MONTE CARLO TECHNIQUES FOR STOCHASTIC NETWORK ANALYSIS

by John M. Burt, Jr.* and Mark B. Garman†

Summary

This paper presents simulation procedures for efficiently obtaining estimates of the distribution function, or parameters thereof, of the maximal flow-time through directed acyclic networks whose activity times are random variables.

Introduction

Project graph analysis (variously called PERT or CPM) has proved to be a useful and broadly applied management tool. In its original form, the completion times for individual activities of a project were assumed fixed and known in advance: this in turn led to the now familar algorithms for computing the critical path, earliest start times, slack, and so forth. The unreality of such an assumption is, of course, apparent in many contexts. Consequently, attempts were made to introduce probabilistic activity completion times, allowing representation of the stochastic nature of most projects.

Methods for solving the stochastic PERT problem have usually followed one of three basic approaches: analytic, approximation, or Monte Carlo methods. All are intended to avoid a difficult multivariate integration, which is the most general solution technique. That this is so may easily be shown as follows: Let a PERT network be defined as usual (a directed, acyclic, weakly-connected graph) with source node s and terminus node z. To the k^{th} arc in the network, $k = 1,2,\ldots,b$, attach a non-negative random variable, T_k, and a cumulative distribution function (c.d.f) $F_k(t)$; these will represent the time of complete the k^{th} activity and probability that this completion time is less than t, respectively. Now let P_j be the subsets of $\{1,2,\ldots,b\}$ such that $k \in P_j$ if and only if the k^{th} arc lies on the j^{th} path, $j = 1, 2,\ldots,m$, from source s to terminus z. The time to complete all the activities on the j^{th} path is a random variable, X_j, where

$$X_j = \sum_{k\epsilon P_j} T_k \; . \qquad (1.1)$$

One variant of the stochastic PERT problem may be stated as that of finding the c.d.f. for the random variable

$$T = \max_{j=1,2,\ldots,m} X_j \; . \qquad (1.2)$$

(The path P_{j*} for which $j*$ maximizes (1.2) is called the critical path and T is called the critical path time or project completion time.) The solution to the problem is simple in appearance, being given by

$$F_T(t) = \int\int\cdots\int_{\substack{x_j\le t\\ j=1,2,\ldots,m}} \prod_{k=1}^{b} dF_k(t_k) \qquad (1.3)$$

where $x_j = \sum_{k\epsilon P_j} t_k$.

In practice, however, expression (1.3) is usually quite difficult to evaluate. This is particularly the case when the number of paths, m, or activities, b, is large or the c.d.f.'s are cumbersome to handle mathematically (e.g., step functions, normal distributions). Other variants of the stochastic PERT problem include calculation of the mean, or expected project completion time,

$$M = \int t\,dF_T(t) = E(T), \qquad (1.4)$$

or more generally, if $c(t)$ is any function (which may be interpreted as a cost function), the calculation of the expected value (cost)

$$C = \int c(t)\,dF_T(t) = E(c(T)), \qquad (1.5)$$

and so on.

The analytic methods tackle expression (1.3) by assuming (1) simple functional forms for the $dF_k(t)$, and (2) special networks for which the multivariate integral (1.3) may be separated into a series of single-variate integrals. Martin (1965) has described a class of networks susceptible to "series-parallel reduction," which he gives as a systematic way of separating (1.3) based on network configuration. Charnes, Cooper, and Thompson (1964) illustrate their method of "chance constrained and stochastic programming" with exponential c.d.f.'s and a network for which (1.3) also separates. Gaver and Burt (1968) have investigated the exponential families of c.d.f.'s in connection with simple stochastic PERT networks.

Generally speaking, we cannot expect in a given problem to find both simple c.d.f.'s and simple network structure. In this case, one alternative is to approximate the given activity completion time c.d.f.'s and network with simpler c.d.f.'s and networks. For example, the c.d.f.'s can be approximated by single step functions, which reduces the stochastic problem to an ordinary deterministic one. Fulkerson (1962) supplies a lower bound on the expected project finish time (1.4), which is shown to be an improvement over the lower bound obtained by solving the deterministic problem in which the random completion times are replaced by their means. Clark (1961) uses the central limit theorem as a rationale for assuming that the X_j's of (1.1) have a joint normal distribution, and gives tables of statistics for this case. Kleindorfer (1969) has derived upper and lower bounds on $F_T(t)$ by successively bounding the c.d.f.'s of incomplete paths from s to z.

One annoying aspect of most approximation methods is that it is difficult to predict the accuracy of the total project approximations, even knowing the accuracy of the individual activity approximations. This might be particularly distressing when, say, the cost function of (1.5) is drastically non-linear, or when an unknown error cannot otherwise be accepted. If analytical methods are unsuitable and the uncertainties of approximation cannot be tolerated, then there is no choice but to turn to numerical integration for the evaluation of (1.3)-(1.5). Due to the multivariate nature of the integrals, their difficult regions of integration, and limitations on computing resources, Monte Carlo simulation is normally employed in their

*University of California, Los Angeles
†University of California, Berkeley

evaluation. In this area, Van Slyke (1963) has treated the case of straightforward Monte Carlo simulations and also gives a method for improving estimates of (1.4) based on his "criticality" index.

In the present paper, we shall advocate some combinations of the analytic and Monte Carlo approaches, giving several special techniques for improving the estimation of (1.3)-(1.5) derived from the general theory of Monte Carlo methods. The first technique to review here is that of straightforward simulation, also referred to as crude Monte Carlo.

