Annals of Operations Research, Vol.53, 199-248, 1994

OPTIMIZATION VIA SIMULATION: A REVIEW

Michael C. FU
University of Maryland, College Park, Maryland 20742 USA

November 1992; revised April 1993

Abstract

We review techniques for optimizing stochastic discrete-event systems via simulation. We discuss both the discrete parameter case and the continuous parameter case, but concentrate on the latter which has dominated most of the recent research in the area. For the discrete parameter case, we focus on the techniques for optimization from a finite set: multiple-comparison procedures and ranking-and-selection procedures. For the continuous parameter case, we focus on gradient-based methods, including perturbation analysis, the likelihood ratio method, and frequency domain experimentation. For illustrative purposes, we compare and contrast the implementation of the techniques for some simple discrete-event systems such as the (s, S) inventory system and the $GI/G/1$ queue. Finally, we speculate on future directions for the field, particularly in the context of the rapid advances being made in parallel computing.

Keywords: simulation optimization, discrete-event simulation, stochastic optimization, stochastic approximation, response surface methodology.

Short running title: Optimization via Simulation

1. Introduction

Optimization in the field of operations research has long been synonymous with the area of mathematical programming. With the rapid advances made in computational efficiency, it is now routine for optimization packages to solve models with thousands of variables. The study in [9] considered a crew scheduling problem with over 12 million variables. On the other hand, the computational advances have also led to increased efforts in the area of optimization for stochastic discrete-event systems, where the complexities necessitate simulation in order to estimate performance. In such cases, one can only observe *samples* through the simulation process.

In this paper, we review the main areas of optimization via simulation and touch on some of the more recent directions that researchers have taken in exploiting computational advances such as parallel computing. The exposition is tutorial in nature, and the target audience includes both those in the operations research and management science community having some familiarity with

discrete-event simulation (especially graduate students), as well as researchers and practitioners specializing in discrete-event simulation.

Throughout this paper, when we use the term *simulation*, we will mean simulation of **stochastic discrete-event systems**, the main features distinguishing these systems being

- the randomness in the model (versus say a system of differential equations which has no randomness but is analytically intractable nonetheless), and
- a "physical" state of the system which experiences "jumps" at discrete points in time upon the occurrences of events.

Such system models can be used to model a very rich set of real-world systems such as manufacturing systems (automated, job shop, and others); computer networks (e.g., LANs); data communications networks (e.g., Internet); airport operations (e.g., runway traffic, baggage handling and transport); and command, control, and communications intelligence (C^3I) operations. The chief analytical tools available to analyze discrete-event systems include the traditional OR areas of queueing theory and inventory theory. A good sampling of other useful models for analyzing discrete-event systems can be found in the Proceedings of the IEEE special issue on Discrete-Event Dynamic Systems [48]. Prior to any attempt to do optimization via simulation, the analyzer would be well-advised to

- use analytical tools to exploit any special *structural* properties inherent in the system of interest which can often reduce the search space by orders of magnitude, and
- do some rough-cut optimization via analytically tractable models.

For expository purposes, we will often illustrate the various techniques for two simple systems: the $GI/G/1$ queue and the (s, S) inventory system. The purpose is not to promote one technique over another, but to give the reader a sense as to how one goes about implementing each of the various techniques in the two most common types of discrete-event systems: queues and inventories. In fact, as we shall see, many of the so-called "competing" techniques can and should be used in conjunction with each other.

The rest of the paper is organized as follows. In Section 2, a more precise definition of the general problem of interest is given, a brief classification of optimization problems described, and some general comments on convergence rates mentioned. The major techniques are described in Sections 3, 4, 5, and 6. In Section 3, we describe finite parameter space procedures based on

multiple comparisons and ranking and selection. Further details on these topics can be found in the review paper by Schmeiser on Output Analysis also contained in this volume. In Section 4, we discuss perhaps the most generally applicable of techniques for simulation optimization, response surface methodology. In Section 5, we focus on gradient-based stochastic approximation algorithms, which have dominated much of the recent research activity in the area of simulation optimization. Included in our discussion are the gradient estimation techniques of perturbation analysis and likelihood ratio, and to a lesser extent, the use of frequency domain experiments. In Section 6, we touch on other methods, including some recent proposals on optimizing individual samples and techniques exploiting advances in parallel computing. Conclusions and future directions are discussed in Section 7.

We end this introductory section with a small disclaimer. In the exposition, it is inevitable that some of the personal research leanings of the author come through, in the form of choosing to emphasize more of the details of one area over another, e.g., Section 5 may be disproportionately long. However, it is hoped that at least enough content is provided to give a flavor of all the major methodologies available for conducting simulation optimization of discrete-event systems, and to give an idea as to current research directions in the fields. It is in this vein that many of the references have been chosen; thus we do not claim to provide a comprehensive bibliography on the literature on simulation optimization, but the choices should suffice to lead one to the sources where such information may be obtained. For example, two reviews which are not intended to be as tutorial in nature as this paper [67, 105], contain a more comprehensive list of the literature on response surface methodology applications to simulation optimization. An older review [21] considers simulation optimization in a broader context, not specifically discrete-event systems. (At that time, none of the gradient estimation techniques were even on the discrete-event simulation scene yet!) Abridged updates of current research in the area can often be found in the state-of-the-art reviews or advanced tutorials of the annual *Proceedings of the Winter Simulation Conference*.

2. The Problem Setting

The general problem setting is the following parametric optimization problem:

$$\min_{\theta \in \Theta} J(\theta), \tag{2.1}$$

where $J(\theta) = E[L(\theta, \omega)]$ is the performance measure of interest, $L(\theta, \omega)$ will be called 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, either defined explicitly (as in our examples below) or implicitly (as in mathematical programming formulations), but assumed to be a closed set. Let us also define the optimum by

$$\theta_* = \arg\min_{\theta\in\Theta} J(\theta). \tag{2.2}$$

In the experimental design literature, the performance measure is usually referred to as the *response* and the parameters as *factors*. In this paper, we will consider only the single response problem. If J is known explicitly, then analytical techniques such as mathematical programming can usually be applied. We hedge a little on the "usually" because there are cases where simulation may still be preferred over analytical techniques due to computational complexity. One such class of problems may be very large (state space) Markov chains which can for example be used to model closed queueing networks. The difficulty arises from the computation of the normalization constant over a state space which increases combinatorially with the number of stations in the network, whereas in simulation, the increase in computation is probably no worse than linear in the number of stations in the network. In such cases, it has been argued [96] that simulation may be preferable to the "exact" method.

Our prototypical examples will be the following two systems:

- a $GI/G/1$ queue,
- an (s, S) inventory system.

These systems are usually used as teaching paradigms in most discrete-event simulation textbooks (e.g., [76, 5]), and will serve to illustrate many of the techniques in the literature. For our two examples, we consider the following problems.

Example 1. For a GI/G/1 queue, find the mean service time of the server that minimizes the sum of expected mean time in system over a given number of customers served and a cost on server speed:

$$\min_{\theta\in\Theta} c_0 E\left[\frac{1}{N}\sum_{i=1}^{N} T_i\right] + c_1/\theta, \tag{2.3}$$

T_i = time in system for the ith customer,

N = number of customers served,

$$
\begin{aligned}
\theta &= 1/\mu \quad = \quad \text{mean service time,} \\
\Theta &= [\delta, 1/\lambda - \delta], \\
A_i &= \text{interarrival time between the } (i-1)\text{st and } i\text{th customer,} \\
X_i &= \text{service time for the } i\text{th customer,} \\
f_1, F_1 &= \text{density and distribution functions of interarrival times,} \\
f_2, F_2 &= \text{density and distribution functions of service times,}
\end{aligned}
$$

where λ is the arrival rate, c_0 and c_1 are given costs, and δ is some small number.

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. We will consider the zero order lead time, periodic review case, where inventory costs are computed only at review epochs, and we explicitly transform our integral into a sum. Furthermore, without loss of generality, we use $1, 2, ..., T$ as the review epochs. For this system, the (s, S) ordering policy is to order up to S if the inventory level falls below s; otherwise, no order is placed.

$$
\min_{\theta \in \Theta} \frac{1}{T} E\left[\sum_{i=1}^{T} C(X_i)\right], \tag{2.4}
$$

$$
\begin{aligned}
X_i &= \text{inventory level (and position) at review epoch } i, \\
T &= \text{number of periods in the horizon,} \\
C(x) &= hx^+ + px^- + I\{x < s\}[K + c(S - x)], \\
q &= S - s, \\
\theta &= (s, q), \\
\Theta &= [0, s_{\max}] \times [0, q_{\max}], \\
D_i &= \text{demand in period } i, \\
g, G &= \text{density and distribution functions of single period demand,}
\end{aligned}
$$

where h, p, K, and c are the holding, backlogging, order set-up, and order per-unit costs, respectively, $x^+ = \max(0, x), x^- = \max(0, -x)$, $I\{\cdot\}$ denotes the indicator function, and $s_{\max}$ and $q_{\max}$ are upper bounds.

Oftentimes, we will be interested in steady-state behavior, when $N \to \infty$ or when $T \to \infty$. However, this is not really crucial, because even in the finite horizon transient case, we can always consider the optimal parameter values for a given set of initial conditions.

2.1. CLASSIFICATION

The optimization problem described by (2.1) is a static one as opposed to the dynamic nature of the following problem:

$$\min_{U(\cdot)} E\left[\frac{1}{T}\int_0^T C(X(t,\omega),U(X,t))dt\right], \tag{2.5}$$

$$\begin{aligned} \text{where } T &= \text{time horizon of interest,} \\ X(t,\omega) &= \text{state of the system at epoch } t, \\ U(x,t) &= \text{control in state } x \text{ at epoch } t, \\ C(x,u) &= \text{cost function for state } x \text{ and control } u. \end{aligned}$$

The time horizon could be fixed or a function of the state and/or events (a stopping time), e.g., the completion of a number of services in a queueing system. Furthermore, although not indicated, there are usually additional constraints, either explicitly or implicitly defined, on the control. In general, this problem is a very difficult one, one of *optimal control*, with complications due to the dynamics and stochastic nature of the system, and the presence of time. In principle, such problems can be attacked by the methods of dynamic programming, the solution of which is characterized by the Hamilton-Jacobi-Bellman partial differential equations. Since the solution is in general a functional, the solution space is infinite dimensional. In discrete time, the theory of Markov decision processes can often be applied.

Our parameter optimization problem can be thought of as the special case of a "stationary" control:

$$U(x,t) = \theta \quad \text{for all} \quad x,t,$$

resulting in a finite-dimensional problem. This relationship is depicted in fig. 1. We note that the classification scheme in [38] defines optimization in a broader sense, instead viewing the control problem as the infinite-dimensional case of optimization.

With optimization problems as a special case of optimal control problems in our classification scheme, optimization problems themselves can be broken down further into two quite distinct categories, based on the nature of the decision space: continuous or discrete. The distinction is important because the tools that are presently used to attack the two categories of problems are quite different. The discrete case could be further subdivided into finite or infinite parameter space. Fig. 1 depicts this classification scheme and includes some associated techniques that can

be applied to each class of problems. In addition, sometimes it is useful to divide the discrete case into the domains of ordered and unordered, a subdivision intended to include the dichotomy between quantitative and qualitative decisions. For example, in the queueing context, parameters of service time distributions may be continuous, whereas the number of servers may be discrete but ordered. Another possible decision, though, may be the queue discipline, which is unordered. (In theory, of course, a decision space could be a hybrid of the above combinations. For example, for the queue discipline selection, besides choosing between FCFS and LCFS, there could also be queue disciplines of the sort that all customers with service times under θ minutes have priority, where θ is a continuous parameter.) The distinction between ordered and unordered can become important in the application of certain procedures such as simulated annealing and genetic algorithms, where a neighborhood structure of some type must be defined.

The case of the (s, S) inventory system is an interesting one. For gradient-based optimization, it is more advantageous to consider a continuous domain than a discrete one, unless the discrete domain is small in number in which case ranking and selection and multiple comparison procedures can be fruitfully applied. However, the analytical methods employed to solve inventory problems based on dynamic programming methodologies assume a countable parameter space.

2.2. CONVERGENCE RATES

As noted in [40], the best possible convergence rate with "pure" stochastic optimization algorithms is generally of the order $n^{-1/2}$, where n represents (roughly) the computational effort. The result stems from the general availability of a central limit theorem characterization of the limiting properties of a best possible estimator, say θ_n, for the optimum:

$$\sqrt{n}(\theta_n - \theta_*) \Longrightarrow \sigma N(0, 1),$$

where σ^2 is the asymptotic variance of the estimate, "$\Longrightarrow$" indicates convergence in distribution, and $N(0, 1)$ indicates the standardized normal distribution. This means that one must expend 100 times more effort to extract an additional decimal point of accuracy in estimating the optimum. However, we make the following remarks in interpreting this rate:

1. It is an *asymptotic* convergence rate. Performance of the algorithm in the beginning may be better (or worse), i.e., we may have a "jump start" effect. Furthermore, in the short-term, the magnitude of the variance term σ^2 may be as important as the convergence rate.

2. This is also the best convergence rate obtainable for the estimation of any (non-trivial)

output random variable from a simulation when i.i.d. samples are taken, i.e., without optimization. Thus, asymptotically, optimization via simulation has the same rate of efficiency (or inefficiency) that estimation via simulation has, the asymptotic rate emerging from the underlying stochastic nature of the system and not from other types of complexity. Practically speaking, this means the asymptotic convergence *rate* for optimizing a simple GI/G/1 queue is the same as for optimizing a network of a million queues with complicated routing and multi-class customers.

3. The rate assumes a linear scaling of computational effort, in a sequential environment. Such measurement of computational effort may not be appropriate in a parallel environment, where massive speed-ups may not be "assessed" as part of the computational effort. For example, if 64K processors are available (simultaneously) on a computational platform, but one algorithm can only use 10% of them, whereas another is able to utilize all of them, the traditional measure would assign a computational "effort" of 10 times more to the latter if the run times are identical. The parallelizability of the algorithm resulting in a factor of 10 speed-up should somehow be reflected in the convergence rate.

The overall message is that simulation for optimization is generally inefficient, but then so is simulation itself, simply for the purposes of modeling and analysis of discrete-event systems. In fact, parallel simulation may make speed-ups for optimization purposes much easier to realize than speed-ups in the simulation process itself (via distributed algorithms). Thus, we reiterate that *before* using simulation at all (as well as during and afterwards), analytical models and methods should be employed to gain insight.

In closing this introduction, we note that whereas linear programming can lay claim to routinely solving problems with thousands of variables, at this point, simulation optimization cannot lay claim to such success stories. It is hoped that the recent surge of interest, coupled with the advent of parallel simulation, will soon change this situation.

3. 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. For example, in the (s, S) inventory case, it may be that order quantities are restricted by the supplier, and through rough analytical

calculations, there are good estimates of upper and lower bounds on the optimal values. In these cases, where the number of choices is not too large, a number of statistical procedures can be applied. These fall into two major categories: ranking and selection and multiple comparisons. 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.

