Proceedings of the 1987 Winter Simulation Conference
A. Thesen, H. Grant, W. David Kelton (eds.)

OPTIMIZATION IN SIMULATION: A SURVEY OF RECENT RESULTS

Marc S. Meketon
AT&T Bell Laboratories
Holmdel, NJ 07733, U.S.A.

ABSTRACT

This paper surveys existing methods, and presents several new ideas, for optimizing performance measures with respect to input parameters for simulation. The usual methods fall into three categories. First, there is the application of traditional non-linear programming techniques, regardless of the stochastic properties of most discrete event simulations. Second, is the application of response surface methodologies. Third, are stochastic approximation techniques, a well known but little used optimization technique. The last two categories account for the stochastic behavior of simulations.

This paper also discusses several developments within the past seven years that promise greater efficiency in optimizing simulations. These developments include: Karmarkar's algorithm, infinitesimal perturbation analysis and likelihood ratios to estimate derivatives of performance measures with respect to parameters, adaptive control and hybrid models.

1. INTRODUCTION

This paper has three goals. The first goal is classify methods of optimization for simulations by the amount of structure or information required by the method. Methods that require detailed knowledge of the simulation usually yield optimal values for parameters faster.

The second goal is to survey the post-1975 literature of optimization techniques in simulation. The popular techniques prior to 1975 have been summarized and compared in the paper by Farrell, McCall, and Russell (1975). One area not covered in that paper is the use of stochastic approximation techniques for simulation optimization. This overlooked subject is recently gaining much attention and deserves a prime spot of its own. The massive theoretical knowledge of stochastic approximation, developed during the past 36 years, contains valuable insights into the do's and do not's, and the asymptotic efficiencies, of several optimization procedures.

Another review of optimization techniques in statistics (not necessarily simulation) is Rustagi (1978).

There has been a lot of exciting work in simulation optimization, especially since 1982, on using perturbation analysis (Section 5.4), and likelihood ratios (Section 5.8). This new work is the focus of the third goal in the paper, namely to present new ideas and directions.

This paper was originally presented as a tutorial in the Winter Simulation Conference, 1983. Unfortunately, it was not written for the proceedings of that conference. This paper also includes the new work on likelihood ratios, more results on perturbation analysis, further results in the application of stochastic approximation when coupled with perturbation analysis (what was coined 'single run optimization' in the WSC 1983 tutorial), and the discussion of the (possible) significance of Karmarkar's algorithm to constrained optimization problems in simulation.

Lastly, this paper is not self contained. In many cases, there will be an informal or non-rigorous discussion of a technique; references that explain the technique in detail are provided.

2. A WORKING DEFINITION OF OPTIMIZATION IN SIMULATION

Simulation is an experiment with inputs, outputs, and a controlled environment. The outputs, often called the *performance measure*, is the (stochastic) function that needs to be optimized. It is assumed throughout the paper that this is a real-valued (i.e. single valued) function. On the other hand, the inputs, called the *parameters*, are usually multivariate. We will denote the inputs by x, the output by $Y(x)$, and the expected value of $Y(x)$ by $M(x) = E[Y(x)]$.

There are two classes of objectives of optimization procedures. The first is the *level crossing*, or root finding, problem. In this case, we wish to find the value of x so that $M(x) = p$, for some prespecified p. For example, let $Y(x)$ be the fraction of customers that wait more that 3 minutes in a queue, and x be the service rate. If the service rate is adjustable (perhaps by adding more servers), then it may be desirable to find the service rate so that customers wait more that 3 minutes only 5% of the time. Hence, we wish to find x so that $M(x) = .05$.

The second class is a minimization (or maximization) problem. Here, we wish to discover an x that minimizes $M(x)$. For example, suppose that $Y(x)$ is the average number of customers in a tandem queue, where x is the service time for the first queue and $6\text{-}x$ for the second queue. (There is a tradeoff between placing resources at the two queues.) We wish to minimize $M(x)$, subject, of course, to $0 \leq x \leq 6$. *Throughout this paper, we will assume minimization* is the default criteria.

Naturally, these two classes of optimizing problems can be unified. The level crossing problem is the same as minimizing $E[(Y(x)-p)^2]$, if the variance of $Y(x)$ is constant. And we could have treated the minimization problem as a level crossing of the first derivative of $M(x)$. However, since there are different algorithms for these two classes we will continue to differentiate these two classes.

3. CLASSIFICATION OF OPTIMIZATION TECHNIQUES

Figure 1 classifies various optimization techniques with regard to the information that is known or assumed about the simulation. These methods will be described in varying levels of detailed in the remainder of the paper.

The first level assumes that the output of the simulation, $Y(x)$, is a deterministic number that can be replicated, and implicitly assumes that $Y(x)$ is a continuous function of x. We note that in stochastic experiments $Y(x)$ is usually considered a random variable, although $Y(x)$ can be duplicated if the random number seed is held constant between runs. However, $Y(x)$ is not continuous, since a small change in x may change the sequence of events and hence $Y(x)$ is not continuous.