In crude Monte Carlo, samples T_k^i are drawn from the populations of each activity time T_k. These are then added a long the possible paths of the networks accordings to (1.1), yielding path time sample X_j^i; the maximum of these path times is then taken according to (1.2) and the result, T^i, yields a single sample of the project's completion time. (We will call T^i a realization of the project completion time T.) If this process is then repeated a number of times for $T^1, T^2, \cdots, T^n$, we may use the resulting empirical distribution of realizations to estimate the actual distribution of the project completion time. In essence we are performing a sampling experiment, constructing draws from the desired distribution function. That there is uncertainty in our estimates is an inevitable consequence of the randomness of the activity time samples. To reduce this uncertainty and increase the confidence of our results, it is often necessary to take a very large sample. Then "on the average," the abnormally high and low values of our estimates will balance out.

In terms of computer time, the cost of taking a very large number of realizations may be prohibitive; for example, it is well known that the variance in the estimation of (1.4) generally decreases with the number of realizations, N, by only a factor of 1/N. Thus improving the confidence in our estimates while limiting the effort involved in computing the realizations is a topic of considerable interest. Hammersley and Handscomb (1967) and Shreider (1966) give several general variance reduction techniques for Monte Carlo. In the following sections we shall apply some of these to the stochastic PERT problem.

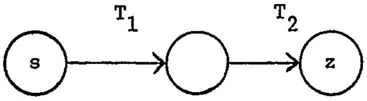


Figure 1.1

To illustrate the advantages of the Monte Carlo techniques to be described in Sections II and III, we will use the simple two-activity series network of Figure 1.1. The success of the techniques to be discussed may be shown for any network configuration; however, this simple network is sufficient for illustrating their utility.

First let us note that a random draw from any known probability function may be expressed as a function of a draw from the uniform distribution, denoted U(0, 1). This fact is an obvious consequence of the monotonicity of cumulative distributions. That is, if T_k is the desired draw from a given distribution, $F_k(\cdot)$, then $T_k = F_k^{-1}(R)$, where R is a random draw from U(0, 1).

Antithetic Variates

The crude Monte Carlo method of estimating $E(T_1 + T_2) = E(T)$ in Figure 1.1 would be to take two independent random draws, R_1 and R_2, from U(0, 1). We would next transform them into independent random draws, T_1 and T_2, by $F_1^{-1}(R_1)$ and $F_2^{-1}(R_2)$ respectively. The first realization would be

$$T^1 = T_1^1 + T_2^1. \qquad (2.1)$$

We would then tabulate n such independent realizations using random numbers and average to obtain the "straightforward" estimate

$$\overline{T} = \frac{T_1^1 + T_2^1 + T_1^2 + T_2^2 + \cdots + T_1^n + T_2^n}{n}. \qquad (2.2)$$

All random numbers are independent, and n samples are generated from their appropriate identical distributions, so we have

$$E(\overline{T}) = E(T_1) + E(T_2) = E(T) \qquad (2.3)$$

and

$$\mathrm{Var}(\overline{T}) = \frac{\mathrm{Var}(T_1) + \mathrm{Var}\ (T_2)}{n}.$$

In other words, the simple procedure described gives an unbiased estimator of E(T) whose variance decreases as 1/n. If the procedure is repeated many times $(n \to \infty)$, then $\overline{T}$ becomes arbitrarily close to E(T) in accordance with a law of large numbers. Unfortunately, if there are many independent serial activities, then the sum of the variances in the numerator of (2.4) becomes larger, and a corresponding larger number of repetitions, n, is required in order to determine E(T) accurately.

We notice first that in order to estimate E(T), the realizations $T_1^i + T_2^i$ and $T_1^j + T_2^j$ need not be independent so long as they have the correct marginal distributions. Intuitively, too, one sees that if, when T_1 is "large" in one realization, it is forced to be correspondingly "small" in another, then the average will tend to be closer to the true value E(T) than in the case of purely independent samples. To achieve this effect, we can construct two realizations of $T = T_1 + T_2$ using the same two random numbers: first generate R_1 and R_2, then T_1 and T_2, and finally T; next make the transformation $R_k' = 1 - R_k$ (k=1,2) and from these transformed draws obtain T_1' and T_2', finally adding to get T'. Lastly average to obtain what we call the antithetic variate estimator,

$$\overline{T}_A = \frac{T_1^1 + T_1^{1'} + T_2^1 + T_2^{1'} + \cdots + T_1^n + T_1^{n'} + T_2^n + T_2^{n'}}{2n}$$

$$= \frac{1}{2}\ \frac{T_1^1 + T_2^1 + \cdots + T_2^n}{n} + \frac{T_1^{1'} + T_2^{1'} + \cdots + T_1^{n'} + T_2^{n'}}{n}$$

$$= \frac{1}{2}\ (\overline{T} + \overline{T}'). \qquad (2.5)$$

Now by construction, $E(\overline{T}) = E(\overline{T}')$, so the estimate $\overline{T}_A$ is unbiased.

