Proceedings of the 1993 Winter Simulation Conference
G.W. Evans, M. Mollaghasemi, E.C. Russell, W.E. Biles (eds.)

GRADIENT/SENSITIVITY ESTIMATION IN DISCRETE-EVENT SIMULATION

Stephen G. Strickland

Dept. of Systems Engineering
University of Virginia
Charlottesville, VA 22903, U.S.A.

ABSTRACT

We consider methods for simultaneously estimating the gradient (resp. sensitivity) of J with respect to a continuous (resp. discrete) parameter θ. While one can always estimate a derivative (or sensitivity) by performing two distinct simulations at different values of θ (and forming the estimate $(J_{\theta'} - J_{\theta_2})/(\theta' - \theta)$), here we focus on methods which compute estimates from a single sample path (though multiple replications can be used to reduce variance and compute confidence intervals). When applicable, these *single run* methods are often computationally more efficient. They also have applications beyond simulation (e.g. in on-line optimization and control).

1 Introduction

In this paper, we give a tutorial overview of methods for estimating gradients and sensitivities of performance measures themselves estimated via simulation. We focus on *single run methods*—where the estimates can be computed from a single simulation run.

To fix the basic idea, let X denote an exponentially distributed random variable with mean θ. Letting F denote its distribution, we can write its expectation in the usual way as

$$\mathbb{E}[X] = \int_X x\, dF_X(x,\theta) = \int_0^\infty x\, \frac{1}{\theta} e^{-x/\theta} dx.$$

Assuming we can interchange differentiation and integration, the derivative $d\mathbb{E}[X]/d\theta$ can be written as

$$\begin{aligned}\frac{d\mathbb{E}[X]}{d\theta} &= \frac{d}{d\theta}\int_X x\, dF_X(x,\theta) \\ &= \int_0^\infty x \frac{d}{d\theta}\left[\frac{1}{\theta}e^{-x/\theta}\right] dx \\ &= \int_0^\infty x\left[\frac{x}{\theta^2} - \frac{1}{\theta}\right]\frac{1}{\theta}e^{1x/\theta} dx \\ &= \int_X x\left[\frac{x}{\theta^2} - \frac{1}{\theta}\right] dF_X(x,\theta).\end{aligned}$$

This gives the derivative as the expectation of a function of X with respect to the original probability distribution of X. Thus we can estimate the derivative $d\mathbb{E}[X]/d\theta$ from samples $\{x_1, \ldots, x_n\}$ of X by

$$\frac{1}{n}\sum_{i=1}^n x_i\left[\frac{x_i}{\theta^2} - \frac{1}{\theta}\right].$$

This is the basis of the *likelihood ratio* (LR) method (also called the *score function* method).

Alternatively, we can view X as being generated from a uniform random variable U via the inverse transform method (the usual case in simulation). Then $x(u) = F^{-1}(u) = -\theta \ln(1-u)$ and we can write the expectation with respect to U as

$$\mathbb{E}[X] = \int_U F^{-1}(u)\, du = \int_0^1 [-\theta \ln(1-u)]\, du.$$

Again, assuming we can differentiate through the integral, we can write the derivative as

$$\begin{aligned}\frac{d\mathbb{E}[X]}{d\theta} &= \frac{d}{d\theta}\int_U x(u)\, du \\ &= \int_U \frac{d}{d\theta}F^{-1}(u)\, du \\ &= \int_0^1 [-\ln(1-u)]\, du \\ &= \int_U \frac{x(u)}{\theta} du.\end{aligned}$$

Again we have the derivative as the expectation of a function of X, but now the expectation is with respect to an underlying, parameter-independent distribution (the uniform). This is the essential idea underlying *infinitesimal perturbation analysis* (IPA).

In both the above methods, we obtain an estimator by differentiating the expectation integrand and in both cases we get an estimator which is a function of the observation x. They differ in the way we write the original expectation. In the LR approach, the parameter θ appears in the probability measure, whereas in the IPA case it appears in the observation functional. Straightforward calculations show that both estimators are unbiased.

It is useful to contrast these approaches with a finite-difference (FD) approach based on simulation runs at two different values of θ. The single run methods avoid both the "twin numerical evils" of approximation error (approximating the slope of a curve by its secant) and numerical noise (dividing by small $\Delta\theta$). Any FD method involves a tradeoff between these two effects. In addition, the computational cost of the single-run estimators is generally much less than the cost of an additional simulation run. This last effect is magnified if multiple derivatives are of interest (i.e. when θ is a vector of parameters).