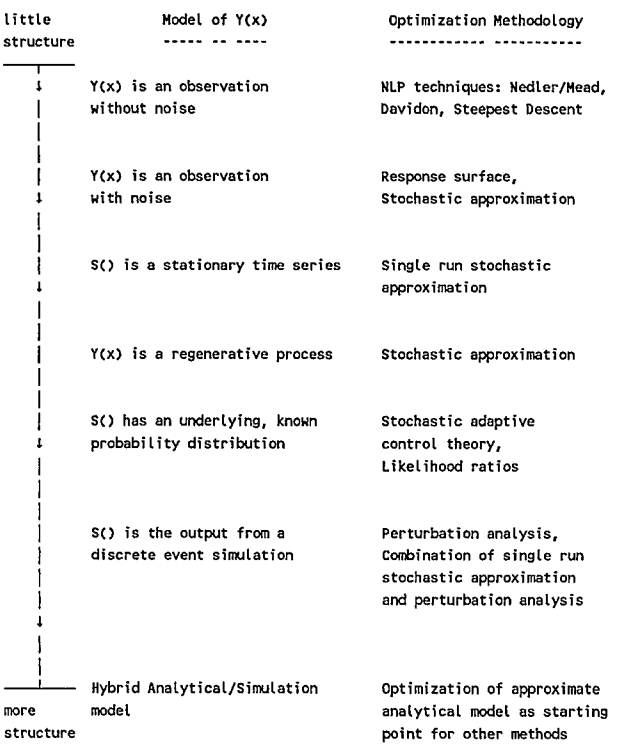

Figure 1. Classification of Optimization Techniques

The second level assumes that $Y(x)$ is a random variable and, with probability one, will optimize $M(x)$. The convergence can be very slow, especially for the minimization problem. Without assuming any more structure, some of these algorithms (e.g. certain modified forms of stochastic approximation) can be shown to be asymptotically optimal.

Let $S(t|x)$ be the underlying stochastic process in the simulation. If that process is stationary (for a given x) then *single run optimization* techniques, where parameters change during the course of the simulation, may be applicable. This reduces the initialization bias substantially.

If we assume that $S()$ is a regenerative simulation, then initialization bias is eliminated and the usual assumptions for stochastic approximation hold. The regenerative assumption is *non-parametric* since in does not involve the estimation of model parameters. If we assumed that $S()$ is an autoregressive / moving average model of known order, which is a *parametric* assumption, then stochastic adaptive control techniques are applicable. Other parametric models lead to likelihood ratio techniques.

The sixth level assumes that $Y(x)$ is the output of a discrete event simulation, and that the simulation can be modified to keep track of virtual changes in $Y(x)$ with respect to x. However, faster convergence for the minimization problem is possible using the techniques of this level.

The remaining level assumes additional analytical models are known that approximate $Y(x)$, such as ARMA models or analytical models.

4. NON-LINEAR PROGRAMMING TECHNIQUES

Since 1975, several authors have tried traditional non-linear programming techniques for the unconstrained minimization problem. The procedures differ by the use of (estimated) first and second derivatives.

The *Nelder & Mead* sequential simplex method does not use any derivative information, and therefore is a natural choice for simulation problems where derivative's may not be practical to calculate. When optimizing in an n-dimensional space (i.e. x is an n-vector), this method first guesses at $n+1$ points, x^1, ..., x^{n+1}, (which define a *simplex*) and runs the simulation for each of those points, yielding $Y(x^1)$, ..., $Y(x^{n+1})$. To maximize the $M(x)$, pick the 'worst' point and choose a new point that is reflected away from the worst point. Details are given in Avriel(1976, pp 245-247).

Convergence is not guaranteed when $Y(x)$ is observed with noise (i.e. $Y(x) = M(x) + e$, where e is a non-degenerate random variable with zero mean) for this and the following traditional non-linear techniques. In fact, these techniques generally converge to a neighborhood of the optimal and bounce around that neighborhood without converging. In order to obtain convergence, stochastic approximation techniques need to be used (Section 5).

Optimization techniques that require derivatives include steepest descent (e.g. Avriel 1975), and Davidon's variable metric technique (Avriel 1975, chapter 11). Often, these methods calculate a descent direction, and then perform a one dimensional search along that descent direction. A good method for performing the line search in the single run optimization method described in Section 5.7. Although a line search is not required for Davidon's variable metric technique, which approximates the second derivative (or Hessian) matrix.

Empirically, Davidon's method is good in the early stages of optimization: If the initial guess is far away from the optimal, this method will get you close to the optimal. The Nelder & Mead method works well close to the optimum. Finally, steepest descent methods that use previous data to estimate the derivatives may outperform the Nelder & Mead. For very 'noisy' problems, the Nelder & Mead method outperforms the other techniques.

These empirical results are based on the works of Segreti, Carter, and Wampler (1979, 1981), Barton(1984), and Deming (1977).

4.1 Implications of Karmarkar's Algorithm for Constrained Optimization

We have avoided the issue of constrained optimization techniques because little, if any, significant new work has been accomplished in this field that was applied to simulations. However, Karmarkar's linear programming algorithm offers the promise of enhanced methods for constrained optimization of simulations.

There are three general approaches to use for constrained optimization. Of course, these approaches are not exclusive and have been mixed and matched. The first is to use penalty or barrier functions that move the constraints into the objective function. For example, if a constraint was of the form $g(x) = b$, then a term $w(g(x)-b)^2$ might be added to the objective function for sufficiently large w.

