Proceedings of the 1992 Winter Simulation Conference
ed. J. J. Swain, D. Goldsman, R. C. Crain, and J. R. Wilson

SIMULATION OPTIMIZATION OF (s, S) INVENTORY SYSTEMS

Michael C. Fu

College of Business and Management
University of Maryland
College Park, MD 20742

Kevin J. Healy

School of Industrial Engineering
Purdue University
1287 Grissom Hall
West Lafayette, IN 47907-1287

ABSTRACT

Various methods have been proposed to conduct simulation optimization for discrete event systems such as manufacturing systems and telecommunications systems. We consider a periodic review (s, S) inventory system and report on simulation experiments with two proposed techniques: a deterministic "retrospective" approach and a gradient-based algorithm.

1 INTRODUCTION

The problem of interest involves inventory control of an item – which we assume is measured in continuous units (e.g., pounds) – where once every period the inventory is reviewed and, if necessary, orders are placed to replenish depleted inventory. An (s, S) ordering policy specifies that an order be placed when the level of inventory on hand plus that on order – called the inventory position – falls below the level s, and that the amount of the order be the difference between S and the inventory position, i.e., order amounts are placed "up to S." In this paper, we consider an infinite horizon problem and use simulation to find the values of the parameters s and S that minimize the average cost per period, where costs are associated with ordering, holding, and shortages. We assume general independent and identically distributed (i.i.d.) continuous demands, zero lead times, full backlogging of orders, and linear ordering, holding and shortage costs. We investigate experimentally two simulation optimization methods: a deterministic "retrospective" algorithm and a gradient-based, steepest-descent algorithm. Reviews of techniques for simulation optimization can be found in Jacobsen and Schruben (1989), Safizadeh (1990), and Fu (1994).

The retrospective approach to simulation optimization was introduced in Healy and Schruben (1991), with a more detailed development to be found in Healy (1992). The basic idea is to solve a deterministic optimization problem with respect to given realizations of the stochastic effects as if the outcomes of all uncertainties were known in advance. A similar idea is proposed in Rubinstein (1991). In the context of the inventory problem, the idea is to estimate the optimal reorder and order-up-to values by solving the deterministic optimization problem associated with an observed or artificially generated n-period realization of the demand process. The applicability of the technique depends on the limiting properties of the estimator as the length of the trajectory on which the solution is based increases as well as the effort involved in solving the corresponding deterministic optimization problem.

The second approach involves estimating the gradient of the performance measure of interest and adjusting the parameters according to the gradient *during* the evolution of the simulation. The simulation is terminated when the gradient is "close enough" to zero. Because the parameters are adjusted during the actual simulation, yielding the optimal values at the conclusion of a single simulation, this approach has been called single-run optimization (Suri and Zazanis, 1988). (In this sense, the retrospective approach could also be deemed a single-run optimization technique.) Two main techniques can be used to do the gradient estimation (see e.g., L'Ecuyer, 1991): perturbation analysis and the likelihood ratio. In this paper, we utilize the perturbation analysis estimators derived in Fu (1990b).

The rest of the paper is organized as follows. In Section 2, we present the problem setting and the test cases considered. In Section 3, we outline the gradient-based algorithm utilizing perturbation analysis and present the results of various simulation experiments. In Section 4, we do the analogous presentation for the retrospective approach. In Section 5, we conclude by commenting briefly on the results from the two proposed approaches.

2 PROBLEM SETTING

The inventory model of interest is a periodic review inventory system with general i.i.d. continuous demands, zero lead times, full backlogging of orders, and linear ordering, holding and shortage costs. An (s, S) ordering policy specifies that an order be placed when the level of inventory on-hand plus that on-order – called the inventory position – falls below the level s, and that the amount of the order be the difference between S and the inventory position, i.e., order amounts are placed "up to S." The optimization problem is to find the values of s and S which minimize long-run average cost per period. A potential drawback of gradient-based optimization techniques is their limitation to finding only local optima. However, for the (s, S) inventory system we consider — with linear order, holding, and backlogging costs — the cost function is convex, so the local optimum is also globally optimal.

We define

W_n = inventory level (on-hand minus on-backorder in period n),
X_n = inventory position (inventory level plus outstanding orders in period n),
D_n = demand in period n (i.i.d. for all n),
$F(\cdot)$ = distribution function of D_n.

We assume throughout that $F(\cdot)$ is absolutely continuous with density function $f(\cdot)$.