The key issues in applying these methods are:

- When do they "work", i.e. yield unbiased and consistent estimators?
- How well do they work, i.e. what is the variance of the resulting estimators? When both methods work, which should we use?
- How can we calculate the estimators for performance measures defined on sample paths of the stochastic processes typically generated in simulation?

The following sections explore these questions. The quick answers are as follows. The question of unbiasedness reduces to justifying the interchange of differentiation and integration used in deriving both estimators. Roughly, this is valid if the integrand is continuous in the parameter θ for each x. This condition is much more broadly satisfied for probability measures than for observation functionals, so the LR method has wider applicability. On the other hand, the variance of IPA estimators is typically much smaller than that of LR estimators. In some simple cases it can be shown that the IPA estimator has the smallest variance among all sample-based estimators. In both cases, consistency of the estimators, as the sample path length (i.e. simulation run duration) increases, requires some form of regeneration of the estimator process. Both estimators can be calculated recursively via relatively straightforward methods.

Thus far we have considered a *distributional* parameter which was *continuous*. A different approach is required for *structural* parameters which are typically *discrete*. Examples include buffer capacities and routing thresholds. Here we must estimate sensitivities as in the finite difference approach. Given an observation $X(\theta)$, we want to construct an observation of $X(\theta + \Delta\theta)$. In the simulation context, this means using the information contained in a sample run at θ to construct a sample path corresponding to $\theta + \Delta\theta$. The principle idea involves the separation of the event generation process from the state updating process. In many cases, systems which differ in a structural parameter have similar event processes and we can transform the event process characterized by θ to obtain an event process consistent with $\theta + \Delta\theta$. We can then apply this process to the state update mechanism for the $\theta + \Delta\theta$ system.

Example 1 Consider a single server queueing system of capacity θ where the server is autonomous, i.e. it performs a service activity whether customers are present or not. Assume that we are interested in the sensitivity of customer waiting time with respect to θ. The arrival and service processes are clearly independent of θ and thus have the same probability law for all θ. If we simulate one such system we can use the resulting event process to simultaneously generate state trajectories for any other values of θ and compute the associated average customer waiting time. In fact we can compute any performance measure of interest. As compared to distinct simulations, we save the computational costs of random number generation, event scheduling, and event list management. Moreover, this approach could be easily applied on-line, where the event process is generated by an actual system rather than a simulation.

Generalizations of this approach involve formulating a fictitious system whose event process can easily be transformed to any of a parametric family of processes. This is the idea behind the

augmented system method and the *standard clock* approach. These techniques are readily adapted to massively parallel simulation on SIMD supercomputers (such as the Connection Machine) where each processor simulates a parametric variant of the system.

The outline of this paper is as follows. In Section 2 we develop the LR and IPA estimators in more detail and describe their implementation in a simulation. In Section 3 we discuss sensitivity estimation for discrete parameters and describe techniques for efficient simultaneous generation of sample paths at a set of parameter values. Section 4 concludes with a brief overview of the literature.

2 Gradient Estimation

The general gradient estimation process in simulation involves extending the ideas above to stochastic processes. In the likelihood ration method, this extension is straightforward—in a sense. While the formulation and calculation of the derivative estimator is simple, the resulting estimator has poor variance properties—the variance is monotonically increasing in the length of the simulation run. Thus some modifications to the basic approach are required, which imposes limitations on the class of systems which can be handled.

The formulation of IPA estimators for performance measures based on stochastic processes is somewhat more complex. The key idea is to model the system by a *generalized semi-Markov process* (GSMP), which represents a sample path by a sequence of (state,holding-time) pairs. The evolution of this sequence is given recursively in terms of the input sequences of random variables (e.g. interarrival times and service times) by a set of dynamic equations. We differentiate these equations to obtain recursive equations for the sequence of holding-time derivatives, from which we can calculate the performance derivative.

2.1 Likelihood Ratio (LR) Methods

As in the introduction, we write the expectation of interest as

$$\begin{aligned}\mathbb{E}[L(X)] &= \int_X L(x)dF_X(x,\theta) \\ &= \left.\frac{d\mathbb{E}[X]}{d\theta}\right|_{\theta_0} \\ &= \lim_{\theta\to\theta_0}\frac{1}{\theta-\theta_0}\left[\int_X L(x)dF_X(x,\theta) - \int_X L(x)dF_X(x,\theta_0)\right]\end{aligned}$$