The second approach is to use *active constraint analysis*, which usually treats constraints of the form $g(x) \leq b$.

The last approach is to linearize the constraints and objective function, and then use a linear programming approach to obtain a direction of decrease in the objective function subject to remaining within the constraints. This is sometimes known as successive linear programming. Within a simulation approach, linearizing the constraints and objective function potentially requires much computational effort, since derivatives must be estimated.

However, we believe that this last approach has promise, due to the Karmarkar linear programming procedure in 1984 (Karmarkar (1985), see also Vanderbei, Meketon, and Freedman (1986) for the popular 'affine scaling' variant). This algorithm has several good properties, but for the purposes of simulation optimization, the most important property is that it requires far fewer iterations than other linear programming algorithms.

When used for constrained simulation optimization, reducing the number of iterations - and therefore reducing the

number of simulation runs - is far more important that reducing the computational effort given the linearized equations. Hence, the promise of Karmarkar's algorithm is a significantly faster method for constrained optimization.

5. OPTIMIZATION METHODS FOR OBSERVATIONS WITH NOISE

The second category of optimization techniques have been designed for noisy experiments. When they are applied to optimization of non-random functions, they will converge very slowly compared to the methods described above. However, they generally will converge to the optimum in the presence of noise unlike the traditional non-linear programming methods.

The best known technique for this is response surface methodology. The technique has been used successfully, and described in many articles, prior to 1975. In particular, Farrell, McCall, and Russell(1975) and Smith(e.g. 1973) spend considerable time on this subject. Post 1975, Biles(1977) described its use for optimization when there are multiple objectives.

A well known statistical technique, apparently custom designed for optimization with noisy functions, but one that has seen little use in the simulation field, is *stochastic approximation*. In Azadivar and Talavage (1980), a general approach that uses stochastic approximation is developed. Rubinstein (1982, 1983) and Glynn (1986a) have recently re-examined the implications of stochastic approximations for simulations.

Due to the importance of stochastic approximation in the following, we will present a short introduction to it.

5.1 Stochastic Approximation - Level Crossing Problem

The original stochastic approximation algorithm was applied to the level crossing problem, i.e. to find the x^* so that $M(x^*) = p$. Assume that x is a real number (i.e. a 1-vector). With mild assumptions on $M(x)$ and on $Y(x)$, it can be proved that the following sequence converges to x^*:

Let x^1 be arbitrary.

$$x^{n+1} = x^n - (a/n)(Y(x^n) - p)$$

It should be understood that this is a sequential procedure. After choosing x^1, the simulation is run and $Y(x^1)$ is observed. Then, according to the formula above, x^2 is calculated, and the simulation re-run to observe $Y(x^2)$.

This was first introduced by Robbins and Monroe (1951) and is termed the *Robbins-Monroe (RM) stochastic approximation* procedure. They showed under rather restrictive assumptions the mean squared convergence to x^*. Since that time, the assumptions have been considerably relaxed and the knowledge of this procedure has been increased substantially. Here we review several of the more important features of this stochastic approximation procedure.

If $\mathbf{x}^n$ is the average of x^1 through x^n, and $\mathbf{Y}^n$ is the average of $Y(x^1)$ through $Y(x^n)$, then

(1) $x^{n+1} = \mathbf{x}^n - a(\mathbf{Y}^n - p)$. If we consider $M(x)$ to be a differentiable function, and if we force $x^{n+1} = x^*$, then $(1/a) \sim (\mathbf{Y}^n - p)/(\mathbf{x}^n - x^*)$ which indicates that the optimal choice of a is the reciprocal of the slope of $M(x)$ at x^*. Let $m = M'(x^*)$. See Lai and Robbins (1979, Lemma 1).

(2) For $a > 1/2m$, the asymptotic variance of x^n is proportional to $1/n$. Indeed, if $\mathrm{Var}[x^n] = s^2$, then

$$(x^n - x^*)/n^{1/2} \sim N(0, 2a^2s^2/(2am - 1))$$

(3) If $Y(x^n) \sim N(M(x^n), s^2)$, then the Cramer-Rao lower bound of the variance of any estimator coincides with the variance of the stochastic approximation estimator. (See Anbar (1973) and Abdelhamid (1973))

Several conclusions may be drawn from these facts. The estimator decreases with variance proportional to s^2/n, only for $a > 1/2m$. If batches of observations are taken for each step in the stochastic approximation procedure, the variance decreases in exact proportion to the batch size, so batching does not affect the asymptotic variance. However, fact (3) indicates that normality of the error term is important. Hence, large batch sizes would be important.

5.2 Single Run Optimization - Level Crossing Problem

Stochastic approximation can be used to adaptively update the parameters in a long, single, simulation run. Perhaps the easiest example is the simulation of an AR(1) process. The particular AR(1) process we choose is defined by

$$W_{n+1} = xW_n + 1 + e_n,$$

where $x < 1$ is a parameter and e_n is an iid sequence of $N(0,1)$ random variables.