We consider the long-run average cost per period, which consists of three components: ordering, holding, and shortage costs. First, we define the one-period cost function by

$$J_n = I\{X_n < s\}(K + c(S - X_n)) + hW_n^+ + pW_n^-,$$

where

h = holding costs/period/unit of inventory,
p = shortage costs/period/unit of inventory,
K = set-up cost for placing an order,
c = per-unit ordering cost,

$x^+ = \max(0, x)$, and $x^- = \max(0, -x)$, and $I\{*\}$ denotes the indicator function of the set. Again, since we have assumed linear costs, a policy of the (s, S) type is optimal. Also, since we have assumed zero lead times, we have $W_n = X_n$. The performance measure of interest is the long-run average cost per period function, i.e., the limit of the the n-period (random) average cost per period function:

$$\mathcal{J} = \lim_{n\to\infty} \mathcal{J}_n, \tag{1}$$

$$\mathcal{J}_n = \frac{\sum_{i=1}^{n} J_i}{n}, \tag{2}$$

where J_i is the one-period cost function defined above. The optimization problem is to find s and S to minimize $\mathcal{J}$. Equivalently, we will define

$$\Delta = S - s,$$

and find s and Δ to minimize $\mathcal{J}$.

In this paper, we chose our test system to have exponentially distributed demands, so that we could easily compare the simulation optimization results of the two methods to analytical results. Letting $\lambda = 1/E[D]$, we have (cf., e.g., Karlin 1958):

$$\Delta_{opt} = \sqrt{\frac{2K}{\lambda h}}, \tag{3}$$

$$s_{opt} = -E[D] \ln\left(\frac{h + \sqrt{2Kh\lambda}}{h + p}\right), \tag{4}$$

$$\mathcal{J}(s, \Delta) = cE[D] + \frac{K + h(s - E[D] + \lambda\Delta(s + \frac{\Delta}{2}))(h + p)E[D]e^{-\lambda s}}{1 + \lambda\Delta}. \tag{5}$$

The eight test cases considered, each with $c = h = 1$, are given in Table 1 below.

Table 1: Test Cases

Case	$E[D]$	p	K	$\mathcal{J}^*$	s_{opt}	Δ_{opt}
1	200	10	100	740.9	341	200
2	200	10	10,000	2200.0	0	2000
3	200	100	100	1184.4	784	200
4	200	100	10,000	2643.4	443	2000
5	5000	10	100	17078	11078	1000
6	5000	10	10,000	21496	6496	10000
7	5000	100	100	28164	22164	1000
8	5000	100	10,000	32583	17582	10000

3 PERTURBATION ANALYSIS AND GRADIENT-BASED OPTIMIZATION

Perturbation analysis is a technique for gradient estimation from a *single* simulation of a discrete-event system (see, e.g., Ho and Cao 1991, Glasserman 1991, as well as the tutorial by Ho and the state-of-the-art review by Glasserman in these proceedings). Perturbation analysis derivative estimators for (s, S) inventory systems were derived in Fu (1990b), representing the first application of perturbation analysis to inventory systems. The technique applied was smoothed perturbation analysis (see Gong and Ho 1987 and Fu and Hu 1992), which uses conditional expectation to overcome difficulties of infinitesimal perturbation

analysis. The perturbation analysis derivative estimators over N periods are given by (Fu 1990b):

$$\left(\frac{\partial \mathcal{J}_N}{\partial s}\right)_{PA} = \frac{1}{N}\left[\sum_{i:W_i>0} h + \sum_{i:W_i<0} p\right], \quad (6)$$

$$\left(\frac{\partial \mathcal{J}_N}{\partial \Delta}\right)_{PA} = \frac{1}{N}\left[\sum_{i:W_i>0} h + \sum_{i:W_i<0} p\right] + \sum_{j\in L^*} \frac{f(z_j)}{1-F(z_j)} \cdot \frac{1}{N+1} \cdot \left[cE[D] + hE[(s-D)^+] + pE[(D-s)^+] - \frac{\sum_{i=1}^{N} J_i}{N}\right]. \quad (7)$$

Note that these estimators correspond to the lead time $L = 1$ in Fu (1990b), because here $\{X_n\}$ is defined after the demand for the period is subtracted, i.e., the recursive equation for $\{X_n\}$ is given by

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

Perturbation analysis derivative estimators for more complicated (s, S) inventory systems can be found in Bashyam and Fu (1991).

An explicit algorithm for both derivatives, (6) and (7), is given below.

PA Algorithm for $\partial\mathcal{J}/\partial s$ and $\partial\mathcal{J}/\partial\Delta$.

