

Simulation System Design

Hossein Arsham¹, Miroljub Kljajić²

¹University of Baltimore, Baltimore, MD, 21201, USA, harsham@ubalt.edu

²University of Maribor, Faculty of Organizational Sciences, Kidričeva cesta 55a, SI-4000 Kranj, Slovenia, miroljub.kljajic@fov.uni-mb.si

The use of simulation as a tool to design complex stochastic systems is often inhibited by cost. Extensive computer processing is needed to find a design parameter value given a desired target for the performance measure of a given system. The designer simulates the process numerically and obtains an approximation for that same output. The goal is to match the numerical and experimental results as closely as possible by varying the values of input parameters in the numerical simulation.

The most obvious difficulty in solving the design problem is that one cannot simply calculate a straightforward solution and be done. Since the output has to be matched by varying the input, an iterative method of solution is implied. This paper proposes a "stochastic approximation" algorithm to estimate the necessary controllable input parameters within a desired accuracy given a target value for the performance function. The proposed solution algorithm is based on Newton's methods using a single-run simulation approach to estimate the needed derivative. The proposed approach may be viewed as an optimization scheme, where a loss function must be minimized. The solution algorithm properties and the validity of the estimates are examined by applying it to some reliability and queueing systems with known analytical solutions.

Key words: system design and simulation, local response surface, goal seeking problem, parameter setting design, gradient estimation, discrete-event systems

Načrtovanje in planiranje z metodo simulacije

Uporaba simulacije kot orodja za načrtovanje kompleksnih stohastičnih sistemov je pogosto časovno zahtevna naloga. Potreben je izdaten računalniški čas da se najde vrednost vhodnih parametrov ki ustrezajo željenim performansam sistema. Načrtovalec simulira proces numerično za izbrane vhodne parametre da dobije oceno želene vrednosti izhoda. Cilj je da dobimo kar se da slične vrednosti experimentalnih in simulacijskih rezultatov z variranjem vhodnih parametrov simulacijskega modela. Pproblem je da ne obstaja enostaven način računanja da direkto dobimo zahtevanno rešitev problema. Ker izhod (rešitev) mora odgovarati enoj od možnih vrednosti vhodnih parametrov metoda reševanja je nujno iterativna kar zahteva veliko računalniškega časa. V tem članku predlagava postopek "stohastičnega približka" za oceno potrebnih controlabinih vhodnih parametrov za določitev željene vrednosti sistema v mejah predpisane zanesljivosti. Predlagani algoritam temelji na Newtonovi metodi, kjer spomočjo (enega) simulacijskega teka ocenimo prvi odvod potreban za optimizacijo kriterijske funkcije. Predlagani postopek lahko razumemo kot optimizacijsko shemo, kjer funkcijo izgube je treba minimizirati. Predlagani postopek je preizkušen in ovrednoten na nekaj primerih zanesljivosti in sistemov strežbe z znanimi analitičnimi rešitvami.

Ključne besede: načrtovanje in simulacija, metoda lokalnega odziva, ciljno optimiziranje, parametrska optimizacija, gradientne ocene, dogodkovna simulacija

1 Introduction

Simulation continues to be the primary method by which engineers obtain information about analysis of complex stochastic systems, such as production assembly lines, flexible manufacturing systems, and reliability systems. Almost all stochastic system performance evaluation can be formulated as an estimation of an expected value. Consider a system with continuous parameter $v \in V \subseteq \mathbb{R}$, where V is an open interval. Let

$$J(v) = E_{Y|v} [Z(Y)], \quad (1)$$

be the steady-state expected performance measure, where Y

is a random vector with known probability density function (pdf), $f(y;v)$ depends on v , and Z is the performance measure. For example, in a reliability system, $J(v)$ might be the mean time to failure; Z is the lifetime of a system; Y is the lifetime of the components; and v might be the components' mean lifetimes. In general, v is the shape or scale parameter of the underlying pdf. Another example could be a queueing system, where Y is the sequence of a two-dimensional vector of inter-arrivals and service times, Z is the delay in the system, and v is the arrival rate.

Before proceeding further, we distinguish between discrete event static systems (DESS) and discrete event dynamic systems (DEDS). Dynamic systems evolve over time; static systems do not evolve over time. Examples of

dynamic systems are the queuing systems; examples of static systems are reliability systems. Note that while in DESS, Y is a multidimensional vector; in DEDS, Y represents a stochastic process. This paper deals with perturbation analysis of both DESS and DEDS.

In systems analysis, we resort to simulation when Z is either unknown or is too complicated to calculate analytically. Simulation is needed to estimate $J(v)$ for most DESS and DEDS. The principal strength of simulation is its flexibility as a systems analysis tool for highly complex systems.