Furthermore,

$$\mathrm{Var}(\overline{T}) = \mathrm{Var}(\overline{T}') = \frac{\mathrm{Var}(T_1) + \mathrm{Var}(T_2)}{n}. \qquad (2.6)$$

However, it is apparent that $\overline{T}$ and $\overline{T}'$ are negatively correlated, i.e., $\mathrm{Cov}(\overline{T}, \overline{T}') < 0$. Since

$$\mathrm{Var}(\overline{T}_A) = \frac{1}{4}\mathrm{Var}(\overline{T}) + \frac{1}{4}\mathrm{Var}(\overline{T}') + \frac{1}{2}\mathrm{Cov}(\overline{T}, \overline{T}') \qquad (2.7)$$

$$= \frac{\mathrm{Var}(\overline{T})}{2} + \frac{\mathrm{Cov}(\overline{T}, \overline{T}')}{2} < \frac{\mathrm{Var}(\overline{T})}{2},$$

this means that the above procedure is more efficient than <u>doubling</u> the total number of independent realizations computed. One can, of course, estimate the variance of $\overline{T}_A$ given specific distributions for T_1 and T_2 by simply computing the sample variance of the n independent averages $\frac{T_1^i + T_2^i}{2}$ for $i=1,2,\cdots,n$. Since $\overline{T}_A$ is the average of n independent terms, confidence limits may be placed on $E(T)$ using the Student t tables.

The significant impact of antithetic variates upon a sum is most apparent when the activity c.d.f.'s are symmetric, for example, if T_k can be assumed uniform, or normal. For Figure 1.1 in the former case, assume $T_1 \sim U(0, 2m_1)$ and $T_2 \sim U(0, 2m_2)$. The antithetic variate estimate T_A is then

$$= \frac{T_1 + T_1' + T_2 + T_2'}{2}$$

$$= \frac{1}{2N}\sum_{i=1}^{N} T_1^i + T_1^{i'} + T_2^i + T_2^{i'} \qquad (2.8)$$

where

$$T_1^i = 2m_1\ R_1^i\ ,$$

$$T_1^{i'} = 2m_1\ (1 - R_1^i),$$

$$T_2^i = 2m_2\ R_2^i,$$

$$T_2^{i'} = 2m_2\ (1 - R_2^i). \qquad (2.9)$$

It follows that

$$E(\overline{T}_A) = \frac{1}{2N}\sum_{i=1}^{N} 2m_1[R_1^i + 1 - R_1^i] + 2m_2[R_2^i + 1 - R_2^i]$$

$$= m_1 + m_2,$$

$$\mathrm{Var}(\overline{T}_A) = 0. \qquad (2.10)$$

Hence the antithetic variate method here yields a zero variance estimate of $E(T_1 + T_2)$; the result would be the same if we had used normal distributions for T_1 and T_2. (The zero-variance result does not, of course, generalize to other than the simple series networks.)

Although the value of antithetics is most pronounced when the underlying activity distributions are symmetric, the technique will result in better (lower variance for the same number of realizations) estimates of mean completion time (1.4) than those from crude Monte Carlo, regardless of the activity time distributions. As an illustration, let T_1 and T_2 of Figure 1.1 be identical and independent exponential distributions with means λ; T_1' and T_2' are the corresponding antithetic exponentials. For each realization, we draw two random numbers, R_1, $R_2 \sim U(0, 1)$, and then compute

$$T = T_1 + T_2 = -\lambda \log R_1 - \lambda \log R_2 \qquad (2.11)$$

and

$$T' = T_1' + T_2' = -\lambda \log(1 - R_1) - \lambda \log(1 - R_2).$$

Our antithetic estimate will then be the simple average

$$\overline{T}_A = \frac{1}{N} \sum_{\substack{\text{over } N\\ \text{realizations}}} \frac{(T + T')}{2}. \qquad (2.13)$$

Thus

$$E(\overline{T}_A) = 2\lambda,$$

$$\mathrm{Var}(\overline{T}_A) = \frac{1}{4}\mathrm{Var}(T) + \frac{1}{4}\mathrm{Var}(T') + \frac{1}{2}\mathrm{Cov}(T, T'), \qquad (2.14)$$

$$\mathrm{Var}(T') = \mathrm{Var}(T) = E(-\lambda \log R_1 - \lambda \log R_2 - 2\lambda)^2 \qquad (2.15)$$

$$= \lambda^2 E\Big((\log R_1)^2 + \log R_1 \log R_2 + 2 \log R_1 + \log R_1$$
$$\log R_1 + (\log R_2)^2 + 2 \log R_2 + 2 \log R_1 + 2 \log R_2$$
$$+ 4\Big) \qquad (2.16)$$

$$= \lambda^2(2 + 1 - 2 + 1 + 2 - 2 - 2 - 2 + 4) = 2\lambda^2,$$

$$\mathrm{Cov}(T, T') = E\Big([-\lambda \log R_1 - \lambda \log R_2 - 2\lambda] \qquad (2.17)$$
$$[-\lambda \log (1 - R_1) - \lambda \log (1 - R_2) - 2\lambda]\Big)$$

$$= \lambda^2 E\Big(\log R_1 \log (1 - R_1) + \log R_1 \log(1-R_2)$$
$$+ 2\log R_1$$
$$+ \log R_2 \log (1 - R_1) + \log R_2 \log (1 - R_2)$$
$$+ 2\log R_2$$
$$+ 2\log (1 - R_1) + 2\log (1 - R_2) + 4\Big). \qquad (2.18)$$

Noting that R_1 and R_2 are independent and that

$$E((\log R)\log(1-R)) = 2\ \frac{\pi^2}{6},^*$$

we obtain

$$\mathrm{Cov}(T, T') = \lambda^2\Big(2 - \frac{\pi^2}{6} + 1 - 2 + 1 + 2 - \frac{\pi^2}{6} - 2$$
$$- 2 - 2 + 4\Big)$$
$$= -1.290\lambda^2.$$

Hence substituting this in equation 2.14,

$$\mathrm{Var}(\overline{T}_A) = \frac{\lambda^2}{2} + \frac{\lambda^2}{2} - .645\lambda^2 = .355\lambda^2 << 2\lambda^2 = \mathrm{Var}(T)$$
$$= \mathrm{Var}(T').$$

*Bierens de Haan; Nouvelles Tables D'Integrals.