The field of ranking and selection is flourishing in the statistics community, with the edited volume [107] dedicated to the 65th birthday of the "father" of the field, Robert Bechhofer, counting well over 700 published papers at that time. (Note that the statistics community may classify multiple-comparisons procedures as a subset of ranking and selection.) The procedures in ranking and selection and multiple comparisons that we discuss in this section are taken chiefly from two sources [76, 130]. The purpose of this section will be to sketch the main ideas of the procedures and their application to simulation optimization over a finite set. More technical details can be found in [113] of this volume. A simple example comparing and contrasting different techniques can also be found in [42].

Again, we wish to solve the parametric optimization problem:

$$\min_{\theta \in \Theta} J(\theta), \tag{3.1}$$

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, which we will denote by

$$\widehat{J}_i = \bar{L}_i = \frac{1}{n} \sum_{j=1}^{n} L_{ij}. \tag{3.2}$$

In multiple-comparisons procedures, the idea will be to run a number of replications to make some inferences on the performance measure of interest by way of confidence intervals. A priori, no decision can be guaranteed. If the confidence intervals are not tight enough to make conclusive statements, then an estimate is made of the number of replications that would be needed to obtain the confidence widths at a desired level and more replications subsequently run. Alternatively, a "pilot" experiment can be used simply for the purposes of estimating the variance and hence the number of replications likely to be necessary to make useful inferences.

Ranking-and-selection procedures, on the other hand, are sequential by construction, for the most part consisting of two stages. In some ways, they formalize the above non-sequential procedure, because the objective is to make a decision based on some criteria. The concept of a *correct selection* is used, and the two-stage procedure guarantees a correct selection at a specified level of confidence.

3.1. MULTIPLE-COMPARISONS PROCEDURES

Procedures based on multiple comparisons are of very basic importance in statistical inference, since almost all applied statistical techniques require comparisons. The text by Hochberg and Tamhane [60] summarizes much of the work up to 1986. We restrict our discussion here to work directly applicable to discrete-event simulation. Like most statistical techniques, the two major assumptions underlying the procedures are *independence* and *normality*. Although the latter may be satisfied for some performance measures of interest (perhaps through batching), or be somewhat robust from small deviations away from it, the former directly conflicts with some of the advantages of discrete-event simulation, where so much is under the modeler's control. In particular, the independence assumption would rule out the implementation of powerful variance reduction techniques such as common random numbers and control variates.

Yang and Nelson [130] have developed procedures to overcome these difficulties, i.e., to allow the use of common random numbers (see [35] for guidelines on its use). Research with the particular application of simulation in mind has immense practical benefits, since the "usual" sort of experimentation assumed by the developers of statistical techniques for experiments would find independence a natural – if not also very convenient – assumption.

We will sketch the main ideas of three multiple-comparisons procedures, all of which assume some form of normality:

- The "brute force" paired-t, Bonferroni, all-pairwise comparisons approach that works particularly well when common random numbers apply.
- An all-pairwise multiple comparisons (MCA) approach that requires both independence and equal variances, and a version that allows the use of control variates and common random numbers, but still requires equal variances for the underlying performance measure of interest.
- A multiple comparisons with the best (MCB) approach that requires both independence and

equal variances, and again a version which allows the use of control variates and common random numbers, but again requires equal variances for the underlying performance measure of interest. In general, one would expect that MCB perform the best for optimization purposes, since fewer comparisons are made.

The idea of the "brute force" approach is quite simple:

1. Define the difference estimate for each possible i-j pair of replications:

$$Z_{ijl} = L_{il} - L_{jl}, i < j,$$

where the subscript l indicates the replication number, and we assume the use of common random numbers to reduce variance.

2. Then, simply form the usual $(1-\alpha)100\%$ confidence intervals for each difference:

$$\bar{Z}_{ij} \pm t_{n-1}^{\alpha/2} s_{ij}/\sqrt{n},$$

$$\bar{Z}_{ij} = \frac{1}{n}\sum_{l=1}^{n} Z_{ijl},$$

$$s_{ij}^2 = \frac{1}{n-1}\sum_{l=1}^{n}(Z_{ijl} - \bar{Z}_{ij})^2,$$

where $\bar{Z}_{ij}$ and s_{ij}^2 are the sample mean and variance, respectively, for the ij-difference of the performance measure, and t_n^α is the upper α quantile of the student-t distribution with n degrees of freedom. Note that $\bar{Z}_{ij}$ is simply $\widehat{J}_i - \widehat{J}_j$ under common random numbers.

3. Using the Bonferroni inequality, the *overall* confidence level is reduced by a factor of K-choose-2 i.e., it is only guaranteed to do no worse than $(1-\alpha K(K-1)/2)100\%$. Thus, for the comparison of four systems ($K = 4$), if individual confidence levels were each chosen at the 95% level ($\alpha = 0.05$), then the overall confidence level has a lower bound of 70%.

After forming all 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. This procedure is essentially the idea of the "Interactive Analysis" procedure in [42]. Strictly speaking, the confidence level statements after subsequent sets are (non-trivially) dependent on the first set of

replications, but in practice this is usually ignored. Also, note because of the drastic reduction in the confidence level, the original α need be very small to get useful inference.

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:

$$\left(\widehat{J}_i - \widehat{J}_j\right) \pm r^{\alpha}_{K,K(n-1)} s/\sqrt{n}, i < j,$$

$$s^2 = \frac{1}{K(n-1)} \sum_{i=1}^{K} \sum_{j=1}^{n} (L_{ij} - \bar{L}_i)^2,$$

here s^2 is the *pooled* sample variance and $r^{\alpha}_{K,K(n-1)}$ is the upper α quantile of the studentized range distribution with K systems and $K(n-1)$ degrees of freedom (found in tables of Hochberg and Tamhane).

Thus, the difference between brute force and MCA is that MCA obtains an overall simultaneous confidence level, with the same confidence half-widths for each pairwise difference, whereas the brute force approach obtains a different confidence half-width for each pairwise difference and uses Bonferroni to obtain a bound on the overall confidence. However, the original version of MCA also requires independence between the replications. Based on control variate estimators, a variation on MCA which would allow the use of common random numbers in a certain form was derived in [130]:

$$\left(\widehat{J}'_i - \widehat{J}'_j\right) \pm r^{\alpha}_{K,K(n-q-1)} s'/\sqrt{n}, i < j,$$

where $\widehat{J}'_i$ denotes the control variate version of $\widehat{J}_i$ The other difference between this interval estimate and the one above is that the degrees of freedom are reduced by a function of the number of control variates used, because covariance estimates are required, and s' is an altered estimate of the pooled variance which involves estimating the covariance matrix of the control variates (see [130] for details).

As an example, in the (s, S) inventory system example, we could use the sample mean of demands as a control variate, with the sequence of demands (the only stochastic component in the system) made identical over the different $\theta = (s, S)$ values by common random numbers. Similarly, in the GI/G/1 queue example, we could use the sample mean of interarrival times as a control variate, with the interarrival times made identical across systems by common random numbers. Note, that in both cases, we would have problems applying the procedure if the parameters appeared in the distributions, e.g., in the demand distribution in the (s, S) inventory system,

and in the interarrival time distribution (simultaneously with the service time distribution) in the GI/G/1 queue.

The intent of the multiple comparisons with the best (MCB) procedure 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 [62]:

1. Form confidence intervals for each choice with the *best of the rest*, where the best of the rest will be the overall best except when the confidence interval with the best is being formed, in which case the pairing is done with the overall second best. There will be K intervals of the form ([61] actually considers the max form)

$$\left[-\left(\widehat{J}_i - \min_{j \neq i} \widehat{J}_j - d^{\alpha}_{K-1,K(n-1)} s \sqrt{2/n} \right)^{-}, \left(\widehat{J}_i - \min_{j \neq i} \widehat{J}_j + d^{\alpha}_{K-1,K(n-1)} s \sqrt{2/n} \right)^{+}, \right],$$

   where $x^+ = \max(0, x), x^- = \max(0, -x)$, and $d^{\alpha}_{K-1,K(n-1)}$ is the upper α quantile of a random variable that is the maximum of $K-1$ equally correlated multivariate-t random variables with correlation 0.5 and $K(n-1)$ degrees of freedom (found in tables of Hochberg and Tamhane).

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. The two situations are illustrated in figs. 2 and 3. In practice, the pooled variance could be used to estimate the additional number of replications needed to make a final determination.

The control variate version derived by Yang and Nelson is given by:

$$\left[-\left(\widehat{J}'_i - \min_{j \neq i} \widehat{J}'_j - d^{\alpha}_{K-1,K(n-q-1)} s' \sqrt{2/n} \right)^{-}, \left(\widehat{J}'_i - \min_{j \neq i} \widehat{J}'_j + d^{\alpha}_{K-1,K(n-q-1)} s' \sqrt{2/n} \right)^{+}, \right],$$

where $\widehat{J}'_i$ and s' have the same meaning as in the MCA procedure.

3.2. RANKING AND SELECTION

We turn now to the ranking-and-selection procedures. In general, two approaches have been taken:

- indifference zone [6],
- subset selection [44].

The method of Dudewicz and Dalal [19] falls into the indifference-zone approach. It has two distinct advantages over other proposed methods 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 (cf. [76]):

1. Take a first-stage set of $n_0 \geq 2$ replications for each of the K different parameter settings to get the first-stage sample means and sample variances for each $J(\lambda_i), i = 1, ..., K$.

2. Use the first-stage sample variances to determine (based on the various parameters K, P^*, n_0) the number of second-stage replications needed for each $\lambda_i, i = 1, ..., K$ (the number of replications usually unequal for each, depending on the variance estimate).

3. Take the second-stage set of replications to get the second-stage sample means.

4. Take a weighted (dependent on K, P^*, n_0, the first-stage sample variance, and the number of second-stage replications) average of the first-stage and second-stage sample means.

5. Choose the λ_i with the smallest weighted average estimate of $J(\lambda_i)$.

Similarly, a subset-selection approach derived in [118] also allows the two critical features of unequal and unknown variances. The procedure guarantees that with probability at least P^* the selected *subset* of at most m (specified by user) λ_i's will contain *at least one* λ_i such that $J(\lambda_i)$ is within δ of the optimal value $J(\theta_*)$, where δ again represents the "indifference zone," i.e., $P\{J(\lambda_i) - J(\theta_*) < \delta\} \geq P^*$. Algorithmically, the procedure closely resembles the procedure of Dudewicz and Dalal, with the notable exception being the last step, where instead of selecting a single λ_i, all λ_i such that the weighted average estimate of $J(\lambda_i)$ is within $d = d(K, m, P^*, n_0)$ of the best are selected, up to a maximum of the best m. (Note that the procedure in Sullivan and Wilson actually considers the maximization problem.)

Although both the indifference-zone and subset-selection approaches fall under a general form of ranking and selection, philosophically they complement each other. Subset-selection approaches would seem to be most useful when the number of choices is quite large, the goal being

to reduce the large number to a manageable random but bounded smaller number. Indifference-zone approaches could then be used to select a single choice that is within a prespecified difference from the true optimum.

The major disadvantage of all the ranking-and-selection procedures are the requirement of independence over competing designs, which precludes the use of most variance reduction techniques in discrete-event simulation such as common random numbers. The assumption of normality can be more easily handled through batching techniques, under which central limit theorem effects can be invoked. In any case, ranking-and-selection and multiple-comparisons procedures are very powerful and easy-to-use tools for optimization when the parameter set is finite.

4. Response Surface Methodology

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. In the context of optimization, it usually takes the form of a sequential procedure 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. However, whereas the discrete-event simulation community is moving towards the goal of optimization, a recent survey of RSM research by the statistics community indicates that RSM statistics researchers may be shifting their emphasis [87]:

> "There is no question that the motivation for the work by Box and his coauthors was the general and perhaps ancient problem of planning and analyzing experiments in a search for desirable conditions on a set of controllable (or design) variables, 'desirable' often being defined more strictly as those conditions that give rise to optimum response. In recent years, much emphasis has been placed by practitioners not on finding optimum response but on finding regions where there is demonstrated *improvement* in response over that achieved by current operating conditions."

In the formal application of RSM for optimization and for design of experiments in general, one of the most important steps is factor screening, the initial identification of the "important" parameters, those factors that have the greatest influence on the response. However, in our discussion of optimization of discrete-event simulation models, we will assume that this has already

been determined. In most discrete-event system applications, this is usually the case, since there are underlying analytic models which can give a rough idea as to the influence of various parameters. For example, in manufacturing systems and telecommunications networks, the analyst knows from queueing network models which routing probabilities and service times have an effect on the performance measures of interest. RSM procedures usually presuppose a more "black box" approach to the problem, so it is unclear a priori which factors are of importance at all.

The literature on RSM is vast. Useful books on the subject include the "classics" such as [88, 18] and newer texts such as [10, 72, 74], the latter being oriented towards simulation. We will briefly outline the general approach in the context of discrete-event simulation. Specific applications with a more detailed discussion, including a more comprehensive list of references, can be found in [67, 105].

The application of RSM to simulation optimization falls into two main categories:

- metamodels,
- sequential procedures.

The use of the former for optimization would imply simply dividing the problem into two separate problems of estimation and optimization. Simulation is used to fit a "global" response curve called the metamodel – a complete functional relationship between the performance measure and the parameters of interest – which is then treated as a deterministic function and optimized using applicable deterministic procedures. This procedure is quite "brute force" in nature, as it will inevitably end up requiring the largest amount of simulation effort in order to characterize the response curve over the entire domain of feasibility. Since the simulation and optimization are decoupled, these problems are tackled separately. One must have quite a bit of faith in the accuracy of the metamodel that is developed in order for this approach to be accurate. In many cases of practical interest, it may be too much to hope for, and in other cases it may be wasteful, in the sense that in order to get such a complete response, information may need to be gathered on portions of the response which may be of little interest to most users. On the other hand, it is also possible to use a metamodel as the final step of an RSM procedure. Once the region of most interest has been determined through exploratory RSM procedures, a more detailed study can be done to get a more accurate picture of the response. Optimization is in fact just one potential use of a metamodel; if that is the primary goal, then this route may not be the most practical. The statistical techniques used for curve fitting and the choice of experimental designs are the critical

features in the metamodeling procedure. Sargent [108] discusses some current research issues in using simulation for constructing metamodels. One of the original proposed uses of frequency domain experimentation, to be discussed in the next section in the context of gradient estimation, was for such metamodels [111]. An extensive discussion of the statistical issues involved can be found in [74].

In the literature, optimization via simulation using RSM usually refers to the second category. 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 – oftentimes the center of the subregion currently being explored – would "represent" the current "best" values of the parameter. The basic algorithm can be described as follows:

- **Phase I**

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

  $$\theta_{n+1} = \theta_n - a_n \nabla J_n, \tag{4.1}$$

  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 representing the 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, by which the interaction effects become larger than the main effects.

- **Phase II**

  A quadratic response surface is fitted using more detailed second-order experimental designs, and then the optimum 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. One will also notice that the iterative algorithm in Phase I is identical in form to the stochastic approximation algorithm to be described in the next section. However, here θ is "representative" of a *subregion* as opposed to the actual "best" single *point* value of the parameter.