The key "trick", borrowed from importance sampling, is to rewrite the first integral as an expectation with respect to the distribution $F_X(x,\theta_0)$. Then assuming we can interchange the limit and integral, we get

$$\begin{aligned}&\left.\frac{d\mathbb{E}[X]}{d\theta}\right|_{\theta_0} \\ &= \lim_{\theta\to\theta_0}\frac{1}{\theta-\theta_0}\left[\int_X L(x)\frac{dF_X(x,\theta)}{dF_X(x,\theta_0)}dF_X(x,\theta_0) - \int_X L(x)dF_X(x,\theta_0)\right] \\ &= \int_X L(x)\lim_{\theta\to\theta_0}\left[\frac{\frac{dF_X(x,\theta)}{dF_X(x,\theta_0)}-1}{\theta-\theta_0}\right]dF_X(x,\theta_0).\end{aligned} \tag{1}$$

The likelihood ratio appearing in the above equations gives the method its name. Defining

$$\psi(x,\theta_0) = \lim_{\theta\to\theta_0}\left[\frac{\frac{dF_X(x,\theta)}{dF_X(x,\theta_0)}-1}{\theta-\theta_0}\right]$$

we get

$$\left.\frac{d\mathbb{E}[X]}{d\theta}\right|_{\theta_0} = \int_X L(x)\psi(x,\theta_0)dF_X(x,\theta_0) \tag{2}$$

which is similar to the basic formula of importance sampling and can be thought of its extension to derivatives. A different but equivalent formulation of ψ is based on the observation that

$$dF\frac{\partial}{\partial\theta}\ln dF = \frac{\partial}{\partial\theta}dF$$

so

$$\begin{aligned}\left.\frac{d\mathbb{E}[X]}{d\theta}\right|_{\theta_0} &= \int_X L(x)\frac{\partial}{\partial\theta}dF_X(x,\theta) \\ &= \int_X L(x)\frac{\partial}{\partial\theta}\ln dF_X(x,\theta)dF_X(x,\theta).\end{aligned}$$

Defining

$$\psi(x,\theta) = (\partial/\partial\theta)\ln dF_X(x,\theta),$$

we get (2). Modulo some regularity conditions, these expressions for ψ are equivalent.

Given samples $\{x^1, \ldots, x^n\}$ of the random sequence x, we compute our estimator as

$$\frac{1}{n}\sum_{k=1}^{n} L(x^k)\psi(x^k, \theta).$$

To compute ψ, we need dF_X. Typically, the x_i are independent (think of independent interarrival times and service times), so $dF_X = \prod_j dF_{X_j}$. Then

$$\begin{aligned}\psi(x^k, \theta) &= (\partial/\partial\theta) \ln \prod_i dF_{X_i^k}(x_i^k, \theta) \\ &= \sum_{i=1}^{n} (\partial/\partial\theta) \ln dF_{X_i}(x_i^k, \theta) \qquad (3) \\ &= \sum_{i=1}^{n} \psi(x_i^k, \theta),\end{aligned}$$

so the problem reduces to obtaining expressions for ψ for each of the simulation input distributions. During the simulation we evaluate the appropriate expression for each random variate generated, accumulating the results. At the end of the simulation, we multiply the result by the sample performance $L(x)$ computed over the simulation run. Thus the implementation of the LR estimator is extremely simple. Note that random variables which do not depend on θ can be ignored. They "cancel" in (1) and evaluate to zero in (3). So we can estimate derivatives for extremely complex systems with minimal computational cost if only one component process depends on θ.

Mitigating these positive features, note that the expression for ψ is a sum of random terms; thus its variance will grow with increasing n (which corresponds roughly to the "length" of the simulation run). Since, at the same time, it is reasonable to expect $L(x)$ to be converging to a constant, it is apparent that the variance of the likelihood ratio derivative estimator will grow uncontrollably with n. There are a few ways of dealing with this. First, we make make short simulation runs. If initialization bias is not significant, or the performance measure is inherently finite horizon, then this may suffice. Alternatively, if the system is regenerative, i.e. a sample path may be broken into a sequence of i.i.d. "cycles", then we can compute and average the derivatives of the individual cycles. Then the variance will decrease inversely with the number of cycles. For example, the single server queue regenerates with an arrival to an empty queue. If we are interested in the derivative of the time-average customer sojourn time, we can compute the derivative for each busy period and average. Although many systems do not regenerate in the usual sense, they can often be show to regenerate in a more generalized sense. For example, if each distribution can be expressed as a random mixture (composition) of an exponential and some other distribution, then we can generate variates so that each variate will be exponential with some probability. If we now consider any fixed state, then a geometric trials argument shows that the system will regenerate with some probability each time the state is entered. More general forms of regeneration are also possible. Finally, several approximations have been developed which break the sample path into cycles a la batch means, ignoring the coupling between cycles.