Here the antithetic variate procedure results in an estimate of the mean whose variance is less than 1/5 that obtained by crude Monte Carlo. In other words, we would have to take almost six times as many independent realizations in a straightforward estimation of the mean to obtain an estimate with the same variance as that of the antithetic variate estimate. We see that even when the activity c.d.f.'s are highly nonsymmetric, a substantial economy can be achieved with this simple technique.

So far we have concentrated upon antithetic variates as a procedure for estimating a parameter, the mean, of the project completion time T. The method of antithetic variates also yields better estimates of the entire function $F_T(t)$ than those obtained via straightforward simulation. If $\hat{F}_1(t)$ and $\hat{F}_2(t)$ are the empirical distribution functions calculated from several realizations of the two parallel antithetic processes, then

$$\hat{F}_A(t) = \frac{\hat{F}_1(t) + \hat{F}_2(t)}{2} \qquad (2.20)$$

will on the average be much "closer" to the actual completion time c.d.f., $F_T(t)$, than will either $\hat{F}_1$ or $\hat{F}_2$. It is intuitively clear that this averaging of c.d.f.'s will "smooth out" the resulting c.d.f.; to see this, we observe that if sampling randomness leads to realizations of Process 1 which are particularly low (i.e., lower than those obtained in case Process 1 were a perfect estimator of $F_T(t)$), then the realizations of Process 2 will be correspondingly high. The degree to which the method of antithetic variates provides better (than straightforward simulation) fit to the actual project completion distribution depends on the underlying activity c.d.f., and upon network configuration. As in estimation of the mean, the best results occur when activity c.d.f.'s are symmetrical.

It should be noted that estimates of Var(T) should not be formed by simple averaging of the sample variances of the antithetic variates; such a procedure would lead to a biased estimate. This undesirable property is common to all of the variance-reduction techniques presented in this paper. However, it appears likely that much of the bias in estimation of Var(T) may be eliminated by applying the Tukey-Quenoville jackknife procedure. The interested reader is referred to W. G. Cochran, Sampling Techniques, p. 180.

Stratification

Stratification is a sampling technique in which the range of the random numbers underlying each activity c.d.f. is broken into k disjoint and exhaustive intervals, (a_i, a_{i+1}), where $0 = a_1 < a_2 < \cdots < a_k = 1$. The division of the ranges need not be the same for all activities in the network; the best method would be to choose the intervals in such a way that the activity c.d.f.'s have equal variance over each of them. However, as a practical matter, it is simpler to split the range into equal parts: $0 < 1/k < 2/k < \cdots < (k-1)/k < 1$. Then k parallel simulations are run in such a manner that for each realization, a sample from each of the intervals is assigned to one of the k processes. The assignment of intervals for each activity to each process is random. An example will clarify the procedure.

By splitting the range of the uniform random variable underlying each activity time into three equal intervals, we will perform 3-way stratification on the network of Figure 1.1. Let us assume that the distributions for activities 1 and 2 are both rectangular with ranges $(0, 2m_1)$ and $(0, 2m_2)$, respectively. For each realization, we will draw two ordinary independent uniform random numbers, R_1 and R_2, and the completion time realizations for the three processes are

$$T^1 = 2m_1R_1 + 2m_2R_2 \qquad (3.1)$$

$$T^2 = 2m_1[(R_1 + \tfrac{1}{3})\text{mod } 1] + 2m_2[(R_2 + \tfrac{1}{3})\text{mod } 1]$$

$$T^3 = 2m_1[(R_1 + \tfrac{2}{3})\text{mod } 1] + 2m_2[(R_2 + \tfrac{2}{3})\text{mod } 1].$$

From equations 3.1 we see that at each realization the three parallel processes receive random times over different intervals of the job distributions. This stratification leads to a dependence between the three simulated processes. Intuitively, if one leads to a very high-valued realization of $T_1 + T_2$, another will be correspondingly low, and the third will be somewhere in between. Note, however, that the draws for each separate process are still independent draws from the uniform distribution and are not serially correlated. The proof of this property is simple. For example, the random number R used for the first activity time realization in process T^2 comes from

$$F_R(r) = P[(R_1 + \tfrac{1}{3})\text{mod } 1 \le r] \qquad (3.2)$$

$$= P[(R_1 + \tfrac{1}{3})\text{mod } 1 \le r | R_1 \epsilon (0,\tfrac{1}{3})] \times P[R_1 \epsilon (0,\tfrac{1}{3})]$$

$$+ P[(R_1 + \tfrac{1}{3})\text{mod } 1 \le r | R_1 \epsilon (\tfrac{1}{3},\tfrac{2}{3})] \times P[R_1 \epsilon (\tfrac{1}{3},\tfrac{2}{3})]$$

