variance for steady-state performance measures. For the $GI/G/1$ queue, where the interarrival times and the service times comprise the random vector, the natural estimator would be given by

$$\left(\frac{dT}{d\theta}\right)_{LR} = \frac{1}{N}\sum_{i=1}^{N} T_i \frac{\partial \ln f}{\partial \theta}, \qquad (10)$$

where X_i is the ith service time. For example, for exponential service times, $\frac{\partial \ln f}{\partial \theta} = \sum_{i=1}^{N}\left(\frac{X_i}{\theta^2} - \frac{1}{\theta}\right)$. The problem with these estimators is that if they are used to estimate *steady state* quantities by increasing the horizon length N, then it is obvious that the variance of the estimator will *increase* linearly. On the other hand, a regenerative estimator does not suffer from this problem, although in practice regenerative cycles could be very long.

The intuitive idea in frequency domain experimentation (FDE) is to *oscillate* the value of the parameter according to a sinusoidal function during the simulation. The magnitude of the performance measure variation gives an indication of the relative sensitivity of the performance measure to the parameter. The vectors of input parameters are modulated as follows:

$$\theta(t) = \theta_0 + \alpha \sin(\tilde{\omega} t), \qquad (11)$$

where θ_0 is the (vector) parameter of interest, α is the vector of oscillation amplitudes, and $\tilde{\omega}$ is the vector of oscillation frequencies called the *driving* frequencies, which are assumed to be distinct in order to be able to discriminate between the contributions of each parameter. Note that the "time" variable t is usually *not* the simulation time. The application of FDE requires the solution of the following problems:

- indexing problem — determination of "t",
- frequency selection problem — determination of $\tilde{\omega}$,
- amplitude selection problem — determination of α.

For FDE, the gradient estimation problem is to estimate the gradient at $\theta(0) = \theta_0$, i.e., $\nabla J(\theta_0)$. By approximating J around θ_0 using a second-order Taylor series expansion, a quadratic dynamic polynomial response surface metamodel (Jacobson and Schruben 1992), call it $Y(\theta(t))$, leads to

$$\nabla_i Y(\theta_0) = \lim_{T\to\infty} \lim_{\omega_i \to 0} \frac{2}{\alpha_i T} \sum_{t=1}^{T} Y(\theta(t)) \sin(\omega_i t), \qquad (12)$$

where ∇_i denotes the partial derivative with respect to $\theta_i, i = 1, ..., p$. FDE estimators are usually referred to as harmonic gradient estimators in the literature. Thus, one simulation run can be used to estimate the gradient. However, since true unbiasedness is achieved only in the limits $\tilde{\omega} \to 0, T \to \infty, \alpha \to 0$, there are implementation trade-offs in the selection of these parameters, analogous to the choice of the difference in the FD estimate.

For the $GI/G/1$ example, we can use the customer index as the "time" variable and take $\omega = 2\pi/N$, where N is the number of customers served, to obtain

$$\left(\frac{dT}{d\theta}\right)_{FDE} = \frac{2}{\alpha N}\sum_{i=1}^{N} T_i \sin(2\pi i/N),$$

where the system is being simulated with the mean service time of the ith customer given by (11). In this example, the only decision that need be made is the choice of α. Sensitivity to this choice is evident in the experiments reported in Jacobson (1993) for an M/M/1 queue (which included the arrival rate as a parameter, as well); three values $\alpha = 0.001, 0.01, 0.1$ led to very disparate mean-squared errors of 2.8, 0.028, 0.018, respectively. For the (s, S) inventory system, FDE gradient estimators can also be derived. The choice of oscillation index this time is quite naturally simulation time as given by the discrete period number. Choices for $\tilde{\omega}$ and α still must be made.

In comparison with LR and PA, FDE gradient estimates require the additional selections of an oscillation index, oscillation frequencies, and oscillation amplitudes. The performance of the estimate will depend heavily on these selections. Moreover, like FD estimates, FDE estimates can never give an unbiased estimate of the gradient in finite time, because the limit $\tilde{\omega} \to 0$ can never be achieved. On the other hand, FDE seems in principle to be more general than IPA or LR, being more akin to the FD estimates. For instance, FDE applies to both the $GI/G/1$ queue and (s, S) inventory system examples, whereas IPA and LR apply only to the former, although SPA can be used for the latter. Overall, when it applies, IPA is usually the most efficient estimator.

We now briefly describe the application of SA to simulation optimization. One of the earliest applications was the work by Azadivar and Talmage (1980), who implemented a version utilizing FD estimates with a number of "practical" heuristics to improve its performance. They empirically compared the performance of their algorithm with an RSM sequential procedure for a number of simple polynomial functions with additive noise and a single discrete-event system. According to their simulation results, for a given computational budget, their algorithm dominated the RSM procedure for every example.