Initialize:
$Jsum = 0$; $haz = 0$; $n^* = 0$;
$H'sum = 0$; $P'sum = 0$; $X = X_0 (< S)$; $z = S - X$;
For each period:
Generate demand D according to $F(*)$;
If $X > 0$ then
  $Jsum = Jsum + hX$; $H'sum = H'sum + hX$;
else
  $Jsum = Jsum - pX$; $P'sum = P'sum - pX$;
If $X < s$ then
  $n^* = n^* + 1$;
  $haz = haz + f(z)/[1 - F(z)]$;
  $Jsum = Jsum + K + c*(S - X)$;
  $z = S - s$; $X = S - D$;
else
  $z = X - s$; $X = X - D$;
At the end of N periods:
$(\partial\mathcal{J})_{IPA} = (H'sum + P'sum)/N$;
$(\partial\mathcal{J})_{SPA} = haz*(cE[D] + hE[(s-D)^+] + pE[(D-s)^+] - Jsum/N)/(N+1)$;
$(\partial\mathcal{J}/\partial s)_{PA} = (\partial\mathcal{J})_{IPA}$;
$(\partial\mathcal{J}/\partial\Delta)_{PA} = (\partial\mathcal{J})_{IPA} + (\partial\mathcal{J})_{SPA}$.

The optimization algorithm is the following two-dimensional Robbins-Munro stochastic approximation algorithm (cf. Kushner and Clark 1978):

$$\begin{bmatrix} s_{n+1} \\ \Delta_{n+1} \end{bmatrix} = \begin{bmatrix} s_n \\ \Delta_n \end{bmatrix} - b_n A \begin{bmatrix} \left(\frac{\partial\mathcal{J}}{\partial s}\right)_{PA} \\ \left(\frac{\partial\mathcal{J}}{\partial\Delta}\right)_{PA} \end{bmatrix}_{s_n,\Delta_n}$$

$$\text{where } A = a\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \quad a_{ij} \in [-1, +1],$$

and $\{b_n\}$ is a scalar series modifying the step sizes. We choose $\{b_n\}$ to be the following two-dimensional adaptation of the accelerated harmonic series:

$$\frac{1}{b_{n+1}} = \begin{cases} \frac{1}{b_n} + 1 & \text{if } \operatorname{sgn}\left(\frac{\partial\mathcal{J}}{\partial s}\right)_{PA,n+1} \neq \operatorname{sgn}\left(\frac{\partial\mathcal{J}}{\partial s}\right)_{PA,n} \\ & \quad \& \operatorname{sgn}\left(\frac{\partial\mathcal{J}}{\partial\Delta}\right)_{PA,n+1} \neq \operatorname{sgn}\left(\frac{\partial\mathcal{J}}{\partial\Delta}\right)_{PA,n} \\ \frac{1}{b_n} & \text{otherwise} \end{cases}$$

("sgn" denoting the sign function) with $b_0 = 1$. Thus, the step size is reduced only if both components of the estimated gradient change sign. A more "accurate" adaptation might involve the angle measure between the previous and new gradients. Since Δ must be positive, we project back to the previous point whenever the algorithm brings Δ less than zero. We also restricted $s \geq 0$, so if this constraint were violated, again a projection was employed. We chose the set of coefficients for the matrix A corresponding to a decoupling of the two gradients, i.e., $a_{11} = a_{22} = 1$ and $a_{12} = a_{21} = 0$, so our algorithm is simply

$$\begin{bmatrix} s_{n+1} \\ \Delta_{n+1} \end{bmatrix} = \begin{bmatrix} s_n \\ \Delta_n \end{bmatrix} - b_n a \begin{bmatrix} \left(\frac{\partial\mathcal{J}}{\partial s}\right)_{PA} \\ \left(\frac{\partial\mathcal{J}}{\partial\Delta}\right)_{PA} \end{bmatrix}_{s_n,\Delta_n} \quad (8)$$

with updates done every N periods, i.e., every N periods, the values of s and Δ are changed according to (8) and the PA algorithm reinitialized. Obviously the performance of the algorithm, in terms of convergence speed, will depend on all of the above choices.

We "naively" chose starting points $s_0 = \Delta_0 = E[D]/2$. The simulation results for the eight cases over 16 replications of 100,000 periods are given in Table 2, presenting mean ± (asymptotically valid) 95% confidence widths for $\mathcal{J}(s_n, \Delta_n)$ at progressive phases of the algorithm, with the value given for # periods equal to 0 corresponding to $\mathcal{J}(s_0, \Delta_0)$ and the theoretical minimum given by $\mathcal{J}^* = \mathcal{J}(s_{opt}, \Delta_{opt})$. Of course, in practice, a stopping rule of some sort would have to be specified.

Various different values of a and N were tried to get some idea as to the sensitivity of the algorithm to the choices of these parameters. The results for Case 1 are also included in Table 2. For the other cases,

the qualitative effects were much the same, and hence only one ("good") choice for each of the other cases is presented in the table.