$$+ P[(R_1 + \tfrac{1}{3})\text{mod } 1 \le r | R_1 \epsilon (\tfrac{2}{3},1)] \times P[R_1 \epsilon (\tfrac{2}{3},1)]$$

$$= \begin{cases} 0 \times \frac{1}{3} + 0 \times \frac{1}{3} + 3r \times \frac{1}{3} = r & \text{for } 0 \le r < \frac{1}{3} \\ (3r - 1) \times \frac{1}{3} + 0 \times \frac{1}{3} + 1 \times \frac{1}{3} = r & \text{for } \frac{1}{3} \le r < \frac{2}{3} \\ 1 \times \frac{1}{3} + (3r - 2) \times \frac{1}{3} + 1 \times \frac{1}{3} = r & \text{for } \frac{2}{3} \le r < 1 \\ 1 \times \frac{1}{3} + 1 \times \frac{1}{3} + 1 \times \frac{1}{3} = 1 & \text{for } r \ge 1 \end{cases} \qquad (3.3)$$

$$= \begin{cases} r & \text{for } 0 \le r < 1 \\ 1 & \text{for } r \ge 1 \end{cases} \qquad (3.4)$$

which is the distribution function, U(0, 1).

The general procedure for stratifying would be to draw k separate (rather than just one) random numbers for each activity, $R_1, R_2, \cdots, R_k$, and place them in the different intervals:

$$\frac{R_1}{k}, \frac{R_2 + 1}{k}, \ldots, \frac{R_k + k - 1}{k}.$$

Or we might use antithetic variates within the k intervals; this latter procedure would lead to 2k parallel processes where for each realization the random draws would be of the form:

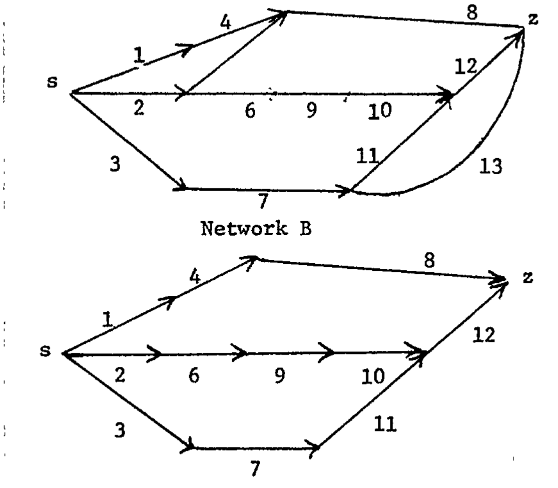
$$\frac{R_1}{k}, \frac{(1-R_1)}{k}, \frac{R_2 + 1}{k}, \frac{(1-R_2) + 1}{k}, \ldots, \frac{R_k + k - 1}{k}, \frac{(1-R_k) + k - 1}{k}.$$

The degree of detail (size of k) that should be used depends upon the purpose of the simulation. For a single experiment on a small network the extra programming required would take a disproportionate amount of time. But for doing a parametric analysis of a network, or for rerunning the simulation several times as activity status data is received, significant savings in computer time can be achieved by increasing k.

Realizations from each of the parallel processes of stratification are serially independent and hence, yield "legitimate" samples in the estimation of $F_T(t)$. The points obtained through the use of this technique again tend to cover, or fill out, this c.d.f. more evenly than those of crude Monte Carlo.

Control Variates

With antithetic variates or stratified simulation, one constructs parallel processes whose realizations are negatively correlated with one another. Thus, if one process results in an unusually high estimate, the other(s) will be correspondingly low, and their average will tend to be close to the actual value. The control variate procedure works in an opposite fashion. Suppose $F_T(t)$ is the (unknown) actual c.d.f. function of project completion time and T^i is the random sample representing the i-th realization from $F_T(t)$. We seek a control variate, CV^i, which has a high positive correlation with T^i, and whose c.d.f., $F_{CV}(t)$, is known exactly. An example will illustrate this method.



Network B

Control Network

Figure 4.1

The c.d.f. of the project completion time T for Network B is difficult to calculate analytically; however, the c.d.f. of the completion time CV for the Control Network may be found by applying the series-parallel reduction technique (Martin, 1965). For each realization, we use the same uniform random numbers to generate T^i and CV^i. By comparing the simulation results of the control variate with the known control variate, c.d.f., we may correct the analytical results for the known c.d.f., $F_{CV}(t)$, to obtain results for the actual network.

In estimating E(T) for Network B, we would take n realizations of T^i and CV^i, using the same set of random numbers for each. We would then compute

$$\overline{T} = \frac{1}{n}\sum_{i=1}^{n} T^i, \qquad \overline{CV} = \frac{1}{n}\sum_{i=1}^{n} CV^i \qquad (4.1)$$

and average the quantities

$$\hat{T}^i = T^i - CV^i + E(CV) \qquad (4.2)$$

to form the control variate estimate

$$\hat{\overline{T}} = \frac{1}{n}\sum_{i=1}^{n} (T^i - CV^i) + E(CV) = \overline{T} - \overline{CV} + E(CV), \qquad (4.3)$$

where E(CV) is the known mean of the c.d.f. $F_{CV}(t)$.