Suppose we wish to find the x such that $E[W] = 2$. One way is to adaptively estimate the x by stochastic approximation. To wit:

$$W_{n+1} = x^n W_n + 1 + e_n$$

$$x^{n+1} = x^n - (a/n)(W_{n+1} - 2).$$

That is, choose an initial W_1 and x^1. Then simulate e_1, calculate W_2, and update x^2. Continue, simulating e_2, calculating W_3, and update x^3.

The advantages of this procedure is that the transient due to initialization bias is reduced. That is, if the parameter x was fixed, then $E[W_n]$ will converge to $(1-x)^{-1}$. The usual way is to run the simulation for a particular x, wait for the transient effects to diminish, estimate $E[W]$, and then change x. However, using single run optimization techniques will decrease the lost time due to transients. Figure 2 plots x^n versus n (jagged curve) and the cumulative average of x^n (smooth curve).

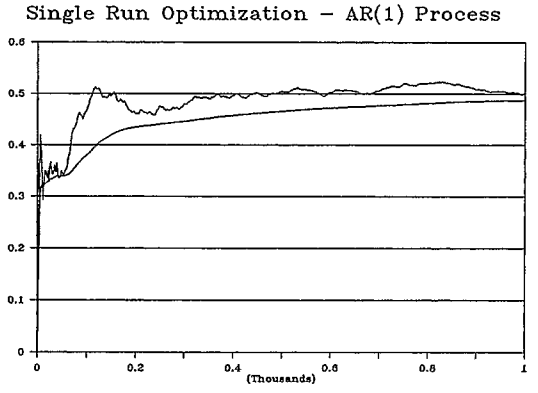

Figure 2: Graph of x^n

A more complicated example is a tandem queue (two queues in series). Queue 2 has a buffer size of 2, and Queue 1 has an infinite buffer. The interarrival time is an exponential random variable with mean length 5, queue 2 has exponential service times of mean length 4. The parameter of interest is the mean service time of queue 1, called x. The object is to find the mean service time of queue 1 so that the average number of customers in the system is 4.23 . The parameter x is updated every 500 customers by the formula:

$$x^{n+1} = x^n - (1/n)(N_n - 4.23)$$

where N_n is the average number of customers in the system based on the last 500 customers that entered the system.

Figure 3 plots x^n (jagged curve) and the cumulative average of the x^n's (smooth curve). Figure 4 plots the cumulative average of the number of customers in the system.

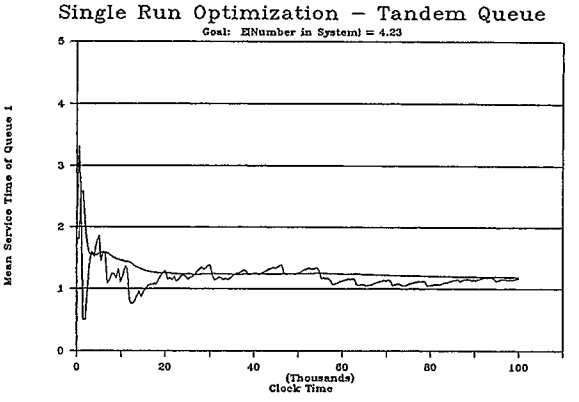

Figure 3: Mean Service Time

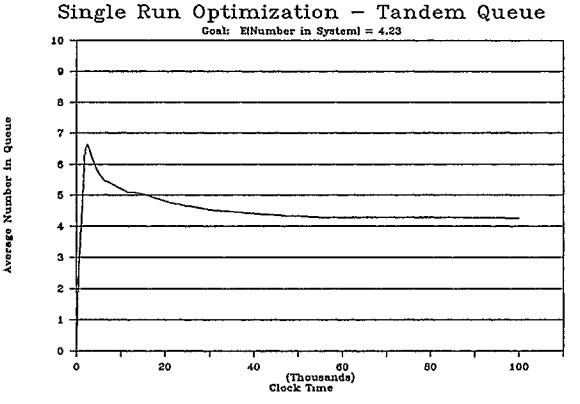

Figure 4: Average Number of Customers in System

Batching customers together is generally a good idea. The batching give approximate normality to N_n, and hence increases the efficiency of the algorithm. Further, there is a slight bias effect if the batch size is 1: if the parameter changes whenever a customer *enters* the system, then the number of customers in the system would tend to be biased high.

5.3 Stochastic Optimization - Minimization Problem

There is a version of stochastic approximation which can be used for the second type of optimization problem, namely to locate the maximum *of M(x)*. Suppose that $M(x)$ has a unique maximum at x^*. The sequential experiment,

$$x^{n+1} = x^n - (a/n)\{(Y(x^n - c_n) - Y(x^n + c_n))/2c_n\}$$

will, with probability one, converge to x^*. In this case, c_n is usually $c/n^{1/4}$.

This technique is known as *Kiefer-Wolfowitz stochastic approximation* technique, after the paper by Kiefer and Wolfowitz (1952). The technique estimates the derivative of the function $M(x)$ by the term $(Y(x^n - c_n) - Y(x^n + c_n))/2c_n$. Unfortunately, at the best $\mathrm{Var}[x^n]$ is proportional to $1/n^{1/2}$, which is very slow.

To speed up the K-W process, we would need to estimate the derivative $M(x)$ more directly. This can be done using the techniques of Perturbation Analysis or Likelihood Ratios.