2.2 Infinitesimal Perturbation Analysis (IPA)

Let $u := \{u_1, u_2, \ldots u_n\}$ denote the finite sequence of uniform random variates generated during a simulation run and $x := \{x_1, x_2, \ldots x_n\}$ the sequence of generally distributed variates generated from u. Further, let $L(x)$ denote the sample performance measure computed by the simulation. As in the introduction, we write the expectation as

$$\mathbb{E}[L(X)] = \int_U L(x(u))du$$

where now

$$\begin{aligned}x(u) &= \{x_1(u_1), x_2(u_2), \ldots x_n(u_n)\} \\ &= \{F_{X_1}^{-1}(u_1), F_{X_2}^{-1}(u_2), \ldots, F_{X_n}^{-1}(u_n).\end{aligned}$$

Again, assuming we can differentiate through the integral, we can write the derivative as

$$\frac{d\mathbb{E}[L(X)]}{d\theta} = \int_U \frac{d}{d\theta} L(x(u))du$$

so our task is to compute $dL(x(u))/d\theta$. To see how this is done, consider the following simple example.

Example 2 Let W_i denote the system time of customer i in a single server queue, and let $\{A_i\}$ and $\{B_i\}$ the interarrival time and service sequences,

respectively. The classical Lindley Equation gives W_i recursively in terms of $\{A_i\}$ and $\{B_i\}$ as

$$W_{i+1} = \begin{cases} W_i - A_{i+1} + B_i & W_i - A_{i+1} + B_i > 0 \\ 0 & \text{otherwise.} \end{cases}$$

Formally differentiating both sides, we get

$$\frac{dW_{i+1}}{d\theta} = \begin{cases} \frac{dW_i}{d\theta} - \frac{dA_{i+1}}{d\theta} + \frac{dB_i}{d\theta} & W_i - A_{i+1} + B_i > 0 \\ 0 & \text{otherwise.} \end{cases}$$

This gives us the waiting time derivative recursively in terms of elementary random variable derivatives. These latter derivatives can be easily calculated in several ways. For example, if the distribution function F_A is continuous and strictly monotone, we can write,

$$\left.\frac{da}{d\theta}\right|_{\theta_0} = \left.\frac{d}{d\theta}F_A^{-1}(u,\theta)\right|_{\theta_0} \quad (4)$$

$$= \left.\frac{d}{d\theta}F_A^{-1}(F_A(a,\theta_0),\theta)\right|_{\theta_0}. \quad (5)$$

Alternatively, if $F(a,\theta)$ is continuously differentiable in both arguments, then

$$\frac{\partial a}{\partial \theta} = \frac{\partial F(a,\theta)/\partial\theta}{\partial F(a,\theta)/\partial a}.$$

More generally, let X_t be a stochastic process, and let the performance measure be given by

$$L_n := \int_0^{t_n} f(X_t)dt$$

where t_n is the occurrence time of the nth event. In typical simulation situations X_t is piecewise constant and can be represented by the sequence $\{(s_i,t_i)\}_{i=1,2,\ldots}$, where s_i is the ith state visited by X_t and t_i is the epoch of the jump to s_i. Then L_n becomes

$$L_n = \sum_{i=1}^{n} f(s_i)(t_i - t_{i-1}).$$

We define the sample derivative as

$$\frac{dL_n}{d\theta} := \sum_{i=1}^{n} f(s_i)\left(\frac{dt_i}{d\theta} - \frac{dt_{i-1}}{d\theta}\right).$$

Under conditions to be discussed below, this is an unbiased estimator of $d\mathbb{E}[L_n]/d\theta$.

It remains to describe how dt_i can be computed. We do this by formulating $\{(s_i,t_i)\}$ as a GSMP. A GSMP is essentially a mathematical formalization of the process of discrete-event simulation. For simplicity we will consider only GSMPs with deterministic transitions, a class which included queueing networks with deterministic routing. Such a GSMP is specified by (S,E,d,F), where S is a discrete set of states, E a finite set of event-types, d a function on $S \times E$ which specifies the state which X_t jumps to when event $e \in E$ occurs in a state $x \in X$, and $F = \{F_e\}_{e\in E}$ is a set of distribution functions—one for each event-type. In a typical application, each event-type corresponds to a particular arrival process or service process, and F_e is the associated renewal distribution (of interarrival times or service times). Note that d is usually a partial function—not all events can occur in each state—so let $E(s)$ denote the subset of E for which $d(s,e)$ is defined, i.e. the set of possible events in state s. Given this specification, a set of recursive equations specifies the construction of $\{(s_i,t_i)\}$ from sample sequences from each F_e. Define the following notation