Clearly, $\hat{\overline{T}}$ is unbiased, since

$$E(\hat{\overline{T}}) = E(\overline{T} - \overline{CV} + E(CV)) = E(\overline{T}) = E(T), \qquad (4.4)$$

but $\hat{\overline{T}}$ often has a much smaller variance than does the straightforward estimate, $\overline{T}$. This may be seen as follows. We have

$$\mathrm{var}(\hat{\overline{T}}) = \mathrm{var}(\overline{T}) + \mathrm{var}(\overline{CV}) - 2\mathrm{cov}(\overline{T}, \overline{CV}) \qquad (4.5)$$

since the same random numbers are used to generate both $\overline{T}$ and $\overline{CV}$. The two networks closely approximate one another, so we may expect the correlation between $\overline{T}$ and $\overline{CV}$ to be close to 1, i.e.,

$$\rho(\overline{T}, \overline{CV}) = \frac{\mathrm{cov}(\overline{T}, \overline{CV})}{\sigma(\overline{T})\sigma(\overline{CV})} \approx 1, \qquad (4.6)$$

where the standard deviations $\sigma(\cdot)$ are also nearly the same, i.e.,

$$\sigma(\overline{T})\sigma(\overline{CV}) \approx \mathrm{var}(\overline{T}) \simeq \mathrm{var}(\overline{CV}). \qquad (4.7)$$

Using approximations (4.6) and (4.7) in (4.5), we have

$$\begin{aligned}\mathrm{var}(\hat{\overline{T}}) &\approx \mathrm{var}(\overline{T}) + \mathrm{var}(\overline{CV}) - \mathrm{var}(\overline{T}) - \mathrm{var}(\overline{CV}) \qquad (4.8)\\ &\approx 0.\end{aligned}$$

In a similar manner, we may use the known c.d.f. of the control variate, $F_{CV}(t)$, to correct for sampling fluctuations in estimating the c.d.f. $F_T(t)$. if $\hat{F}_{CV}(t)$ and $\hat{F}_T(t)$ are empirical estimates of the $F_{CV}(t)$ and $F_T(t)$, obtained by forming the cumulative frequency functions of several realizations of CV^i and T^i, then an improved estimate would be

$$\hat{\hat{F}}(t) = \hat{F}_T(t) - \hat{F}_{CV}(t) + F_{CV}(t). \qquad (4.9)$$

For a given value of t, if $\hat{F}_T(t)$ is much larger (smaller) than the actual c.d.f., $F_T(t)$, then $\hat{F}_{CV}(t)$ will be larger (smaller) than $F_{CV}(t)$. On the average, the corrected estimate, $\hat{F}(t)$, will be closer to $F_T(t)$.

The success of the control variate technique depends upon how closely the control variate, CV^i, mimics the realizations of T^i in the given network. The control network must be sufficiently simple, so that we can calculate $F_{CV}(t)$ exactly; however, we must be sure that realizations of the control network are highly correlated with those of the original network. The concept of "criticality" (Van Slyke, 1963) was mentioned in Section I. In brief, the criticality of any activity of a given network is the probability that it will lie on the critical path. (It should be intuitively clear that criticality is a function of activity time's variance, as well as its mean.) If an activity has a high criticality, it should be included in the control network. In constructing the Control Network B above, for example, it was implicitly assumed the probabilities of activities 5 and 13 being on the critical path were relatively small. If, however, activity 13 had a much higher criticality than activity 11, a better choice for the control network would be given in Figure 4.2:

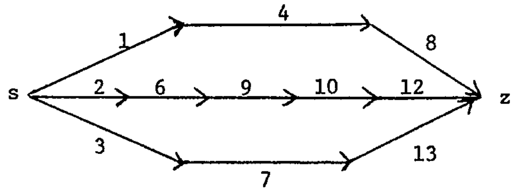
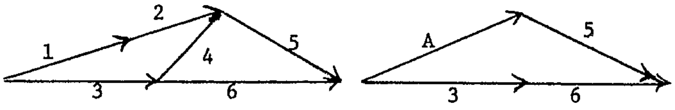


Figure 4.2

The control variate network need not be an exact subnetwork of the original configuration. In Network C of Figure 4.3, let us assume that activity time 3 has a relatively small variance. Then it seems plausible that a good control network can be formed by constructing activity A with c.d.f., $F_A(t) = P[\max(T_1 + T_2, T_4 + E(T_3)) \le t]$. If T_j^i is the i-th sample of activity j, then



Network C

Control for Network C

Figure 4.3

$T_A^i = \max[T_1^i + T_2^i, T_4^i + E(T_3)]$ may be used as the i-th sample for activity A of the control network.

Also, there is no necessity for restricting ourselves to a single control variate. If we construct two different control networks and label their critical path realizations $CV1^i$ and $CV2^i$, then a good mean estimate is

$$\overline{T} = w_1 \frac{1}{n} \sum_{i=1}^{n} (T^i - CV1^i) + E(CV1) + w_2 \frac{1}{n} \sum_{i=1}^{n} (T^i - CV2^i) + E(CV2). \tag{4.10}$$

The weights, w_1 and w_2, could be set at 1/2 or computed by the regression technique discussed in the next section. A similar procedure could be used for estimating the unknown critical path distribution function.