In discrete event systems, Monte Carlo simulation is usually needed to estimate $J(v)$ for a given value $v = v_0$. By the law of large numbers

$$\hat{J}(v_0) = 1/n \sum_{i=1}^n Z(y_i), \quad (2)$$

converges to the true value, where y_i , $i = 1, 2, \dots, n$ are independent, identically distributed random vector realizations of Y from $f(y; v_0)$, and n is the number of independent replications. The numerical result based on (2) is only a point estimate for $J(v)$ at $v = v_0$. The numerical result based on (2) is a solution to a **system analysis**: Given the underlying pdf with a particular parameter value v_0 , estimate the output function $J(v_0)$. The direct problem is widely used in stochastic systems analysis. Now we pose the **system design problem**: Given a target output value of the system and a parameterized pdf family, find an input value for the parameter which generates such an output. The solution to the design problem has potential application in stochastic systems analysis and design. Mathematical formulation of the design problem is as follow:

Given τ , find $v \in V \subseteq R$ subject to $J(v) = \tau$, where

$$J(v) = E_{Y|v} [Z(Y)] = \int Z(y) f(y; v) dy, \quad (3)$$

$Z: R^m \rightarrow R$ is a system performance measure

$Y \in R^m$ is a random vector (or a truncated stochastic process) with pdf $f(y; v)$

The design problem is essentially backwards. The output is given, but the input must be determined. This is easiest to appreciate when a designer wants to match experimental data in order to obtain some basic parameters. The designer simulates the process numerically and obtains an approximation for that same output. The goal is to match the numerical and experimental results as closely as possible by varying the values of input parameters in the numerical simulation. Analyzing this, clearly, the output is there, and it is the input quantity that needs to be determined. The most obvious difficulty in solving the design problem is that one cannot simply calculate a straightforward solution and be done. Since the output must be set by varying the input, an iterative method of solution is implied. Our approach may be viewed as an optimization scheme where a loss function must be minimized. Therefore, the process of solving a design problem often comes down to finding the best method of minimizing the loss function. The key part

of optimization is to compute the derivative of the output with respect to an input parameter.

There are strong motivations for both problems. In the case when v is any controllable or uncontrollable parameter, the designer is interested in estimating $J(v)$ for a *small* change in $v = v_0$ to $v = v_0 + \delta v_0$. This is the so-called what-if problem which is a direct problem. However, when v is a controllable input the decision maker may be interested in the goal-seeking problem; i.e., "What perturbation of the input parameter will achieve a desired change in the output value?" Another application of the design problem is where we may want to adapt a model to satisfy a new constraint with stochastic function. While the what-if problem has been extensively studied, the goal-seeking simulation problem is relatively new. Design interpolation based on regression models provides an indirect approach to solve the design problem. In this treatment, one simulates the system for many different values of $v = v_0$ and then one approximates the response surface function $J(v)$, see e. g. (Kleijnen, 1979). Finally, the fitted function is used to interpolate to obtain the unknown parameter v . Since the shape of $J(v)$ function is unknown, this approach is tedious, time-consuming and costly. Moreover, in random environments, the fitted model might have unstable estimates for the coefficients. The only information available about $J(v)$ is general in nature, for example, continuity, differentiability, invertability, and so on.

The simulation models based on (2), although simpler than the real-world system, are still a very complex way of relating input (v) to output $J(v)$. Sometimes a simpler analytic model may be used as an auxiliary to the simulation model. This auxiliary model is often referred to as a local response surface model (known also as a metamodel (Friedman, 1996). Local response surface models may have different goals: model simplification and interpretation (Yu & Popplewell, 1994), optimization (Arsham, 1996), what-if analysis (Arsham, 1996a), and generalization to models of the same type. The following polynomial model can be used as an auxiliary model.

$$J(v) = J(v_0) + \delta v J'(v_0) + (\delta v)^2 J''(v_0) / 2 + \dots, \quad (4)$$

where $\delta v = v - v_0$ and the primes denote derivatives. This local response surface model approximates $J(v)$ for small δv . To estimate $J(v)$ in the neighborhood of v_0 by a linear function, we need to estimate the nominal $J(v_0)$ based on (2) and its first derivative. Traditionally, this derivative is estimated by crude Monte Carlo; i.e., finite difference which requires rerunning the simulation model. Methods which yield enhanced efficiency and accuracy in estimating, at little additional cost, are of great value.

There are few ways to obtain efficiently the derivatives of the output with respect to an input parameter (Arsham, 1998). The most straightforward method is the Score Function (SF). The SF approach (Arsham et al., 1989) is the major method for estimating the performance measure and its derivative, while observing *only a single* sample path from the underlying system (Rubinstein & Melamed, 1998). The basic idea of SF is that the derivative of the performance function, $J'(v)$, is expressed as expectation with respect to the *same* distribution as the performance measure itself.

This paper treats the design problem as a simulation (as opposed to regression) problem. By this approach, we are able to apply variance reduction techniques (VRT) used in the direct problem. Specifically, we embed a stochastic version of Newton's method in a recursive algorithm to solve the stochastic equation $J(v) = J$ for v , given J at a nominal value v_0 .

The explicit use of a linear local response surface model is the target parameter design: Given a desired value $J = J(v)$, find the prerequisite input parameter v .

Most engineering design methods essentially involve a framework for arriving at a target value for product, process, and service attributes through a set of experiments which include Monte Carlo experiments. To solve the product design problem, we will restrict our model to the first order expansion. For a given $J(v)$ the estimated δv using (4) is

$$\hat{\delta v} = [J(v) - \hat{J}(v_0)] / \hat{J}'(v_0), \tag{5}$$

provided that the denominator in (5) does not vanish for all v_0 in interval V .

The remainder of this article is divided into eight sections. The next section contains the construction of a polynomial local response surface model using estimated derivatives of $J(v)$. Section 3 deals with the target setting problem in design of a system. This is followed by construction of an accuracy measure. Section 5 develops an iterative solution algorithm for the parameter selection problem. Sections 6 and 7 illustrate the proposed method for reliability and queueing systems, respectively. Finally, Section 8 provides some concluding remarks and ideas for further research and extensions.