The first application of PA to optimization was contained in the paper by Ho and Cao (1983). An IPA gradient estimate for throughput of a queueing network was incorporated into a simple stochastic approximation algorithm on an objective function with Lagrangian multipliers. The approach was to use long simulation runs to get a good estimate of the gradient; thus, the number of iterations was relatively small. In contrast, the work of Suri and Zazanis (1988) introduced the idea of "single-run" optimization using IPA. Instead of completing a long simulation run before updating the parameter, and repeating the procedure for just a few iterations, the parameter was updated after a very short observation horizon, and the simulation *continued*; between iterations the simulation mechanism was not reinitialized and restarted. The single simulation run was terminated when it was determined that the gradient was "close enough" to zero according to a given stopping criterion. Thus, a *single* run of approximately the same length it would take to estimate the performance itself also yielded an estimate of the *optimal* value of the parameter, providing significant computational savings over the previous implementation. The procedure was applied to the steady-state version of Example 1 for various interarrival time and service time distributions, and empirically, the algorithm worked quite well. The first theoretical convergence proof was provided in Fu (1990) for the case where updates are done at the beginning of regenerative periods.

Examples incorporating LR estimators in SA algorithms are presented in Rubinstein and Shapiro (1993). In L'Ecuyer et al. (1994), a very comprehensive set of numerical experiments on the M/M/1 queue example are reported. Various algorithms utilizing IPA, LR, and FD estimates with CRN are considered and compared, with the IPA-based algorithms clearly superior. The numerous simulation results also show the obvious effect of step size selection.

5 CONSTRAINED OPTIMIZATION

Although a rich body of techniques are available for the problem (1) when the constraint set Θ is known, the literature on constrained optimization with "noisy" constraints is limited. For instance, in practice, managers are often uncomfortable with estimating backlogging costs in situations such as Example 1, instead preferring to work with a cost function involving only ordering and holding costs combined with an additional constraint of some type of service level such as fraction of demand filled from on-hand stock. This constraint then would itself require estimation by simulation. The only reported algorithm we know of that handles noisy constraints is the Lagrangian approach in Kushner and Clark (1978). One potential problem with this technique is that it guarantees a feasible solution only in the limit. In Bashyam and Fu (1994), an algorithm based on the feasible directions approach from nonlinear programming is proposed. The algorithm generates the sequence $\{\theta_n\}$ as

$$\theta_{n+1} = \theta_n + A_n D(\theta_n), \tag{13}$$

with the normalized direction vector $D(\theta_n)$ given by

$$D(\theta) = \begin{cases} -D_C(\theta) & \text{if } J(\theta) < \beta_l, \\ -D_J(\theta) & \text{if } J(\theta) > \beta_u, \\ D_f(\theta) & \text{otherwise,} \end{cases} \tag{14}$$

with $\beta_l \leq \beta \leq \beta_u$ and the normalized *feasible directions* vector D_f to be specified. The basic motivation of this algorithm is to generate a subsequence $\{\theta_{n_i}\}$ of *feasible and improving solutions.* To achieve this, we force the $\{J(\theta_n)\}$ process to visit a suitably constructed interval $I = [\beta_l, \ \beta_u]$ infinitely often. The construction of I ensures that every θ such that $J(\theta) \in I$ is a feasible solution. Within this interval, the direction of movement given by $D_f(.)$ is such that it points strictly towards the interior of the feasible region, and represents a reduction in cost. Whenever $J(\theta) > \beta_u$, the direction $-D_J(\theta)$ forces the process back into the feasible region, whereas $J(\theta) < \beta_l$ indicates that the process is well within the feasible region, and the algorithm in this case, proceeds in an unconstrained fashion. Given the observed values of $\nabla C(\theta_n)$ and $\nabla J(\theta_n)$ associated with θ_n, a number of options are available to determine a $D_f(\theta_n)$ with the desired properties. One standard approach is to use a linear program.

The proposed algorithm consists of three stages:

- Stage 1: analytical approximation;
- Stage 2: line search on s, with Q kept constant;
- Stage 3: update via (13).

Stages 1 and 2 are carried out once each, and are considered preprocessing steps to Stage 3, which is iterative. Computational experience reported in Bashyam and Fu (1994) for the (s, S) inventory problem described at the beginning of this section was very promising, showing substantial improvements for cases where analytical approximations fare poorly.

