Proceedings of the 1995 Winter Simulation Conference
ed. C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman

TRANSFER OPTIMIZATION VIA SIMULTANEOUS PERTURBATION STOCHASTIC APPROXIMATION

Stacy D. Hill

The Johns Hopkins University
Applied Physics Laboratory
Laurel, Maryland 20723, U.S.A.

Michael C. Fu

College of Business and Management
University of Maryland at College Park
College Park, Maryland 20742, U.S.A.

ABSTRACT

We consider the problem of optimizing a transit network with respect to customer service, when simulation of the network is necessary to accurately characterize performance. In particular, we consider the transfer optimization problem, where the goal is to minimize the total expected waiting time of riders by coordinating transfers in the network. We apply the technique of simultaneous perturbation stochastic approximation to optimize system performance. For a simple test case, we provide simulation results and discuss difficulties in applying the technique to this problem, specifically with regard to the smoothness of the objective function.

1 INTRODUCTION

If we wish to optimize some performance measure of a discrete event system (see, e.g., Cassandras 1993 or Fu 1994), then under suitable conditions, this problem reduces to finding the zero of the performance measure gradient, so that gradient-based techniques based on stochastic approximation can be applied. Techniques such as perturbation analysis or likelihood ratio provide efficient means for estimating the performance measure gradient, but these techniques are not universally applicable. This paper considers a technique called simultaneous perturbation stochastic approximation (SPSA) which requires minimal assumptions on the system of interest (Spall 1992). SPSA uses the simultaneous perturbation (SP) method to estimate the gradient with only estimates of the performance measure itself. Furthermore, in each update step, SPSA requires only **two** sample estimates of the performance measure to calculate a gradient estimate, regardless of the dimension of the vector of parameters. This sharply contrasts with the method of finite differences which grows linearly with the dimension. Thus, SPSA requires substantially less data — in our application, meaning significantly fewer simulations — than finite differences for estimating gradients in high dimensions.

To be more specific, let $\theta \in \Theta \subset \mathcal{R}^p$ denote a vector of controllable (adjustable) parameters and ω the stochastic effects. Let $L(\theta,\omega)$ denote the sample path performance of interest and $J(\theta) = E[L(\theta,\omega)]$ expected system performance. The problem is to find $\arg\min\{J(\theta) : \theta \in \Theta\}$. The stochastic approximation algorithm for solving $\nabla J = 0$ is given by the following iterative scheme:

$$\theta_{(n+1)} = \theta_{(n)} - a_n \widehat{g}_n, \tag{1}$$

where $\theta_{(n)}$ represents the nth iterate, $\widehat{g}_n$ represents an estimate of the gradient ∇J at $\theta_{(n)}$, and $\{a_n\}$ is a positive sequence of numbers converging to 0.

2 SIMULTANEOUS PERTURBATIONS

Let e_i denote the unit vector in the ith direction, and t_n the simulation length of the nth iteration. Let $\{\Delta_1, ..., \Delta_p\}$ be a set of i.i.d. perturbations satisfying the conditions given in Spall (1992), and define the vector $\Delta = [\Delta_1 ... \Delta_p]$. Let $(\widehat{g}_n)_i$ denote the ith component of $\widehat{g}_n$, and let $J_{t_n}(\theta,\omega)$ denote the observed (sample) system performance at θ on sample path ω for duration t_n. Then, the SP estimator is given by

$$(\widehat{g}_n)_i = \frac{J^+ - J^-}{2c_n\Delta_i}, \tag{2}$$

where (3)

$$J^+ = J_{t_n}(\theta_{(n)} + c_n\Delta, \omega_n^+),$$
$$J^- = J_{t_n}(\theta_{(n)} - c_n\Delta, \omega_n^-),$$

where J^+ and J^- are performance estimates at the parameter value $\theta_{(n)}$ *simultaneously* perturbed in all directions.

Compare this estimator with the usual symmetric difference (SD) estimator:

$$(\widehat{g}_n)_i = \frac{J_{t_n}(\theta_{(n)} + c_n e_i, \omega_n^{i+}) - J_{t_n}(\theta_{(n)} - c_n e_i, \omega_n^{i-})}{2c_n}. \tag{4}$$

The SD estimator requires a different pair of estimates in the *numerator* for each parameter, thus requiring $2p$ simulations, whereas in SPSA, the *same* pair is used in the numerator for all parameters, and instead the *denominator* changes; thus, only **two** discrete-event simulations are required at each iteration. For common random numbers, we take $\omega_n^+ = \omega_n^- = \omega_n$ (or as close as possible in the simulation implementation).