From the results in this admittedly limited test set, we see rapid improvement in the beginning for "good" choices of a and N, with much slower improvement as the optimum is neared. Figures 1 and 2 graphically illustrate the effect of the choices of a and N on the early behavior of the algorithms. A single run from Case 1 is represented by each curve on the graph. Figure 5 gives the $N = 50$ case for the four different values of a, whereas Figure 5 gives the $N = 500$ case. It is clear that the early behavior depends heavily on the choice of a, with too small a choice leading to very slow movement, whereas too large a choice may lead to erratic behavior. For example, for the $N = 5, a = 1000$ case, 9 of the 16 replications ended (at 100,000 periods) with $\mathcal{J}_n$ under 751 — reasonably close to optimal, whereas the remaining 7 replications were above 2500, which is actually worse than the starting value!

We have not attempted to give any convergence proofs here. For a regenerative version of the algorithm — where order points constitute regenerative points, a proof along the lines of Fu (1990a) could be obtained. For the version of the algorithm studied here, it is likely that convergence could be established using martingale methods as in Chong and Ramadge (1991). It should be noted, however, that both Fu (1990a) and Chong and Ramadge (1991) consider infinitesimal perturbation analysis (IPA) estimators, not SPA estimators, but it does not appear that this difference should introduce any substantive difficulties in the proofs.

4 A RETROSPECTIVE APPROACH

For a given realization of demand, $\mathcal{J}_n$ in (2) represents a functional estimate of the long-run average cost per period for all feasible values of (s, S). With this in mind, define the n-period retrospective estimator of the optimal (s, S) pair to be

$$(\hat{s}, \hat{S}) = \underset{(s,S)\in\Re_+^2}{argmin}\ \mathcal{J}_n \tag{9}$$

Realizations of $(\hat{s}, \hat{S})$ are obtained by simulating n period demands, $\{d_1, d_2, ..., d_n\}$, and optimizing over the corresponding deterministic cost function. In the the process, it is necessary to restrict ourselves to stationary (s, S) solutions since the solution to the unconstrained sample path problem is trivially nonstationary. That is, knowing the demand values we can do no better costwise than ordering exactly the amount in each period necessary to satisfy the demand in the coming period. In general, a key requirement of the retrospective technique is that the underlying sample space of the stochastic component (the demand values in this case) and its driving distribution be independent of the decision parameters. Otherwise, realizations of stochastic effects could only be obtained by fixing values of the decision parameters a priori.

We briefly outline the solution to the deterministic problem, referring the reader to Healy (1992) for details. A key observation is that the periods in which an order is placed are completely determined by the value of $\Delta = S - s$. Furthermore, for fixed Δ, $\mathcal{J}_n$ is continuous, piecewise linear, and convex in S with probability one. These observations suggest an algorithm employing a search over values of Δ starting with $\Delta = 0$ (in which case an order is placed in each of the n periods) and ending with $\Delta = \sum_{i=1}^{n} d_i$ (beyond which point no orders are placed).

Let $\hat{S}_\Delta$ represent the optimal order-up-to quantity for a given value of Δ in the n-period sample path problem and $\mathcal{J}_n(\hat{S}_\Delta, \Delta)$ be the corresponding cost of this policy. The structure of the deterministic problem is such that $\hat{S}_\Delta$ is a piecewise constant function of Δ so that there exists only a finite number of subintervals intervals over $[0, \infty)$ on which the value of $\hat{S}_\Delta$ differs. The idea is to identify these intervals, compute the corresponding values $\hat{S}_\Delta$ and $\mathcal{J}_n(\hat{S}_\Delta, \Delta)$ and choose the solution with minimum cost.

If $d_{i:n}$ represents the i^{th} smallest of the n demand values, then for $0 \le \Delta < d_{1:n}$, orders will be placed in each of the n periods and

$$\begin{aligned} n\mathcal{J}_n = \quad & nK + c\textstyle\sum_{i=1}^{n}(S - X_i) + \\ & h\textstyle\sum_{i=1}^{n}(S - X_i)^+ + p\sum_{i=1}^{n}(S - X_i)^- \end{aligned} \tag{10}$$

Since orders are placed in each period, $X_i = S - d_i$ (with $X_0 = S$ by assumption) so that

$$\begin{aligned} n\mathcal{J}_n = \quad & nK + (c - h)\textstyle\sum_{i=1}^{n} d_i - npS + \\ & (h + p)\textstyle\sum_{i=1}^{n} max(S, d_{i:n}) \end{aligned} \tag{11}$$

The continuity, piecewise linearity and convexity in S can be justified by examining the behavior of $\mathcal{J}_n$ over ranges of S corresponding to the intervals between the ordered values of the demands $[0, d_{1:n})$, $[d_{1:n}, d_{2:n})$,..., $[d_{n-1:n}, d_{n:n})$, $[d_{n:n}, \infty)$. It then becomes straightforward to establish the following result:

$$\hat{S}_\Delta = d_{k:n} \quad \forall\ \Delta \in [0, d_{1:n}) \tag{12}$$

where $k = \lceil \frac{np}{h+p} \rceil$.

