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

VARIANCE REDUCTION FOR SIMULATION PRACTITIONERS

Barry L. Nelson
Department of Industrial and Systems Engineering
The Ohio State University
Columbus, OH 43210 USA

ABSTRACT

A comprehensive guide to applying three well-known variance reduction techniques is given, including point and interval estimators, software requirements, and guidelines for experiment design.

1. INTRODUCTION

Variance reduction techniques (VRTs) are experiment design techniques that increase the precision of simulation-based point estimators without a corresponding increase in computer effort. However, increased analyst effort is probably the primary reason that VRTs are rarely used in practice, especially if the time needed to understand and select a VRT is included in the "analyst effort."

The goal of this tutorial is to provide a comprehensive guide to applying a limited number of VRTs. The presentation is comprehensive in the sense that it covers the mathematical principles behind the VRTs, the software (in addition to what is available in standard simulation languages) needed to apply the VRTs, the output analysis methods used in conjunction with the VRTs, and guidelines for sample size determination. The number of VRTs, however, is limited to three: *antithetic variates (AV), control variates (CVs),* and *common random numbers (CRN).* These VRTs were selected because they are useful in many simulation experiments, they are easy to apply, and they are not likely to be counterproductive. Some commercial simulation languages have built-in features that facilitate AV and CRN. CVs usually require additional software support, but it is support that is available on many computers.

Textbook treatments of the topics in this tutorial can be found in Bratley, Fox and Schrage (1983) and Law and Kelton (1982), where many of the terms and concepts defined loosely here are defined precisely. Surveys of VRTs include Kleijnen (1974), Nelson (1987a) and Wilson (1984).

The paper is organized as follows: Section 2 introduces examples and notation that will be used throughout the tutorial. Section 3 states the mathematical principles behind the three VRTs. Section 4 presents the three VRTs as they apply to finite-horizon simulation experiments. Section 5 discusses modifications for infinite-horizon experiments. Finally, section 6 offers some concluding comments, including references to more advanced material.

2. EXAMPLES AND NOTATION

This section introduces two simulation problems that will be used as illustrations throughout the tutorial, and also defines the mathematical notation that will be needed. We distinguish between *finite-horizon* simulation experiments, in which the unknown system parameters of interest are defined with respect to fixed starting and ending conditions for the simulated system, and *infinite-horizon* simulation experiments, in which the parameters are defined as limits as the length of the simulation run goes to infinity, limits that are independent of the starting conditions.

Example 1 (finite-horizon experiment): A cardiovascular risk reduction program runs risk screening in the morning hours beginning at 9 am. Thirty patients are scheduled per day at five minute intervals, and daily screening ends when the last patient leaves. Arriving patients first complete a questionnaire with the help of a Clerk. Risk screening tests are then administered by two Medical Technicians working in series. Those patients with high blood pressure must be examined by a Nurse. All patients make an appointment with the Clerk prior to leaving. A simulation experiment is conducted to find a good division of duties between the two Medical Technicians. Performance measures of interest include the average number of patients in the facility, the average time for a patient to complete risk screening, and the utilization of the Medical Technicians.

Example 2 (infinite-horizon experiment; see Law and Kelton (1982), p. 281): A computer manufacturer wants to test several proposed operating systems under prolonged periods of peak loading, periods long enough that the effects of the initial load on the computer has no impact. In particular, long-run expected job response time and device utilization are of interest. Notice that such prolonged peak periods may never actually

occur in the real system, but simulating alternative operating systems under these conditions provides a basis for comparison.

Since section 4 presents VRTs for finite-horizon simulations we define notation with respect to example 1.

Let $Y_j^{(l)}$ represent a simulation output of interest from the jth replication, $j = 1, 2, \cdots, k$, of the lth system design, $l = 1, 2, \cdots, r$. Thus, a subscript represents a replication number and a subscript or superscript in ( ) represents a system design alternative. Frequently, $Y_j^{(l)}$ is itself an average of outputs within the jth replication. For example, $Y_3^{(1)}$ might be the utilization of a Medical Technician on the third day (replication) of simulated risk screening under the first proposed system design, or it might be the average time to screen patients on the third day of screening under the first design.

Let $\theta^{(l)}$ represent the unknown parameter of interest for the lth system design. We assume that replications provide independent and identically distributed observations of system performance, and specifically that $E[Y_j^{(l)}] = \theta^{(l)}$ and $\mathrm{Var}[Y_j^{(l)}] = \sigma_{(l)}^2$ for all j. Thus, if $Y_3^{(1)}$ is the utilization of a Medical Technician on the third day under the first system design, then $\theta^{(1)}$ is the expected utilization under the first design.