2 Polynomial Local Response Surface Model Construction by Single-Run Simulation

Simulation models, although simpler than real-world systems, are still a very complex way of relating input parameters (v) to performance measures $J(v)$. Sometimes a simple analytical model may be used as an auxiliary to the simulation model. This auxiliary local response surface model is often referred to as a metamodel (Friedman, 1996). In this treatment, we have to simulate the system for some different values of (v) and then use a "goodness-of-fit" regression. We fit a response surface to these data (Kleijnen, 1979). Clearly, coupling the simulation model with the Score Function method enhances the efficiency of local response surface model construction. A local response surface model can also be constructed by using sensitivities in a Taylor expansion of $J(v)$ in the neighborhood of $v = v_0$. The resulting local response surface model can be used for characterization (such as increasing/decreasing, and convexity/concavity) of the response surface.

Let

$$J(v) = E_{Y|v} [Z(Y)] = \int Z(y) f(y; v) dy, \tag{6}$$

Z is a system performance measure
 $Y \in R^m$ is a random vector (or a truncated stochastic

process) with pdf $f(y; v)$
 be the steady state performance measure, then

$$J'(v) = \int [Z(y) \cdot f'(y; v)] dy, \tag{7}$$

where the prime (') denotes the derivative with respect to v . Note that despite the fact that y depends on v , only the function $Z \cdot f$ is subject to differentiation with respect to v . From (7) it follows that

$$J'(v) = \int Z(y) f'(y; v) dy = E_{Y|v} [Z(Y) S], \tag{8}$$

where $S = f'(y; v) / f(y; v)$ is the Score Function. Differentiation is with respect to v . This is subject to the assumptions that the differentiation and the integration operators are interchangeable, $f'(y; v)$ exists, and $f(y; v)$ is positive for all $v \in V$, where V is an open interval. A necessary and sufficient condition for the interchangeability used above is that there must be no discontinuity in the distribution with position depending on the parameter v (Arsham, 1996a). Similarly, the second derivative is

$$J''(v) = \int [Z(Y) S'(y; v) + Z(Y) S f'(y; v)] dy = E_{Y|v} [Z(Y) H] \tag{9}$$

where

$$H = S' + S^2. \tag{10}$$

In the multidimensional case, the gradient and Hessian of $J(v)$ could be obtained in a straightforward manner by generalizing these results (Arsham, 1998). The estimator for the first and second derivatives based on (8) and (9) are given by:

$$\hat{J}'(v_0) = \frac{1}{n} \sum_{i=1}^n Z(y_i) \cdot S(y_i; v_0) \tag{11}$$

$$\hat{J}''(v_0) = \frac{1}{n} \sum_{i=1}^n Z(y_i) \cdot H(y_i; v_0) \tag{12}$$

where

$$S(y_i; v_0) = f'(y_i; v_0) / f(y_i; v_0) \tag{13}$$

and

$$H(y_i; v_0) = f''(y_i; v_0) / f(y_i; v_0). \tag{14}$$

Notice that both (11) and (12) estimators are evaluated at $v = v_0$, and y_i 's are the same n independent replications used in (2) for estimating the nominal performance $J(v_0)$; therefore they are quite efficient in terms of CPU cost. Estimates obtained by using (11) and (12) are unbiased, consistent, and they converge to the true values in the sense of the mean squared error (Arsham, 1998). The estimated gradient can also be used in solving optimization problems by simulation using the stochastic version of the classical nonlinear programming algorithms (Arsham, 1996). Other applications of sensitivity information include stability analysis (Arsham, 1996a).

3 Design-setting Problem

Most engineering system designs such as product, process, and service design, involve a framework for arriving at a target value for a set of experiments, which may include Monte Carlo experiments. A random quality loss function $L(Z_i)$ for a given system τ can be expanded in the neighborhood of the target value τ as follows:

$$L(Z_i) = L(\tau) + (Z_i - \tau)L'(\tau) + (Z_i - \tau)^2 L''(\tau)/2 + \dots \quad (15)$$

It can be shown that $L(Z_i)$ converges in *mean squared error* if $*Z_i - \tau* < 1$ and derivatives are finite. Since the *optimal* loss is zero at τ , equation (15) reduces to the following quadratic approximation

$$L(Z_i) = K (Z_i - \tau)^2 \quad (16)$$

In (16), K is some constant which can be determined in terms of the customer's tolerance limit $(\tau - \delta)$, which suggests that the product performs unsatisfactorily when Z_i slips below this limit. Given that the cost to customer is A dollars, then $K = A/\delta^2$. Without loss of generality, for simplicity let $K=1$.

The goal of parameter design is to choose the setting of the design parameter v that minimizes the average loss (the risk function). The risk function $R(\tau)$ is the expected value of the loss function, which can be shown as:

$$R(\tau) = E \{L(Z_i)\} = (J - \tau)^2 + \text{Var} (Z_i), \quad (17)$$

This risk function measures the average loss due to a product performance which is proportional to the square of the deviation from the target value τ .

The non-adjustable variational noise; i.e.;

$$\text{Var} (Z_i | v) = \text{Var} (Z_i), \quad (18)$$

is a measure of variation among products. However, the role of product design is to reduce the $(J - \tau)^2$ part of risk, which is our interest in this paper. Note that all estimates involved in computing δv based on (5); i.e., in

$$\hat{\delta v} = [J(v) - J(v_0)] / J'(v_0) \quad (19)$$

are computed *simultaneously* from a *single-run simulation* of the nominal system ($v = v_0$). This was achieved by transforming all probability space to the nominal one. Note that to estimate the derivative we do not need to rerun the simulation. To estimate the derivatives adds only moderate computational cost to the base simulation.