$\alpha(e,k)$		the kth sample variate from F_e
e_n		the n-th event
t_n		the epoch of the n-th event;
τ_n		$t_n - t_{n-1}$; the holding time in s_n
C_n	=	$\{C_n(e) : e \in E(s_n)\}$, where $C_n(e)$ is the remaining clock time for e at t_n
$\mathfrak{N}(s';s,e)$	=	$E(s')\backslash(E(s)-\{e\})$, the set of new events following a transition from s to s' triggered by event e;
$\mathfrak{O}(s';s,e)$	=	$E(s')\cap(E(s)-\{e\})$, the set of old events at such a transition.
$N(e,n)$		the number of instances of e among $e_1,\ldots,e_n$

C_n corresponds to the state of the event-list at epoch n in a discrete-event simulation of X_t. Given an initial state s_1, $\{(s_i,t_i)\}$ is given recursively by

Initialization:

$$C_1(e) = \alpha(e,1) \text{ for } e \in E(s_1)$$

$$\tau_1 = \min_{e\in E(s_1)} C_1(e)$$

$$
\begin{aligned}
e_1 &= \arg \min_{e \in E(s_1)} C_1(e) \\
t_1 &= \tau_1 \\
N(e,1) &= \begin{cases} 1 & e = e_1 \\ 0 & \text{otherwise} \end{cases}
\end{aligned}
$$

Evolution:

$$
s_n = d(s_{n-1}, e_{n-1})
$$

$$
C_n(e) = \begin{cases} X(e, N(e,n-1)+1), & \text{if } e \in \mathcal{N}(s_n; s_{n-1}, e_{n-1}), \\ C_{n-1}(e) - \tau_{n-1}, & \text{if } e \in \mathcal{O}(s_n; s_{n-1}, e_{n-1}); \end{cases}
$$

$$
\begin{aligned}
\tau_n &= \min_{e \in E(s_n)} C_n(e); \\
e_n &= \arg \min_{e \in E(s_n)} C_n(e); \\
t_n &= t_{n-1} + \tau_n \\
N(e,n) &= \begin{cases} N(e,n-1)+1, & \text{if } e = e_n, \\ N(e,n-1), & \text{otherwise.} \end{cases}
\end{aligned}
$$

These equations can be regarded as a generalization of the Lindley equation. We define the derivative sequence by simply differentiating these equations. This yields

$$
\frac{dC_n(e)}{d\theta} = \begin{cases} \frac{dX(e,N(e,n-1)+1)}{d\theta}, & \text{if } e \in \mathcal{N}(s_n; s_{n-1}, e_{n-1}), \\ \frac{dC_{n-1}(e)}{d\theta} - \frac{d\tau_{n-1}}{d\theta}, & \text{if } e \in \mathcal{O}(s_n; s_{n-1}, e_{n-1}); \end{cases}
$$

$$
\begin{aligned}
\frac{d\tau_n}{d\theta} &= \frac{dC_n(e_n)}{d\theta}; \\
\frac{dt_n}{d\theta} &= \frac{dt_{n-1}}{d\theta} + \frac{d\tau_n}{d\theta}
\end{aligned}
$$

Having *defined* this derivative, we now consider when it is an unbiased estimator of $d\mathbb{E}[L_n]/d\theta$. As noted in the introduction, this question essentially reduces to the continuity of the sample path performance with respect to θ. This question is addressed in detail in Glasserman (1990). Intuitively, the idea is as follows. First, we assume that each sample from each F_e is continuous in θ. This is true, for example, if θ is a scale or location parameter and variates are generated by inversion (as in the example in the introduction). Then small changes in θ produce small changes in the $\alpha(e,k)$. Inspection of the dynamic equations shows that each t_i is the sum of a finite subset of the $\alpha(e,k)$. Moreover, Each C_i is determined by the $\alpha(e,k)$ and past values of t_i, so small changes in θ will cause small changes in t_i, and thus the C_i. For sufficiently small changes in C_i, the state sequence will be unchanged. Small changes in the t_i along with no change in s_i means a small change in L_n. On the other hand, this only holds for sufficiently small $\Delta\theta$. For any fixed $\Delta\theta$, the set of sample paths for which the above argument does not hold has positive probability. These discontinuities result from changes in the result of the "min" operations, and thus the event sequence. In sum, events will interchange order. Assume that the transition function d satisfies the following two conditions:

A1 If $e, e' \in E(s)$ with $e \neq e'$, then $e \in E(d(s,e'))$, and

A2 $d(d(s,e),e') = d(d(s,e'),e)$.