Two statistics that will arise frequently are the sample mean and sample variance, which are, respectively,

$$\bar{Y}^{(l)} = \frac{1}{k}\sum_{j=1}^{k} Y_j^{(l)}$$

and

$$s_{(l)}^2 = \frac{1}{k-1}\sum_{j=1}^{k}(Y_j^{(l)} - \bar{Y}^{(l)})^2.$$

In addition to simulation outputs, we define simulation inputs as data generated from experimenter-specified probability distributions. Let $\underline{X}_j^{(l)} = (X_{j1}^{(l)}, X_{j2}^{(l)}, \cdots, X_{jq}^{(l)})'$ represent a column vector of q (not necessarily all) simulation inputs on the jth replication of the lth system design; the $'$ indicates the matrix transpose operation. For fixed l the inputs are assumed to be independent and identically distributed across all replications j. Examples of inputs in the risk screening simulation are the patient arrival times, the time required by the Medical Technicians to administer tests, and the number of patients that must consult with the Nurse, if these quantities are modeled as random variables with known distributions. Since the distributions are known, the expected value of $\underline{X}^{(l)}$ is also known and is denoted by $\underline{\mu}^{(l)} = (\mu_1^{(l)}, \cdots, \mu_q^{(l)})'$.

3. SOME PRINCIPLES OF VARIANCE REDUCTION

Before discussing any particular VRT or how to apply it, we state some basic results that underlie the VRTs. This section is based on Nelson (1985).

Suppose we estimate θ with the point estimator $\bar{Y}$ as defined above (we drop the superscript here and consider only one system design). Then if $\{Y_1, \cdots, Y_k\}$ is covariance stationary

$$\mathrm{Var}[\bar{Y}] = \frac{\sigma^2}{k} + \frac{2}{k}\sum_{h=1}^{k-1}(1 - \frac{h}{k})\gamma_h \qquad (1)$$

where $\gamma_h = \mathrm{Cov}[Y_i, Y_j]$ when $|i - j| = h$. If the $\{Y_j\}$ are independent, as they are for independent replications, then (1) reduces to σ^2/k.

Now let $\{X_1, X_2, \cdots, X_k\}$ be a sequence of identically distributed scalar random variables. Let $Z_j = Y_j \pm bX_j$, where b is a constant. Then

$$\mathrm{Var}[Z_j] = \sigma^2 + b^2\sigma_x^2 \pm 2b\,\mathrm{Cov}[Y_j, X_j] \qquad (2)$$

where $\sigma_x^2 = \mathrm{Var}[X_j]$.

Result (1) shows that there are three components that determine the variance of a sample mean: σ^2, γ_h, and k. Decreasing σ^2 and γ_h, or increasing k, reduces $\mathrm{Var}[\bar{Y}]$. Result (2) shows that the combination of Y_j with another random variable X_j may yield a random variable with smaller variance, provided the covariance between them is large enough and has the correct sign. Variance reduction is achieved by designing a simulation experiment to take advantage of these results. It is important to stress that *variance reduction* refers to reducing the population variance of an **estimator** of θ, where θ may be a variance. Variance reduction does not necessarily affect the variability of the simulated stochastic process; e.g. σ^2 may not be changed.

4. THREE VARIANCE REDUCTION TECHNIQUES

In this section three VRTs are presented in detail. We consider finite-horizon experiments, and subdivide the section into two cases: (1) When θ is the parameter of a single system design ($r = 1$), and (2) when we are interested in determining the system with the largest $\theta^{(l)}$ for $l = 1, \cdots, r \geq 2$ different systems. In each subsection we define the standard or "crude" experiment and then present the VRT experiment. In all cases point and interval estimators are specified, implementation instructions are given, and experiment design issues are discussed. The interval

estimators are confidence intervals based on the t distribution; see any introductory statistics text for a discussion of the assumptions behind these estimators. If, as is typically the case, there are several performance measures of interest for each system design, then the estimators presented here can be applied individually for each parameter, perhaps using a Bonferroni procedure to achieve a specified overall confidence level. We do not consider multivariate procedures in this tutorial.

4.1 Absolute Parameter

The purpose of the experiment is to estimate θ, a performance measure for a single system. For example, θ could be the expected utilization of the first Medical Technician over the course of a day in example 1, and $\{Y_1, \cdots, Y_k\}$ could be the utilization of the Medical Technician on k simulated days of risk screening.

For the purposes of this section the "crude" experiment is:

number of independent replications: k
point estimator: $\bar{Y}$
interval estimator: $\bar{Y} \pm t_{\alpha/2}(k-1)s/\sqrt{k}$

where $t_{\alpha/2}(k-1)$ is the $1-\alpha/2$ quantile of the t distribution with $k-1$ degrees of freedom and s is the square root of the sample variance s^2. The interval estimator is often referred to as a $(1-\alpha)100\%$ confidence interval, and α is traditionally .10, .05 or .01. We assume that the experimenter will make k as large as the available time and/or budget allows.