The subsequent intervals for Δ are identified as follows. Let δ_l and δ_u be the lower and upper endpoints

of the i^{th} interval. At stage $i+1$, we set δ_l to be the upper end of the previous interval. We then make a pass through all n period demands, in the process accumulating the values of demands in consecutive periods until their sum *exceeds* the new value of δ_l. This step identifies the periods in which orders are placed in the original problem when $\Delta = \delta_l$. Let $D' = \{d'_j : j = 1, 2, ..., n'\}$ be this set of accumulated or coalesced demands where n' depends on the value of δ_l. Note that the quantity of total demand may not be sufficient to guarantee that the last of the coalesced demands is strictly greater than δ_l. If this is the case, this last value is omitted from the the set D'. We view these values as a set of demands from a subproblem of length n' in which an order is placed in every period. Once the number of orders is fixed the problem is reduced to a tradeoff between holding and shortage costs. Consequently, the structure of the resultant cost function at each stage is identical to (11). The only distinctions are the amount of order costs, $n'K$, and the holding costs that are incurred in those periods from the original problem in which an order is not placed. These periods can be identified during the execution of the accumulation pass and the holding costs accounted for as follows. Say the k^{th} coalesced demand consists of demands from periods l through $l+m$ in the original problem. Since the beginning of period l constitutes a reorder point, holding costs are incurred in periods l through $l+m-1$ in the amounts of $h(S-d_l), h(S-(d_l+d_{l+1})), ..., h(S-(d_l+...+d_{l+m-1}))$. For each pseudo period $k = 1, 2, ..., n'$, let

$$A_k = d_l + (d_l + d_{l+1}) + ... + (d_l + ... + d_{l+m-1}) \quad (13)$$

be the sum of the intermediate accumulations of the demand values l through $l+m-1$. If during the accumulation pass we are left with any demands the sum of which did not exceed δ_l, say periods l through n, then there will also be holding costs in each of these periods in the amounts of $h(S-d_l), h(S-(d_l+d_{l+1})), ..., h(S-(d_l+...+d_n))$. In this case, define

$$A_{n'+1} = d_l + (d_l + d_{l+1}) + ... + (d_l + ... + d_n) \quad (14)$$

and $A_{n'+1} = 0$ otherwise. In general, for all $\Delta \in [\delta_l, d'_{1:n'})$

$$n\mathcal{J}_n = n'K + c\sum_{i=1}^{n'} d'_i - h\sum_{i=1}^{n'+1} A_i + h(n-n')S + (h+p)\sum_{i=1}^{n'} max(S, d_{i:n'}) \quad (15)$$

and

$$\hat{S}_\Delta = d'_{k:n'} \quad \forall\, \Delta \in [\delta_l, d'_{1:n'}) \quad (16)$$

where

$$k = \lceil \tfrac{n'p-(n-n')h}{p+h} \rceil \quad (17)$$

The ideas of this section are brought together in the following algorithm.

Initialize:
$\delta_l = \delta_u = 0$; $min\_cost = \infty$; $dsum[0] = 0$;
For each period: $i = 1, 2, ..., n$
  Generate d_i,
  $dsum[i] = dsum[i-1] + d_i$

Enumerate Δ intervals:
While $\delta_l < dsum[n]$
  $n' = 0$
  $reorder = 0$
  $A = 0$
  Coalesce demands:
  For each period: $i = 1, 2, ..., n$
    $d' = dsum[i] - dsum[reorder]$
    if ($d' < \delta_l$)
      $A = A + d'$
    else
      $n' = n' + 1$
      $reorder = i$
      insert d' into ordered list of demands
  Optimize over current interval:
  $\delta_u = d'_{1:n'}$
  $k = \lceil \frac{n'p-(n-n')h}{h+p} \rceil$
  $\hat{S}_{\delta_l} = d'_{k:n'}$
  $cost = n'K + c(dsum[n] - dsum[reorder]) - hA$
    $+ h(n-n')\hat{S}_{\delta_l} +$
    $+ h\sum_{i=1}^{k-1} d'_{i:n'} + p\sum_{i=k+1}^{n'} d'_{i:n'}$
  Update min cost solution:
  if ($cost < min\_cost$)
    $min\_cost = cost$
    $\hat{S} = \hat{S}_{\delta_l}$
    $\hat{\Delta} = \delta_l$

  $\delta_l = \delta_u$

One is tempted to conclude that after each stage in which the demands are coalesced there is one less period in which an order is placed so that there are exactly $n+1$ separate intervals of Δ over which $\mathcal{J}_n$ is constant. If this were the case, the running time of the algorithm would be $O(n^2 \log n)$ since the *While* loop (within which an $O(n' \log n')$ sort of the pseudo demands is performed) would be executed only n times. A careful analysis, however, reveals that there may exist for a given realization of demands as many as $(n - n' + 1)$ alternate configurations or intervals of

Δ in which exactly n' orders are placed, although in practice much fewer are typically encountered. As a consequence, the worst case computational complexity is necessarily $O(n^3 \log n)$.