5.4 Perturbation Analysis

One method for obtaining estimates of derivatives of $M(x)$ is *perturbation analysis* (see, e.g. Suri (1983)). There are two classifications of perturbation analysis (PA): finite perturbation analysis (FPA) is designed for inherently discrete parameters such as buffer size. FPA is an heuristic that approximates the difference of a performance measure when the discrete parameter(s) is perturbed by one unit, and has minimal known theoretical properties. Infinitesimal perturbation analysis (IPA) is used to obtain derivatives of continuous parameters. In this paper, we consider only IPA, for which several theorems can be proved.

The key to IPA is the concept of an *event tree*. An event tree is a tree where the nodes are events. Whenever an event is extracted from the event list, it causes other events to be placed on the event list in the simulation. The directed arcs of the tree point from the extracted event to the events that it directly generates.

Let's give an example of IPA using a G/G/1 queue where the interarrival times for customer j is distributed as $q + W_j$; q is non-random and W_j is an exponential random variable with mean 30. We wish to use IPA to estimate the derivative of the expected number of customers in the system with respect to q. The parameter q was chosen for simplicity - changing the parameter q by h will uniformly change all the interarrival times by h.

The event tree for the G/G/1 has two types of nodes: arrivals and departures. An arrival node always has an arc that leads to the next arrival. An arrival node may also have an arc that leads to a departure node if the arrival found the queue empty. A departure node may have an arc that leads to the departure of the next customer in the queue if the queue is not empty, otherwise the departure node is terminal - it has no arcs leading out from it.

Figure 5 is an example of an event tree for a realization of the G/G/1. The time of the event is depicted on the left-hand side of the middle of a box. The number in the right-hand side of the event boxes are the cumulative number of customers in the system up till the time of the event: the integral, from 0 to the current time, of the number of customers in the system.

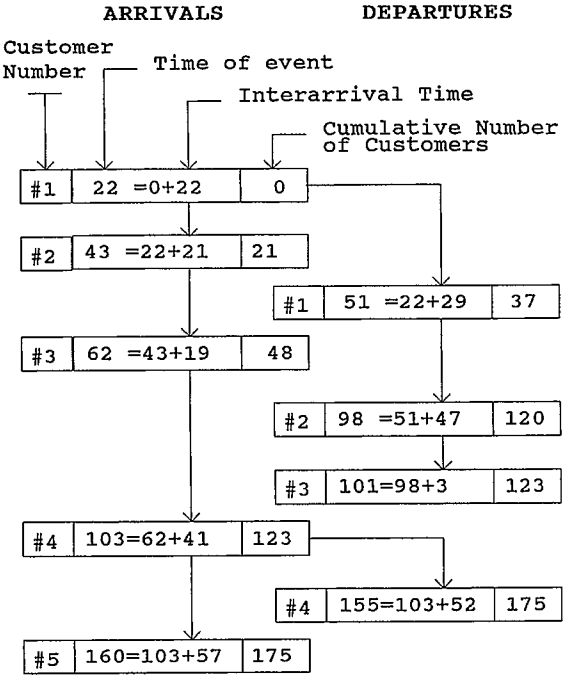

Figure 5: Nominal Event Tree

Figure 6 depicts the event tree when q is increased by 1. Note that the graph of the event tree has not changed. This would not be true if the interarrival time was increased by 10 instead of 1. For then customer #1 would depart the system at time 61, while customer #2 would enter the queue at time 63. Hence, when customer #1 left the system the queue would be empty, and there would be no arc from the departure of #1 to the departure of #2.

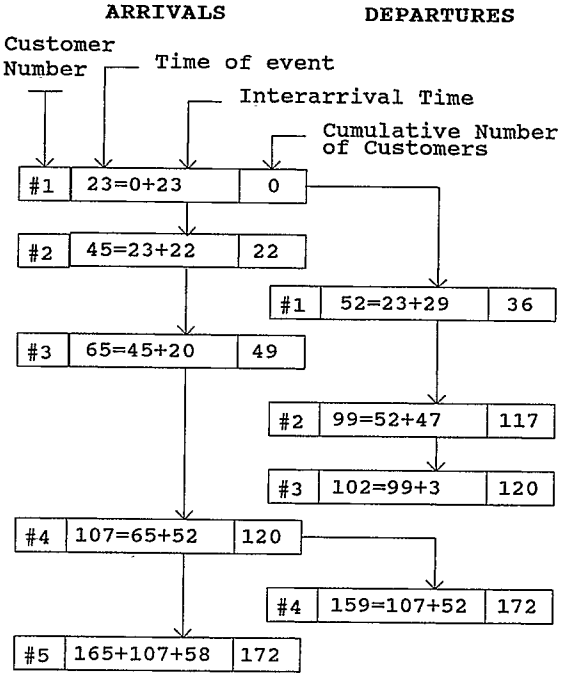

Figure 6: Event Tree with Interarrival Times Increased by 1

IPA computes the estimate by adding an accumulator, A, to the simulation, and by adding an attribute, S_j, to the event list. Recall that event lists have a number of attributes, such as time of event and type of event. Now we are adding another attribute. The algorithm works as follows.