In both the Phase I and Phase II models, there are two sources of error in the fit: bias and variance. The bias is due to the inadequacy of the polynomial fit, e.g., a linear approximation to a function which is of course not linear, whereas the variance is simply the sampling error. Usually, bias error is less critical in Phase I, because presumably one may be far from the optimum so the goal is just to get an indication of where potential improvements lie. When Phase II is reached, reduction of bias becomes a more important consideration.

In Phase I, an orthogonal first-order, factorial design is usually employed, oftentimes with a center point (simplex designs are also sometimes used to reduced the number of points in the design). If the parameters have been standardized appropriately such that "+1" represents the "high" value and "−1" represents the "low" value, then a 2^p factorial design consists of all possible settings of the parameters to $+1$ and -1, of which there are 2^p points. The center point would correspond to $(0, \cdots, 0)$.

In Phase II, a denser design is used to get a better fit, e.g., a central composite design, which is the central factorial design with star points added, i.e., by adding the following $2p$ points:

$$\pm\alpha e_i, i = 1, ...p,$$

where e_i denotes the unit vector in the ith direction, i.e., with 1 in the ith place and 0 otherwise. Thus, $\pm\alpha e_i$ is the vector with $\pm\alpha$ in the ith place and 0 otherwise. The value of α used depends on the desired tradeoff between variance and bias.

We describe the linear model in a little more detail. With

$$\begin{aligned} p &= \#\text{ parameters}, \\ m &= \#\text{ observation points}, \\ \theta &= (\theta^{(1)} \quad \cdots \quad \theta^{(p)})^T, \\ \beta &= (\beta_1 \quad \cdots \quad \beta_p)^T, \end{aligned}$$

where the "T" superscript denotes transpose, the first-order model is given by

$$J(\theta) = \beta_0 + \sum_{i=1}^{p} \beta_i \theta^{(i)} = \beta_0 + \theta^T \beta. \tag{4.2}$$

Defining the experimental matrix for the m settings of the parameters by

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & & & & \\ 1 & x_{m1} & x_{m2} & \cdots & x_{mp} \end{bmatrix} \tag{4.3}$$

where

$$x_{ij} = \text{the } i\text{th setting of } \theta^{(j)},$$

and letting

$$\begin{aligned} \widehat{J}_i &= \text{estimate of } J(\theta) \text{ at } \theta = (x_{i1} \quad \cdots \quad x_{ip})^T, \\ \widehat{J} &= (\widehat{J}_1 \quad \cdots \quad \widehat{J}_m)^T, \end{aligned}$$

the least squares estimate of the coefficients is given by

$$\hat{\beta} = (X^T X)^{-1} X^T \widehat{J}, \tag{4.4}$$

whenever $X^T X$ is nonsingular. In terms of simulation, the latter means that multiple replications must be incorporated into a *single* estimated value for $\widehat{J}_i$, e.g., the sample mean. As is customary in regression, an analysis of variance (ANOVA) can be done to partition the sum of squares attributes into regression error, lack of fit error (due to deviations from linearity), and pure error.

Under the linear model, the gradient estimate is simply

$$\widehat{\nabla} J = \hat{\beta}.$$

If the step size is determined by a line search, the line search itself can be formulated as a one-dimensional optimization problem. A set of experiments is performed along the estimated steepest descent direction $\widehat{\nabla} J$ to determine a higher order polynomial in the step size a, which can then be easily optimized. For example, a second-order polynomial fit would be of the form

$$J(a) = J(\theta_n) + \gamma_1 a + \gamma_2 a^2,$$

where at least two additional points (along the gradient direction) would be needed to get the least-squares coefficient estimates. It is clear that the convergence rate of the RSM method will depend on this step size selection problem, just as it does for the stochastic approximation procedures we describe in the next section.

We now graphically illustrate the RSM sequential procedure. Since Example 1 with the GI/G/1 queue contains only a single parameter, we consider Example 2, the (s, S) inventory system, which can illustrate the procedure nicely in two dimensions. Fig. 4 depicts the progression of $\theta = (s, S)$ values in the two-dimensional $s - S$ plane, in a five-step realization $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$. For the four iterations of Phase I, a centered complete (2^2) factorial design is illustrated, and in the final Phase II stage, a central composite design is used, which require four (2×2) additional

points. Furthermore, in our depicted situation, the Phase II region is also smaller than the Phase I regions, signifying confidence that the procedure has homed in on the region of the optimum, where reduction of bias becomes more critical. See also [63, 106] for more details on similar inventory examples, where the second parameter is the actual re-order quantity.

As was noted in [127], RSM has the advantage of having an arsenal of well-known and well-studied statistical tools such as regression analysis and the analysis of variance at its disposal. One can (and many do) argue that discrete-event simulation models simply constitute another set of stochastic models, and thus naturally fall under the purview of the vast body of research done on RSM, as reported, e.g., in the *Technometrics* survey paper of [87]. However, upon perusal of the contents of the survey, one finds that philosophically researchers in the field of RSM usually face a very different set of circumstances than researchers in discrete-event simulation. A major advantage of RSM is its transparency to the user, but even this is a two-edged sword for discrete-event systems. In particular, the amount of control the simulator has over the model is generally much greater than the relative lack of control the traditional user of RSM has. This, for example, eliminates the need for blocking and randomization in the order of conducting experiments at different values of the parameters, assuming of course that the random variate generation is operating correctly. Furthermore, a lot more is known about the underlying discrete-event system model than is usually assumed in traditional RSM applications. Among other things, this understanding makes possible simplifications such as converting a control problem to an optimization problem and the development of gradient estimates from a single observation of the system.

One of the earliest works applying the RSM sequential methodology to discrete-event simulation was reported in [63], where the procedure was demonstrated on an inventory model. Following this, Smith [115, 116, 117] did a lot of work attempting to automate the procedure and bring it to widespread use in the simulation community. The book by Biles and Swain [8] contains a summary of the RSM sequential approach, including large portions devoted to both the design of experiments aspects and deterministic optimization algorithms. However, despite these attempts, it was recently reported that in the simulation community, "RSM has yet to receive much attention and respect from practitioners and academicians"[106]. Further references can be found in the review paper [105], which concentrates primarily on RSM techniques.

In summary, RSM sequential procedures provide a very general methodology for optimization via simulation. RSM's biggest advantage is its generality, but it should be evident that its biggest drawback if applied blindly is its computational requirements. Design of experiments

with a "black box" perspective can go only so far in addressing this problem. For this reason, other techniques or analyses based on the nature of the actual discrete-event 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 needed.

5. Gradient-Based Algorithms

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, it should be clear from the discussion in the last section that the sequential RSM procedure also implements a gradient-based algorithm in Phase I, where the gradient is found from the regression model.

5.1. STOCHASTIC APPROXIMATION

The bulk of research in recent years for optimization of discrete-event systems via simulation has been on stochastic approximation (SA) methods, first introduced by Robbins and Monro [95] and Kiefer and Wolfowitz [73]. Whether this is because it will become the most useful technique for simulation optimization or because it is simply an academically challenging research area remains to be seen. In any case, new algorithms have been proposed, much theoretical research on convergence properties have appeared, and work on gradient estimation has flourished (at least in the academic research community).

For background, probably the most referenced book on stochastic approximation is [75]. Other books on the subject include [2, 7, 89, 126].

The basic underlying assumption of stochastic approximation is that the original problem given by (2.1) can be solved by finding the zero of the gradient, i.e., by solving

$$\nabla J(\theta) = 0. \tag{5.1}$$

Of course, in practice, this may lead only to local optimality. The problem of global optimality is a difficult one to handle theoretically in an efficient manner (i.e., the global convergence algorithms that *provably* converge often perform quite poorly in applications), although heuristically there are a variety of approaches, e.g., applying procedures used for global optimization of combinatorial optimization problems such as simulated annealing, genetic algorithms, and various "learning" algorithms. We discuss this issue briefly in the Section 6.

The general form of the stochastic algorithm takes the following form:

$$\theta_{n+1} = \Pi_\Theta \left(\theta_n - a_n \widehat{\nabla} J_n \right), \tag{5.2}$$

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 an unbiased estimator is used for $\nabla J(\theta_n)$, (5.2) is called a Robbins-Monro [95] algorithm and when a finite difference estimate is used, it is called a Kiefer-Wolfowitz [73] algorithm. Sometimes, the term Robbins-Monro-like algorithm is used for those procedures that estimate the gradient with some bias but without resorting to finite differences.

The main factors are the following:

- Getting a gradient estimate $\widehat{\nabla} J_n$.
- Choosing a step size a_n.
- Choosing an observation length for each iteration.

The usual assumptions needed for the convergence of the algorithm are the requirements that (1) the step size goes 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 (2) that the bias of the gradient estimate goes 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 the harmonic series sequence of step sizes, a decrease is taken at every iteration. In practice, this often leads to rather slow convergence. A sequence proposed by Kesten [71] only decreases the step size if the gradient direction has changed from the previous iteration and appears to work better in practice.

We now discuss some features of the stochastic approximation approach as applied to simulation optimization that set it apart from the previous approaches. One major difference is contained in the last item in the list above. Since in general the conditional (on the information known at the beginning of the iteration) bias in the gradient estimation must go to zero, for the finite horizon case, eventually the estimates must be taken over the entire finite horizon. One approach would be to follow a similar procedure as in RSM: use an unbiased finite horizon gradient estimator over a *fixed* observation length – the length of the entire horizon. In other words, an unbiased estimate is utilized at each step of the algorithm. This is the traditional version of the Robbins-Monro SA algorithm.

For the steady-state problem, on the other hand, RSM would approach it as the limit of the finite horizon problem, i.e., take long runs to get estimates. However, in theory, this approach would lead to biased results, since the finite horizon estimator would usually be a biased estimate of the steady-state value. Of course, one would expect the bias to decrease with horizon length. In RSM, this is the only choice. In stochastic approximation, though, one can take the observation length small in the beginning and gradually increase it, so that in a single run, the algorithm converges to the optimum. In fact, in general, this is what is needed for steady-state optimization. However, there are cases where even this is not needed, i.e., a *fixed* observation length will still lead to convergence to the optimum [80]. This phenomenon was first observed in the experiments of Suri and Zazanis (1988), where an infinitesimal perturbation analysis (IPA) gradient estimate was used. Thus, while the parameter is being updated, the system approaches steady state, where IPA is unbiased over a finite observation length. However, aside from the IPA case, algorithms implemented in this fashion involve both biased estimators and dependent noise, which remove them from the Robbins-Monro class of algorithms. Kushner and Clark [75] deal with these under weak convergence, whereas strong (those that hold almost surely) limit theorems are considered in [84, 85, 102]. The technique used is the association of a deterministic differential equation with the stochastic approximation algorithm. As a result, the convergence of the stochastic approximation algorithm is analyzed via stability analysis of the differential equation.

The bulk of the remainder of this section will be on discussing some applications of SA to optimization of discrete-event systems and to the gradient estimation problem itself, with some further discussion on the step size problem at the end of the section. We will describe in greater detail the two gradient estimation techniques of perturbation analysis (PA) and the likelihood ratio (LR) method (also known as the score function method), and to a lesser extent the frequency domain experimentation (FDE) method. Another technique called the weak derivative [91] will not be discussed. Conceptually, it bears some similarity to the LR/SF method, in that it differentiates the probability measure, but the differentiated quantity is then replaced by its weak derivative. The implementation of these derivative estimates in a stochastic approximation method for Markov processes can be found in [90]. See also [78] for a good overview of gradient estimation techniques.

5.2. GRADIENT ESTIMATION TECHNIQUES FOR DISCRETE-EVENT SYSTEMS

We discuss four gradient estimation techniques: finite differences (FD), perturbation anal-

ysis (PA), the likelihood ratio (LR) method, and frequency domain experimentation (FDE). The traditional "brute-force" finite difference implementations is represented schematically in fig. 5, whereas the newer methods are depicted in fig. 6. Finite differences and FDE both alter the input and analyze the resulting input, 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.

5.2.1. Finite differences

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

$$\widehat{\nabla} J_n = [\widehat{\nabla_1} J_n \quad \cdots \quad \widehat{\nabla_p} J_n]^T, \tag{5.3}$$

$$\widehat{\nabla_i} J_n = \frac{\widehat{J}(\theta_n - c_n e_i) - \widehat{J}(\theta_n + c_n e_i)}{2c_n}, \tag{5.4}$$

where recall that 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 (see, e.g., [39].

When the finite difference estimate is used in the stochastic approximation algorithm (5.2), it is called the Kiefer-Wolfowitz algorithm [73]. In order to get convergence to the correct value, we must have $c_n \to 0$. Because of this, aside from exceptional cases (which includes the cases where the method of common random numbers "works"; see [79, 82]), the best convergence rate achievable by the Kiefer-Wolfowitz algorithm is $O(n^{-1/3})$ versus $O(n^{-1/2})$, when an unbiased estimate is used [75]. 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.

5.2.2. Perturbation analysis

Perturbation analysis was introduced by Ho, Eyler, and Chien [54] in the context of buffer allocation of a serial production line. Its use as a gradient estimation technique was formulated in [51, 53, 52]. The books by Ho and Cao [50] and Glasserman [32] cover perturbation analysis up to 1990, with some updated references contained in [49]. The former text takes a more algorithmic

view of the problem ("written by engineers for engineers") and covers a broader range of the PA techniques not limited to just gradient estimation, whereas the latter monograph concentrates much more on the gradient estimation problem and focuses on the theoretical problem of finding structural conditions for verifying unbiasedness and consistency of the estimators.

We begin by pointing out that the LR method is a well-defined technique, whereas PA taken as a whole is more of a philosophical approach of using **sample path** analysis for gradient estimation, whereby the effects of small changes in the parameter are traced on the sample path. Thus, the term PA has now come to embody a whole class of different techniques. *Infinitesimal* perturbation analysis (IPA), on the other hand, *is a well-defined technique* which is easy to derive and implement with very little computational overhead. Intuitively, it is the estimator resulting from assuming that small changes cause no event order changes. Roughly speaking, when the sample performance is a.s. (almost surely) continuous, the IPA estimator is unbiased. If the parameter is a *distributional* parameter, then we have the IPA estimate considered in detail in [32, 33]. The underlying requirement is uniform integrability of difference quotients. For IPA w.r.t. distributional parameters, Glasserman [33] gives an easily checked set of sufficient conditions on the structure of the system of interest for a certain class of performance measures. The framework used is the generalized semi-Markov process (GSMP) model for discrete-event simulation. The main condition is the commuting condition, which we discuss briefly here. Milder conditions on Markov chains and extensions such as the idea of relevance will not be discussed in detail here. The commuting condition is illustrated by the state transition diagram shown in fig. 7, where s, s_1, s_2, s_3, s_4 represent states of the system and α, β represent events. In the diagram, the commuting condition requires $s_3 = s_4$. In words, the resulting state of the system after the occurrence of two events α and β will be the same regardless of the order of occurrence (assuming both are possible in the original state). A system is said to satisfy the commuting condition if this holds for all states s. An example is the FCFS single server queue. The state of the GSMP modeling the queue is the number in system. There are only two events possible, a departure and an arrival. Other than the empty state, when only an arrival is possible and so the condition need not be checked, one can easily see that the state reached after the sequence of events departure-arrival is the same as with the order reversed, namely the original state ($n \to n-1 \to n$ and $n \to n+1 \to n$).