Experimental results for the set of problem instances defined in Table 1 are given in Table 3. The entries represent (asymptotically valid) .95 level interval estimates for $\mathcal{J}(\hat{s}, \hat{S})$ for various choices of n up to 10,000, each one based on 16 independently seeded replications. Conspicuously absent are entries for $n = 100{,}000$ which proved to be computationally prohibitive given the running time of the algorithm. It should be noted that solutions for intermediate values of n are obtained at considerable expense since the problem must be completely re-solved unlike the gradient-based approach in which incremental updates to the estimator can be made as the run length is extended.

We refer the reader to Healy (1992) for a general discussion of convergence properties of retrospective estimators and their application to the inventory problem presented here.

5 CONCLUSIONS

It is difficult to give a very meaningful comparison between the two methods, because in some sense they solve different versions of the problem. The single-run stochastic approximation algorithm described here is designed to give the optimal values for the infinite horizon problem. The parameter values are continuously changed as the simulation progresses, with the final value obtained when the algorithm "settles down." In contrast, the retrospective approach determines the optimum for a *given* finite horizon, with the optimal values for the infinite horizon problem found by taking the horizon large enough. However, the computational burden increases with the horizon length, so that in some sense it is not practical for very lengthy horizons. On the other hand, application of stochastic approximation for finite horizon length is handled differently from the infinite horizon problem, because each iteration should "restart" from the initial conditions, with the simulation length for each iteration being the length of the finite horizon, so that no choice of updating sequence length (N in our algorithm) need be made. Furthermore, since the finite horizon gradient estimator can be proven unbiased, convergence proofs are quite straightforward for the finite horizon problem. However, since each iteration requires a simulation of length of the horizon, it is likely that for "moderate" horizons, the retrospective approach is computationally more efficient. In the decision-theoretic framework proposed by Glynn and Whitt (1992), the efficiency of an estimator is taken to be inversely proportional to the product of the sampling variance and the amount of work expended in obtaining the estimate. Under this criteria, the results in Tables 2 and 3 indicate that the retrospective approach has considerably lower sampling variance than the perturbation analysis-based gradient optimization algorithm, but the amount of work expended in obtaining the estimates increases rapidly for lengthy horizons.

Another topic worthy of further investigation is how the two techniques might be combined to exploit the most desirable properties of each. Perhaps the most appealing aspect of the retrospective technique is that it uses simulation to get ideas, not just evaluate ideas, by effectively concentrating the sampling on the most promising solutions. Since the performance of a prospective search technique is often dependent on the quality of the initial solution, it would be fairly easy in this case to conduct a small retrospective experiment whose solution would serve as a starting point for the gradient search.

In terms of implementation, it should be fairly clear that the gradient-based algorithm is quite easily incorporated into a simulation, due to the ease in calculating the perturbation analysis gradient estimate. In approximately the same amount of simulation effort that would be required to get a reasonable estimate of the cost function, the algorithm terminates with near-optimal values of the parameters. The main difficulties are the two choices that have to be made as to initial step size (a) and update period (N). The experimental results indicate that selecting too short of an update period can lead to highly unpredictable results. For many runs, it may achieve extremely quick convergence to the optimal values, but for a few runs, especially when the initial step size a is also "too large," the algorithm is initially led far astray, leading to very poor results. On the other hand, selecting too long of an update period may sacrifice some computational efficiency, in the sense of waiting unnecessarily "too long" before making a move. The initial step size has much the same effect in the inverse relationship to its value, i.e., too small an initial step size leads to slow convergence, but too large an initial step size leads to erratic behavior. Probably the best practical implementation would be to start with a relatively small and N relatively large, and get some initial "feel" as to a reasonable size for a, upon which the algorithm can be "restarted" with the estimate of a and with L reduced to obtain quicker convergence. Another possibility is to use some sort of gradient averaging, which in Fu and Ho (1988) tended to reduce the sensitivity to the choices of a and N.