(1) Let $A = 0$, give initial event attribute $S_0 = -1$.

(2) If event i schedules a departure event j, let $S_j = S_i$. Subtract S_j from A.

(3) If event i schedules an arrival event j, $S_j = S_i - 1$. Add S_j to A.

The algorithm is depicted in Figure 7.

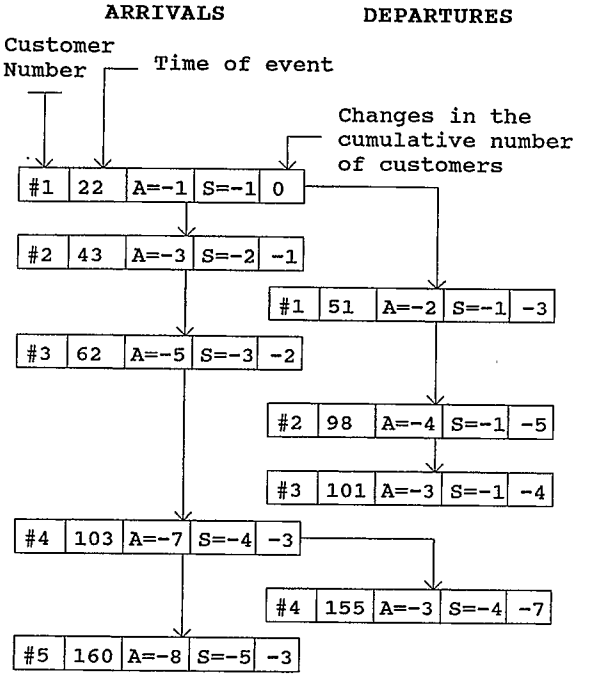

Figure 7: Illustration of IPA Algorithm

5.5 Common Random Numbers and Perturbation Analysis

Common random numbers is another technique that can be used to estimate derivatives. Indeed, if by actually changing the parameter q to $q+1$, then re-running the simulation with the same random number seed, we could obtain another estimate of the change in the number of customers in the system. If the event tree did not change, this estimate will coincide with the perturbation analysis estimate.

Unfortunately, if the simulation is run long enough the event tree will change. This dramatically increases the variance of the estimate of the derivative. IPA does not suffer from such problems. Also, when using common random numbers, the simulation will have to be re-run for every parameter that is changed. IPA only uses the one simulation run to simultaneously estimate all the derivatives.

5.6 Limitations of Perturbation Analysis

Heidelberger (1986) reports of situations where perturbation analysis would lead to the wrong answers - in situations such as multi-class queueing networks. In response, Gong and Ho (1987) suggests several changes to the perturbation algorithms that yield correct estimates. Though

IPA correctly calculates the infinitesimal change of $Y(x)$, it is not easy to prove that estimate is unbiased or ergodic.

5.7 Single Run Optimization - Minimization Problem

The single run optimization method used for the level set problem can be used for the minimization problem by incorporating IPA to estimate the derivatives.

For an example we consider a tandem queue, each with infinite buffers. The first queue has mean service length of x second queue has mean service length of $6\text{-}x$. Both service times are exponentially distributed. The interarrival times are exponentially distributed with mean length of 10. We wish to find the x that minimizes the expected number of customers in the system. Theoretically, this is $x^* = 3$, to give $E[N] = .8$. Figures 8 and 9 show how well the combination of Robbins-Monroe stochastic approximation (updating the estimates every 1000 customers) and IPA to estimate the gradient works in obtaining the optimal solution. After a clock time of 200,000 (approximately 20,000 customers), $x = 3.24$.

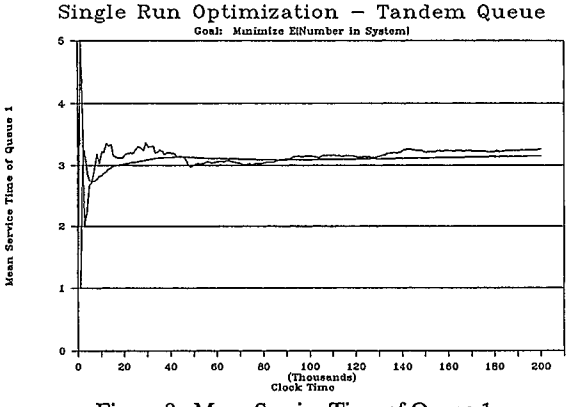

Figure 8: Mean Service Time of Queue 1

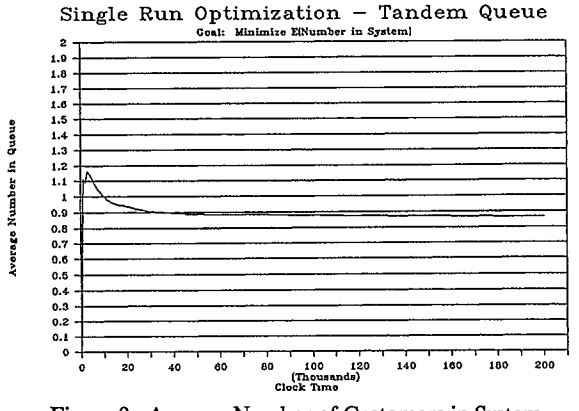

Figure 9: Average Number of Customers in System

Suri and Leung (1987) have recently corroborated these results for an M/M/1 queue, using *accelerated stochastic approximation* (Kesten (1958)).