3 TRANSPORTATION APPLICATION

In Hill and Fu (1994a,b), SPSA was applied to simple single-server queues and open queueing networks with general routes. Here, we consider a transportation application. The model we consider is a transit network with bus lines traveling in four directions on a grid: east, west, north, and south. Transfers occur, for instance, from a west-bound line to a north-bound line, and multiple transfers are possible. Undesirable delays occur for passengers due to waiting for a transfer. As summarized in Bookbinder and Désilets (1992), there are two basic approaches to this problem: timed transfer and transfer optimization. The former focuses on coordinating the transfer points, and is more applicable for networks where transfers constitute a relatively smaller proportion of overall traffic, e.g., intercity trains and planes. This approach would not be appropriate for a large transit network, such as is found in a downtown bus network, where transfers are decentralized. In this case, transfer optimization is usually employed, whereby the decisions to be made have to do with the departure times of the first bus on a line, and also possibly the headways.

In transfer optimization, the following are usually assumed to be given: the network, i.e., no re-routing is allowed; the headways, defined as the times between adjacent buses on the same line (assumed to be constant and equal); the transfer points; the passenger traffic and transfers. Traffic on a route can be given either as a point-to-point total or equivalently as a Markovian routing matrix at each stop. Stochastic elements of the network — incorporated indirectly in Bookbinder and Désilets (1992) — include the arrival process of passengers at each stop, both timing and number; and the travel times of buses.

Let N be the number of transit lines, M be the number of transfer points, H_i be the headway for transit line i, $i = 1, \ldots, N$, Θ_i be the set of allowable offset times for transit line i, $i = 1, \ldots, N$, $\theta = \{\theta_i\}_{i=1}^N$ be the timetable for the transit network, $\Theta = \{\Theta_i\}_{i=1}^N$ be the allowable timetables for the transit network. Note that a *transfer point* in the network model is quite different from a *stop* in the physical real world. In particular, if a given "single" stop occurs at an intersection of two bi-directional routes and allows all possible transfers, then this would generate eight separate transfer points in the network model.

We wish to minimize the total expected waiting time for transfers in the network. This problem is usually formulated as a mathematical program, which requires the assumption that the sets $\Theta_i, i = 1, \ldots, N$ be **discrete and finite**, and yields the integer program:

$$\min_{\theta \in \Theta} \sum_{i=1}^{N} \sum_{j=1}^{N} C_{ij\theta_i\theta_j}, \tag{5}$$

where n_k is the transfer flow at transfer connection k, $k = 1, \ldots, M$, $C_{ijrs} = \sum_{k \in A_{ij}} n_k W_k(r, s)$, $r \in \Theta_i, s \in \Theta_j$, $A_{ij} = \{k$: connection k goes from line i to line $j\}$, $W_k(r, s)$ is the mean waiting time at connection k, for offset times $r \in \Theta_i$ and $s \in \Theta_j$, for lines i and j, respectively. The key elements are the waiting times, which must somehow be estimated. This problem is equivalent to the 0-1 quadratic optimization problem:

$$\min \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{r \in \Theta_i} \sum_{s \in \Theta_j} C_{ijrs} x_{ir} x_{js},$$

$$\text{subject to} \quad \sum_{r \in \Theta_i} x_{ir} = 1, \; i = 1, \ldots, N, \quad x_{ir} \in \{0, 1\}.$$

The 0-1 variables x_{is} take the value 1 if and only if offset time s is chosen for link i, and the equality constraints insure that exactly one of the allowable offset times is chosen for each line. This formulation is equivalent to the well-known quadratic assignment problem (QAP) in facilities layout planning, and hence is $\mathcal{NP}$-complete.

A more realistic model includes the following features: the feasible set of offset times is continuous; headways need not be constant nor deterministic, e.g., they could be closer during rush hours; travel times need not be constant nor deterministic, i.e., they are likely to be random and higher during rush hours; the passenger arrival process need not be deterministic. However, incorporating such factors into a model leads to analytical intractability in determining the mean waiting times, in which case the best approach is a stochastic discrete-event simulation model. Thus, we incorporated these more realistic features into a simulation model, and then addressed the optimization problem (5) by allowing Θ_i to be continuous, in particular, corresponding to intervals $[0, K_i]$, where K_i is the maximum allowable offset time on transit line i. Our assumption is that the optimum (at least local) is found at a zero gradient point of the objective function in (5), so that the method of SPSA is applicable.