Table 2: Experimental Results for Stochastic Approximation Algorithm

			# periods					
Case	N	a	0	100	1000	10,000	100,000	$\mathcal{J}^*$
1	5	1	1139.6	1013.9±20.1	925.6±15.9	876.7±9.5	848.6±6.2	740.9
1	5	10	1139.6	803.6±18.4	761.6±4.8	750.4±1.7	746.0±0.6	740.9
1	5	100	1139.6	812.5±36.6	767.0±18.0	759.9±13.3	755.9±10.1	740.9
1	5	1000	1139.6	1888.9±533.6	1856.7±695.9	1915.3±802.2	2028.8±897.6	740.9
1	50	1	1139.6	1112.5±3.5	1000.1±4.8	783.5±4.5	765.2±1.9	740.9
1	50	10	1139.6	944.8±20.1	774.4±6.0	745.6±0.7	742.9±0.2	740.9
1	50	100	1139.6	833.1±40.6	747.8±3.8	742.9±1.1	741.7±0.4	740.9
1	50	1000	1139.6	4758.4±884.8	1044.0±129.3	818.3±110.6	787.8±99.2	740.9
1	500	1	1139.6	—	1113.5±1.0	1001.2±1.3	771.1±0.4	740.9
1	500	10	1139.6	—	946.6±5.8	765.6±1.0	742.0±0.2	740.9
1	500	100	1139.6	—	788.6±11.3	743.2±0.9	741.5±0.2	740.9
1	500	1000	1139.6	—	4365.7±224.9	765.7±17.8	741.3±0.2	740.9
2	50	100	7739.6	2589.9±99.6	2281.9±46.5	2269.1±39.4	2269.7±39.3	2200.0
3	50	10	8417.9	1391.3±114.6	1257.6±15.7	1225.3±6.4	1215.3±6.0	1184.4
4	50	10	15017.9	3802.1±298.0	3345.1±11.4	3063.1±45.0	2968.1±26.3	2643.4
5	50	1000	26890	18110±460	17432±46	17314±43	17258±33	17078
6	50	1000	33490	22617±659	21718±79	21600±30	21553±14	21496
7	50	100	208849	69356±7729	37052±443	31207±262	30161±97	28164
8	50	100	215449	73678±8159	41345±860	35108±314	33954±116	32583

Table 3: Experimental Results for Retrospective Algorithm

	# periods			
Case	100	1000	10,000	$\mathcal{J}^*$
1	797.7±13.2	743.6±4.8	741.0±.04	740.9
2	2276.2±20.7	2214.0±2.8	2200.0±.01	2200.0
3	1861.0±127.2	1207.5±10.7	1184.7±.64	1184.4
4	3208.0±77.7	2678.5±15.4	2643.7±.67	2643.4
5	19015.2±435.7	17152±92	17079±1.2	17078
6	22591.5±401.6	21553±58	21497±1.1	21496
7	46172.4±3345.0	28802±378	28165±17	28164
8	47625.2±2803.5	33733±199	32594±12	32583

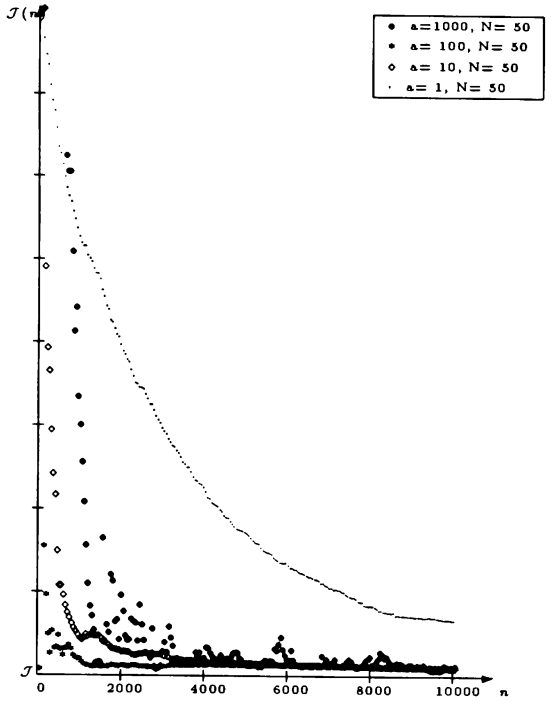

Figure 1: Comparison of Effect of Choice of a for $N = 50$.