5.8 Likelihood Ratio Methods

Suppose that there is an underlying Poisson process, $N(t)$, with known rate x, within the simulation. Reiman and Weiss (1986) showed that $d\,M(x)/dx = E[(N(T)/x - T)Y(x)]$, where T is the run length of the simulation. Hence, the statistic, $(N(T)/x - T)Y(x)$, gives a useful estimate of the derivative of $M(x)$ with respect to x. However, $Y(x)$ tends to $M(x)$ as T tends to infinity, so the variance of $(N(T)/x - T)Y(x)$ tends to be $O(T)$. To overcome this situation, the authors suggest using regenerative techniques to batch the sensitivity estimates. That is, calculate the statistic in each regeneration cycle and then average the statistics. Reiman and Nguyen (1987) were able to improve the estimators using control variables.

Glynn (1986b, 1986c) and Rubinstein (1987b) have independently used likelihood ratios (and variants) to estimate the sensitivities. Rubinstein (1987a, Section 3) compares a likelihood ratio method to IPA for monte carlo simulations. He concludes that (1) likelihood ratios are less sensitive to the complexity of the model, (2) the variance of IPA estimates is generally lower than likelihood ratio estimates, and (3) IPA assumptions are hard to verify (relating to interchanging expectation and derivative operators) and that its conditions for convergence are generally harder to achieve than in likelihood ratio methods.

6. OTHER OPTIMIZATION TECHNIQUES

6.1 Adaptive Control Techniques

One relatively unexplored technique for optimization of simulations is to use adaptive control theory. Essentially, the idea is to assume that $Y(x)$ is an autoregressive / moving average (ARMA) process with adjustable inputs. So for parameters $\{a_i\}$, $\{b_i\}$, and $\{c_i\}$,

$$Y(x) = Y(t|x) = a_1 Y_{t-1} + \ldots + a_p Y_{t-p}$$
$$+ e_t + c_1 e_{t-1} + \ldots + c_p e_{t-p}$$
$$+ b_0 x^t + b_1 x^{t-1} + \ldots + b_p x^{t-p}.$$

Here, the parameters $\{a_i\}$, $\{b_i\}$, and $\{c_i\}$ are unknown, but can be adaptively estimated using variants of the stochastic approximation technique. The goal is to input values of x^t so

that the average of $||Y(t) - Y^*(t)||^2$, from $t = 1, ..., n$ is minimized.

Two recent papers are Becker, Kumar, and Wei (1983) and Goodwin, Ramadge, and Caines (1981).

6.2 Hybrid Analytical/Simulation Models

Using analytical models to assist in optimization is discussed in Hanssman, Dinif, Fisher, and Ramer (1980). An overview of hybrid models is given in Shantikumar and Sargent (1982).

7. CONCLUDING REMARKS

The investigation of optimization techniques for simulation is still in its infancy. During the past seven years research on perturbation analysis and likelihood ratios have shown that derivatives of performance measures may be estimated for many simulations without re-running the simulation. The integration of these derivative estimators with stochastic approximation or non-linear programming techniques has not been adequately explored for unconstrained optimization. Using single run optimization, and thereby reducing the computations needed to overcome initial transients, is promising but needs better theoretical results to understand the dynamics and efficiencies of the algorithms. Constrained optimization is also largely untouched - although using variants of Karmarkar's algorithm should provide interesting research.

Optimization for simulation, to date, remains an art, not a science.

REFERENCES

Abdelhamid, S. N. (1973). Transformations of Observations in Stochastic Approximation. *The Annals of Statistics* **1**, 1158-1174.

Anbar, D. (1973). On Optimal Estimation Methods using Stochastic Approximation Procedures. *The Annals of Statistics* **1**, 1175-1184.

Avriel, M. (1975). *Non-Linear Programming: Analysis and Methods*, Prentice-Hall, Englewood Cliffs, New Jersey.

Azadivar, F. and Talavage, J. (1980). Optimization of Stochastic Simulation Models, *Mathematics and Computation in Simulation* **XXII**, 231-241.

Barton, R. R. (1984). Minimization Algorithms for Functions with Random Noise. *American Journal of Mathematical and Management Sciences* **4**, 109-138.

Becker, A., Kumar, P. R., and Wei, C. E. (1983). Adaptive Control the Stochastic Approximation Algorithm. University of Maryland - Baltimore County, Department of Mathematics, Report #83-8.

Biles, W. E. (1977). Strategies for Optimization of Multiple-Response Simulation Models, *Proceedings of the 1977 Winter Simulation Conference.*

Deming, S. N. (1977). Optimization of Experimental Parameters in Chemical Analysis, *Validation of Measurement Process*. 162-175.

Farrell, W., McCall, C. H., and Russell, E. C. (1975). Optimization Techniques for Computerized Simulation Models. CACI, Inc. Technical Report 1200-4-75.

Glynn, P. E. (1986a). Stochastic Approximation for Monte-Carlo Optimization. *Proceedings of the 1986 Winter Simulation Conference*, 356-365.