4.1.1 Antithetic Variates

AV exploits (1) by inducing favorable covariance terms γ_h; all the $\gamma_h = 0$ in the crude experiment. The required covariances are realized indirectly by inducing dependence between previously independent simulation inputs. There are two major problems: First, it is not possible to make all of the covariances negative. Thus, AV often settles for inducing negative covariance between pairs of replications $\{Y_{2j-1}, Y_{2j}\}$, $j = 1, 2, \cdots, k/2$, leaving different pairs independent; this is the only form of AV considered here. The second problem is that inducing negative covariance between inputs does not guarantee negative covariance between outputs. We discuss this issue under **Implementation** below.

AV Experiment. To define the AV experiment, let $\hat{Y}_j = (Y_{2j-1} + Y_{2j})/2$ for $j = 1, 2, \cdots, k/2$. Let $\hat{Y}$ be the sample mean of these outputs, which is algebraically the same as $\bar{Y}$. Then the simulation experiment is:

number of independent replications: $k/2$
point estimator: $\hat{Y}$
interval estimator: $\hat{Y} \pm t_{\alpha/2}(k/2-1)\hat{s}\sqrt{2/k}$

where the sample variance of the $\{\hat{Y}_j\}$ is

$$\hat{s}^2 = \frac{1}{k/2-1}\sum_{j=1}^{k/2}(\hat{Y}_j - \hat{Y})^2.$$

Implementation. Inducing negative covariance between pairs of simulation outputs $\{Y_{2j-1}, Y_{2j}\}$ depends on being able to induce dependence between pairs of simulation inputs. Suppose X_{2j-1} and X_{2j} are inputs corresponding to replications $2j-1$ and $2j$, respectively, both having cumulative distribution function (cdf) $F(x)$. For example, X_1 and X_2 might be the time required to draw a blood sample from the first patient on simulated days 1 and 2, respectively. Then if U is a sample from the uniform distribution on the interval (0, 1), denoted $U(0, 1)$, setting

$$X_{2j-1} = F^{-1}(U)$$

$$X_{2j} = F^{-1}(1-U)$$

yields values of X_{2j-1} and X_{2j} with the correct marginal distributions and minimal achievable covariance. This method of random variate generation is known as the *inverse transform.* If the simulation code maps the inputs $\{X_{2j-1}, X_{2j}\}$ into $\{Y_{2j-1}, Y_{2j}\}$ monotonically, then the negative covariance between the inputs is preserved in the outputs. Of course, the outputs from each replication are usually functions of many inputs, so this procedure must be applied to each input random variable in replication $2j-1$ and corresponding replication $2j$ for $j = 1, \cdots, k/2$.

Many commercial simulation languages have antithetic sampling of inputs built in to the random variate generators. Typically, specifying a negative random number stream causes the antithetic values of the uniform random variates to be used; i.e. $1-U$ instead of U. However, some random variate generators do not use the inverse transform method to generate X from U, and in those cases the antithetic sampling procedure is not guaranteed (or likely) to produce negatively correlated inputs.

The reason that the inverse transform is not always used is because it may be computationally slow and/or because no closed-form expression for the inverse exists. However, the variance reduction from AV can more than offset the computational cost of slower approximate inverse transform methods, and the approximations can be quite accurate. Bratley, Fox and Schrage (1983, Chapter 5) emphasize inverse transform methods of variate generation and give variate generation code that could be used in place of the routines contained in a simulation language.

Even when dependence can be induced between pairs of inputs, there is no guarantee that the dependence will be realized in the outputs where it is needed. Monotonicity of the input-output mapping can be enhanced by synchronizing the random numbers used in antithetic replications. The idea of synchronization is to ensure that if, say, U is used to generate some input in the first replication of the antithetic pair, then $1-U$ is used for the corresponding input in the second replication. One standard trick is to use a different random number stream (sequence of $U(0,1)$ values) for each input process in the simulation. In example 1, for instance, one stream might be used to generate the testing time at the first Medical Technician, another for the second Technician, and so on. Bratley, Fox and Schrage (1983) discuss approaches for synchronization at some length.

Design Issues. Let $\rho = \text{Corr}[Y_{2j-1}, Y_{2j}]$, the induced correlation between antithetic pairs of replications. If $\rho < 0$, then the AV point estimator $\hat{Y}$ has smaller variance than the crude point estimator $\bar{Y}$. However, the performance of the AV interval estimator relative to the crude interval estimator is not as certain. The performance of the interval estimator depends on the number of replications, k, the magnitude of the achieved negative correlation, ρ, and the confidence level α. Unfortunately, ρ is never known. However, Nelson (1987b) showed that if $k \geq 20$ then ρ as large as $-.12$ gives the AV interval estimator better expected performance than the crude interval estimator in terms of the length of the confidence interval (a shorter interval is better). On the other hand, if $k < 8$ the correlation requirement for improved performance is $\rho < -.3$, going rapidly to $\rho < -.92$ for $k = 4$. Achieving such correlations is optimistic for practical problems. Thus, if the budget is very tight and the interval estimate is important, then the AV experiment may not be superior to the crude experiment. However, if there is sufficient budget to obtain 20 or more replications (10 antithetic pairs), then interval estimator performance is likely to be improved by AV.