4 EXPERIMENTAL RESULTS

We considered a four-line transit network model, which comprises transit lines traveling in four directions on a grid: east, west, north, and south. These four lines are represented by the two bi-directional routes in Figure 1: an east-west route and a north-south route. There are just three stops on each line: an origin point, a potential transfer point which we will call the *center* point, and a destination point. Bus travel between stops is represented by eight different traffic "links" labeled in Figure 1 on the four bi-directional segments. Henceforth, we will refer to the transit vehicles as "buses" and the transit lines as "bus lines."

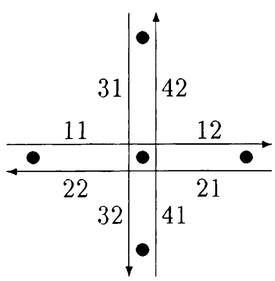

Figure 1: Schematic of Four-Line Traffic Network

In the implementation of SPSA, two positive sequences converging to zero at the appropriate rate are required:

- the step size multiplier sequence $\{a_n\}$, and
- the gradient estimate difference sequence $\{c_n\}$.

In our experiments, we took $a_n = a/n^\alpha$ and $c_n = c/n^\beta$, where α, β, a and c are constants to be selected. Also, we took the Δ_i's to be symmetric Bernoulli in all of our simulation experiments.

We built a simulation program of the transit network using the SIMAN simulation language, and then attached the SPSA optimization shell to it. The details, including excerpts of code, are given in Appendix A.

We began by considering a very simple experiment where only four customers rode in the entire day, one at each of the origin points, with routes chosen randomly. We chose a fixed headway of 10, uniformly distributed bus travel times, and considered various arrival time distributions. We began with deterministic arrivals, with all four customers arriving exactly at time 10. However, the optimization scheme failed in this case, because the objective function was insufficiently smooth.

4.1 OBJECTIVE FUNCTION BEHAVIOR

The smoothness difficulty is intuitively clear to most transit (or even airline) riders: if you just miss a connection – whether transfer or origination – your wait jumps from small to large, as you must wait for the next vehicle. We can easily see this analytically by considering just a single isolated rider in the system, for whom we will define the following:

$$\begin{aligned} W &= \text{wait time}, \\ X &= \text{arrival time}, \\ H &= \text{headway of vehicles} = 10, \\ \theta &= \text{offset of first vehicle} \in [1, 19]. \end{aligned}$$

Let us consider various distributions for X, and consider just the initial wait time (before transfers).

Deterministic. For $X = \theta$, it is obvious that

$$W = \begin{cases} 10 - \theta & 0 \le \theta \le 10 \\ 20 - \theta & 10 \le \theta \le 20 \end{cases} .$$

As shown in Figure 2, this function is discontinuous and has a constant slope, so stochastic approximation techniques are not applicable.

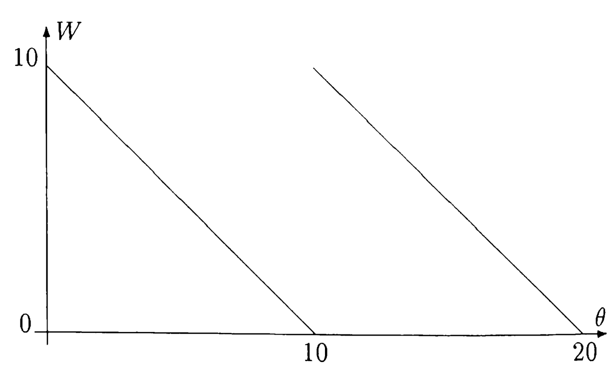

Figure 2: Discontinuous Objective Function

Uniform. For $X \sim U(\theta - 0.5, \theta + 0.5)$, it can be easily verified that

$$E[W] = 0.5 + \begin{cases} 10 - \theta & 1 \le \theta \le 10 \\ 9\ (\theta - 10) & 10 \le \theta \le 11 \\ 20 - \theta & 11 \le \theta \le 20 \end{cases}.$$

In this case, shown in Figure 3, the objective function is continuous, but the derivative is still discontinuous, and the minimum does not occur at a zero gradient point (no such point exists).