4 Accuracy of the Estimate

In the design problem, input parameter is random, while the output is fixed and given as a target value. Upon estimating the input parameter, we must provide a measure, such as a confidence interval, to reflect the precision of the estimate. To construct a confidence interval for δv using the estimator (19), let

$$A_i = J(v) - Z(y_i; v_0), \quad (20)$$

$$B_i = Z(y_i; v_0) S(y_i; v_0) \quad (21)$$

and denote

$$A = \sum A_i / n, \text{ and } B = \sum B_i / n, \quad (22)$$

then

$$S^2 = S_{11}^2 - 2 \wedge v S_{12} + (\wedge v)^2 S_{22} \quad (23)$$

where

$$S_{11} = \sum (A_i - A)^2 / (n-1), S_{22} = \sum (B_i - B)^2 / (n-1), \quad (24)$$

and

$$S_{12} = \sum (A_i - A)(B_i - B) / (n-1), \quad (25)$$

An exact 100 $(1 - \alpha)$ % confidence interval for δv is given by

$$P \left[n^{1/2} \frac{|\delta v - v|}{S/B} \leq t_{n-1, 1-\alpha/2} \right] \geq 1-\alpha, \quad (26)$$

where $t_{n-1, 1-\alpha/2}$ is the 100 $(1 - \alpha / 2)$ percentile of Student's t distribution with $(n-1)$ degrees of freedom (Kleijnen & Van Groenendaal, 1992).

5 A Recursive Solution Algorithm

The solution to the design problem is a solution of the stochastic equation $J(v) = J$, which we assume lies in some bounded open interval V . The problem is to solve this stochastic equation by a suitable experimental design to ensure convergence as δv approaches zero. The following algorithm involves placing experiment $j+1$ according to the outcome of experiment j immediately preceding it. That is,

$$v_{j+1} = v_j + d_j [\tau - \hat{J}(v_j)] / \hat{J}'(v_j), \quad (27)$$

where d_j is any sequence of positive numbers satisfying the following conditions:

$$\sum_{j=1}^{\infty} d_j = \infty, \quad (28)$$

and

$$\sum_{j=1}^{\infty} d_j^2 < \infty, \quad (29)$$

The first condition is a necessary condition for the convergence δv to approach zero, while the second condition asymptotically dampens the effect of the simulation random errors. These conditions are satisfied, for example, by the harmonic sequence $d_j = 1/j$. With this choice, the rate of reduction of d_j is very high initially but may reduce to very

small steps as we approach the root. Therefore, a better choice is, for example

$$d_j = 9 / (9 + j).$$

Since the adjustments are made in proportion to the recent value, we must be sure that the results remain finite. This requires that $J'(v)$ does not vanish for $v \in V$, where V is an open interval. To prevent excessive over-correction, we assume further that the solution lies in some finite interval V . Under these not unreasonable conditions, this algorithm will converge in mean square; moreover, it is an almost sure convergence. For some generalizations and studies concerning speed of convergence and acceleration techniques, see (Dippon & Renz, 1997). Finally, as in Newton's root-finding method, it is impossible to assert that the method converges for just any initial $v = v_0$, even though $J'(v)$ may satisfy the Lipschitz condition over V . Indeed, if the initial value v_0 is sufficiently close to the solution, which is usually the case, then this algorithm requires only a few iterations to obtain a solution with very high accuracy.

ALGORITHM

Step 0. INPUTS

- τ = Desired output
- j = Iteration number
- v_j = Controllable input parameter v
- n = Sample size
- U = Desired upper limit for absolute increment $u = \frac{v_{j+1} - v_j}{j}$
- α = A desired significance level

Step 1. INITIALIZATION

- Set $j=1$
- Set $v_j = v_0$

Step 2. ESTIMATIONS

- $J(v_j)$ using (2)
- $J'(v_j)$ using (9)

Step 3. COMPUTATIONS

- $u = 9[\tau - J(v_j)] / [(9+j) J'(v_j)]$
- If $|u| < U$
- Construct $100(1 - \alpha)\%$ confidence interval for v using (20)
- Stop.
- Otherwise
- set $v_{j+1} = v_j + u$ and $j \rightarrow j+1$

Step 4. RESET: Reset the seeds of random number generators to their initial values. Go to step 2.

Note that, by resetting the seeds to their initial values, we are using the Common Random Variate approach as a variance reduction technique.

6 Design of a Reliability System

For most complex reliability systems, the performance measures such as mean time to failure (MTTF) are not available in analytical form. We resort to Monte Carlo Simulation (MCS) to estimate MTTF function from a family

of single-parameter density functions of the components life with specific value for the parameter. The purpose of this section is to solve the design problem which deals with the calculation of the components' life parameters (such as MTTF) of a homogeneous subsystem, given a desired target MTTF for the system. A stochastic approximation algorithm is used to estimate the necessary controllable input parameter within a desired range of accuracy. The potential effectiveness is demonstrated by simulating a reliability system with a known analytical solution.

Consider a coherent reliability sub-system consists of 4 homogeneous elements; i.e., manufactured by an identical process, components having independent random lifetimes $Y_1, Y_2, Y_3,$ and Y_4 , which are distributed exponentially with rates $v = v_0 = 0.5$.