4.1.2 Control Variates

CV exploits the covariance between the simulation output of interest, Y_j, and a $q \times 1$-vector of (usually averages of) input random variables $\underline{X}_j$ via (2). Suppose $q = 1$, meaning $\underline{X}_j$ is scalar, and $\mu = \text{E}[\underline{X}_j]$. For example, Y_j may be the utilization of the Nurse and X_j the number of patients that consulted with the Nurse on the jth simulated day of screening. The expected value of X_j is known if the percentage of patients with high blood pressure is specified by the experimenter. The CV estimator is

$$\bar{Z} = \bar{Y} - b(\bar{X} - \mu)$$

For any constant b, $\bar{Z}$ is an unbiased estimator of θ. The variance of $\bar{Z}$ can be minimized by setting $b = \text{Cov}[Y_j, X_j]/\text{Var}[X_j]$. This value is seldom known, but it can be estimated; see **Implementation** below. The CV estimator readily generalizes to $q > 1$ controls and a $q \times 1$-vector control multiplier $\underline{b} = (b_1, \cdots, b_q)'$. The variance of $\bar{Z}$ can be reduced substantially by using more than one control, but some deterioration in performance is also possible; see **Design Issues** below.

CV Experiment. From the jth replication of the simulation experiment we observe $\{Y_j, X_{j1}, \cdots, X_{jq}\}$. Let $\bar{\underline{X}} = (\bar{X}_1, \cdots, \bar{X}_q)'$, the sample averages of the control variates. Then the control variate point estimator is

$$\bar{Z} = \bar{Y} - \hat{\underline{b}}'(\bar{\underline{X}} - \underline{\mu})$$

where $\hat{\underline{b}}$ is the estimated control multiplier. The simulation experiment is

number of independent replications: k
point estimator: $\bar{Z}$
interval estimator: $\bar{Z} \pm t_{\alpha/2}(k-q-1)\bar{s}$

where the computation of $\bar{s}^2$ is discussed below.

Implementation. Computing the CV point and interval estimators can be viewed as estimating the intercept term in a least-squares regression of Y_j on $\underline{X}_j - \underline{\mu}$. This means that any available software for regression can be used. The regression model is

$$Y_j = \theta + \sum_{i=1}^{q} b_i(X_{ji} - \mu_i) + \varepsilon_j$$

for $j = 1, \cdots, k$. The point estimator $\bar{Z}$ is just the least-squares estimator of θ, and $\bar{s}^2$ is the estimated variance of $\bar{Z}$; both are computed by standard regression packages.

Unlike AV, CVs do not affect the execution of the simulation experiment, they only affect the calculation of the point and interval estimators. Since selecting control variates from a set of potential controls may be necessary, data base capabilities for storing the simulation output and the potential control variates may be useful. Then the regression routine can be used to select the CVs and compute the estimators without complicating the simulation program.

Design Issues. The variance of the CV estimator $\bar{Z}$ depends on three factors: (1) the number of replications, k, (2) the number of control variates, q, and (3) the square of the multiple correlation coefficient of Y_j on $\underline{X}_j$, R^2. The magnitude

of R^2 depends on both q and the particular control variates that compose $\underline{X}_j$. Mathematically,

$$\mathrm{Var}[\bar{Z}] = \frac{k-2}{k-q-2}(1-R^2)\mathrm{Var}[\bar{Y}]$$

The factor $(k-2)/(k-q-2)$ is nondecreasing in q, while $(1-R^2)$ is nonincreasing in q.

Since almost any input random variable is a potential control variate, most simulations have many potential controls. Selecting a subset of controls that balances the above factors is difficult. Some researchers have suggested stepwise regression procedures to select the control variates that appear to make the most significant contribution to variance reduction. However, since many regression packages automatically calculate R^2, an approach suggested by Nelson (1986) that accounts for the performance of the CV point and interval estimators can be used.