6 FINANCIAL DERIVATIVES

In finance parlance, "derivatives" are financial instruments (or contracts) that derive their value from some underlying commodity, e.g., a call option on a stock gives the right to buy the stock at a specified price

(the *strike* price) within a specified period of time (the *expiration date*). A European option can only be exercised at the end of the contract, whereas an American option can be exercised at any time up to and including the end of the contract. Determining the appropriate monetary values for these instruments — **derivative pricing** — is a matter of very practical concern, as well as a very active area of research in the finance community. The focus in the research literature has been on numerical methods. However, due to the complexity of the underlying commodities which violate many of the assumptions of analytical models, simulation is often a practical alternative. Here, we illustrate how simulation optimization techniques can be used in derivative pricing by considering an American call option, refuting the claim that "Monte Carlo simulation can only be used for European-style options" (Hull 1993, p.363). Basically, if the pricing problem is viewed as an optimization problem, there are American-style options which can also be priced by using Monte Carlo simulation in conjunction with gradient-based optimization techniques.

We begin by defining the following variables:
S_t is the stock price at time t, S_0 is the initial stock price, r is the annualized riskless interest rate (compounded continuously), μ, σ are parameters of the distribution of the underlying stock, K is the striking price of the option contract, T is the lifetime (expiration date) of the option contract, J_T is the net present value return of the option on its expiration. Aside from S_t and J_T, the rest of the variables are assumed to be constants.

We consider a stock which distributes dividend D_j at time $t_j, j = 1, \ldots, \eta(T)$, where $\eta(T)$ is the number of ex-dividends distributed during the lifetime of the call contract. Following standard models, we assume that after each ex-dividend, the stock price drops by the amount of the ex-dividend, i.e., $S_{t_j^+} = S_{t_j^-} - D_j$. For notational convenience, we also denote $t_0 = 0, t_{\eta(T)+1} = T$. We will assume that the ex-dividend amounts $\{D_j\}$ are known (deterministic). Although an American call option can be exercised at any time before the expiration date T, under the assumption of a frictionless market, it is well-known that the option should only be exercised, if at all, right before an ex-dividend date or at the expiration date, i.e., only at one of the t_j's. Thus, we can assume that a **threshold** exercise policy is adopted: there is a stock price $s_j (\geq K)$ associated with t_j such that the option is exercised if (and only if) $S_{t_j^-} > s_j$. The European call option can be thought of the special case of $s_j = \infty$ for all $j \leq \eta(T)$.

The sample performance can be written as

$$J_T = e^{-rT} \left(\sum_{i=1}^{\eta(T)} \left[\prod_{j=1}^{i-1} \mathbf{1}\{S_{t_j^-} \leq s_j\} \right] \mathbf{1}\{S_{t_i^-} > s_i\} \cdot \left(S_{t_i^-} - K\right) e^{r(T-t_i)} + \prod_{j=1}^{\eta(T)} \mathbf{1}\{S_{t_j^-} \leq s_j\} (S_T - K)^+ \right).$$

The option pricing problem can then be viewed as an optimization problem, whereby the option value is the point at which the expected return $E[J_T(\theta)]$ is maximized with respect to the vector of threshold parameters $\theta = s_1, \ldots, s_{\eta(T)}$. To determine the optimal setting of the threshold parameters, we incorporate a gradient estimate $\partial E[J_T]/\partial\theta$ into a stochastic approximation algorithm, where the "best guess" of the optimal setting is updated iteratively via (5), where $g(\theta) = \nabla_\theta E[J_T(\theta)]$. The basic underlying assumption of the stochastic approximation algorithm is that the original problem can be solved by finding the zero of the gradient, i.e., by finding θ_*, the optimal exercise threshold level, such that $g(\theta_*) = 0$. Of course, in practice, this may lead only to local optimality. Since the problem is a maximization problem, the stochastic approximation iteration (5) is the positive version of the recursion. The necessary unbiased gradient estimator is derived by using SPA in Fu and Hu (1994), and then incorporated into the SA algorithm. Simulation results reported there indicate that the algorithm converges quite quickly, using much less effort than is needed to simply estimate an option payoff to within a penny. In other words, the additional effort needed to estimate an American option using Monte Carlo simulation over what was needed to estimate a European option was negligible.

7 OTHER RECENT WORK

Other approaches, some proposed recently, include:
- Utilizing non-gradient-based algorithms such as pattern search methods and random search methods, e.g., Jacobson and Schruben (1989);
- Using *each sample* to derive an entire performance curve and optimize the resulting curve using deterministic methods, e.g., Healy and Schruben (1991);
- Combining techniques, e.g., the proposed Gradient Surface Method (GSM) by Ho et al. (1992a) combines RSM and SA;
- Replacing cardinal optimization with ordinal optimization (Ho et al. 1992b), i.e., instead of trying to find the best in a possibly uncountable infinite state space, just try to find better "satisficing" solutions;
- Employing massively parallel simulation for exploring a response surface in parallel.

For more in-depth reviews, the interested reader is referred to Fu (1994) — which expands upon most of the topics touched on here — Jacobson and Schruben (1989), and Safizadeh (1990). Together these contain a comprehensive literature review, the first concentrating on gradient-based methods, and the latter two emphasizing response surface methods.