The first 2, and the last two elements are in series, while these two series each with two components are in parallel. The system lifetime is $Z(Y_1, Y_2, Y_3, Y_4; v_0) = \max[\min(Y_3, Y_4), \min(Y_1, Y_2)]$. It is readily seen that the theoretical expected lifetime of this system is $J(v_0) = 3/(4 v_0)$, (Barlow & Proschan, (1975). Now we apply our results to compute a necessary value for v to obtain a particular value for $J(v)$, say $J(v) = 2$. For this reliability system, the underlying probability density function is:

$$f(y;v) = v^4 \exp(-v \sum y_i), i = 1, 2, 3, 4. \tag{30}$$

The Score Function is

$$S(y) = f'(y; v) / f(y; v) = 4/v - \sum y_i, i = 1, 2, 3, 4. \tag{31}$$

$$H(y) = f''(y; v) / f(y; v) = [v^2 (\sum y_i)^2 - 8v (\sum y_i) + 12] / v^2, i = 1, 2, 3, 4. \tag{32}$$

The estimated average lifetime and its derivative for the nominal system ($v = v_0 = 0.5$) based on (2) and (9) are

$$J(v_0) = \sum \max[\min(Y_{3j}, Y_{4j}), \min(Y_{1j}, Y_{2j})] / n, \tag{33}$$

and

$$J'(v_0) = \sum \max[\min(Y_{3j}, Y_{4j}), \min(Y_{1j}, Y_{2j})] \cdot S(Y_{ij}) / n, \tag{34}$$

$$J''(v_0) = \sum \max[\min(Y_{3j}, Y_{4j}), \min(Y_{1j}, Y_{2j})] \cdot H(Y_{ij}) / n, \tag{35}$$

respectively where Y_{ij} is the j^{th} observation for the i^{th} component ($i = 1, 2, 3, 4$). We have performed a Monte Carlo experiment for this system by generating $n = 10000$ independent replications using SIMSCRIPT II.5 random number streams 1 through 4 to generate exponential variates Y_1, Y_2, Y_3, Y_4 , respectively, on a VAX system. The estimated performance is $J(0.5) = 1.5024$, with a standard error of 0.0348. The first and second derivative estimates are -3.0933 and 12.1177 with standard errors of 0.1126 and 1.3321, respectively.

The response surface approximation in the neighborhood $v = 0.5$ is:

$$J(v) = 1.5024 + (v - 0.5)(-3.0933) + (v - 0.5)^2 (12.1177)/2 + \dots + 6.0589v^2 - 9.1522v + 4.5638 \tag{36}$$

A numerical comparison based on direct simulation and local response surface model (36) is given in Table 1. Notice that the largest error in Table 1 is 0.33% which could be reduced by either more accurate estimates of the derivatives and/or using a higher order Taylor expansion. A comparison of the errors indicates that the errors are smaller and more stable in the direction of increasing v . This behavior is partly due to the fact that lifetimes are exponentially distributed with variance $1/v$. Therefore, increasing v causes less variance than the nominal system (with $v = 0.50$).

TABLE 1: A second order polynomial local response surface model and direct simulation

| v | Analytic | Simulation | Metamodel | Abs.error(%) |
|-------------|---------------|---------------|---------------|--------------|
| 0.40 | 1.8750 | 1.8780 | 1.8723 | 0.14 |
| 0.42 | 1.7857 | 1.7885 | 1.7887 | 0.17 |
| 0.44 | 1.7045 | 1.7072 | 1.7098 | 0.31 |
| 0.46 | 1.6304 | 1.6330 | 1.6359 | 0.33 |
| 0.48 | 1.5625 | 1.5650 | 1.5667 | 0.27 |
| 0.50 | 1.5000 | 1.5024 | 1.5024 | 0.16 |
| 0.52 | 1.4423 | 1.4446 | 1.4430 | 0.05 |
| 0.54 | 1.3889 | 1.3911 | 1.3884 | 0.04 |
| 0.56 | 1.3393 | 1.3414 | 1.3386 | 0.05 |
| 0.58 | 1.2931 | 1.2951 | 1.2937 | 0.05 |
| 0.60 | 1.2500 | 1.2520 | 1.2537 | 0.29 |

Now assume that the manufacturer wants to improve the average lifetime of the system to $J(v) = \tau = 2$. To achieve this goal, we have set $v_0 = 0.5$ and $U = 0.0001$ in the proposed algorithm. The numerical results are tabulated in Table 2.

TABLE 2: Iterative decision parameter estimate for the reliability system

| (1) Iteration number j | (2) Fixed input v_j | (3) Estimated MTTF | (4) Estimated derivative | (5) Change in v_j | (6) New input parameter v_{j+1} |
|--------------------------|-----------------------|--------------------|--------------------------|---------------------|-----------------------------------|
| 1 | 0.5000 | 1.5024 | -2.9598 | -0.1513 | 0.3487 |
| 2 | 0.3487 | 2.1544 | -6.0862 | -0.0208 | 0.3694 |
| 3 | 0.3694 | 2.0333 | -5.4217 | +0.0046 | 0.3740 |
| 4 | 0.3740 | 2.0083 | -5.2888 | +0.0011 | 0.3751 |
| 5 | 0.3751 | 2.0025 | -5.2583 | +0.0003 | 0.3754 |
| 6 | 0.3754 | 2.0009 | -5.2498 | +0.0001 | 0.3755 |
| 7 | 0.3755 | 2.0003 | -5.2471 | +0.0000 | 0.3756* |