The first condition is referred to as *non-interruption*, and means that once an event becomes active (i.e. in $E(s)$) it remains active until it occurs. The second condition is called the *commuting condition* and says that each s_i is independent of the order of the preceding events. The commuting condition implies that the effect of an event order interchange is to change the state during the interval between the events. But the continuity of the t_i ensures that this interval is small. Thus the effect on L_n is small and L_n is continuous in θ. Under conditions (A1) and (A2), the IPA estimator will be unbiased (presuming continuity of the $\alpha(e,k)$).

One can largely determine if an IPA estimator is unbiased by considering the effect of event order interchanges. If the interchange of events which are infinitesimally spaced in time causes a jump in the performance measure of interest, then IPA will be biased. Thus IPA will be biased for L_n for queueing systems with finite buffers. In addition, IPA will generally be biased when the performance measures involve hitting times (e.g. the length of a busy period).

3 Sensitivity Estimation

In this section we consider estimating performance sensitivities with respect to *discrete* parameters, such as buffer capacities. This implies estimating

the system performance at two or more values of θ. We can pose this problem in two ways:

1. Given a simulation sample path at parameter θ, how can we use the information it contains to *construct* a sample path at θ'?
2. How can we *efficiently* generate sample paths at multiple values of θ *simultaneously*?

So-called *"cut & paste"* (C&P) methods address formulation 1 while the *standard clock* (SC) approach begins with formulation 2. As mentioned in the introduction, both techniques rely on the separation of the event process from the state update process. Given the GSMP formulation developed above, observe that the components (S, E, d) constitute an automaton describing the state transition behavior of the system. This automaton makes transitions in response to an event process consisting of the "superposition" of the individual renewal process characterized by F. The dependence of the event process on the automaton is captured by $E(s)$ which, roughly speaking, switches the renewal processes on and off as a function of the state.

The goal of the methods described in this section is to construct, for each value of θ, an event process with a probability law characterized by θ. This event process is then used to update the state for each system individually. The key idea is to efficiently "merge" the generation of these individual event processes.

3.1 "Cut & Paste" Methods

C&P methods do exactly that—they "cut out" portions of the event process and "paste" the remaining pieces together. Cutting corresponds to deleting the associated events in a time interval, and pasting corresponds to shifting the time of the subsequent events by the size of the deleted interval. This process is based on characterizing the probability law of an event process by $(E(s), \{a_e\}_{e \in E_s})$, where $E(s)$ is the set of active events (as described above) and a_e is the *age*, or backward renewal time of event e. This information suffices to characterize the probability law of the type of the next event and the time until it occurs. Note that the age of Poisson event processes can be ignored due to the memoryless property.

Given the event process corresponding to θ, we simultaneously update s_i and t_i for the θ' system as long as the characterizations are identical. When the characterizations deviate, we suspend the updating of the θ' system until they again coincide. This simple idea is surprisingly effective. Consider the $M/G/1/\theta$ system (θ is the buffer capacity) and let (s, s') denote the joint state of the θ and $\theta + 1$ system. If we cut the interval from the time (s, s') enters state $(0, 1)$ until the θ system exits state 0, then the resulting event process satisfies the probability law of the $\theta + 1$ system. Note that the cut points need not coincide with event occurrence epochs. This latter fact allows us to extend the above idea to the $G/G/1/\theta$ queue to construct the event process for $\theta - 1$. We cut from $(s, s') = (0, 1)$ until the θ system is in state 0 *and* the arrival age matches the value at the cut point. Of course the θ system may make some number of visits to state 0 before the age condition can be satisfied, but under mild conditions, the expected number of visits will be finite. In general, the C&P method may be used when $E(s)$ contains no more than one non-Poisson event process, and each $E(s)$ which occurs in the θ' system also occurs in the θ system (a few minor technical conditions must also be satisfied). The $G/G/1/\theta$ example just mentioned shows that this condition is not necessary.