REFERENCES

Azadivar, F. and J.J. Talavage. 1980. Optimization of stochastic simulation-models, *Mathematics and Computers in Simulation* **22**:231-241.

Bashyam, S. and M.C. Fu. 1994. Optimization of (s,S) inventory systems with random lead times and a service level constraint, submitted for publication.

Cassandras, C.G. 1993. *Discrete Event Systems: Modeling and Performance Analysis*, Irwin.

Fu, M.C. 1990. Convergence of a stochastic approximation algorithm for the GI/G/1 queue using infinitesimal perturbation analysis, *Journal of Optimization Theory and Applications* **65**: 149-160.

Fu, M.C. 1994. Sample path derivatives for (s,S) inventory systems. *Oper. Res.* **42**: 351-364.

Fu, M.C. 1994. Optimization using simulation: a review. to appear in *Ann. Oper. Res.*

Fu, M.C. and K. Healy. 1992. Simulation optimization of (s,S) inventory systems, *Proc. Winter Simulation Conference*: 506-514.

Fu, M.C. and J.Q. Hu. 1994. Sensitivity analysis for Monte Carlo simulation of option pricing, submitted for publication

Glasserman, P. 1991. *Gradient Estimation Via Perturbation Analysis.* Kluwer Academic.

Goldsman, D., B. L. Nelson and B. Schmeiser. 1991. Methods for selecting the best system, *Proc. Winter Simulation Conference*: 177-186.

Healy, K. and L.W. Schruben. 1991. Retrospective simulation response optimization. *Proc. Winter Simulation Conference*: 901-906.

Hill, S.D. and M.C. Fu. 1994. Simulation optimization via simultaneous perturbation stochastic approximation, *Proc. Winter Simulation Conference.*

Ho, Y. C. and X. R. Cao. 1983. Optimization and perturbation analysis of queueing networks, *Journ. Optimization Theory and App.* **40**: 559-582.

Ho, Y.C. and X.R. Cao. 1991. *Perturbation Analysis and Discrete Event Dynamic Systems.* Kluwer.

Ho, Y.C., L. Shi, L. Dai and W. B. Gong. 1992. Optimizing discrete event systems via the gradient surface method, *Discrete-Event Dynamic Systems: Theory and Applications*: 99-120.

Ho, Y.C., R. Sreenevas and P. Vakili. 1992. Ordinal optimization of DEDS, *Discrete-Event Dynamic Systems: Theory and Applications*: 61-88.

Hsu, J. C. and B.L. Nelson. 1988. Optimization over a finite number of system designs with one-stage sampling and multiple comparisons with the best, *Proc. Winter Simulation Conference*: 451-457.

Hull, J.C. 1993. *Options, Futures, and Other Derivative Securities,* 2nd edition. Prentice Hall.

Jacobson, S. H. 1993. Variance and bias reduction techniques for harmonic gradient estimators, *Applied Mathematics and Computation*: 153-186.

Jacobson, S. H. and L.W. Schruben. 1989. A review of techniques for simulation optimization, *Operations Research Letters* **8**: 1-9.

Kushner, H. J. and D. C. Clark. 1978. *Stochastic Approximation Methods for Constrained and Unconstrained Systems.* Springer-Verlag.

Law, A. M. and W. D. Kelton. 1991. *Simulation Modeling and Analysis*, 2nd edition. McGraw-Hill.

L'Ecuyer, P., N. Giroux, and P. W. Glynn. 1994. Stochastic optimization by simulation: numerical experiments with a simple queue in steady-state, *Management Science.* to appear.

Rubinstein, R.Y. and A. Shapiro. 1993. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method.* Wiley.

Safizadeh, M. H. 1990. Optimization in simulation: current issues and the future outlook. *Naval Research Logistics* **37**: 807-825.

Suri, R. and M. Zazanis. 1988. Perturbation analysis gives strongly consistent sensitivity estimates for the M/G/1 queue, *Management Sci.* **34**: 39-64.

Yang, W.N. and B.L. Nelson. 1991. Using common random numbers and control variates in multiple-comparison procedures, *Oper. Res.* **39**: 583-591.

AUTHOR BIOGRAPHY

MICHAEL C. FU is an Associate Professor of Management Science & Statistics in the College of Business and Management, with a joint appointment in the Institute for Systems Research, at the University of Maryland at College Park. He received S.B. and S.M. degrees in electrical engineering and an S.B. degree in mathematics from MIT in 1985, and M.S and Ph.D. degrees in applied mathematics from Harvard University in 1986 and 1989, respectively. His research interests include stochastic derivative estimation and stochastic optimization, particularly with applications towards manufacturing systems, inventory control, and the pricing of financial derivatives. He teaches courses in applied probability, stochastic processes, discrete-event simulation, and operations management. He is a member of **IEEE** and **ORSA**.