It should be noted that antithetic variates and stratification could be used in conjunction with the control variate procedure. This would be accomplished by antitheticizing or stratifying within each of the two processes of control variates. For instance, the application of antithetic variates would lead to four parallel processes; for each realization, a T^i, T'^i, CV^i, and CV'^i would be computed. A low variance mean estimate, $\hat{\overline{T}}$, would then be

$$\hat{\overline{T}} = \frac{1}{N} \sum_{i=1}^{N} \overline{T}^i \tag{4.11}$$

where

$$\overline{T}^i = \frac{T^i + T'^i}{2} - \frac{CV^i + CV'^i}{2} + E(CV). \tag{4.12}$$

Regression

Let us suppose that we have run N experiments employing antithetic variates to obtain estimates of some parameter of the critical path, $T_A^1, T_A^2, \cdots, T_A^N$. Using the same set of underlying random numbers, we have also computed N estimates of this parameter using stratification $T_S^1, T_S^2, \cdots, T_S^N$ and the same for control variates $T_C^1, T_C^2, \cdots, T_C^N$. The question arises as to how to combine these 3N estimates into a single "best" estimate, $\hat{T}$, of the given parameter. A reasonable procedure would be to take their simple average

$$\overline{T} = \frac{1}{3} \overline{T}_A + \frac{1}{3} \overline{T}_S + \frac{1}{3} \overline{T}_C = \frac{1}{3} \frac{1}{N} \sum_{i=1}^{N} (T_A^i + T_S^i + T_C^i). \tag{5.1}$$

Instead of weighting the results of the three techniques equally, a better estimate can be found by computing weights c_a, c_s, c_c, which yield the lowest variance estimator,

$$\hat{\overline{T}} = c_a \overline{T}_A + c_s \overline{T}_S + c_c \overline{T}_C , \tag{5.2}$$

where the three coefficients are found by minimizing the variance of the identically distributed

$$\hat{T}^i = c_a T_A^i + c_s T_S^i + c_c T_C^i \tag{5.3}$$

subject to the constraint,

$$c_a + c_s + c_c = 1. \tag{5.4}$$

For simplicity, let $V_j = \mathrm{var}(T_j^i)$ and $C_{jk} = \mathrm{cov}(T_j^i, T_k^i)$; then

$$\mathrm{var}(T^i) = c_a^2 V_A + c_s^2 V_S + c_c^2 V_C + 2c_a c_s C_{AS} + 2c_a c_c C_{AC} + 2c_s c_c C_{SC}. \tag{5.5}$$

Forming the LaGrangian by adding the constraint and differentiating leads to a set of simultaneous equations whose solution is

$$c_s = \frac{C_{AC}C_{SC} + C_{AS}V_{AC} - C_{AC}^2 + V_A V_C - V_C C_{AS}}{K} \tag{5.6}$$

where

$$K = V_A V_S + V_A V_C + V_S V_C - C_{AS}^2 - C_{AC}^2 - C_{SC}^2 - 2(V_A C_{SC} + V_S C_{AC} + V_C C_{AS} - C_{AS}C_{AC} - C_{AS}C_{SC} - C_{AC}C_{SC}). \tag{5.7}$$

By symmetry, the coefficients c_a and c_c may be found. Hence the best linear unbiased estimate of the parameter, provided we know the variances V_j and covariances C_{jk}, is given by (5.2). Since this proviso can seldom be met in practice, the best we can do is to approximate V_j and C_{jk} using sample variances and covariances throughout (5.5)-(5.7). In essence, this is a "bootstrap" technique since we combine a number of estimates with coefficients based upon the estimates themselves. The coefficients are chosen in such a manner that the resulting combination, $\hat{T}^i$, will have as small a variance as possible, given our state of knowledge regarding the sampling distributions. If one particular estimator is very "stable," (low variance), or tends to counteract the fluctuations in the other estimators (large negative covariances), it will receive a relatively large weighting.

Conditioned Sampling

In the introductory section, it was noted that analytic methods assume simple activity completion c.d.f.'s and simple networks to evaluate the integral (1.3). In a network possessing various simple subnetworks, we then might reduce these subnetworks via series-parallel reduction, and derive an equivalent network for which estimations may be completed by Monte Carlo methods. This has the effect of diminishing the number of random samples required to effect a realization of the project completion time T. There is, however, another way of doing this, without using series-parallel reduction, which we will call conditioned sampling.

To illustrate the procedure, let us suppose that we wish to estimate the c.d.f. $F_T(t)$ for the critical path in the network of Figure 6.1.

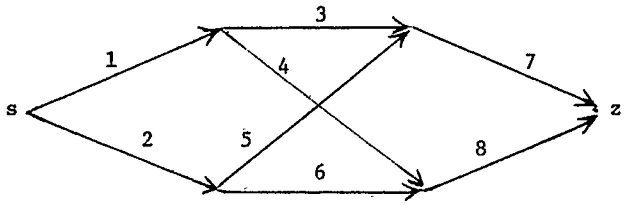


Figure 6.1

(We note first that the network in Figure 6.1 may not be further simplified by series-parallel reduction, being a "multiply-crossed" (Hartley and Wortham, 1966) network.)

The crude Monte Carlo approach would be to take samples $T_1^i, T_2^i, \cdots, T_8^i$, $i=1,2,\cdots,n$, and form the estimate

$$\hat{F}_T(t) = \frac{1}{n} \sum_{i=1}^{n} G_i(t) \qquad (6.1)$$

where

$$G_i(t) = \begin{cases} 1 & \text{if } \max[X_1^i, X_2^i, X_3^i, X_4^i] < t, \text{ where} \qquad (6.2) \\ & X_1^i = T_1^i + T_3^i + T_8^i, \quad X_2^i = T_1^i + T_4^i + T_8^i, \\ & X_3^i = T_2 + T_5 + T_8, \quad X_4^i = T_2^i + T_6^i + T_8^i \\ 0 & \text{otherwise} \end{cases}$$

In this case, eight samples per realization are necessary.