Suppose that at some stage in the control variate selection process q controls have been chosen. The decision is whether or not to add another control. Let $R^2(k, q)$ be the square of the multiple correlation coefficient for an experiment with k replications and a fixed set of q controls. For the remaining potential controls compute the marginal improvement ratio

$$\frac{1-R^2(k, q+1)}{1-R^2(k, q)}. \qquad (3)$$

This is the ratio of the unexplained variation in $\bar{Y}$ after adding the $(q+1)$st control to the unexplained variation before adding it. Table 1 gives bounds such that if (3) is less than or equal to the bound the additional control variate will not degrade point and interval estimator performance; if (3) is strictly less than the bound then performance is improved (smaller point estimator variance and shorter expected interval length). Notice that as the number of controls increases each additional control is more difficult to add. However, large k makes it easier to add controls. Of course, $R^2(k, q)$ is never known, so we can only estimate the marginal improvement ratio. Stepwise regression procedures based on R^2 are discussed in Neter and Wasserman (1974).

Table 1: Marginal improvement ratio for adding a control variate

k / q+1	1	2	3	4	5
10	.78	.76	.75	.73	.69
30	.96	.95	.93	.93	.93
60	.97	.97	.97	.97	.97

4.2 Relative Difference

Suppose that there are two competing system designs and we are interested in $\theta^{(1)} - \theta^{(2)}$, the difference between corresponding parameters of the two systems. This situation is common in simulation experiments. For example, consider estimating the difference between the utilization of the first Medical Technician under two different allocations of tasks in example 1.

Let $D_j = Y_j^{(1)} - Y_j^{(2)}$ for $j = 1, 2, \cdots, k$, and let $\bar{D} = \bar{Y}^{(1)} - \bar{Y}^{(2)}$. If $\sigma^2_{(1)} = \sigma^2_{(2)}$, then define the pooled sample variance

$$s_P^2 = \frac{1}{2k-2}\sum_{i=1}^{2}\sum_{j=1}^{k}(Y_j^{(i)} - \bar{Y}^{(i)})^2.$$

The crude experiment is

number of independent replications: k from each system
point estimator: $\bar{D}$
interval estimator: $\bar{D} \pm t_{\alpha/2}(2k-2)s_P/\sqrt{2k}$

If the variances are not known to be equal, define the difference sample variance

$$s_D^2 = \frac{1}{k-1}\sum_{j=1}^{k}(D_j - \bar{D})^2.$$

The crude experiment in this case is

number of independent replications: k from each system
point estimator: $\bar{D}$
interval estimator: $\bar{D} \pm t_{\alpha/2}(k-1)s_D/\sqrt{k}$

We assume an equal number of replications from each system, which can always be guaranteed in simulation experiments.

4.2.1 Common Random Numbers

CRN exploits (2) -- where we identify $Y_j^{(1)}$ with Y_j, $Y_j^{(2)}$ with X_j, and set $b = 1$ -- by inducing positive covariance between the pairs of outputs $(Y_j^{(1)}, Y_j^{(2)})$, leaving different pairs independent. Again, the covariance is induced by inducing dependence between simulation inputs.

CRN Experiment ($r = 2$). The simulation experiment is

number of independent replications: k
point Estimator: $\bar{D}$
interval estimator: $\bar{D} \pm t_{\alpha/2}(k-1)s_D/\sqrt{k}$

Implementation. As with AV, the inverse transform method of variate generation is still the method of choice for inducing dependence between corresponding simulation inputs. However, instead of using antithetic (U and $1-U$) random

number streams on pairs of replications, the same ("common") random numbers are used on pairs of replications from each system design. Synchronization is also important to ensure that dependence induced between inputs is realized in the outputs. Unlike AV, positive correlation may result even if the inverse transform in not used; e.g., if the distribution of an input is the same under both system designs then precisely the same inputs will be generated for each system no matter what variate generation method is used.

Design Issues. Let $\rho = \text{Corr}[Y_j^{(l)}, Y_j^{(2)}]$. If the crude experiment does not assume equal variances then both the CRN point and interval estimators dominate the crude estimators when $\rho > 0$. However, if the crude experiment uses the pooled variance estimator then guidelines similar to AV apply. If we obtain $k \geq 10$ replications from each system (20 total), then the CRN interval estimator is likely to be better than the crude estimator even if ρ is small. However, if $k < 4$ then superior performance is not as certain unless ρ is known to be large.

We next consider the situation when there are $r > 2$ system designs and we want to identify l^* such that

$$\theta^{(l^*)} = \max_l \{\theta^{(l)}\}$$

That is, we want the system design that has the largest parameter. The procedure we present guarantees a minimum probability $1/r < p < 1$ of selecting the best system when the difference between $\theta^{(l^*)}$ and the second best $\theta^{(l)}$ is greater than or equal to δ. The constants p and δ must be specified by the experimenter.

In the interest of space we do not present a crude procedure; see Dudewicz and Dalal (1975) and the survey by Goldsman (1983). The CRN-based procedure CY below is due to Clark and Yang (1986). Their procedure assumes that common random numbers are employed when sampling from all r systems. The goal of CRN is to reduce the total number of replications needed to ensure the specified probability of correct selection.