If IPA is not applicable, extensions are necessary, and these extensions are usually not as straightforward. Perhaps the most general extension is *smoothed* perturbation analysis (SPA),

whereby discontinuities are smoothed by conditional expectation. This technique was first used in [132], and developed more formally in [43]. However, unlike IPA, this is not an "automatic" procedure. As a result, many other variants usually also involving conditioning arguments have been proposed, including rare perturbation analysis [11], discontinuous perturbation analysis [114], and augmented perturbation analysis [31]. Under the GSMP framework used for IPA in [33], Glasserman and Gong [34] considered systems satisfying the commuting condition and applied SPA to certain types of performance measures which were not amenable to IPA. Fu and Hu [26] generalized these results to a broader class of performance measures and for systems not necessarily satisfying the commuting condition, with the results in [34] and the IPA results in [33] falling out as corollaries of the main theorems.

We present an overview of IPA and a brief discussion of SPA. We will illustrate the technique via Example 1, the GI/G/1 queue; see also in [119] for a more detailed discussion of this example. The system time of a customer for a first come, first-served (FCFS) single-server queue satisfies the well-known recursive Lindley equation:

$$T_{n+1} = X_{n+1} + \begin{cases} T_n - A_{n+1} & \text{if } T_n \geq A_{n+1} \\ 0 & \text{if } T_n < A_{n+1} \end{cases} . \tag{5.5}$$

The IPA estimator for the first derivative is given by the sample path derivative, which in this case falls out from differentiation of both sides of (5.5):

$$\frac{dT_{n+1}}{d\theta} = \frac{dX_{n+1}}{d\theta} + \begin{cases} dT_n/d\theta & \text{if } T_n \geq A_{n+1} \\ \\ 0 & \text{if } T_n < A_{n+1} \end{cases} , \tag{5.6}$$

where we have under mild conditions [121]

$$\frac{dX}{d\theta} = -\frac{dF_2(X;\theta)/d\theta}{dF_2(X;\theta)/dX}. \tag{5.7}$$

For example, for scale parameters, such as if θ is the mean of an exponential distribution, we have $dX/d\theta = X/\theta$. Using the above recursion, the IPA estimator for the derivative of mean system time is given by

$$\left(\frac{dT}{d\theta}\right)_{IPA} = \frac{1}{N}\sum_{i=1}^{N}\frac{dT_i}{d\theta}. \tag{5.8}$$

For example, for M busy periods in an M/M/1 queue with N customers served, we would use

$$\left(\frac{dT}{d\theta}\right)_{IPA} = \frac{1}{N}\sum_{m=1}^{M}\sum_{i=1}^{n_m}\sum_{j=1}^{i}\frac{dX_{(j,m)}}{d\theta}, \tag{5.9}$$

where n_m is the number of customers served in the mth busy period, $N = \sum_{m=1}^{M} n_m$ is the total number of customers served, and the (j,m) subscript denotes the jth customer in the mth busy

period, i.e., $(j,m) = j + \sum_{i=1}^{m-1} n_i$. Thus, implementation of the estimator involves the addition of accumulators; *no alteration of the underlying simulation is required.*

The assumption used in deriving an IPA estimator is that small changes in the parameter will result in small changes in the performance measure. Thus, in the above, this means that the boundary condition in (5.6) is unchanged by differentiation. For general sample performances L, the interchange

$$\frac{dE[L]}{d\theta} = E\left[\frac{dL}{d\theta}\right] \tag{5.10}$$

is justified roughly if L is a.s. continuous with respect to θ. For the Lindley equation, although T_{n+1} in (5.5) has a "kink" at $T_n = A_{n+1}$, it is still continuous at that point. This intuitively explains why IPA works. Unfortunately, the "kink" means that the derivative given by (5.6) has a discontinuity at $T_n = A_{n+1}$, so that IPA will fail for the second derivative.

The idea of smoothed perturbation analysis (SPA) is to use conditional expectation to "smooth" out such discontinuities [43]. After conditioning on a set of sample path quantities z called the *characterization*, oftentimes $E[L|z]$, as opposed to L itself, will satisfy the interchange in (5.10), i.e.,

$$\frac{dE\left[E[L|z]\right]}{d\theta} = E\left[\frac{dE[L|z]}{d\theta}\right]. \tag{5.11}$$

There are two potential difficulties in applying SPA: finding an appropriate z [27] and determining an estimator for $E[L|z]$ from the original sample path [26].

For the GI/G/1 queue, however, SPA works nicely for the second derivative of mean system time, resulting in the following estimator [132]:

$$\left(\frac{d^2T}{d\theta^2}\right)_{SPA} = \frac{1}{N}\sum_{m=1}^{M}\sum_{i=1}^{n_m}\sum_{j=1}^{i}\frac{d^2X_{(j,m)}}{d\theta^2} + \frac{1}{M}\sum_{m=1}^{M}\frac{f_1(z_m)}{1-F_1(z_m)}\left(\sum_{i=1}^{n_m}\frac{dX_{(i,m)}}{d\theta}\right)^2, \tag{5.12}$$

where z_m is the age of the interarrival time at the end of the mth busy period, and

$$\frac{d^2X}{d\theta^2} = -\frac{\partial^2F_2/\partial\theta^2(\partial F_2/\partial X)^2 + \partial^2F_2/\partial X^2(\partial F_2/\partial\theta)^2 - 2\partial^2F_2/\partial X\partial\theta(\partial F_2/\partial\theta)(\partial F_2/\partial X)}{(\partial F_2/\partial X)^3},$$

when $F_2(X;\theta)$ is twice differentiable.

The recursive version is derived informally by first rewriting (5.6):

$$\frac{dT_{n+1}}{d\theta} = \frac{dX_{n+1}}{d\theta} + \frac{dT_n}{d\theta}\cdot I\{T_n \geq A_{n+1}\} + 0\cdot I\{T_n < A_{n+1}\}, \tag{5.13}$$

where recall that $I\{\cdot\}$ denotes the indicator function on the given set. Both T_n and A_{n+1} are random. To derive the second derivative, we condition on T_n. The important thing to note here

is that if customer n is the last customer served in the busy period, then T_n is in fact equal to the lifetime of the interarrival time at the end of the busy period, which we denoted by z before. Thus, we write

$$\begin{aligned} E\left[\frac{dT_{n+1}}{d\theta}|T_n = z\right] &= E\left[\frac{dX_{n+1}}{d\theta}|T_n = z \geq A_{n+1}\right] && (5.14)\\ &\quad + E\left[\frac{dT_n}{d\theta}|T_n = z \geq A_{n+1}\right] P(T_n \geq A_{n+1}|T_n = z) \\ &= \frac{dX_{n+1}}{d\theta} + E\left[\frac{dT_n}{d\theta}\right] F_1(T_n)|_{T_n=z}, && (5.15) \end{aligned}$$

where the conditioning has been removed for those random variables which are independent of T_n and A_{n+1}. Differentiating, and assuming exchangeability for the "smoothed" expectation, we get

$$\begin{aligned} \left(\frac{d^2T_{n+1}}{d\theta^2}|T_n = z\right)_{SPA} &= \frac{d^2X_{n+1}}{d\theta^2} + \frac{d^2T_n}{d\theta^2}F_1(T_n)|_{T_n=z} + \frac{dT_n}{d\theta}F_1'(T_n)|_{T_n=z}\frac{dT_n}{d\theta} && (5.16)\\ &= \frac{d^2X_{n+1}}{d\theta^2} + \frac{d^2T_n}{d\theta^2}F_1(T_n)|_{T_n=z} + \left(\frac{dT_n}{d\theta}\right)^2 f_1(T_n)|_{T_n=z} && (5.17)\\ &= \frac{d^2X_{n+1}}{d\theta^2} + \frac{d^2T_n}{d\theta^2}P(T_n \geq A_{n+1}|T_n = z) && (5.18)\\ &\quad + \left(\frac{dT_n}{d\theta}\right)^2 \frac{f_1(z)}{1-F_1(z)}P(T_n < A_{n+1}|T_n = z), && (5.19) \end{aligned}$$

which rewritten as

$$\left(\frac{d^2T_{n+1}}{d\theta^2}\right)_{SPA} = \begin{cases} \left(d^2T_n/d\theta^2\right)_{SPA} & \text{if } T_n \geq A_{n+1} \\ (dT_n/d\theta)^2 f_1(T_n)/(1-F_1(T_n)) & \text{if } T_n < A_{n+1} \end{cases}, \quad (5.20)$$

gives the recursive form of Equation (5.12).

For the (s, S) inventory system example, SPA can be used to derive the following estimator over N periods [23]:

$$\begin{aligned} \left(\frac{\partial J(s,q)}{\partial s}\right)_{PA} &= \frac{1}{N}\left[\sum_{i:X_i>0} h - \sum_{i:X_i<0} p\right], && (5.21)\\ \left(\frac{\partial J(s,q)}{\partial q}\right)_{PA} &= \frac{1}{N}\left[\sum_{i:X_i>0} h - \sum_{i:X_i<0} p\right] + \frac{1}{N+1}\sum_{j:X_j<s}\frac{g(Z_j)}{1-G(Z_j)} \\ &\quad \cdot \left[cE[D] + hE[s-D]^+ + pE[D-s]^+ - \frac{\sum_{i=1}^N C(X_i)}{N}\right], && (5.22) \end{aligned}$$

where $q = S - s$, $J(s, q)$ denotes the long-run average cost per period, $Z_j = X_{j-1} - s$, and D denotes the (generic r.v.) single period demand with density and distribution functions $g(\cdot)$ and $G(\cdot)$, respectively. We note that the above estimator is for the system where the review occurs

before demand for the period has been subtracted, i.e., X_n satisfies (corresponding to the $L = 1$ case in [23])

$$X_{n+1} = \begin{cases} S - D_{n+1} & X_n < s \\ X_n - D_{n+1} & X_n \geq s \end{cases} .$$

5.2.3. Likelihood ratio method

The likelihood ratio (LR) method, also known as the score function (SF) method, has its roots in a 1968 Russian paper [1], but it did not come into the consciousness of the discrete-event simulation community until around 1986 [100, 99, 93, 94, 36, 37]. Basically the idea is to differentiate the underlying probability measure of the system, but it can more generally be viewed as a special case of importance sampling. Conceptually, the difference between LR and PA is that perturbation analysis tries to study the behavior of *individual* sample paths as a function of a parameter. Once conditioning is added to PA, though, the line becomes blurred between the two. The "unified" view contained in [77] actually combines both of the above philosophies by allowing simultaneous consideration of both changes on individual sample paths and entire sample path changes.

The LR estimator is as easily implementable as IPA, and it often works for systems where IPA fails. However, the resulting estimator may have variance problems for some systems; when IPA works, it usually has much lower variance. Variance comparisons between SPA and LR, on the other hand, seem to be quite problem dependent (see, e.g., [124]). Also, 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 [77]. Rubinstein [98] has proposed a "push-in" approach to extend the applicability of the LR/SF methodology, which attempts to move structural parameters into the probability measure by an appropriate transformation. At present, though, the existence of such a transformation which can be practically applied is very problem-dependent.

We present a brief overview of the LR technique, and derive regenerative LR estimators for our problem. 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:

$$E[L(X)] = \int L(x) dF(\theta, x). \tag{5.23}$$

Differentiating (5.23), we have

$$\begin{aligned}\frac{\partial E[L]}{\partial\theta} &= \frac{\partial}{\partial\theta}\int L(x)f(\theta,x)dx \quad = \quad \int L(x)\frac{\partial f(\theta,x)}{\partial\theta}dx \\ &= \int L(x)\frac{\partial f(\theta,x)}{\partial\theta}\frac{f(\theta,x)}{f(\theta,x)}dx \quad = \quad \int L(x)\frac{\partial \ln f(\theta,x)}{\partial\theta}f(\theta,x)dx \\ &= E\left[L(X)\frac{\partial \ln f(\theta,X)}{\partial\theta}\right]. \end{aligned} \tag{5.24}$$

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. We note that the above derivation is an application of the more general importance sampling concept to derivative estimation, where in the second line $g(x)/g(x)$ could have been used in place of $f(\theta,x)/f(\theta,x)$ for appropriate $g(\cdot)$. A set of mild assumptions relating to the differentiability of the performance measure (cf., e.g., [99, 77]) allows the interchange of differentiation and integration in the first line. However, as we shall see, the "naive" estimator for (5.24) leads to unbounded variance for steady-state performance measures.

To give a flavor of the method, we demonstrate the ease of the method for Example 1, the GI/G/1 queue, where the interarrival times and the service times comprise the random vector. Assume that the system starts empty, so that the times of the first N service completions are completely determined by the first N interarrival times and first N service times. Since these times are all independently generated, the joint density function f on X will simply be the product of the density functions of the interarrival and service time distributions given by

$$f(\theta,A_1,...,A_N,X_1,...,X_N) = \prod_{i=1}^{N} f_1(A_i)\prod_{i=1}^{N} f_2(X_i), \tag{5.25}$$

where $A_i,\ X_i,\ i=1,...,N$ are the interarrival times and service times, respectively. For example, in the Markovian case (M/M/1 queue) with arrival rate λ and service rates μ, and taking $\theta = 1/\mu$, (5.25) becomes

$$f(\theta,A_1,...,A_N,X_1,...,X_N) = \prod_{i=1}^{N} \lambda e^{-\lambda A_i}\prod_{i=1}^{N} \mu e^{-\mu X_i} \tag{5.26}$$

and we have

$$\ln f(\theta,A_1,...,A_N,X_1,...,X_N) = \sum_{i=1}^{N}(\ln\lambda - \lambda A_i) + \sum_{i=1}^{N}(\ln\mu - \mu X_i) \tag{5.27}$$

and

$$\frac{\partial \ln f}{\partial\theta} = \sum_{i=1}^{N}\left(\frac{X_i}{\theta^2} - \frac{1}{\theta}\right). \tag{5.28}$$

Taking mean system time as our performance measure of interest, the natural estimators would then be given by

$$\left(\frac{dT}{d\theta}\right)_{LR} = \frac{1}{N}\sum_{i=1}^{N} T_i \sum_{i=1}^{N}\left(\frac{X_i}{\theta^2} - \frac{1}{\theta}\right). \tag{5.29}$$

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 with N, resulting quickly in a useless estimator. To resolve this problem, we demonstrate a regenerative estimator which does not have this problem. Other approaches that tackle this problem include the use of control variates and the use of conditional expectation [29]. A completely different and very novel scheme for reducing variance is proposed in [133], where simulation results indicate great promise.