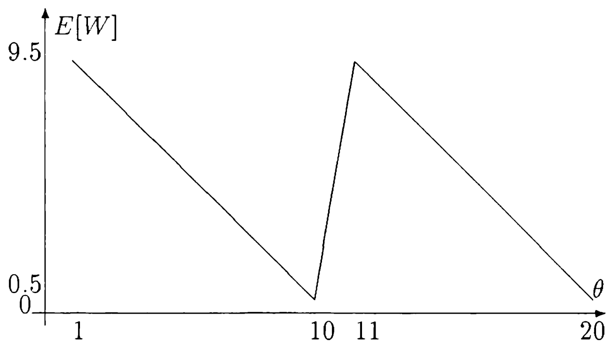

Figure 3: Discontinuous Derivative

Triangular. For $X \sim TR(\theta - 0.5, \theta, \theta + 0.5)$, we have

$$E[W] = \begin{cases} 10 - \theta & 1 \le \theta \le 9 \\ 10 - \theta + 5\ (\theta - 9)^2 & 9 \le \theta \le 10 \\ 20 - \theta - 5\ (\theta - 11)^2 & 10 \le \theta \le 11 \\ 20 - \theta & 11 \le \theta \le 19 \end{cases},$$

which has continuous derivatives, but discontinuous second derivatives at the points $\theta = 9, 10, 11$ (i.e., at these 3 points modulo 10). This function does possess a zero gradient point (9.1 mod 10), but as can be seen in Figure 4, the function is not very well behaved, having very steep approaches to the minimum.

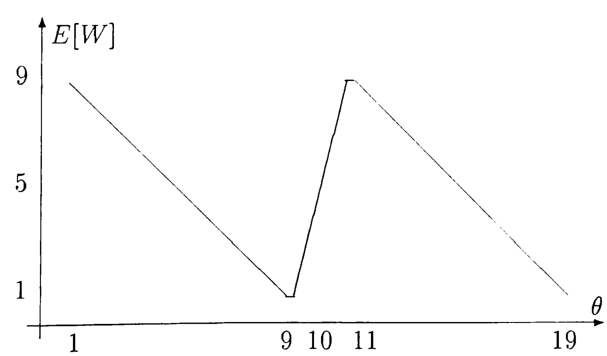

Figure 4: Discontinuous Second Derivative

4.2 EXAMPLE 1

We conducted simulation experiments using the triangular distribution for the parameter values shown in Table 1. Three different step sizes, two different starting points, and two different values for the exponent of the finite difference step size were considered. The results are given in Tables 2 and 3, where the estimated values of the objective function are given with 95% confidence half-widths. At $\theta_{(0)} = (9,9,9,9)$, the average wait is approximately 6.73±0.32. At $\theta_{(0)} = (9,7,13,11)$, the average wait is approximately 5.62±0.05. Based on the analysis of the last section, which was not complete, as it did not consider the transfers, the optimum was guessed to be at $\theta_* =$ (10.9,10.9,10.9,10.9), which yielded an estimated average wait of approximately 2.19±0.29. The best results for the parameter values occur at $a = c = 1.0$, with $\alpha = 0.751$, yielding estimates of the average wait which have lower means than the estimate for the optimum; however, the confidence intervals overlap, so the experiments were not statistically conclusive.

Table 1: Example 1 Parameter Values (36 cases total)

Θ	(0,19)		
$\theta_{(0)}$	(9,9,9,9)	(9,7,13,11)	
α	1.0	0.751	
β	0.25		
a	1.0	3.0	10.0
c	1.0	0.5	0.1

Table 2: Example 1 Results
$\theta_{(0)} = (9,9,9,9)$; $E[\bar{W}] = 6.73 \pm 0.32$;
$\theta_* = (10.9, 10.9, 10.9, 10.9)$; $E[\bar{W}] = 2.19 \pm 0.29$;
$\beta = 0.25$

$E[\bar{W}]$		a		
		1.0	3.0	10.0
	1.0	3.01±0.19	3.03±0.16	4.21±0.06
c	0.5	2.10±0.20	2.17±0.20	4.91±0.07
α=1.0	0.1	2.32±0.16	2.73±0.24	3.87±0.07
	1.0	2.04±0.15	4.10±0.19	5.45±0.05
c	0.5	2.43±0.07	2.98±0.13	5.36±0.04
α=0.751	0.1	3.41±0.33	4.48±0.07	5.78±0.05