CRN Experiment ($r > 2$). The simulation experiment proceeds in two stages, beginning with a sample of k_0 replications from each system, and followed by a second stage of sampling. The number of initial replications k_0 must also be specified by the experimenter. The system with the largest sample mean is chosen as the best system.

Define the difference variance for system designs $l \neq i$ as

$$s^2_{D(l,i)} = \frac{1}{k-1}\sum_{j=1}^{k}((Y_j^{(l)} - Y_j^{(i)}) - (\bar{Y}^{(l)} - \bar{Y}^{(i)}))^2.$$

Procedure CY

0. Select k_0, δ and p.

1. Sample k_0 replications $Y_1^{(l)}, \cdots, Y_{k_0}^{(l)}$ from each system, $l = 1, \cdots, r$, using CRN.

2. Compute $\bar{Y}^{(l)}$, $l = 1, \cdots, r$.

3. Compute $s^2_{D(l,i)}$ for $l = 1, \cdots, r$, $i = 1, \cdots, r$, $l \neq i$. For $l = 1, \cdots, r$ let

$$s^2_{D(l)} = \max_{i \neq l}\{s^2_{D(l,i)}\}.$$

4. Let $k_{(l)} = \max\{k_0, [s^2_{D(l)}h^2/\delta^2 + 1/2]\}$ for $l = 1, \cdots, r$, where h comes from Tables 2 or 3 below, and $[a]$ is the largest integer less than or equal to a.

5. Sample $k_{(l)} - k_0$ additional replications from system l and compute

$$\bar{Y}^{(l)} = \frac{1}{k_{(l)}}\sum_{j=1}^{k_{(l)}} Y_j^{(l)}.$$

6. Select the system design l^* with the largest sample mean.

Table 2: Values of h for Clark and Yang's Procedure ($p = .9$)

k_0 / r	3	4	5	6	7	8
5	2.13	2.50	2.78	3.00	3.19	3.35
10	1.83	2.09	2.26	2.40	2.51	2.60
20	1.73	1.95	2.09	2.20	2.29	2.37
30	1.70	1.91	2.05	2.15	2.23	2.30

Table 3: Values of h for Clark and Yang's Procedure ($p = .95$)

k_0 / r	3	4	5	6	7	8
5	2.78	3.19	3.50	3.75	3.96	4.15
10	2.26	2.51	2.69	2.82	2.93	3.03
20	2.09	2.29	2.43	2.54	2.63	2.70
30	2.05	2.23	2.36	2.46	2.54	2.61

Additional tables can be generated by finding h such that $\Pr\{T(k_0 - 1) < -h\} = (1 - p)/(r - 1)$, where $T(k_0 - 1)$ is a random variable with a t distribution and $k_0 - 1$ degrees of freedom.

Implementation. To implement a two-stage procedure such as CY requires the facility to save the random number seeds at the end of k_0 replications of each system so that the simulations can be restarted for the second stage.

Design Issues. We recommend a minimum initial sample size of $k_0 \geq 10$ to ensure reasonable performance of the first-stage variance estimators. Empirical results in Yang (1986) suggest that the procedure will be more efficient than the Dudewicz and Dalal procedure in terms of total number of replications when the induced positive correlation is large or the number of system designs is no more than 8.

5. VRTS FOR INFINITE-HORIZON SIMULATION EXPERIMENTS

In this section we describe modifications of the VRTs, output analysis methods, and experiment designs in section 4 for infinite-horizon simulation experiments. Recall that infinite-horizon experiments estimate a system parameter defined as simulated time becomes infinite; see example 2 in section 2.

Let $Y_{ij}^{(l)}$ denote the ith output from the jth replication for the lth system design, with $i = 1, 2, \cdots, m$. In example 2, $Y_{73}^{(2)}$ might be the response time of the seventh job on the third replication under the second operating system. The additional subscript is necessary because output analysis methods for infinite-horizon experiments are affected by the number of replications and the length of the replications. The length of a replication is not an issue in finite-horizon simulations because replications have well-defined starting and ending conditions.

The outputs within a replication may be neither independent nor identically distributed. Let $\theta^{(l)} = \lim_{i \to \infty} E[Y_{ij}^{(l)}]$ be the parameter of interest for the lth system design; e.g. long-run expected response time in example 2. Due to initial conditions (such as the number of jobs initially loaded on the computer), $E[Y_{ij}^{(l)}]$ may not equal $\theta^{(l)}$ for any finite i, and may be severely biased for small i. This suggests an experiment design with m as large as possible, even to the point of only $k = 1$ replication. However, for $h \neq i$, $Y_{i1}^{(l)}$ and $Y_{h1}^{(l)}$ are not independent, in general, which makes interval estimation from a single replication difficult.