We describe the regenerative estimator for the GI/G/1 queue. Using regenerative theory, we can express the mean steady-state system time as a ratio of expectations:

$$E[T] = \frac{E[Q]}{E[\eta]}, \tag{5.30}$$

where η is the number of customers served in a busy period and Q is the sum of the system times of customers served in a busy period. Differentiation of (5.30) yields

$$\frac{dE[T]}{d\theta} = \frac{dE[Q]/d\theta}{E[\eta]} - \frac{dE[\eta]/d\theta}{E[\eta]}E[T]. \tag{5.31}$$

Now, employing (5.24) in conjunction with (5.28), we have the following regenerative estimators over M busy periods:

$$\left(\frac{dT}{d\theta}\right)_{LR} = \frac{1}{N}\sum_{m=1}^{M}\left\{\sum_{i=1}^{n_m} T_{(i,m)}\frac{\partial \ln f}{\partial \theta}\right\} - \frac{1}{N}\sum_{m=1}^{M}\left\{n_m\frac{\partial \ln f}{\partial \theta}\right\}\frac{1}{N}\sum_{j=1}^{N} T_j, \tag{5.32}$$

where $N = \sum_{m=1}^{M} n_m$ is the total number of customers served and recall that the subscript (i,m) denotes the ith customer in the mth busy period. The advantage of these estimators is that the summations are bounded by the length of the busy periods, so as long as the busy periods are not too long, the variance of the estimators should be reasonable.

5.2.4. Frequency domain experimentation

Frequency domain experimentation (FDE) was introduced by Schruben and Cogliano [112, 111] for the purposes of factor screening in simulation experiments. In this context, Sargent and Som [109] report some precautions that should be taken in applying the technique and interpreting the results. In [110], the technique was proposed as a gradient estimation technique, and a model

for obtaining such estimates and using them in optimization was presented in [68]. It is in this latter context that we are interested.

The intuitive idea in the 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. Initially, the main thrust of FDE was for factor screening, i.e., as a tool to identify the most significant parameters by oscillating a large number of parameters at different frequencies in a *single* simulation, thus providing significant computational savings over traditional design of experiment approaches such as factorial design. Thus, the vectors of input parameters are modulated as follows:

$$\theta(t) = \theta_0 + \alpha \sin(\tilde{\omega} t), \tag{5.33}$$

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. In fact, in order to apply FDE, the determination of an appropriate "time" is a non-trivial problem called the *indexing* problem. Thus, it is clear that 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.

Much of the research in the FDE area has been to address one of these three problems; cf. [69] for the frequency selection problem, [65, 64] for amplitude selection problem, and [70, 86] for the indexing problem. The indexing problem is probably the most "open" of the three problems. In queueing systems, natural candidates for indexing would be events like customer arrivals. However, this index present problems in all but the simplest queueing systems, because outputs such as customer departures will not be ordered the same as the input arrival events, so some re-ordering of the output may need to be done before the spectral analysis. Furthermore, for closed queueing networks, an obvious index may not be evident at all. Contrary to previous work on the problem, Mitra and Park [86] have recently proposed the use of the global simulation clock as the ideal index. Their approach differs from previous work using the simulation clock in that

they in fact discretize time for uniform sampling purposes, instead of sampling at the occurrence of the usual events generated by the simulation. Because this alters the simulation procedure itself, making it even more "intrusive" than any of the other gradient estimation techniques, it remains to be seen whether this will be a viable solution.

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, we obtain a quadratic dynamic polynomial response surface metamodel [68]:

$$J(\theta(t)) \approx Y(\theta(t)) = g_0 + \sum_{j=1}^{p} \sum_{\tau=-\infty}^{\infty} \theta_j(t-\tau) g_j(\tau) + \sum_{j=1}^{p} \sum_{m=j}^{p} \sum_{\tau=-\infty}^{\infty} \theta_j(t-\tau)\theta_m(t-\tau) g_{j,m}(\tau) \quad (5.34)$$

where $|g_0| < \infty, g_j(\tau), g_{j,m}(\tau) = g_{m,j}(\tau), j, m = 1, ..., p$, are memory filters satisfying $\sum_{\tau=-\infty}^{\infty} |g_j| < \infty$ and $\sum_{\tau=-\infty}^{\infty} |g_{j,m}| < \infty$. Then, one can show that [68]

$$\nabla_i Y(\theta_0) = \lim_{T\to\infty} \lim_{\omega_i \to 0} \frac{2}{\alpha_i T} \sum_{t=1}^{T} Y(\theta(t)) \sin(\omega_i t), \quad (5.35)$$

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, assuming the driving frequencies have been chosen properly to ensure that no confounding occurs (e.g., using the tables contained in [69]). A number of variance reduction techniques that require an additional simulation run are proposed in [66].

In order to get asymptotic unbiasedness for the quadratic model, one needs $\tilde{\omega} \to 0$, which in turn means that the simulation horizon T must be made very large. Furthermore, to get unbiasedness for the gradient of the true performance measure $J(\theta)$ – as opposed to polynomial metamodel $Y(\theta)$– the oscillation amplitudes α must also be driven to zero. As a result, there are trade-offs in the selection of α, which are quite analogous to the choice of the difference in the FD estimate. Since true unbiasedness requires $\alpha \to 0$, on one hand α should be made as small as possible. However, as in the FD estimates, making α too small will lead to higher variance. Thus, as in FD estimation, there is a point at which one can minimize the mean-squared error, i.e., the sum of the variance and squared bias; analysis along this line is done in [64]. Of course, most of the time one would not know this value a priori.

For the GI/G/1 example, we have only a single parameter, so there is no problem with confounding. Using the customer index as the "time" variable, we can choose, for example, $\omega = 2\pi/N$, where N is the number of customers served, and so our estimator at $\theta = \theta_0$ is simply

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

where T_i is again the time in system for the ith customer, but remember that the system is being simulated with the mean service time of the ith customer given by

$$\theta(i) = \theta_0 + \alpha \sin(2\pi i/N).$$

In this example, the only decision that need be made is the choice of α. As an example of the sensitivity to this choice, in the experiments reported in [66] for an M/M/1 queue (which included the arrival rate as a parameter, as well), the three values $\alpha = 0.001, 0.01, 0.1$ were used. The mean-squared error between the highest and lowest of the three differed by two orders of magnitude, e.g., 2.8 vs. 0.028 vs. 0.018.

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 (see [66] for details).

5.3. COMPARISON OF THE TECHNIQUES

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. However, 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. In fact, Jacobson [64] shows that the convergence rate of the FDE estimate (of the mean-squared error of the estimator) is identical to the rate of the symmetric FD estimate. Even in the limit of $\tilde{\omega} \to 0$, the estimator is biased, because the convergence is to the gradient of an underlying model which is quadratic. In order to get true unbiasedness, we need in addition $\alpha \to 0$. In terms of stochastic approximation, this means that *three* quantities must go to zero in the limit: the step size, as well as both the oscillation frequencies and oscillation amplitudes.

On the other hand, FDE seems in principle to be more general than IPA or LR, being as we noted already 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. However, for more complicated systems, the indexing problem for FDE may not be so easy to solve. Using the global simulation clock as proposed in [86] requires alteration of the simulation by discretizing time, so it remains to be seen if this is a practical solution.

In the cases where LR applies, it is the easiest of the methods to extend to higher derivative estimates. For the GI/G/1 queue example, the analysis is trivial, whereas for PA, again SPA was

needed, and for FDE, although second derivatives can be obtained for the second-order model, higher order derivatives would require higher order extensions to the model.

Finally, as we noted already, the implementation of all three procedures — PA, LR and FDE — are "add-ons" which usually require fairly minimal additional code such as statistical counters (extensions of IPA sometimes being notable exceptions), but do not require alteration of the underlying event-generation code; however, in the case of FDE, if the simulation clock is used for indexing, as proposed in [86], modification of the event-generation code would be required. Furthermore, since FDE alters the input stream, it cannot be used for on-line optimization of an actual system. Overall, when it applies, IPA is usually the most efficient estimator. If IPA is not applicable, then an LR estimate, a PA variant such as one based on SPA, or an FDE gradient estimate should be tried.

5.4. APPLICATION TO SIMULATION OPTIMIZATION

One of the earliest applications of stochastic approximation methods to simulation optimization was the work by Azadivar and Talmage [4], 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 of Smith [116] 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; see also [81] for computational results.

The first application of PA to optimization was contained in the paper by Ho and Cao [51]. 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. There have been a number of applications along this line. Caramanis and Liberopolous [12] applied an IPA-based gradient algorithm to determine the parameters of a flow controller for failure-prone manufacturing systems. The system state was composed of a continuous component containing the buffer levels and a discrete component containing the machine up-down states, with the dynamics of the continuous component driven by a differential equation determined by the discrete-event model of the machine states. The sample paths of the buffer levels were continuous, which lent itself nicely to IPA.

In contrast, the work of Suri and Zazanis [121] introduced the idea of "single-run" optimiza-

tion 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, thus 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. Empirically, the algorithm worked quite well, but no theoretical convergence proofs were given.

Suri and Leung [120] conducted a more comprehensive simulation study of Example 1 focusing exclusively on the steady-state M/M/1 queue, where the analytical optimum was available for comparison:

$$\theta_* = (\lambda + \sqrt{c_0/c_1})^{-1},$$

λ being the arrival rate of the Poisson arrival process. The aim was to provide an empirical comparison of the convergence rates of the IPA-driven gradient algorithm and an FD-based algorithm. The FD-based algorithm, however, used a biased estimate, because the difference was kept constant instead of being forced to zero, so aside from its inferior convergence rate, it also converged to the wrong value, although from experimental results, the bias was relatively small. In [81], an even more comprehensive set of numerical experiments on the same M/M/1 queue example were reported. Various algorithms utilizing IPA, LR, and FD estimates with common random numbers are considered and compared, with the IPA-based algorithms clearly superior.

The use of FDE gradient estimates in stochastic optimization algorithms is relatively new. Jacobson and Schruben [68] provided some numerical results also for the steady-state M/M/1 queue example. However, the algorithm was of the long-run, small number of replications type, intended to mimic the Newton algorithm of deterministic optimization by using the Hessian (inverse) for step size. Since the step size was not decreasing nor the time horizon increasing in the algorithm, strictly speaking there could be no theoretical convergence proof for such an algorithm, but in practice it may work well.

Of course, aside from empirical results, numerous theoretical convergence proofs have been provided for the various algorithms. Some of the earlier work includes Glynn [40], who gave some convergence results for FD-based and LR-based algorithms for Markov chains, and Fu [22],

who gave a convergence proof for a regenerative IPA-based algorithm (see also [125]). Other more recent work along these lines includes Chong and Ramadge [15, 16] – who looked at the general update time issue, L'Ecuyer and Glynn [80], Leung and Suri [83], and Andradóttir [3]. It is interesting to note that in order to establish the a.s. convergence of the various algorithms, with the exception of special regenerative-based algorithms and some IPA-based algorithms, the simulation length between parameter updates must be increasing (see also [20]). In terms of on-line optimization this result demonstrates another notable advantage of the IPA-based version, since it is undesirable in practical terms to have the observation length go to infinity.

One of the major difficulties in applying stochastic approximation algorithms is the choice of various parameters in the algorithm such as step size and the observation horizon, and to a lesser extent, the projection rule used in (5.2), the latter of which we will not address. Also, for practical implementation, a stopping rule must be implemented. As we have noted already for Example 1, except for certain IPA-based versions of the algorithm, the observation horizon must increase with each iteration. From the numerous simulation results contained in [81], the effect of step size is quite apparent. Fig. 8 (taken from [24]) illustrates this effect for Example 2, the periodic review (s, S) inventory system, where convergence of the value of the performance measure is depicted as a function of time for four different values of the initial step size a. $\mathcal{J}$ is the long-run average cost per period, $\mathcal{J}^*$ is the optimal (minimal) value, and n is the number of periods simulated (not the iteration number). A single replication for each case is depicted, with an observation horizon of 50 periods, indicating that iterates of the algorithm were done every 50 periods. Of course, it has long been known [17, 104] that the optimal (in terms of asymptotic convergence rate) initial step size for the harmonic series $1/n$ is the inverse of the second derivative (Hessian in vector case). In theory, this could also be estimated, but it is unclear whether the extra effort would be worthwhile. For systems where the LR method applies, this is a straightforward extension. For applying PA, however, it is not always so easy. For example, for the GI/G/1 queue, IPA suffices for the first derivative, but SPA is needed for the second derivative. Fu and Ho [25] use second derivative information to improve the early convergence rate of the stochastic approximation algorithm. However, even for systems as simple as the GI/G/m queue, applying PA methods can be a formidable task [28]. On the other hand, the system studied by Caramanis and Liberopolous [13] had very amenable sample path properties, which made it possible to apply IPA for determining second derivatives and speed up the convergence of their algorithm. In general, another consideration is the increased variance that comes with estimating higher derivatives.

Averaging is another proposal to try to reduce the effects of the above choices; see e.g., [103, 92, 131]. The tradeoff is between the variance of taking noisy observations versus the bias of using "old" observations taken at previous values of the parameter. However, the experimental results reported in [25] did seem to indicate improvement in terms of stability, but theoretical work to support the empirical observations is still scarce, although Kushner and Yin have reported some results just recently being disseminated. Of course, it is well-known, that the asymptotic convergence rate is limited by $n^{-0.5}$, but to the practitioner, this theoretical result may not be of much interest, since the run may well be terminated before the asymptotic convergence rate becomes paramount. Leung and Suri [83] have done some studies on this transient convergence behavior topic.

In concluding this section, there are two main messages:

- The ability to compute the gradient directly (e.g., PA or LR vs. "brute-force" FD estimates) provides substantive opportunities to increase the efficiency of optimization techniques.
- The "single run" optimization implementation may do better in the early stages of optimization – when the parameter value is likely to be far from the optimum – than the traditional long-run, small number of iteration implementation of stochastic approximation algorithms, because it does not spend so much time trying to get an accurate estimate of the gradient when it is not necessary.

Gradient estimation techniques are not universally applicable and, aside from FD estimators, probably never will be, but when they work, their use in stochastic approximation algorithms make them a formidable tool. A simple analogy is linear programming. Like RSM procedures in simulation optimization, we always have simplex (or interior point methods) for linear programming, but if there is special structure like in an assignment problem, we should exploit it to full advantage. That is the role gradient estimation techniques will probably play. For example, many queueing systems exhibit a structure which makes them amenable to perturbation analysis, so that instead of applying the general RSM procedure for optimizing the system, it may be more efficient to apply a gradient-based technique, or combine the two as has been done in [58].

6. Other Methods

In this section, we touch on some other methods proposed recently which have some novel features. Before doing so, we note that for the uncountable state space problem we have concen-

trated on gradient-based algorithms. For more on other non-gradient-based algorithms such as pattern search methods and random search methods, see [67].

Thus far, deterministic algorithms have come into play in two major versions: metamodeling, which decouples simulation and optimization, and stochastic approximation and sequential response surface methodology procedures, which involve iterative interaction between simulation and optimization. Another approach developed by Rubinstein [97] and Healy and Schruben [46] independently is to use *each sample* to derive an entire performance curve "sample" and optimize the resulting curve using deterministic methods. Repeating this over a number of independent replications, the sample average is then used as the estimate of the optimum, where confidence intervals can also be constructed. This is a kind of dual to the metamodeling procedure where *all* the simulation replications are carried out before doing the deterministic optimization *once*. It is clear that in order for the proposed procedures to work, we need the following to hold:

$$E[\min_{\theta\in\Theta} L(\theta,\omega)] = \min_{\theta\in\Theta} E[L(\theta,\omega)], \tag{6.1}$$

since by the usual strong law of large numbers, the proposed procedures give the quantity on the left-hand side, whereas we are interested in the right-hand side. Sufficient conditions under which this exchange are justified are given in [47]. A more general setting has been reported in [101].