3.2 Standard Clock Methods

When all the F_e are exponential, the characterization reduces to the set $E(s)$ so we cut when $E(s') \neq E(s)$. This suggests generating an "augmented" event process containing all the events in $E(s) \cup E(s')$. This is the idea behind the *augmented system* approach. The natural extension of this idea when a large number of values of θ are to be evaluated, is to generate an event process with all the events in $\cup_i E_{\theta_i}$, where E_{θ_i} is the set of events for the system at θ_i. Thus, event e is active if it is active in any of the system variants. From the perspective of a particular system, this is a form of *uniformization*. Notice that the Poisson intensity of this aggregate process is constant. Let m_e be the mean of F_e and define $M = \sum_e m_e$ and $\Lambda = \sum_e (1/m_e)$, where the sums are taken over all events in the aggregate process. We can generate the process by first generating a sequence of exponential variates $\{\omega_i\}$ with mean 1 (this is the *standard clock*). We then set $\tau_i = M\omega_i$ and $e_i = e$ with probability $1/(m_e\Lambda)$ (recall that τ_i is the ith interevent time and e_i the ith event). This event process can be used to update the state for each

variant with the proviso that events not in a variant's $E(s)$ are ignored by that variant.

The standard clock approach is effectively limited to Markovian systems. On the other hand, it fits nicely with parallel simulation on massively parallel supercomputers of the SIMD (single-instruction, multiple-data) architecture. These computers consist of a front-end computer that generates instructions which are executed simultaneously by an array of processors—each with its own data. In implementing a parallel simulation we can use the front end to generate the event process and associate each processor with a distinct parametric variant of the system. The processor array updates the state and computes the performance of all the variants simultaneously. Note that this approach differs substantially from older paradigms of parallel simulation which use multiple processors to generate a single sample path. The multiple sample path approach is appropriate when we need to explore the performance at a large number of parameter settings—a common situation in optimization settings, particularly when θ is a vector. Note that we trivially obtain $O(n)$ speedup.

4 Literature Overview

In these notes we emphasize references for further study of the above techniques rather than a chronological accounting of their development. The standard reference on IPA is now the book by Glasserman (1991). Many extensions exist. See the book by Ho and Cao (1991) and the papers by Ho (1987), Suri (1989), Gong and Ho (1987), Glasserman and Gong (1990), Shi (1992), and Gong, et. al. (1992). Unbiasedness and consistency properties are studied in Cao (1985), Heidelberger (1988), Suri & Zazanis (1988), Glasserman (1991), Hu & Strickland (1990), Glasserman, et al. (1991), Hu (1992), and Glasserman (1993). The GSMP is described in Glynn (1989) who also formulates the random regeneration idea described at the end of Section 2.

The likelihood ratio method is described in Rubinstein (1989), Reiman and Weiss (1998), and Glynn (1987). Reiman and Weiss (1998), Zhang & Ho (1989, 1991) and Rubinstein (1992) present variance reduction modifications. See also the new book by Rubinstein & Shapiro (1992).

The Cut & Paste approach was first presented in Ho & Li (1987), with subsequent developments in Cassandras and Strickland (1989a, 1989b, 1989c) and Ho et. al. (1988). The standard clock was described in Vakili (1991) and its adaption to massively parallel simulation is detailed in Vakili (1992).

ACKNOWLEDGMENTS

The work of the author was supported in part by the National Science Foundation under contract No. ECS-9209902.

REFERENCES

Cao, X.R. 1985. "Convergence of Parameter Sensitivity Estimates in a Stochastic Experiment," *IEEE Trans. on Automatic Control*, vol. AC-30, no. 9, pp. 845-854.

Cassandras, C.G. and Strickland, S.G. 1989a. "On-Line Sensitivity Analysis of Markov Chains," *IEEE Trans. on Automatic Control*, vol. AC-34, no. 1, pp. 76–86.

Cassandras, C.G. and Strickland, S.G. 1989b. "Sample Path Properties of Timed Discrete Event Systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 59- 71.

Cassandras, C.G. and Strickland, S.G. 1989c. "Observable Augmented Systems For Sensitivity Analysis of Markov and Semi-Markov Processes," *IEEE Trans. on Automatic Control*, vol. 34, no. 10, pp. 1026-1037.

Glasserman, P. 1990. *Gradient Estimation via Perturbation Analysis*, Kluwer, Boston.

Glasserman, P. 1993. Regenerative Derivatives of Regenerative Sequences. *Advances in Applied Probability*,**25**, pp. 116-139.

Glasserman, P. and Gong, W.B. 1990. "Smoothed Perturbation Analysis of Discrete-Event Systems," *IEEE Transactions on Automatic Control*, **AC-35**, 11, pp. 1218-1230.

Glasserman, P., Hu, J.Q., and Strickland, S. 1991. "Strong Consistency of Sample Path Derivative Estimates," *Probability in the Engineering and Informational Sciences*, 5, pp. 391-413.