The problem of initial-condition bias is one of the most difficult in simulation output analysis. The standard recommendation is to discard some of the outputs from the beginning of each replication. Let

$$\bar{Y}_j(d) = \frac{1}{m-d} \sum_{i=d+1}^{m} Y_{ij}$$

where $d < m$, and we drop the superscript representing different system designs. This is the *truncated replication average*. If we treat $\{\bar{Y}_1(d), \bar{Y}_2(d), \cdots, \bar{Y}_k(d)\}$ as the basic data, then the VRTs in section 4 can be applied directly to the truncated replication averages. (We have ignored the difficult question of determining d; see Law and Kelton (1982).) If the simulation budget is not limited or d is very small, then the experiment design recommendations above apply. If, however, the total available budget is fixed at, say, $km = n$ total observations and d is large, then the tradeoff between the number of replications k and length of each replication m becomes more crucial. The larger k is the more outputs are discarded and the shorter each replication is. However, if k is too small the variance of the point estimator may be large and there will be too few degrees of freedom for the interval estimators.

Single-replication output analysis methods have been developed for situations when d is large relative to the total budget. We describe one method, *nonoverlapping batch means*, and give guidelines for applying it in conjunction with the VRTs in section 4.

Assume that initial conditions have already been eliminated. For $k = 1$ replication let

$$\bar{Y}_j = \frac{1}{m} \sum_{i=(m-1)j+1}^{jm} Y_{i1}$$

be the jth *batch mean* of size m, for $j = 1, \cdots, k$, where $km = n$. The batch means partition a single replication of length n into k nonoverlapping batches of size m. The idea behind batch means is that there is an (unknown) number of batches k^* (equivalently batch size m^*) such that for $k \leq k^*$ ($m \geq m^*$) the batch means are nearly independent. Thus, the batch means take the place of the replication means in the VRTs described in section 4.

The most difficult design decision is selecting the batch size. Batching algorithms that employ tests of independence have been advocated (e.g. Fishman, 1978). In principle we want k as large as possible, but not larger than k^*. Although the value of k^* is not known, Schmeiser (1982) showed that it is not necessary to determine if $k^* > 30$ because interval estimator performance is not improved substantially beyond 30 batches no matter how large k^* or n is, and the point estimator is unaffected by the number of batches.

To estimate the difference between the performance of two system designs, sample one replication from each system and batch. If CRN is employed, we recommend looking for an acceptable number of batches in the range of 10 to 30 from each system, where we assume the same number of batches and batch size for both systems. This recommendation takes into account the number of batches needed to expect improved interval

estimator performance from CRN and the number required to obtain nearly all the benefits from degrees of freedom. If a minimum of 10 approximately independent batches cannot be achieved with the given sample size, then the experimenter should consider increasing the replication length, if possible. On the other hand, no matter how long the replication is there is little additional benefit from dividing it into more than 30 batches, and possible harm since k^* is not known. For procedure CY a minimum of $k_0 \geq 10$ batches for the initial sample is recommended.

In the case of CVs, both the outputs and the controls should be batched together; Anonuevo and Nelson (1986) describe one algorithm. When $q \leq 5$ controls are used, Nelson (1986) showed that $30 \leq k \leq 60$ batches protects against degradation in estimator performance due to selecting ineffective controls, while providing nearly the maximum benefit from degrees of freedom. Again, if a minimum of 30 acceptable batches cannot be obtained then the experimenter should consider increasing the replication length, but no matter what the length is there is little benefit from dividing it into more than 60 batches.

6. CONCLUSIONS

The VRTs presented in this tutorial are versatile and relatively easy to apply. The price of generality is that problem specific VRTs will almost always be more effective. However, the time needed to develop a problem specific VRT may be considerable. Nelson (1985) proposes an algorithm for selecting VRTs from among the many that have been developed.

CRN should be a standard VRT for all experimenters, even if confidence interval procedures that take advantage of it are not used. Remember that CRN reduces the variance of the difference $\bar{Y}^{(1)} - \bar{Y}^{(2)}$, not the individual point estimators. For experiment designs that take advantage of CRN see Schruben and Margolin (1978) and Tew and Wilson (1985)

AV should also be applied routinely in practice. However, it is important that the inverse transform method of variate generation is employed. Because variate generation is such a small part of the computation in dynamic simulation experiments, and because of the potential for variance reduction, we recommend that simulation language designers adopt inverse or approximate inverse transform methods in commercial languages.

Any stochastic simulation contains potential control variates. However, the theory of CVs depends on multivariate normality assumptions, and it is not yet known how robust the theory is to nonnormality. Based on our experience, we recommend CVs. The theory of CVs and multivariate extensions are discussed in Lavenberg and Welch (1981), Nozari, Arnold and Pegden (1984), Porta Nova and Wilson (1986), and Venkatraman and Wilson (1986).