Conceptually, the ideas in [97] and [46] are similar, but the method in [97] relies on likelihood ratio (importance sampling) ideas to generate an estimate of the entire curve from a single sample path and then optimizes based on the assumption (needed for the LR method) that only the probability measure depends on the parameter and not the sample function (the "dual" of the IPA assumption). Under this condition, the exchanged required above is automatically satisfied. In [97], numerical results are provided for many examples, including the very well-worn steady-state M/M/1 queue of Example 1.

Healy's procedure, on the other hand, is in principle more general since it does not require the above assumption. However, then it must somehow establish the given interchange in another manner in order to guarantee that the resulting estimate is unbiased. Also, Healy's procedure requires a case by case analysis of each problem. In other words, a different optimization technique is potentially applied for each problem of interest, so in some sense it is not a complete "procedure" in the true sense of the work, but a proposed approach. Queueing and inventory examples are contained in [47], including instances of both Example 1 and Example 2.

Most of this review has focused on the local optimization problem. What can we do in terms of global optimization? At this point, although the literature on global optimization is

quite large, there is very little on global optimization specifically for discrete-event simulation. Obvious proposals include using multiple starting points for the gradient-based algorithms or adapting global optimization procedures from the deterministic nonlinear programming pages, such as random and pattern search strategies, simulated annealing, genetic algorithms, Tabu search, and "learning" algorithms; see, e.g., [129, 128] for some applications of learning algorithms to discrete-event simulation models.

Some may feel that optimization over an entire (especially uncountable) space may be asking for too much. One approach advocated is to systematically reduce the number of alternatives (whether they be countably or uncountably infinite) to a finite few, upon which the procedures of Section 3 can be applied. An implementation of this approach using pattern search methods (i.e., no stochastic approximation nor response surface methods) and its application to an inventory system problem was reported in [45]. Another related proposal in [59] is to replace cardinal optimization with ordinal optimization. In other words, instead of trying to find the best in a possibly uncountable infinite state space, just try to find "better" solutions. The measure then is not true optimality but for example the probability of being within some range of the best. The concept of "satisficing" is used to formalize this over the previous concept of optimality. Their proposal also seeks to exploit properties of massively parallel simulation, to be described later on in this section.

Combining techniques is another possible fruitful avenue of research. A recently proposed method by Ho et al. [58] called the Gradient Surface Method (GSM) combines the approaches of RSM and SA by utilizing gradient estimates. In RSM, the early phase involves utilizing a first-order design, and each point in the design requires multiple replications. In SA, only the most recent information is kept, and only a single replication is taken before a move is taken. Like RSM, the GSM procedure is essentially a two-phase procedure, where in the first phase, the method fits a surface to a set of points in the early phase. However, GSM differs by *implicitly* utilizing a second-order design by considering the *gradient* surface modeled by a first-order design. Like SA, only a single replication is used to get the gradient estimate used to determine the usual first-order least squares fit, but unlike SA, multiple points (a "window") are used. When the optimum is approached, the procedure switches to another phase which uses SA only, the justification being that SA is a much more efficient procedure than curve fitting once one is in the vicinity of the optimum. Thus, gradient estimation is used for two different purposes in the GSM procedure: in the first phase providing the "point estimates" in the gradient surface least-squares fit, and in

the second phase, providing the search direction in the gradient-based stochastic approximation algorithm. The essential gain from this procedure is p^2-fold for larger p, where p is the number of parameters. Like both SA and RSM, the procedure is sequential in spirit. Like SA, it tries to explore the region quickly, but like RSM, it also uses more than just the current information. The difficulty of choosing step size in SA is replaced with the seemingly easier choice of choosing (a possibly varying) window size for GSM.

Perhaps the greatest promise in stochastic optimization via simulation lies in the vast unexplored potential of massively parallel simulation. It would appear that one can achieve massive speed-ups with little effort simply by doing independent replications on separate processors, unlike the enormous effort that it takes to parallelize the simulation of a single discrete-event simulation called *distributed* simulation (see, e.g., [30] for an introduction to this area). In fact, this author would go so far as to say that it is far more advantageous to utilize this scheme than trying to implement any complicated parallelization of the code. One trade-off that might be argued is that if it is inherently a lengthy simulation, parallelizing a single simulation means that simulation would end earlier, and hence be more amenable to sequential decision-making, unlike the case where one would have to wait longer for the termination of all the replications done in parallel.

One must be careful, however, in implementing such a scheme naively, because of the difference between simulated time and computer time. The problem here is the following: say you wanted to optimize the (s, S) inventory system over one month. If you used one processor for each simulation replication — probably not practical at this point — and then at the end utilize some ranking and selection or multiple comparisons analysis, then the procedure presents no problems. However, it is more likely that the process will be sequential, i.e., you may want to simulate as many replications as possible for a given computer budget. The naive implementation would be to run all the replications on each of the processors until the termination condition on budget is reached. Unfortunately, as pointed out in [41], this leads to bias. Intuitively, the reason is that due to the difference between simulated time and computer run time, each processor is likely to end up processing a different number of replications, and the estimates from small — in terms of events – runs will be biased over long runs. So, for a given CPU time constraint, on one processor, it may produce six replications, whereas on another it may produce twelve replications. In a sense, it is like the inspection paradox in renewal theory. There, if one "arrives" at random, one is likely to land in a longer interval; here, if one picks a replication at random, one is likely to pick a smaller (in terms of computation time) replication. A simple example is the following.

Let X be the output random variable from a simulation. If X is Bernoulli with parameter p, i.e.,

$$X = \begin{cases} 1 & \text{w.p. } p \\ 0 & \text{w.p. } 1-p \end{cases},$$

then $E[X] = p$. Now assume that X is jointly distributed with CPU time as follows:

$$X = \begin{cases} 1 \text{ and simulation run time} < T & \text{w.p. } p \\ 0 \text{ and simulation run time} > T & \text{w.p. } 1-p \end{cases}.$$

If we run simulations in parallel for CPU time constraint T and take the sample mean of our output random variable over all completed replications, then our estimate will be 1 w.p. 1, which is biased (unless $p = 1$, of course).

An even simpler example is a geometrically distributed random variable, where the simulation is to determine, say, the expected value. If we simulate the process as a sequence of independent Bernoulli trials, and terminate at some finite number of tosses on each processor, we will encounter the same problem of long values becoming under-represented minorities.

Efficiently generating these parallel replications for optimization purposes is another important topic of research. "Cut-and-paste" variants of perturbation analysis, where portions of one sample path at a given parameter value can be used to construct sample paths at other values of the parameter, are one approach [56, 14]. Another approach uses a single clock mechanism, called the Standard Clock, which drives all replications simultaneously, but at different values of the parameter [57, 122, 123]. The Standard Clock is based upon the ideas of uniformization (sometimes also called randomization) for Markov chains, and is thus valid for Markov chain representation of systems. The resulting replications are coupled, which can result in variance reduction, as well as orders of magnitude of speedup. The scheme has been implemented on a Single Instruction Multiple Data (SIMD) Thinking Machines Connection Machine CM-2, with each processor element simulating one variant of the discrete-event system. In fact, we note that a key feature in the proposed method of ordinal optimization touched on earlier [59] was the exploitation of induced positive correlation between parallel replications to make the "weeding out" of poor candidates easier and quicker, massively parallel processors being the implementation in mind.

7. Conclusions and Future Directions

Simulation is the most general technique for modeling stochastic discrete-event systems, but that does not mean that each simulation model should be treated as a "black box." Exploitation of system structure is as pertinent for stochastic models as it is for deterministic models, where

for example, almost every problem can be formulated as a mathematical program, but without exploiting special structure this would oftentimes lead to computational intractability. Response surface methodology and ranking-and-selection procedures can be viewed as general purpose tools, which can always be applied, whereas gradient estimation techniques such as perturbation analysis and the likelihood ratio method can be viewed as special purpose tools, which when they are applicable increase efficiency of optimization algorithms tremendously; as we have said, they are the potential assignment problem modules in a general purpose linear programming code. The development of special purpose tools for discrete-event simulation and optimization involve more detailed analysis of discrete-event simulation models, whereas the general purpose tools have been traditionally more detached from the application, being based on an arsenal of well-developed statistical procedures.

The state of the art is far from satisfying. Practical implementations have not attacked large problems, because some of the procedures find themselves computationally impractical once the number of parameters becomes too large. In this author's opinion, the future directions in the optimization of discrete-event systems using simulation lies in two directions:

- **Gradient-based algorithms** (whether stochastic approximation or response surface methodology).

  The important areas of research are work on the algorithms themselves, including implementation with global optimization schemes and step size determination in conjunction with observation lengths; work on gradient estimation techniques – both new algorithms and finding a systematic way to determine when each is applicable; combining various algorithms, e.g., RSM and SA, or using those two to reduce the number of alternatives and then simply applying the ranking-and-selection procedures. In addition, there has been little work on second-order techniques, due chiefly to the difficulty in obtaining the necessary estimates efficiently and accurately. Perhaps ideas from nonlinear deterministic optimization could be fruitfully adapted to the stochastic setting.

- **Massively parallel processors**.

  This is a new and very open problem, and we have barely scratched the surface here. It is not clear whether the major advances will come from distributed simulation or from simply exploiting simultaneously generated parallel replications through induced correlation implementations such as the Standard Clock. These advances should open up exciting new research possibilities, and potential directions include either adapting existing techniques to

this new environment, or in fact coming up with new paradigms more suitable for exploiting parallelism.

Lastly, it may be useful to put the optimization problem itself into an optimization context. In addition to proving convergence of the algorithms as the number of iterations goes to infinity (the focus of a large number of papers on stochastic approximation), it may be more practical to consider the problem of doing the best under a constraint on resources. For example, for a given computational budget, which algorithm will be most likely to find the parameter value that minimizes the expected value of the performance of interest, or comes with a certain range of it? Although some work has been done on this topic in the past, it has been mostly empirical. A formal framework for comparing algorithms would be useful.

Bibliography

[1] V.M. Aleksandrov, V.I. Sysoyev and V.V. Shemeneva, Stochastic optimization, *Engineering Cybernetics*, **5**, (1968) 11-16.

[2] A.E. Albert and L.A. Gardner, Jr., *Stochastic Approximation and Nonlinear Regression* (Research Monograph No. 42, M.I.T. Press, Cambridge, MA, 1967).

[3] S. Andradóttir, A new algorithm for stochastic approximation, *Proceedings of the 1990 Winter Simulation Conference*, (1990), pp. 364-366.

[4] F. Azadivar and J.J. Talavage, Optimization of stochastic simulation-models, *Mathematics and Computers in Simulation*, **22**, (1980), 231-241.

[5] J. Banks and J.S. Carson, *Discrete-Event System Simulation*, (Prentice-Hall, Englewood Cliffs, NJ, 1984).

[6] R.E. Bechhofer, A single-sample multiple decision procedure for ranking means of normal populations with known variances, *Annals of Mathematical Statistics*, **25** (1954) 16-39.

[7] A. Benveniste, M. Metivier and P. Priouret, *Adaptive Algorithms and Stochastic Approximation*, (Springer-Verlag, New York, 1990).

[8] W.E. Biles and J.J. Swain, *Optimization and Industrial Experimentation*, (Wiley-Interscience, New York, 1980).

[9] R.E. Bixby, J.W. Gregory, I.J. Lustig, R.E. Marsten, and D.F. Shanno, Very large-scale programming: a case study in combining interior point and simplex methods, *Operations Research*, **25**, (1992) 885-897.

[10] G E.P. Box and N.R. Draper, *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, (1987).

[11] P. Brémaud and F. Vázquez-Abad, On the pathwise computation of derivatives with respect to the rate of a point process, *Queueing Systems: Theory and Applications,* to appear (1993).

[12] M. Caramanis, and G. Liberopolous, Perturbation analysis for the design of flexible manufacturing flow controllers, *Operations Research*, **40**, (1992) 1107-1126.

[13] G. Liberopolous and M. Caramanis, Infinitesimal perturbation analysis for second derivative estimation and design of manufacturing flow controllers, *Journal of Optimization Theory and Applications*, submitted (1993).

[14] C. G. Cassandras, and S. G. Strickland, On-line sensitivity analysis of Markov chains, *IEEE Transactions on Automatic Control,* **AC-34**, (1989) 76-86.

[15] E. K. P. Chong and P. J. Ramadge, Convergence of recursive optimization algorithms using infinitesimal perturbation analysis, *Discrete Event Dynamic Systems: Theory and Applications*, **1** (1992) 339-372.

[16] E. K. P. Chong and P. J. Ramadge, Optimization of queues using an ipa based stochastic algorithm with General Update Times, *SIAM Journal on Control and optimization*, to appear (1993).

[17] K. L. Chung, On a stochastic approximation method, *Annals of Mathematical Statistics*, **25** (1954) 463-483.

[18] W. G. Cochran and G. M. Cox, *Experimental Designs*, 2nd Ed., Wiley, New York, (1957).

[19] E. J. Dudewicz and S. R. Dalal, Allocation of measurements in ranking and selection with unequal variances, *Sankhya* **B37**, (1975) 28-78.

[20] P. Dupuis and R. Simha, On sampling controlled stochastic approximation, *IEEE Transactions on Automatic Control*, **36**, (1991) 915-924.

[21] W. Farrell, Literature review and bibliography of simulation optimization, *Proceedings of the Winter Simulation Conference*, (1977), pp.117-124.

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

[23] M. C. Fu, Sample path derivatives for (s,S) inventory systems, to appear in *Operations Research*, (1993).

[24] M. C. Fu and K. Healy, Simulation optimization of (s,S) inventory systems, *Proceedings of the Winter Simulation Conference*, (1992), pp.506-514.

[25] M. C. Fu and Y. C. Ho, Using perturbation analysis for gradient estimation, averaging, and updating in a stochastic approximation algorithm, *Proceedings of the Winter Simulation Conference*, (1988), pp. 509-517.

[26] M. C. Fu and J. Q. Hu, Extensions and generalizations of smoothed perturbation analysis in a generalized semi-Markov process framework, *IEEE Transactions Automatic Control*, **37**, (1992) 1483-1500.

[27] M. C. Fu, and J. Q. Hu, On choosing the characterization for smoothed perturbation analysis, *IEEE Trans. Automatic Control*, **36**, (1991) 1331-1336.

[28] M. C. Fu and J. Q, Hu, Second derivative sample path estimators for the GI/G/m Queue, *Management Science*, **39**, (1993) 359-383.

[29] M. C. Fu, J. Q. Hu and L. Shi, Likelihood ratio methods with conditional Monte Carlo and splitting, *ACM Transactions on Modeling and Computer Simulation*, submitted (1992).

[30] R. Fujimoto, Parallel discrete event simulation, *Communications of the ACM*, **33**, (1990) 30-53.