4.3 EXAMPLE 2

This differs from Example 1 only in the number of customers simulated, which was increased from 4 to 80 (20 on each line). We fixed $a = 1.0$ for all the runs, and just considered the first starting point

Table 3: Example 1 Results
$\theta_{(0)} = (9, 7, 13, 11)$; $E[\bar{W}] = 5.62 \pm 0.05$;
$\theta_* = (10.9, 10.9, 10.9, 10.9)$; $E[\bar{W}] = 2.19 \pm 0.29$;
$\beta = 0.25$

$E[\bar{W}]$		a		
		1.0	3.0	10.0
	1.0	3.04±0.21	2.97±0.04	4.75±0.07
c	0.5	4.04±0.20	3.49±0.05	4.96±0.05
α=1.0	0.1	3.25±0.16	5.23±0.33	3.65±0.05
	1.0	2.10±0.18	3.31±0.21	4.25±0.15
c	0.5	3.42±0.16	3.96±0.05	3.82±0.11
α=0.751	0.1	3.84±0.08	3.96±0.05	4.44±0.04

$\theta_{(0)} = (9, 9, 9, 9)$; otherwise, we again used the parameter values shown in Table 1. The results are given in Tables 4 and 5, where the estimated values of the objective function are given with 95% confidence half-widths. At $\theta_{(0)} = (9,9,9,9)$, the average wait is approximately 5.80±0.18. The only difference between the two tables is that a different seed set was used. It can be seen that the algorithm exhibits a lot of randomness, as the second seed set performed much better than the first. The final values for the offsets in the two best results in Table 5 ($\alpha = 1.0$ with $c = 1.0$ and $c = 0.5$) were $\theta_{(500)} = (11.60,10.86,11.56,10.94)$ and $\theta_{(500)} = (11.34,10.64,11.52,10.78)$, which are quite close to the guessed optimal of $\theta_* = (10.9,10.9,10.9,10.9)$. In fact, in these two cases, the estimated performance was better than the guessed optimal, and quite an improvement over that at $\theta_{(0)}$.

Table 4: Example 2 Results (seed set 1)
$\theta_{(0)} = (9, 9, 9, 9)$; $E[\bar{W}] = 5.80 \pm 0.18$;
$\theta_* = (10.9, 10.9, 10.9, 10.9)$; $E[\bar{W}] = 2.46 \pm 0.16$;
$\beta = 0.25$; $a = 1.0$

$E[\bar{W}]$	c		
	1.0	0.5	0.1
$\alpha = 1.0$	2.99±0.12	3.03±0.10	4.34±0.05
$\alpha = 0.751$	3.53±0.07	3.45±0.07	5.07±0.13

Table 5: Example 2 Results (seed set 2)
$\theta_{(0)} = (9, 9, 9, 9)$; $E[\bar{W}] = 5.80 \pm 0.18$;
$\theta_* = (10.9, 10.9, 10.9, 10.9)$; $E[\bar{W}] = 2.46 \pm 0.16$;
$\beta = 0.25$; $a = 1.0$

$E[\bar{W}]$	c		
	1.0	0.5	0.1
$\alpha = 1.0$	2.03±0.10	2.07±0.13	4.01±0.05
$\alpha = 0.751$	4.10±0.07	3.25±0.07	3.75±0.13

5 CONCLUSIONS

We have applied the technique of SPSA to the transfer optimization problem of minimizing average waiting time in the system with respect to the scheduled offset times of the transit vehicles. Naive application revealed discontinuities in the objective function, leading to the inapplicability of gradient-based stochastic approximation methods. In particular, we found that the interarrival time densities must be continuous, which implies that the performance measure is twice differentiable. However, even in this case, the objective function does not satisfy the sufficient conditions of Spall (1992), which requires a continuous third derivative. We conducted preliminary simulation experiments for a triangular distribution, which demonstrated the sensitivity to the various parameters of the algorithm, such as the step size and finite difference step. However, at certain settings of these parameters, the algorithm performed very well, improving upon the initial schedule significantly. Further experiments are currently underway.