Glynn, P. 1987. "Likelihood Ratio Gradient Estimation: An Overview," *Proc. of the 1987 Winter Simulation Conference*, pp. 366-375.

Glynn, P. 1989. "A GSMP Formalism For Discrete-Event Systems," *Proc. of the IEEE*, Vol. 77, No. 1, pp. 14-23.

Gong, W.B. and Ho, Y.C. 1987. "Smoothed (conditional) Perturbation Analysis of Discrete-

Event Dynamic Systems," *IEEE Trans. on Automatic Control*, vol. AC-32, no. 10, pp. 858-866.

Heidelberger, P, Cao, X.R., Zazanis, M.A., and Suri, R. 1988. "Convergence Properties of Infinitesimal Perturbation Analysis Estimates," *Management Science*, **34**, 11, 1281-1301.

Ho, Y.C. 1987. 'Performance evaluation and perturbation analysis of discrete event dynamical systems', *IEEE Transactions on Automatic Control*, Vol.7, pp.563-572.

Ho, Y.C. and Li, S. 1988. "Extensions of Perturbation Analysis of Discrete Event Dynamic Systems," *IEEE Trans. on Automatic Control*, vol. AC-33, no. 10, pp. 427-438.

Ho, Y.C., Li, S. and Vakili, P. 1988. "On the Efficient Generation of Discrete Event Sample Paths Under Different Parameter Values," *Mathematics and Computers in Simulation*, **30**, 347-370.

Ho, Y.C. and Strickland, S.G. 1991. "A Taxonomy of Perturbation Analysis Techniques", *Discrete Event Dynamic Systems*, IEEE Press.

Hu, J.Q. and Strickland, S.G. 1990. "Strong Consistency of Sample Path Derivative Estimates," 1990, *Applied Mathematics Letters*, Vol. 3, No. 4.

Hu, J.Q. 1992. "Convexity of Sample Path Performance and Strong Consistency of Infinitesimal Perturbation Analysis Estimates," *IEEE Transactions on Automatic Control*, **37**, pp. 258-262.

Reiman, M. and Weiss, A. 1989. "Sensitivity Analysis for Simulations Via Likelihood Ratios," *Operations Research*, Vol. 37, No. 5, pp. 830-844.

Rubinstein, R. 1989. "Sensitivity Analysis and Performance Extrapolation for Computer Simulation Models," *Operations Research*, **37**, pp. 72-81.

Rubinstein, R. 1992a. "Decomposition Score Function Estimators For Sensitivity Analysis and Optimization of Queueing Networks," *Annals of Operations Research*, **39**, Baltzer.

Rubinstein, R. and Shapiro, A. 1992b. *Discrete Event Systems: Sensitivity Analysis and Stochastic Optimization via the Score Function Method*, Wiley.

Shi, Leyuan. 1992. "Discontinuous Perturbation Analysis," subm. to *IEEE Trans. on Automatic Control.*

Suri, R. 1989. "Perturbation Analysis: The State of the Art and Research Issues Explained via the GI/G/1 Queue," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 59- 71.

Suri, R. and Zazanis 1988. M., "Perturbation Analysis Gives Strongly Consistent Sensitivity Estimates For the M/G/1 Queue," *Management Science*, Vol. 34, No. 1, pp. 39-64.

Vakili, P. 1991. "A Standard Clock Technique For Efficient Simulation," *Operations Research Letters*, **10**, pp. 445-452.

Vakili, P. 1992. "Massively Parallel and Distributed Simulation of a Class of Discrete Event Systems: A Different Perspective," subm. to *ACM Trans. on Modeling and Computer Simulation.*

Wardi, Y., Gong, W.B., Cassandras, C.G., and Kallmes, M.H. 1992. "Smoothed Perturbation Analysis for a Class of Piecewise Constant Sample Performance Functions," *Discrete Event Dynamic Systems: Theory and Applications*, **1**, pp. 393-414.

Zhang, B. and Ho, Y.C. 1989. "Variance Reduction for Likelihood Ratio Methods," *Proceedings of the 28th IEEE Conference on Decision and Control*, 145-150.

Zhang, B. and Ho, Y.C. 1991. "Performance Gradient Estimation for Very Large Markov Chains," *IEEE Trans. on Automatic Control*, **AC-11**, 1218- 1227.

AUTHOR BIOGRAPHY

STEPHEN G. STRICKLAND has been an Assistant Professor in the Department of Systems Engineering at the University of Virginia since 1990. His research interests include gradient/sensitivity estimation and rare event simulation.