[31] A. Gaivoronski, Optimization of stochastic discrete event dynamic systems: a survey of some recent results, *Lecture Notes in Economics and Mathematical Systems*, **374**, Proceedings of a workshop on Simulation and Optimization, Laxenburg, Austria, August 1990, G. Pflug and U. Dieter (Eds.), (1992), pp. 24-44.

[32] P. Glasserman, *Gradient Estimation Via Perturbation Analysis*, Kluwer Academic, (1991).

[33] P. Glasserman, Structural conditions for perturbation analysis derivative estimation: finite-time performance indices, *Operations Research*, **39**, (1991) 724-738.

[34] P. Glasserman and W. B. Gong, Smoothed perturbation analysis for a class of discrete event systems, *IEEE Transactions on Automatic Control*, **AC-35**, (1990) 1218-1230.

[35] P. Glasserman and D. D. Yao, Some guidelines and guarantees for common random numbers, *Management Science*, **38**, (1992) 884-908.

[36] P. W. Glynn, Likelihood ratio gradient estimation: an overview, *Proceedings of the Winter Simulation Conference*, (1987), pp. 366-374.

[37] P. W. Glynn, Likelihood ratio gradient estimation for stochastic systems, *Communications of the ACM*, **33**, (1990) 75-84.

[38] P. W. Glynn, Optimization of stochastic systems, *Proceedings of the Winter Simulation Conference*, (1986), pp. 52-59.

[39] P. W. Glynn, Optimization of stochastic systems via simulation, *Proceedings of the Winter Simulation Conference*, (1989), pp. 90-105.

[40] P. W. Glynn, Stochastic approximation for Monte Carlo optimization, *Proceedings of the Winter Simulation Conference*, (1986), pp. 356-364.

[41] P. W. Glynn and P. Heidelberger, Analysis of parallel replicated simulations under a completion time constraint, *ACM Transactions on Modeling and Computer Simulation*, **1**, (1991) 3-23.

[42] D. Goldsman, B. L. Nelson and B. Schmeiser, Methods for selecting the best system, *Proceedings of Winter Simulation Conference*, (1991), pp. 177-186.

[43] W. B. Gong and Y. C. Ho, Smoothed perturbation analysis of discrete-event dynamic systems, *IEEE Transactions on Automatic Control*, **AC-32**, (1987) 858-867.

[44] S.S. Gupta, On a decision rule for ranking means, Inst. Statist. Mimeo. Ser. No. 150, University of North Carolina, Chapel Hill, NC (1956).

[45] J. Haddock and G. Bengu, Application of a simulation optimization system for a continuous review inventory model, *Proceedings of the Winter Simulation Conference*, (1987), pp. 382-390.

[46] K. Healy and L.W. Schruben, Retrospective simulation response optimization. *Proceedings of the 1991 Winter Simulation Conference*: (1991), pp. 901-906.

[47] K. Healy, Retrospective simulation response optimization, Ph.D. dissertation, Cornell University, (1992).

[48] Y. C. Ho, Special issue on dynamics of discrete event systems, *Proceedings of the IEEE*, **77**, No.1 (1989).

[49] Y. C. Ho, Perturbation analysis: concepts and algorithms, *Proceedings of the Winter Simulation Conference*, (1992), pp. 231-240.

[50] Y. C. Ho and X. R. Cao, *Discrete Event Dynamic Systems and Perturbation Analysis,* Kluwer Academic, (1991).

[51] Y. C. Ho, and X. R. Cao, Optimization and perturbation analysis of queueing networks, *Journal of Optimization Theory and Applications,* **40**, (1983) 559-582.

[52] Y. C. Ho, X. R. Cao and C. G. Cassandras, Infinitesimal and finite perturbation analysis for queueing networks, *Automatica,* **19**, (1983) 439-445.

[53] Y. C. Ho and C. Cassandras, A new approach to the analysis of discrete event dynamic systems, *Automatica*, **19**, (1983) 149-167.

[54] Y. C. Ho, A. Eyler and T. T. Chien, A gradient technique for general buffer storage design in a serial production line, *International Journal of Production Research,* **17**, (1979) 557-580.

[55] Y. C. Ho, M. A. Eyler and T. T. Chien, A new approach to determine parameter sensitivity of transfer lines, *Management Science*, **29**, (1983) 700-714.

[56] Y. C. Ho and S. Li, Extensions of perturbation analysis of discrete-event dynamic systems, *IEEE Transactions on Automatic Control*, **AC-33**, (1988) 427-438.

[57] Y. C. Ho, S. Li and P. Vakili, On the efficient generation of discrete event sample paths under different system parameters, *Mathematics and Computers in Simulation*, **30**, (1988) 347-370.

[58] Y. C. Ho, L. Shi, L. Dai and W. B. Gong, Optimizing discrete event systems via the gradient surface method, *Discrete-Event Dynamic Systems: Theory and Applications*, **2**, (1992) 99-120.

[59] Y. C. Ho, R. Sreenevas and P. Vakili, Ordinal optimization of DEDS, *Discrete-Event Dynamic Systems: Theory and Applications*, **2**, (1992) 61-88.

[60] Y. Hochberg and A. C. Tamhane, *Multiple Comparison Procedures*, Wiley, (1987).

[61] J. C. Hsu, Simultaneous confidence intervals for all distributions from the best, *Annals of Statistics,* **9**, (1981) 1026-1034.

[62] J. C. Hsu and B.L. Nelson, Optimization over a finite number of system designs with one-stage sampling and multiple comparisons with the best, *Proceedings of the Winter Simulation Conference*, (1988), pp. 451-457.

[63] J. S. Hunter and T. H. Naylor, Experimental designs for computer simulation experiments, *Management Science*, **16**, (1970) 422-434.

[64] S. H. Jacobson, Optimal mean squared error analysis of the harmonic gradient estimators, to appear in *Journal of Optimization Theory and Applications*, (1994).

[65] S. H. Jacobson, Oscillation amplitude consideration in frequency domain experiments, *Proceedings of Winter Simulation Conference*, (1989), pp.406-410.

[66] S. H. Jacobson, Variance and bias reduction techniques for harmonic gradient estimators, to appear in *Applied Mathematics and Computation*, (1993).

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

[68] S. H. Jacobson, and L.W. Schruben, A simulation optimization procedure using harmonic analysis, submitted to *Operations Research*, (1992).

[69] S. H. Jacobson, A. Buss and L. W. Schruben, Driving frequency selection for frequency domain simulation experiments, *Operations Research*, (1991) 917-924.

[70] S. H. Jacobson, D. Morrice and L. W. Schruben, The global simulation clock as the frequency domain experiment index, *Proceedings of Winter Simulation Conference*, (1988), pp.558-563.

[71] H. Kesten, Accelerated stochastic approximation, *Ann. Math. Stat.,* **29**, (1958) 41-59.

[72] A. I. Khuri and J. A. Cornell, *Response Surfaces*, Marcel Dekker, (1987).

[73] J. Kiefer and J. Wolfowitz, Stochastic estimation of the maximum of a regression function, *Ann. Math. Stat.,* **23**, (1952) 462-466.

[74] J. P. C. Kleijnen, *Statistical Tools for Simulation Practitioners*, New York, Marcel Dekker, (1987).

[75] H. J. Kushner and D. C. Clark, *Stochastic Approximation Methods for Constrained and Unconstrained Systems,* Springer-Verlag, New York, (1978).

[76] A. M. Law and W. D. Kelton, *Simulation Modeling and Analysis*, 2nd edition, McGraw-Hill, New York, (1991).

[77] P. L'Ecuyer, A unified view of the IPA, SF, and LR gradient estimation techniques, *Management Science*, **36**, (1990) 1364-1383.

[78] P. L'Ecuyer, An overview of derivative estimation. *Proceedings of the 1991 Winter Simulation Conference*: (1991), pp. 207-217.

[79] P. L'Ecuyer, Convergence rates for steady-state derivative estimators, *Annals of Operations Research*, **39** (1992), 121-136.

[80] P. L'Ecuyer and P. W. Glynn, Stochastic optimization by simulation: convergence proofs for the GI/G/1 queue in steady-state, submitted to *Management Science* (1993).

[81] P. L'Ecuyer, N. Giroux, N., and P. W. Glynn, Stochastic optimization by simulation: numerical experiments with a simple queue in steady-state, submitted to *Management Science* (1993).

[82] P. L'Ecuyer and G. Perron, On the convergence rates of IPA and FDC derivative estimators for finite-horizon stochastic simulations, 1993, submitted to *Operations Research*.

[83] Y. T. Leung and R. Suri, Finite-time behavior of two simulation optimization algorithms, *Proceedings of the Winter Simulation Conference*, (1990), pp.372-376.

[84] L. Ljung, Analysis of recursive stochastic algorithms, *IEEE Transactions Automatic Control*, **AC-22**, (1977) 551-575.

[85] L. Ljung, Strong convergence of a stochastic approximation algorithm, *Annals of Statistics*, **6**, (1978) 680-696.

[86] M. Mitra and S. K. Park, Solution to the indexing problem of frequency domain simulation experiments, *Proceedings of the Winter Simulation Conference*, (1991), pp.907-915.

[87] R. H. Myers, A. I. Khuri and W. H. Carter, Response surface methodology: 1966-1988, *Technometrics*, **31**, (1989) 137-157.

[88] R. H. Myers, *Response Surface Methodology*, Allyn and Bacon, Boston, (1971).

[89] M. B. Nevelson and R. Zalmanovich, Stochastic approximation and recursive estimation, translation of mathematical monographs, Vol. 47, (1973).

[90] G. Ch. Pflug, On-line optimization of simulated Markovian processes, *Math. of Operations Research*, **15**, (1990) 381-395.

[91] G. Ch. Pflug, Sampling derivatives of probabilities, *Computing*, **42**, (1989) 315-328.

[92] B. T. Polyak, New method of stochastic approximation type, *Automation and Remote Control*, **51** (1990) 937-946.

[93] M. I. Reiman and A. Weiss, Sensitivity analysis via likelihood ratios, *Proceedings of the Winter Simulation Conference,* (1986), pp.285-289.

[94] M. I. Reiman and A. Weiss, Sensitivity analysis for simulations via likelihood ratios, *Operations Research,* **37**, (1989) 830-844.

[95] H. Robbins and S. Monro, A stochastic approximation method, *Ann. Math. Stat.,* **22**, (1951) 400-407.

[96] K. W. Ross and J. Wang, Solving product form stochastic networks with Monte Carlo summation, *Proceedings of the Winter Simulation Conference,* (1990), pp.270-275.

[97] R. Y. Rubinstein, How to optimize discrete-event systems from a single sample path by the score function method. *Annals of Operations Research* **27**, (1991), pp.175-212.

[98] R. Y. Rubinstein, The push out method for sensitivity analysis of discrete event systems, *Annals of Operations Research*, (to appear) (1993).

[99] R. Y. Rubinstein, Sensitivity analysis and performance extrapolation for computer simulation models, *Operations Research,* **37**, (1989) 72-81.

[100] R. Y. Rubinstein, *Monte-Carlo Optimization: Simulation and Sensitivity of Queueing Networks*, Wiley, (1986).

[101] R. Y. Rubinstein and A. Shapiro, *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization by the Score Function Method*, to appear, Wiley, (1992).

[102] D. Ruppert, Almost sure approximations to the Robbins-Monro and Kiefer-Wolfowitz processes with dependent noise, *Annals of Statistics*, **16**, (1982) 178-187.

[103] A. Ruszczynski and W. Syski, Stochastic approximation method with gradient averaging for unconstrained problems, *IEEE Transactions on Automatic Control*, **AC-28**, (1983) 1097-1105.

[104] J. Sacks, Asymptotic distribution of stochastic approximation procedures, *Ann. Math. Stat.,* **29**, (1958) 373-405.

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

[106] M. H. Safizadeh and B. M. Thornton, Optimization in simulation experiments using response surface methodology, *Computers and Industrial Engineering*, **8**, (1984) 11-27.

[107] J. Santner and A. C. Tamhane, *Design of Experiments: Ranking and Selection*, Marcel Dekker, (1984).

[108] R. G. Sargent, Research issues in metamodeling, *Proceedings of the Winter Simulation Conference*, (1991), pp.888-893.

[109] R. G. Sargent and T. K. Som, Current issues in frequency domain experimentation, *Management Science*, **38**, (1992) 667-687.

[110] L. W. Schruben, Simulation optimization using frequency domain methods, *Proceedings of the Winter Simulation Conference*, (1986), pp. 366-369.

[111] L. W. Schruben and V. Cogliano, An experimental procedure for simulation response surface model identification, *Communications of the Association for Computing Machinery*, **30**, (1987). 716-730.

[112] L. W. Schruben and V. J. Cogliano, Simulation sensitivity analysis: a frequency domain approach, *Proceedings of the Winter Simulation Conference*, (1981), pp.455-459.

[113] B. Schmeiser, Simulation output data analysis, *Annals of Operations Research*, in this volume (1994).

[114] L. Y. Shi, Discontinuous perturbation analysis of discrete event dynamic systems, submitted to *IEEE Transactions on Automatic Control*, (1993).

[115] D. E. Smith, An empirical investigation of optimum-seeking in the computer simulation situation, *Operations Research*, **21**, (1973) 475-497.

[116] D. E. Smith, Automatic optimum-seeking program for digital simulation, *Simulation* **27**, (1976) 27-32.

[117] D. E. Smith, Requirements of an optimizer for computer simulation, *Naval Research Logistics Quarterly*, **20**, (1973) 161-179.

[118] D. W. Sullivan and J. R. Wilson, Restricted subset selection procedures for simulation, *Operations Research*, **37**, (1989) 52-71.

[119] R. Suri, Perturbation analysis: the state of the art and research issues explained via the G/G/1 queue, *Proceedings of the IEEE,* **77**, (1989) 114-137.

[120] R. Suri and Y. T. Leung, Single run optimization of discrete event simulations – an empirical study using the M/M/1 queue, *IIE Transactions*, **21**, (1991) 35-49.

[121] R. Suri and M. Zazanis, Perturbation analysis gives strongly consistent sensitivity estimates for the M/G/1 queue, *Management Science*, **34**, (1988) 39-64.

[122] P. Vakili, Using a standard clock technique for efficient simulation, *Operations Research Letters*, (1991) 445-452.

[123] P. Vakili, Massively parallel and distributed simulation of a class of discrete event systems : a different perspective, *ACM Transactions on Modeling and Computer Simulation*, to appear (1993).

[124] F. Vázquez-Abad and P. L'Ecuyer, Comparing alternative methods for derivative estimation when IPA does not apply directly, *Proceedings of the Winter Simulation Conference,* (1991), pp.1004-1011.

[125] Y. Wardi, Simulation-based stochastic algorithm for optimizing GI/G/1 queues, manuscript, Department of Industrial Engineering, Ben Gurion University, (1988).

[126] M. T. Wasan, *Stochastic Approximation*, Cambridge University Press, (1969).

[127] J. R. Wilson, Future direction in response surface methodology for simulation, *Proceedings of the Winter Simulation Conference*, (1987), pp. 378-381.