APPENDIX A: SIMAN CODE

In Figures 5 and 6, we list portions of the SIMAN program for the transfer network, for the MODEL code only. There are the two main parts of the code: the network simulation itself, and the SPSA optimization shell.

As delineated in Figure 5, the network simulation consists of three portions: rider arrivals, bus travel, and rider travel and transfers. For each of these, there is a corresponding general form, which is duplicated for the different lines. To save space, we just included one portion of each to illustrate the main constructs used in simulating the network.

Passengers are created for each line, and randomly assigned a route upon arrival. These routes are specified in the SEQUENCES element in the experiment (EXP) file.

Buses travel along the links of each line. Line i (i=1,...,4) consists of LINKi1 and LINKi2. The time interval between bus departures equals headway(i), with the first bus departing at t=offset(i). Travel time along LINKij takes TravTime(i,j) time units, randomly distributed. When a bus completes travel along a link, it signals riders that that portion of its trip is complete. For SPSA implementation, the epsilon term is needed to READ in new values of the parameters th(i) *before* a bus is scheduled; otherwise a delay of th(i) is identical to CREATE offset of th(i).

Passengers flow through the network by moving from link to link, according to their preassigned

```
;*******************************************************************
;GENERATE RIDER ARRIVALS TO EACH LINE,
;ASSIGN ROUTES & SEND TO ORIGINATION
        CREATE,       BatchSize,TR(InterArrv1,1):TR(InterArrv1,1),
                      maxnum;
        ASSIGN:       NS = DP(Type1,2);            Assign route
        COUNT:        No. Riders Line #1;
        ROUTE:        ;
;*******************************************************************
;SIMULATE BUS RUNS ALONG EACH LINE
        CREATE,       ,epsilon:headway(1),maxbus:MARK(TimeIn);
        DELAY:        th(1);
        ASSIGN:       TravTime(1,1)=TR(range11,3);
        SIGNAL:       11,BusCap;
        DELAY:        TravTime(1,1)+slack(1,1);
        ASSIGN:       TravTime(1,2)=TR(range12,3);
        SIGNAL:       12,BusCap;
        DELAY:        TravTime(1,2):DISPOSE;
;*******************************************************************
;DEFINE STATIONS STOPS/LINKS, WAITING QUEUES & SIMULATE PASSENGER FLOW
        STATION,      Link11;
        QUEUE,        Transit11:MARK(XferTime);
        WAIT:         11,BusCap;
        TALLY:        Waits,TNOW-XferTime;
        ROUTE:        TravTime(1,1)-epsilon;
;*******************************************************************
```

Figure 5: The Three Main Modules of the Network Simulation Model

```
          CREATE;      !create single entity for initialization
          READ,       Offsets,,1:  !read in parameter values
                        offset(1),offset(2),offset(3),offset(4);
          CLOSE,      Offsets;
          ASSIGN:     iter=AINT((NREP+1)/2):
                      cn=cmult/SQRT(SQRT(iter));
 Continu0 BRANCH,     1:IF, NREP/2 .EQ. ANINT(NREP/2), NegRun:
                      ELSE, PosRun;
;;;;;;;;calculate J+;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 PosRun   ASSIGN:     run=1: i=1;
 LoopBern ASSIGN:     pert(i)=DP(Bernoulli,4):
                      th(i)=offset(i)+cn*pert(i): i=i+1;
          BRANCH,     1:IF, i .LE. numpar, LoopBern: ELSE, WaitEnd;
;;;;;;;;calculate J-;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 NegRun   ASSIGN:     run=2: i=1;
 Loop2    ASSIGN:     th(i)= offset(i)-cn*pert(i): i=i+1;
          BRANCH,     1:IF, i .LE. numpar, Loop2: ELSE, WaitEnd;
;;;;;;;;;updates;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
 WaitEnd  DELAY:      endarrv;
          ASSIGN:     maxnum=0;   turn off arrival of riders
          DELAY:      endday;     end day shortly afterwards
          ASSIGN:     pm(run)=TAVG(Waits): i=1;
          BRANCH,     1:IF, run .EQ. 2, Loop3:  !update parameters
                      ELSE, EndRep;           !next rep
 Loop3    ASSIGN:     dpm(i)=(pm(1)-pm(2))/(cn*pert(i)): i=i+1;
          BRANCH,     1:IF, i .LE. numpar, Loop3: ELSE, Continu1;
 Continu1 ASSIGN:     i=1;
 LoopSPSA ASSIGN:     thOld(i)=offset(i):
                      offset(i)=offset(i)-dpm(i)*mult/iter:
                      temp=(offset(i) .GT. minoff .AND.
                            offset(i) .LT. maxoff)+
                      (offset(i) .LE. minoff)*(minoff-thOld(i))/
                        (offset(i)-thOld(i)-epsilon)+
                      (offset(i) .GE. maxoff)*(maxoff-thOld(i))/
                        (offset(i)-thOld(i)+epsilon):
                      proj=temp*(temp .LT. proj) +
                        proj*(temp. GE. proj);
          ASSIGN:     i=i+1;
          BRANCH,     1:IF, i .LE. numpar, LoopSPSA:
                      IF, proj .EQ. 1, Continu2:
                      ELSE, Projectn;
 Projectn ASSIGN:     i=1;       projection;;;;;;;;;;;;;;;;;;;;;;;;;;
 Loop4    ASSIGN:     offset(i)=thOld(i)+
                        0.9*proj*(offset(i)-thOld(i)): i=i+1;
          BRANCH,     1:IF, i .LE. numpar, Loop4: ELSE, Continu2;
 Continu2 ASSIGN:     proj=1;
 EndRep   WRITE,      Offsets,"(6F6.2,4F4.0)":
                        offset(1),offset(2),offset(3),offset(4);
          CLOSE,      Offsets;
          ASSIGN:     TFIN=TNOW:DISPOSE;   !end replication now
```