ACKNOWLEDGMENTS

This research was partially supported by the National Science Foundation under Grant No. ECS-8707634. Tables 2 and 3 were prepared by Wei-Ning Yang. The author received helpful comments from Jane Fraser and Charles Reilly of The Ohio State University.

REFERENCES

Anonuevo, R. and B.L. Nelson (1986). Automated Estimation and Variance Reduction for Steady-State Simulations. *Proceedings of the 1986 Winter Simulation Conference,* 871-875.

Bratley, P., B.L. Fox, and L.E. Schrage (1983). *A Guide to Simulation.* Springer-Verlag, NY.

Clark, G.M. and W. Yang (1986). A Bonferroni Selection Procedure When Using Common Random Numbers with Unknown Variances. *Proceedings of the 1986 Winter Simulation Conference,* 313-315.

Dudewicz, E.J. and D.R. Dalal (1975). Allocation of Observations in Ranking and Selection with Unequal Variances. *Sankhya* **37**, 28-78.

Fishman, G.S. (1978). Grouping Observations in Digital Simulation. *Management Science* **24**, 510-521.

Goldsman, D. (1983). Ranking and Selection in Simulation. *Proceedings of the 1983 Winter Simulation Conference,* 387-393.

Kleijnen, J.P.C. (1974). *Statistical Techniques in Simulation, Part I.* Marcel Dekker, NY.

Lavenberg, S.S. and P.D. Welch (1981). A Perspective on the Use of Control Variables to Increase the Efficiency of Monte Carlo Simulations. *Management Science* **27**, 322-335.

Law, A.M. and W.D. Kelton (1982). *Simulation Modeling and Analysis.* McGraw-Hill, NY.

Nelson, B.L. (1985). A Decomposition Approach to Variance Reduction. *Proceedings of the 1985 Winter Simulation Conference,* 23-32.

Nelson, B.L. (1986). Batch Size Effects on the Efficiency of Control Variates in Simulation. Working Paper Series No. 1986-001, Dept. of Industrial and Systems Engineering, Ohio State University.

Nelson, B.L. (1987a). A Perspective on Variance Reduction in Dynamic Simulation Experiments. *Communications in Statistics - Simulation and Computation* **B16**, in press.

Nelson, B.L. (1987b). Some Properties of Simulation Interval Estimators Under Dependence Induction. *Operations Research Letters* **6**, in press.

Neter, J. and W. Wasserman (1974). *Applied Linear Statistical Models.* Irwin, Homewood, IL.

Nozari, A., S.F. Arnold and C.D. Pegden (1984). Control Variates for Multipopulation Simulation Experiments. *IIE Transactions* **16**, 159-169.

Porta Nova, A.M.O. and J.R. Wilson (1986). Using Control Variates to Estimate Multiresponse Simulation Metamodels. *Proceedings of the 1986 Winter Simulation Conference,* 326-334.

Schmeiser, B. (1982). Batch Size Effects in the Analysis of Simulation Output. *Operations Research* **30**, 556-568.

Schruben, L.W. and B.H. Margolin (1978). Pseudorandom Number Assignment in Statistically Designed Simulation and Distribution Sampling Experiments. *Journal of the American Statistical Association* **73**, 504-525.

Tew, J.D. and J.R. Wilson (1985). Validation of Correlation Induction Strategies for Simulation Experiments. *Proceedings of the 1985 Winter Simulation Conference,* 190-195.

Venkatraman, S. and J.R. Wilson (1986). The Efficiency of Control Variates in Multiresponse Simulation. *Operations Research Letters* **5**, 37-42.

Wilson, J.R. (1984). Variance Reduction Techniques for Digital Simulation. *American Journal of Mathematical and Management Sciences* **4**, 277-312.

Yang, W. (1985). A Bonferroni Selection Procedure when using Common Random Numbers with Unknown Variances. Unpublished M.S. thesis, Dept. of Industrial and Systems Engineering, Ohio State University.

AUTHOR'S BIOGRAPHY

BARRY L. NELSON is an Assistant Professor in the Department of Industrial and Systems Engineering at The Ohio State University. He is also associated with the Ohio State University Statistical Consulting Service. His Ph.D. is from the School of Industrial Engineering at Purdue University, and his research interests center on design and analysis of simulation experiments, particularly methods for statistically efficient simulation. He teaches courses in simulation and stochastic processes. Continuing memberships include ASA, IIE, ORSA, TIMS and SCS. Dr. Nelson is an active member of the TIMS College on Simulation and Gaming, and is editor of its Newsletter.

Barry L. Nelson
Department of Industrial and Systems Engineering
The Ohio State University
Columbus, OH 43210, USA
(614) 292-0610