[128] S. Yakowitz, A globally convergent stochastic approximation, *SIAM Journal on Control and Optimization*, , to appear, (1993).

[129] S. Yakowitz, T. Jayawardena and S. Li, Theory for automatic learning under partially observed Markov-dependent noise, *IEEE Transactions on Automatic Control*, **37**, (1992) 1316-1324.

[130] W.N. Yang and B.L. Nelson, Using common random numbers and control variates in multiple-comparison procedures, *Operations Research*, **39**, (1991) 583-591.

[131] G. Yin, Stochastic approximation via averaging: the Polyak approach revisited, *Lecture Notes in Economics and Mathematical Systems*, **374** Proceedings of a workshop on Simulation and Optimization, Laxenburg, Austria, August 1990, G. Pflug and U. Dieter (Eds.), (1990), pp.119-134.

[132] M. Zazanis and R. Suri, Estimating first and second derivatives of response time for G/G/1 queues from a single sample path, Technical Report, Northwester University, Evanston, Illinois, also submitted to *Queueing Systems,* (1986).

[133] B. Zhang and Y. C. Ho, Performance gradient estimation for very large finite Markov chains, *IEEE Transactions on Automatic Control*, **36**, (1991) 1218-1227.

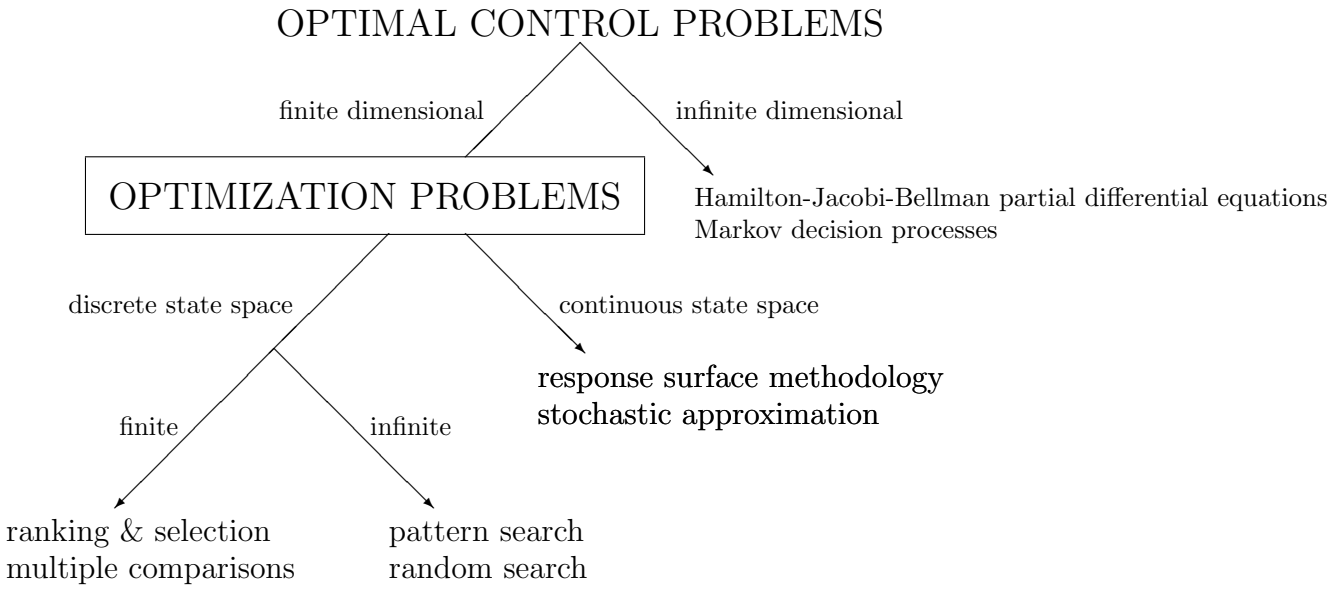

Figure 1: Classification of Optimization Problems and Solution Methodologies.

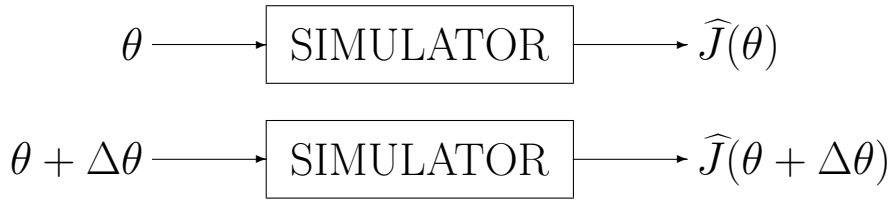

Figure 2: Traditional Approach to Gradient Estimation.

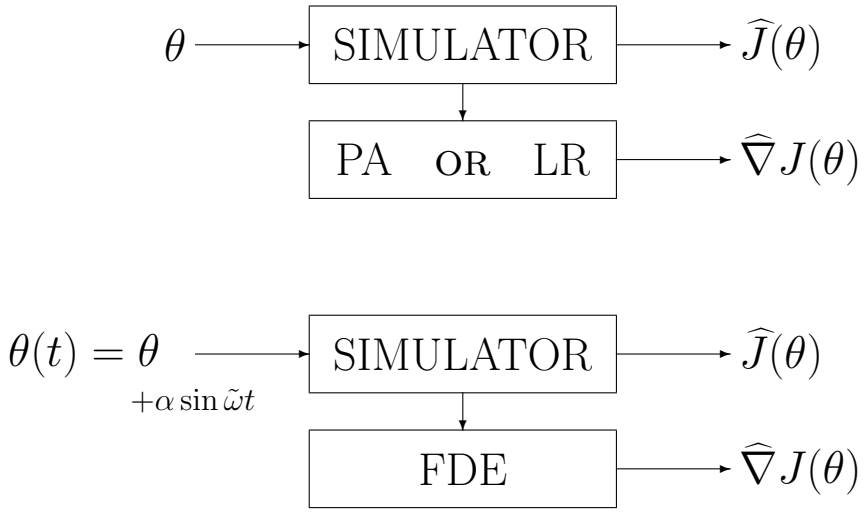

Figure 3: New Wave Approaches to Gradient Estimation.

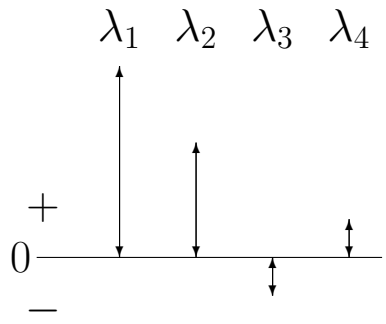

Figure 4: Confidence Intervals for MCB: λ_3 is optimal.

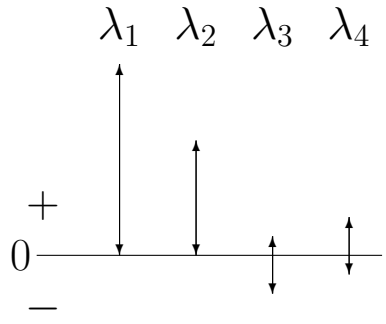

Figure 5: Confidence Intervals for MCB: either λ_3 or λ_4 could be optimal.

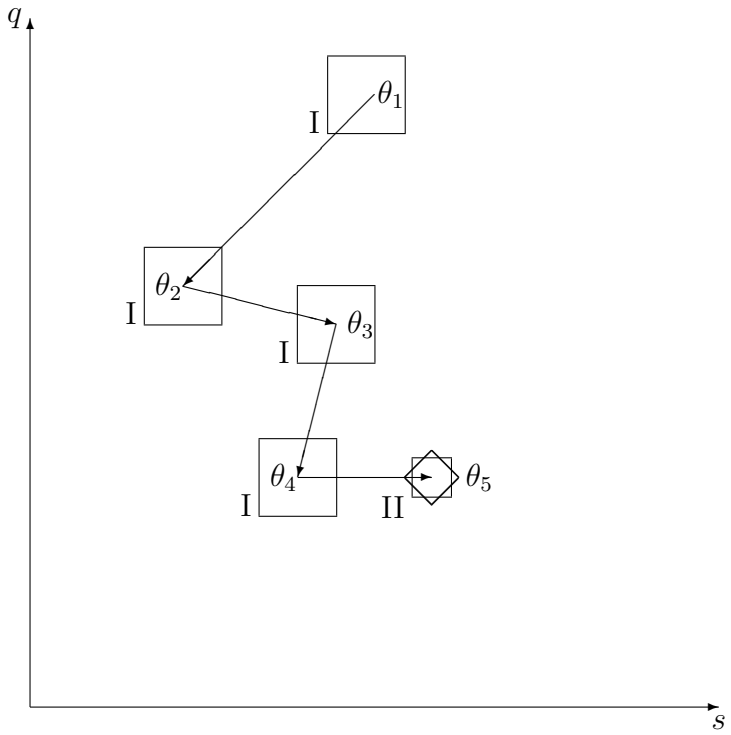

Figure 6: Illustration of the Sequential RSM Procedure.

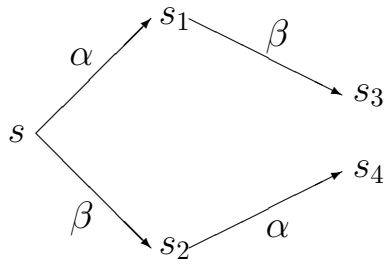

Figure 7: The Commuting Condition.

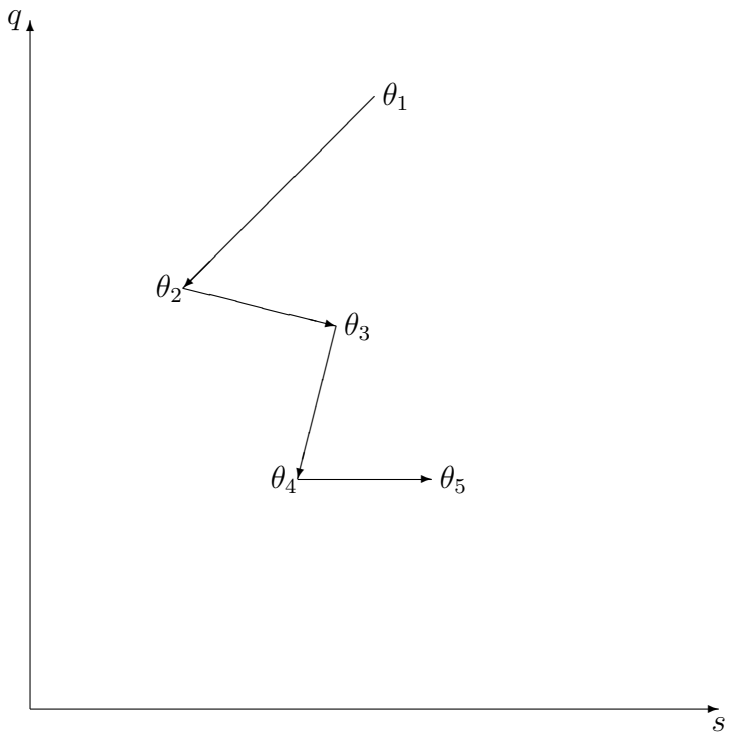

Figure 8: Illustration of the Traditional Stochastic Approximation Algorithm.

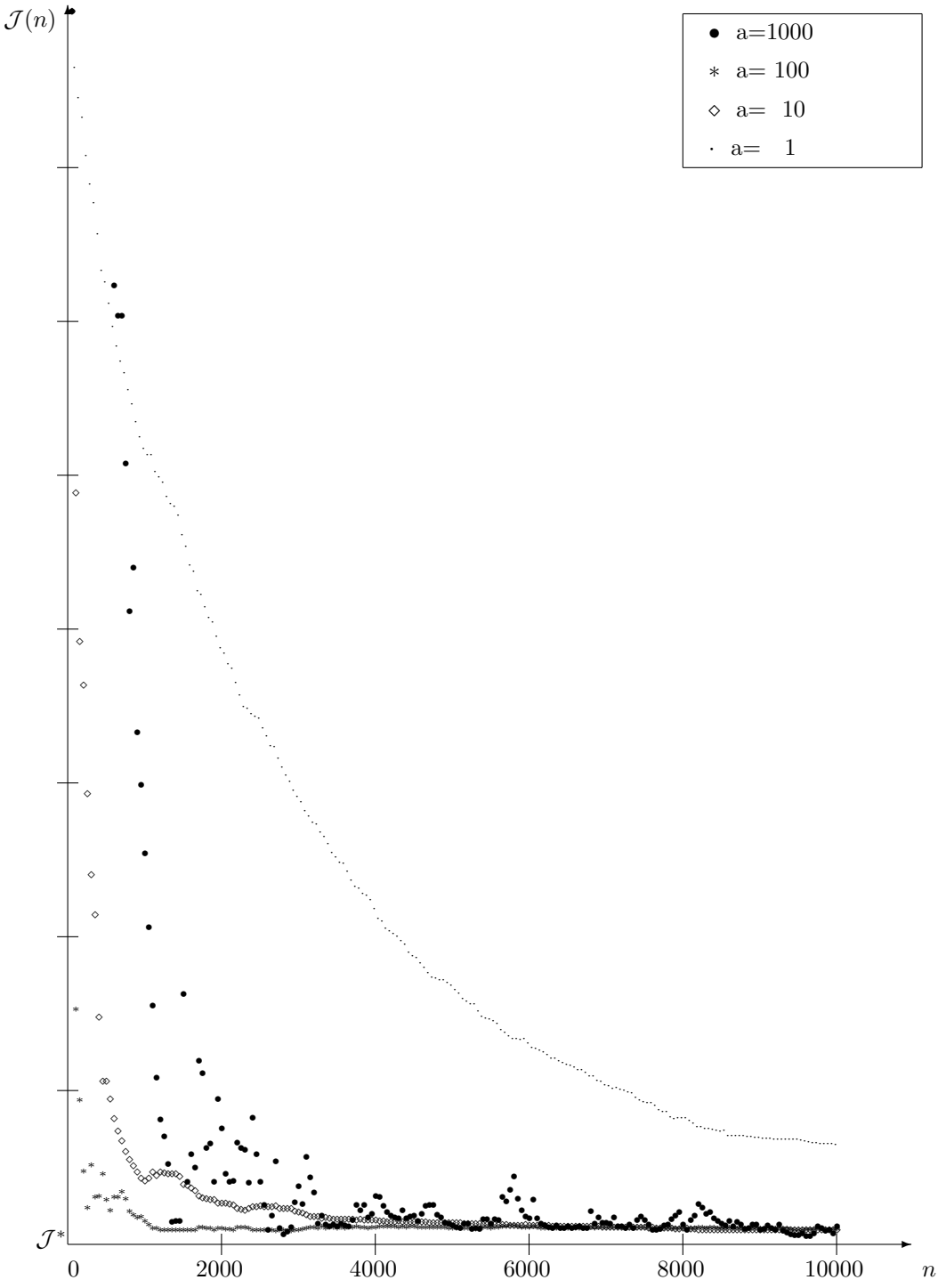

Figure 9: Effect of Choice of Initial Step Size a (parameter updates every 50 periods).