Glynn, P. E. (1986b). Sensitivity Analysis for Stationary Probabilities of a Markov Chain. *Proceedings of the 4th Army Conference on Applied Mathematics and Computers.*

Glynn, P. E. (1986c). Optimization of Stochastic Systems. *Proceedings of the 1986 Winter Simulation Conference.*

Gong, W. B. and Ho, Y. C. (1987). Smoothed Perturbation Analysis of Discrete Event Dynamic Systems. To appear in *IEEE Transactions on Automatic Control.*

Goodwin, G. C., Ramadge, P. J., and Caines, P. E. (1981). Discrete Time Stochastic Adaptive Control. *Siam Journal of Control and Optimization*, **19**, 829-853.

Goodwin, Ramadge, and Caines (1982). Corrigendum: Discrete Time Stochastic Adaptive Control. *Siam Journal of Control and Optimization*, **20**, 893.

Hanssman, Dinif, Fisher, and Ramer (1980). Analytical Search Model for Optimum Seeking in Simulations. *OR Spektrum* **2.**

Heidelberger, P. (1986). Convergence Properties of Infinitesimal Perturbation Analysis Estimates. Manuscript, IBM Thomas J. Watson Research Center, Yorktown Heights, NY.

Karmarkar, N. (1984). A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica* **4**, 373-395.

Kesten, H. (1958). Accelerated Stochastic Approximation. *Annals of Mathematical Statistics* **29**, 41-59.

Kiefer, J. and Wolfowitz (1952). Stochastic Estimation of the Maximum of a Regression Function, *Annals of Mathematical Statistics* **23**, 462-466.

Lai, T. L., and Robbins, H. (1979). Adaptive Design and Stochastic Approximation. *The Annals of Statistics* **7**, 1196-1221.

Montgomery, D. C., and Bettencourt, V. M. (1977). Multiple Response Surface Methods in Computer Simulation. *Simulation*, 113-121.

Reiman, M. I., and Weiss, A. (1986). Sensitivity Analysis for Simulations via Likelihood Ratios. *Operations Research*, to appear.

Reiman, M. I., and Nguyen, P. (1987). Variance Reduction for Sensitivity Estimates Obtained from Regenerative Simulation. AT&T Manuscript.

Robbins, H., and Monroe, S (1951). On a Stochastic Approximation Technique, *Annals of Mathematical Statistics* **22**, 400-407.

Rubinstein, R. Y. (1983). Smoothed Functionals in Stochastic Optimization, *Mathematics of Operations Research* **8**, 26-33.

Rubinstein, R. Y., and Kreimer, J. (1982). Stochastic Optimization via Stochastic Approximation. Operations Research Statistics and Economics Mimeograph, Series N.350, Technion, Haifa, Israel.

Rubinstein, R. Y. (1987). Sensitivity Analysis of Computer Simulation Models via the Efficient Score. George Washington University, Department of Operations Research.

Rubinstein, R. Y. (1987b). The "What If" Problem in Simulation Analysis. GWU/IMSE/Serial T-514/87. The George Washington University, School of Engineering and Applied Sciences.

Rustagi, J. S. (1978). Optimization in Statistics: An Overview. *Communications of Statistics - Simulation and Computation* **B7**, 303-307.

Shanthikumar, J. G., and Sargent R. G. (1982). A Unifying View of Hybrid Simulation/Analytical Models and Modeling. Syracuse University, Department of IE/OR, Working Paper #82-003.

Segreti, A. C., Carter, W. H., and Wampler, G. L. (1979). Monte Carlo Evaluation of Several Sequential Optimization Techniques when the Response is Time to an Event. *Journal of Statistical Computation and Simulation* **9**, 289-301.

Segreti, Carter, and Wampler (1981). A Monte Carlo Evaluation of the Robustness of Several Sequential Optimization Techniques when the Response is Time to an Event. *Journal of Statistical Computation and Simulation* **12**, 209-216.

Smith, D. E. (1973). An Empirical Investigation of Optimum-Seeking in the Computer Simulation Situation. *Operations Research* **21**, 475-497.

Suri, R. (1983). Infinitesimal Perturbation Analysis of Discrete Event Dynamic Systems: A General Theory. *Proceedings of the 22nd IEEE Conference on Decision and Control.*

Suri, R. and Leung, Y. T. (1987). Single Run Optimization of Discrete Event Simulations - An Empirical Study Using the M/M/1 Queue. Working paper, University of Wisconsin - Madison.

Vanderbei, R., Meketon, M., and Freedman, B. (1986). A Modification of Karmarkar's Linear Programming Algorithm. *Algorithmica* **1**, 395-407.

AUTHOR BIOGRAPHY

Marc Meketon is with the Network Design and Optimization department of AT&T Bell Laboratories. He is currently involved in the implementation of Karmarkar based linear programming algorithms. His research has been in optimization, statistical analysis of simulation output processes, and network design. Prior to joining AT&T in 1982, Marc taught at Syracuse University, the University of Rochester, and worked at the Xerox Corporation. Marc obtained a PhD in Operations Research from Cornell University in 1980, and a Bachelors of Science from Villanova University in 1976. He is a member of ORSA, TIMS, and ASA.