Restating (1.2) for the network of Figure 6.1, we have

$$\begin{aligned} T &= \max[X_1, X_2, X_3, X_4] \qquad (6.3) \\ &= \max[T_1+T_3+T_8,\ T_1+T_4+T_8,\ T_2+T_5+T_8,\ T_2+T_6+T_8]. \end{aligned}$$

Now the only apparent reason that the c.d.f. of T cannot be expressed as a product of the c.d.f.'s of X_1, X_2, X_3, X_4 is that the latter are not independent; otherwise this would be possible. On the other hand, if T_1, T_2, T_7, T_8 were constants rather than random variables, X_1, X_2, X_3, X_4 would be independent. Let us therefore proceed as if such were the case; then we would have

$$\begin{aligned} F_T(t) = {} & F_3(t - T_1 - T_7)F_4(t - T_1 - T_8)F_5(t - T_2 - T_7) \\ & F_6(t - T_2 - T_8). \qquad (6.4) \end{aligned}$$

The manner in which we may set T_1, T_2, T_7, T_8 to "constants" is simply by sampling them, i.e., fixing the random variables at some sample values.

Expression (6.4) is then said to be "conditioned" on the fact that T_1, T_2, T_7, T_8 have taken on these sample values. We can now form the conditional sampling estimate

$$\hat{\hat{F}}_T(t) = \frac{1}{n} \sum_{i=1}^{n} H_i(t) \qquad (6.5)$$

where

$$H_i(t) = \begin{cases} F_3(t-T_1^i-T_7^i)F_4(t-T_1^i-T_8^i)F_5(t-T_2^i-T_7^i)F_6(t-T_2^i-T_8^i) \\ \text{if } \max[T_1^i+T_7^i,\ T_1^i+T_8^i,\ T_2^i+T_7^i,\ T_2^i+T_8^i] < t \\ 0 \quad \text{otherwise} \qquad (6.6) \end{cases}$$

When using the conditioned sampling estimate (6.5), we need take only four samples per realization, that is, half as many as demanded by the crude Monte Carlo estimate of (6.1). Furthermore, it would appear that the "variation" in $\hat{\hat{F}}_T(t)$ from the true $F_T(t)$ is less than for $\hat{F}_T(t)$, since in each realization we are using information on the entire c.d.f.'s for T_3, T_4, T_5, T_6, rather than only the information contained in single samples of these. The sole price to be paid for these gains is the multiplication of $F_3(\cdot)$, $F_4(\cdot)$, $F_5(\cdot)$, and $F_6(\cdot)$ in (6.6).

The general outline of the method is then as follows: first, perform all series-parallel reductions that are to be done on a given network, deriving an equivalent network; second, examine all paths from s to z in the equivalent network and select one activity per path which is unique to that path (if such exists); third, form the appropriate product distribution as in (6.4) and "condition" it upon the remaining activities; lastly, generate realizations by repeatedly sampling these unselected activities, using the product form estimate as in (6.6). The method will reduce the required number of samples per realization by a number somewhat less than the total number of paths in the network, which may be an important savings in effort.

Conclusion

The choice of analytic, approximation, or Monte Carlo methods (and combinations thereof) as means of solving the stochastic PERT problem depends on the activity c.d.f.'s, network configuration, computing resources available, and desired accuracy. When Monte Carlo methods are indicated, there exist several techniques for improving accuracy and diminishing computational effort. In this paper we have described the application of five such techniques: antithetic variates, stratified sampling, control variates, regression, and conditioned sampling. These techniques may be used separately or in combinations; their existence makes Monte Carlo methods much more attractive in the investigation of stochastic PERT networks.

References

Charnes, A., Cooper, W. W. and Thompson, G. L., "Critical Path Analyses via Chance Constrained and Stochastic Programming," Operations Research, 12, May-June 1964, pp. 460-470.

Gaver, D. P. and Burt J. M., "Simple Stochastic Networks: Some Problems and Procedures," Management Sciences Research Report No. 142, Graduate School of Industrial Administration, Carnegie-Mellon University.

Hammersley, J. and Handscomb, D. C., Monte Carlo Methods, London, Methuen and Co., Ltd., 1967.

Hartley, H. O. and Wortham, A. W., "A Statistical Theory for PERT Critical Path Analysis," Manage-Science, 12, No. 10, June 1966, pp. 469-481.

Jewell, W., "Risk Taking in Critical Path Analysis," Management Science, 11, No. 3, January 1965, pp. 437-443.

Kelley, J. E., "Critical Path Planning and Scheduling: Mathematical Basis," Operations Research, 9, 1961, pp. 296-320.

Kleindorfer, G. B., "Bounding Distributions for Stochastic Acyclic Networks," Working Paper, School of Education, University of California,. Berkeley, 1969.

Martin, J. J., "Distribution of the Time Through a Directed Acyclic Network," Operations Research, 13, 1965, pp. 46-66.

Shrieder, Y. A. (ed.), The Monte Carlo Method, Oxford, Pergamon Press, 1966.

Van Slyke, R. M., "Monte Carlo Methods and the PERT Problem," Operations Research, 11, 1963, pp. 839-860.