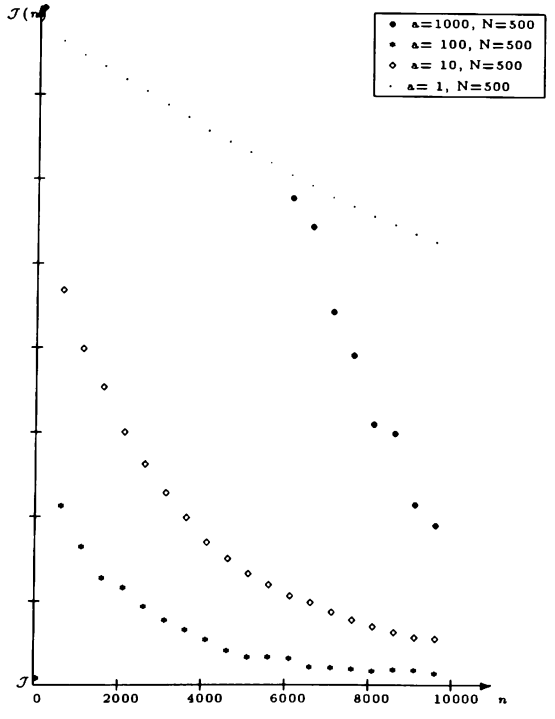

Figure 2: Comparison of Effect of Choice of a for $N = 500$.

REFERENCES

Bashyam, S. and M.C. Fu. 1991. Application of Perturbation Analysis to (s,S) Inventory Systems. *Proceedings of the 1991 Winter Simulation Conference*: 993-1004.

Chong, E.K.P. and P.J. Ramadge. 1991. Optimization of Queues Using an Infinitesimal Perturbation Analysis based Stochastic Algorithm with General Update Times. Submitted to *SIAM Journal on Control and Optimization.*

Fu, M.C. 1990a. Convergence of a Stochastic Approximation Algorithm for the GI/G/1 Queue Using Infinitesimal Perturbation Analysis. *Journal of Optimization Theory and Applications* **65**: 149-160.

Fu, M.C. 1990b. Sample Path Derivatives for (s,S) Inventory Systems. Submitted to *Operations Research.*

Fu, M.C. 1994. Optimization Using Simulation: A Review. to appear in *Annals of Operations Research.*

Fu, M.C. and Y.C. Ho. 1988. Using Perturbation Analysis for Gradient Estimation, Averaging, and Updating in a Stochastic Approximation Algorithm. *Proceedings of the 1988 Winter Simulation Conference*: 509-517.

Fu, M.C. and J.Q. Hu. 1992. Extensions and Generalizations of Smoothed Perturbation Analysis in a Generalized Semi-Markov Process Framework. *IEEE Transactions on Automatic Control*: to appear.

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

Glynn, P.W. and W. Whitt. 1992. The Asymptotic Efficiency of Simulation Estimators. *Operations Research* **40**: 505-520.

Gong, W.B. and Y.C. Ho. 1987. Smoothed Perturbation Analysis of Discrete-Event Dynamic Systems. *IEEE Transactions on Automatic Control* **32**: 858-867.

Healy, K. 1992. Optimizing Stochastic Systems: A Retrospective/Deterministic Approach. Ph.D. dissertation. Cornell University.

Healy, K. and L.W. Schruben. 1991. Retrospective Simulation Response Optimization. *Proceedings of the 1991 Winter Simulation Conference*: 901-906.

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

Jacobsen, S. and L.W. Schruben. 1989. A Review of Techniques for Simulation Optimization. *Operations Research Letters* **8**: 1-9.

Karlin, S. 1958. Steady State Solutions. *Studies in the Mathematical Theory of Inventory and Production.* K.J.Arrow, S.Karlin and H.Scarf (eds.), Stanford University Press.

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

L'Ecuyer, P. 1991. An Overview of Derivative Estimation. *Proceedings of the 1991 Winter Simulation Conference*: 207-217.

Rubinstein, R.Y. 1991. How to Optimize Discrete-Event Systems From a Single Sample Path by the Score Function Method. *Annals of Operations Research* **27**: 175-212.

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

Suri, R. and M.A. Zazanis. 1988. Perturbation Analysis Gives Strongly Consistent Sensitivity Estimates for the M/G/1 Queue. *Management Science* **34**: 39-64.

AUTHOR BIOGRAPHIES

MICHAEL C. FU is an assistant professor in management science in the College of Business and Management at the University of Maryland. He holds S.B. and S.M. degrees in electrical engineering and an S.B. degree in mathematics from MIT (1985) and a Ph.D. in applied mathematics from Harvard University (1989). His research interests include stochastic derivative estimation and stochastic optimization, particularly with applications towards manufacturing systems and inventory control.

KEVIN J. HEALY is an assistant professor in the Industrial Engineering Dept. at Purdue University. He holds B.S. (1980) and M.S. (1982) degrees in Industrial Engineering from The Pennsylvania State University and a Ph.D. (1992) in Operations Research from Cornell University. Formerly, he was Vice-President of the Systems Modeling Corporation and a key developer of the SIMAN and Cinema simulation software. His research interests include stochastic optimization, simulation output analysis, and modelling formalisms.