Figure 6: SPSA Optimization Shell

routes. To simulate movement through the network, the riders wait in queue until their bus departs for travel along a link. At the time of departure, the bus transmits a signal to its riders. When riders receive receive this signal, they travel to the next link of their trip and wait for the next departure along the link. Riders continue in this fashion until they reach their destination.

The optimization shell is built by making a replication correspond to an iteration, where an outside file is used to alter the values of the parameters between replications by reading at the beginning and then writing at the end. In this way, common random numbers can easily be employed to reduce the variance of the SP estimates. A "dummy" entity is created each iteration in order to execute the optimization shell. This entity first initializes the iteration, and then waits until the end of the simulation, when it does the updating and finally terminates the replication/iteration.

REFERENCES

Bookbinder, J.H. and Désilets. A., Transfer optimization in a transit network. *Transportation Science* 26, No. 2, May 1992: 106-118.

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

Fu, M.C.. 1994. Optimization via simulation: a review. *Annals of Operations Research* 53: 199-248.

Fu, M.C. and Y. C. Ho. 1988. Using perturbation analysis for gradient estimation, averaging, and updating in a stochastic approximation algorithm. In *Proceedings of the Winter Simulation Conference,* ed. M.A. Abrams, P.L. Haigh, and J.C. Comfort, 509-517.

Hill, S.D. and M. C. Fu. 1994. Optimizing discrete event systems with the simultaneous perturbation stochastic approximation algorithm, *Proceedings of the 33rd IEEE Conference on Decision and Control*: 2631-2632.

Hill, S.D. and M. C. Fu. 1994. Simulation optimization via simultaneous perturbation stochastic approximation. In *Proceedings of the Winter Simulation Conference*, ed. J.D. Tew, S. Manivannan, D.A. Sadowski, and A.F. Seila, 1461-1464.

Spall, J.C. 1992. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE Transactions on Automatic Control 37*: 332-341.

Spall, J.C. and J.A. Cristion. 1994. Nonlinear Adaptive Control Using Neural Networks: Estimation with a Smoothed Simultaneous Perturbation Gradient Approximation, *Statistica Sinica 4*:1-27.

AUTHORS BIOGRAPHIES

STACY D. HILL is a Systems Engineer on the Senior Professional Staff of the Johns Hopkins University Applied Physics Laboratory. He received the D.Sc. degree in control systems engineering and applied mathematics from Washington University in 1983. He has been project and technical lead in developing, testing, and applying statistical estimation techniques and software, and has authored and co-authored journal articles on estimation theory and techniques, including an article in the cross-disciplinary book *Bayesian Analysis of Time Series and Dynamic Models.* His present research interests include optimization and simulation of discrete event systems, with application to transportation problems. Dr. Hill is a member of **IEEE**, **ASA**, and **Sigma Xi**.

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