The estimated input parameter to achieve the output $J(v) = \tau = 2$ is 0.3756. A 90% confidence interval based on this estimate using (20) is:

$$P[0.3739 \leq v \leq 0.3773] \geq 0.90. \tag{37}$$

Comparing the theoretical value $v_0 = 0.3750$, obtained from $J(v) = 3/4v_0 = 2$, with our computational value suggests that the results based on the proposed algorithm are quite satisfactory. In fact, running this system with $v = 0.3756$, and $n = 10000$, we obtained an estimated MTTF of $J(v) = 2.0000$. Hence the discrepancy in the estimated input parameter by this algorithm must be considered as a pure random error which can be reduced by increasing n . The metamodel (36) could also be applied to $J(v) = 2$ to estimate the desirable v . Solving the resulting quadratic equation, the relevant root is $v = 0.3725$. This result is an inferior estimate for v compared with the iterative method, although the accuracy of the latter comes with greater computational cost.

7 Design of a Service System

This section presents implementation details and some statistical results on the efficiency of the proposed technique for a discrete event dynamic system. To evaluate the proposed single-run technique to solve the design problem, we have chosen to implement it on an M/G/1 queueing system with a *known* analytical solution. Consider, a single-server, first-come-first-served, Poisson input queue with arrival rate of 1 customer per unit of time. The server works according to a Gamma density

$$f(y;v) = y e^{-y/v} / v^2, v > 0, y \geq 0. \tag{38}$$

The analytic solution for the expected *steady-state* waiting time as a performance measure, in this system is:

$$J(v) = \rho + (\rho^2 + \sigma^2)/[2(1-\rho)] \tag{39}$$

which is obtained by using the Pollaczek-Khintchin formula (Gross & Harris, 1998), where $\sigma^2 = \text{Var}(y) = 2v^2$ and $\rho = \text{traffic intensity} = 1/\text{service rate} = 2v$. If we set the nominal value $v = 0.25$ for the nominal system, then we have $\sigma^2 = 0.125$ and $\rho = 0.5$ resulting in $J(0.25) = 0.875$.

To estimate $J'(v)$ for the nominal system, we will use the method of Batch Means. Other methods, such as Independent Replications or Regenerative Method could also be used.

Batch Means is a method of estimating the steady-state characteristic from a single-run simulation. The single run is partitioned into equal size batches large enough for estimates obtained from different batches to be approximately independent. In the method of Batch Means; it is important to ensure that the bias due to initial conditions is removed to achieve at least a covariance stationary waiting time process. An obvious remedy is to run the simulation for a period (say R customers) large enough to remove the effect of the initial bias. During this warm-up period, no attempt is made to record the output of the simulation. The results are thrown away. At the end of this warm-up period, the waiting time of customers are collected for analysis. The practical question is "How long should the warm-up period be?" Abate and Whitt (Abate & Whitt, 1987) provided a

relatively simple and nice expression for the time required (t_p) for an M/M/1/4 queue system (with traffic intensity ρ) starting at the origin (empty) to reach and remain within 100(1-p)% of the steady-state limit as follows:

$$t_p(\rho) = 2C(\rho) \text{Ln} \{1/[(1-p)(1+2C(\rho))]\}/(1-\rho)^2 \quad (40)$$

where

$$C(\rho) = [2 + \rho + (\rho^2 + 4\rho)^{1/2}] / 4. \quad (41)$$

Some notions of $t_p(\rho)$ as a function of r and p , are given in Table 3.

TABLE 3: Time (t_p) required for an M/M/1 queue to reach and remain with 100(1-p)% limits of the steady-state value

| Traffic Intensity ρ | 100(1-p)% | | | |
|--------------------------|-----------|--------|--------|--------|
| | 95.0 | 99.0 | 99.9 | 99.99 |
| 0.10 | 3.61 | 6.33 | 10.23 | 14.12 |
| 0.20 | 5.01 | 8.93 | 14.53 | 20.14 |
| 0.30 | 7.00 | 12.64 | 20.71 | 28.79 |
| 0.40 | 10.06 | 18.39 | 30.31 | 42.23 |
| 0.50 | 15.18 | 28.05 | 46.47 | 64.89 |
| 0.60 | 24.70 | 46.13 | 76.79 | 107.45 |
| 0.70 | 45.51 | 85.87 | 143.61 | 201.36 |
| 0.80 | 105.78 | 201.53 | 338.52 | 475.51 |
| 0.90 | 435.74 | 838.10 | 1413.7 | 1989.4 |

Although this result is developed for M/M/1 queues, it has already been established that it can serve as an approximation for more general; i.e., GI/G/1 queues (Whitt, 1989). To compute the Score Function S, we need the density function of the steady-state process.

Clearly, for computational implementation, we need a truncated (say m-truncated) version of this process. The waiting time of customer t at steady state depends on values of the $(m - 1)$ previous customers interarrival and service times. The dependency order m must be chosen so that the correlation between the waiting time of customer t and $(t-m)$ is negligible. Notice that the order of dependency m is equivalent to the "Batch Size" widely discussed in simulation literature in connection with the method of Batch Means. We have chosen $m = R$ large enough to ensure independency and not too large to create the singularity problem.

Let X_k and Y_k be the interarrival and service times of the k^{th} customer at steady state, $k \geq R+1$. The underlying density function for the j^{th} customer, $j \geq 2R+1$, in batch number i is:

$$f(v) = \sum_{k=j-m+1}^j f(y_k) f(x_k), \quad j = (i+1)R+1, (i+1)R+2, \dots, (i+2)R \quad (42)$$

where

$$f(x_k) = \exp(-x_k)$$

and

$$f(y_k) = [y_k \exp(-y_k / v)] / v^2.$$

The expected waiting time for the nominal system is:

$$J(v) = \sum_{i=1}^n \sum_{j=(i+1)R+1}^{(i+2)R} L_{i,j} / (Rn) \quad (43)$$

where $L_{i,j}$ is the waiting time of the j^{th} customer in the i^{th} batch. The Score Function S is:

$$S_{j,i} = -2m / v + \sum x_{j,k} / v^2 \quad (44)$$

For the nominal system ($v = v_0 = 2$), we have used $n = 500$ independent replications. In each run, we set $k = m = T = 100$. The estimated delay in the system and its derivative based on these simulation parameters are 1.007 and -0.951 with computed variance 0.001 and 0.012, respectively. Clearly, derivative estimators discussed in this paper work much better for terminating models for which only small number of observations are generated.

Consider the system described above. Assume we want to find a value for the controllable input parameter, service rate v , such that $J(v) = J = 0.8$. We have set $v_0 = 2$ and $U = 0.0001$ in the proposed algorithm. The simulation results are contained in Table 4. Our computations are performed on a PC computer using streams 1 and 2 of SIMSCRIPT II.5 to generate the inter-arrival and service times, respectively.

Table 4: Estimated service rate to achieve a desirable steady state average delay in an M/G/1/ ∞ queue.

| Iteration Number | Fixed Input Parameter v_0 | Estimated δv_0 | Updated v_0 |
|------------------|-----------------------------|------------------------|---------------|
| 1 | 2.000 | 0.236 | 2.236 |
| 2 | 2.236 | 0.001 | 2.237 |
| 3 | 2.237 | 0.001 | 2.238 |
| 4 | 2.238 | 0.001 | 2.239 |
| 5 | 2.239 | 0.000 | 2.239 |

The estimated input parameter to achieve the output $J(v) = 0.8$, is $v = 2.239$ with standard error 0.128. A 95% confidence interval for δv at the fifth iteration, based on the usual t-statistic is:

$$P[-0.016 \leq \delta v \leq 0.016] \geq 0.95 \quad (45)$$

A comparison of the analytical value $v = 2.25$, obtained from (39) with our estimated value suggests that the results based on the proposed algorithm are quite satisfactory. In fact, solving the direct problem using the same simulation parameters with $v_0 = 2.239$, the estimated expected waiting time turned out to be 0.800 with variance equal to 0.001. Hence the discrepancy in the estimated input parameter by this algorithm must be considered as a random error which can be reduced by increasing n .

The method of Independent Replication has lower efficiency than the method of Batch Means for the steady-state perturbation analysis. In the Independent Replication method, the output data are collected over a period of length T in a simulation run over a period of length $R + m + T$; (T could be as small as 1). The ratio $T/(R+m+T)$, which is the fraction of CPU time generating useful data, would be very small. Clearly, the method of Batch Means is more efficient.

8 Conclusions

Almost all discrete event systems simulation computation can be formulated as an estimation of an expected value of the system performance measure, which is a function of an input parameter of the underlying probability density function. In the ordinary system simulation, this input parameter must be known in advance to estimate the output of the system. From the designer's point of view, the input parameters can be classified as controllable and uncontrollable (Morris & Bardhan, 1995). The influential controllable input can be recognized by factor screening methods (Ruppert, 1985). In this paper, we considered the design problem: "What must be the perturbation of the current controllable input parameter value to achieve a desired output value?" The approach used in this study was:

- To estimate the derivative of the output function with respect to the input parameter for the nominal system by a single-run, and on-line simulation;
- To use this estimated derivative in a Taylor's expansion of the output function in the neighborhood of the parameter; and finally,
- To use a recursive algorithm based on the Taylor's expansion to estimate the necessary controllable input parameter value within a desired accuracy.

Under some mild and reasonable conditions, the algorithm converges to the desired solution with probability 1. The efficiency of the proposed algorithm in terms of accuracy is tested using an $M/G/1/\infty$ queuing service, as well as a reliability product designs with satisfactory results. The approach may have major implications for simulation modelers and practitioners in terms of time and cost savings. As always, since this experiment was done on these specific numerical examples, one should be careful in making any other generalizations.

In the course of future research:

1. We expect to introduce other efficient variance reduction techniques (VRT). The Common Random Variates as a VRT is already embedded in the algorithm. Notice that since

$$E[S] = E[\ln f]' = \int [\ln f]' f dx = \int f' dx = \left[\int f dx \right]' = 0. \tag{46}$$

We can express the gradient in terms of covariance between Z and S

$$J'(v) = \text{Cov}[Z(Y), S] = E[Z.S] + E[Z].E[S]. \tag{48}$$

and

$$J'(v) = E[Z(Y).S] + \alpha E[S] \tag{49}$$

where α could be the optimal linear control. Note also that (6) can be written as:

$$J'(v) = \int Z(y) f'(y; v) dy = \int Z(y) [f'(y; v) / \varphi(y; v)] \varphi(y; v) dy. \tag{50}$$

The best choice for φ is the one proportional to $Z(y).f'(y; v)$. This minimizes the variance of $J'(v)$; however, this optimal φ depends on the performance function $Z(y)$, which is not known in advance for most cases. One may use the empirical version of $Z(y).f'(y; v)$. We recommend a pilot run to study the effectiveness of these and other variance reduction techniques before implementing them.

2. We expect to extend our methodology to higher order Taylor's expansion. We believe that there is a tradeoff between number of iterations, sample size in each iteration; and the order of Taylor's expansion. Clearly, estimating the second derivative requires a larger sample size n , but a fewer iterations to achieve the same accuracy.
3. We also expect to extend our methodology to the design problems with two or more unknown parameters by considering two or more relevant outputs to ensure uniqueness. By this generalization, we could construct a linear system of stochastic equations to be solved simultaneously by multidimensional versions of the stochastic approximation proposed in (Ruppert, 1985; Wei, 1987) as well as the Newton method in (Polak, 1997; Tyrtshnikov, 1997) using the second order derivatives (e.g., Hessian).
4. The algorithms in this paper are presented in English-like step-by-step format to facilitate implementation in a variety of operating systems and computers, thus improving portability. However, there is a need to develop an expert system that makes the algorithms more practically applicable to stimulation in system design (Clymer, 1995).

Acknowledgment: This work is supported by The National Science Foundation Grant CCR-9505732.

Literature

Abate, J. & Whitt, W. (1987). Transient behavior of $M/M/1$ queue: Starting at origin, *Queueing Systems*, 2: 41-65.

Arsham, H. (1996). Stochastic optimization of discrete event systems simulation, *Microelectronics and Reliability*, 36: 1357-1368.

Arsham, H. (1996a). Performance extrapolation in discrete-event systems simulation, *International Journal of Systems Science*, 27: 863-869.

Arsham, H. (1998). Algorithms for sensitivity information in discrete-event systems simulation, *Simulation Practice and Theory*, 6: 1-22.

- Arsham, H., Feuerverger, A., McLeish D., Kreimer J. & Rubinstein, R. (1989). Sensitivity analysis and the 'what-if' problem in simulation analysis, *Mathematical and Computer Modelling*, **12**: 193-219.
- Barlow, R. & Proschan F. (1975). *Statistical Theory of Reliability and Life Testing Probability Models*, Holt Rinehart and Winston, New York.
- Clymer, J. (1995). System design and evaluation using discrete event simulation with AI, *European Journal of Operational Research*, **84**, 213-225.
- Dippon, J. & Renz J. (1997). Weighted means in stochastic approximation of minima, *SIAM Journal of Control and Optimization*, **35**: 1811-1827.
- Friedman, L. (1996). *The Simulation Metamodel*, Kluwer Academic Publishers, Norwell, Massachusetts.
- Gross, D. & Harris C. (1998). *Fundamentals of Queueing Theory*, John Wiley and Sons, New York.
- Kleijnen, J. & Van Groenendaal, W. (1992). *Simulation: A Statistical Perspective*, Wiley, Chichester, U.K.
- Kleijnen, J. (1979). Regression metamodel for generalizing simulation results, *IEEE Transactions on Systems, Man and Cybernetics, SMC*, **9**: 93-96.
- Morrice, D. & Bardhan, I. (1995). A weighted least squares approach to computer simulation factor screening, *Operations Research*, **43**(5): 792-806.
- Polak, E. (1997). *Optimization: Algorithms and Consistent Approximations*, Springer, New York.
- Rubinstein, R., & Melamed, B. (1998). *Modern Simulation and Modeling*, Wiley, New York.
- Ruppert, D. (1985). A Newton-Raphson version of the multivariate Robbins-Monro procedure, *Annals of Statistics*, **13**: 236-245.
- Tyrtshnikov, E. (1997). *A Brief Introduction to Numerical Analysis*, Birkhauser, Berlin.
- Wei, C. (1987). Multivariate adaptive stochastic approximation, *Annals of Statistics*, **15**: 1115-1130.

- Whitt, W. (1989). Planning queueing simulation, *Management Science*, **35**: 1341-1366.
- Yu, B. & Popplewell, K. (1994). Metamodels in manufacturing: A review, *International Journal of Production Research*, **32**: 787-796.

Hossein Arsham is the Harry Wright Distinguished Professor of Decision Science and Statistics at the Merrick School of Business, University of Baltimore, USA. Education: D.Sc. in Operations Research, M.Sc. in Management Science, B.Sc. in Physics. His teaching, research, and consulting activities are multidisciplinary, interdisciplinary, and transdisciplinary which include: General Optimization, Systems Simulation, and Statistical Data Analysis. His aim in constructing model-based decision support systems is to provide models for the managers which are easy to understand, easy to use, while providing useful managerial information. He has published over 40 articles in recognized world journal. For more information about Dr. Arsham, visit his home page at: <http://home.ubalt.edu/ntsbarsh/index.html>.

Miroljub Kljajić is Professor at the Faculty of Organizational Sciences, University of Maribor in the field of System Theory, Decision Theory and Computer Simulation. He completed his Dipl. Eng., M.Sc, and D.Sc. at the Faculty of Electrical Engineering in Ljubljana. He developed a method of quantitative gait evaluation and a simulation system for decision making support in the business systems. He has been the principal investigator of many national and international modeling and simulation projects. As author and co-author he has published 26 scientific articles recognized in SCI.
