Probability in the Engineering and Informational Sciences, **13**, 1999, 209–220. Printed in the U.S.A.

AVERAGE RUN LENGTHS FOR MOVING AVERAGE CONTROL CHARTS

SHELDON M. ROSS*
Department of Industrial Engineering and Operations Research
University of California
Berkeley, California 94720

We are interested in $E[N]$, the mean time until the most recent k values of a sequence of independent and identically distributed random variables exceeds a specified constant. Using recent results, we present a simulation procedure for determining $E[N]$. These results are also used to obtain upper and lower bounds for $E[N]$. These bounds, however, are in terms of a quantity ω that is not easily calculated. A recursive procedure for evaluating ω when the data distribution is Bernoulli is given. Efficient simulation procedures for estimating ω in the cases of normal and exponential population distributions are also presented, as is a Markov chain monte carlo procedure when the distribution is general.

1. INTRODUCTION

Let $X_1, X_2, \ldots$ be a sequence of independent random variables having a common distribution function F. For a fixed positive integer k and constant c, let

$$M_n = \sum_{i=n-k+1}^{n} X_i, \qquad n \geq k$$

and define

$$N = \min\{n \geq k : M_n \geq c\}$$

and

$$\alpha = P\{M_n \geq c\}.$$

We are interested in bounding and approximating $E[N]$ in situations where α is small.

*This research was supported by National Science Foundation grant DMI-9610046 with the University of California.

© 1999 Cambridge University Press 0269-9648/99 $12.50

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

To motivate the preceding, imagine that X_i represents the measurable value of the ith item produced by a certain process. Suppose that when the process is in control, these items tend to have small values; furthermore, suppose that the producer is employing a moving average control rule that declares the process to be out of control whenever the average of the most recent k values is at least c/k. In this scenario, $E[N]$ represents the mean number of items that will be produced before the process is declared out of control when the underlying distribution is F (see Crowder [1] and Roberts [4]). Its computation for various distributions F is thus important in enabling us to evaluate the control rule.

A natural approach to computing $E[N]$ is to regard it as a mean first passage time of the Markov chain that continually keeps track of the most recent $k-1$ values of the X's until the moving sum is at least c. However, as this results in a Markov chain with a $k-1$ dimensional state space, its analysis is only computationally feasible for very small values of k. An approach to approximate $E[M]$ by a straightforward simulation also leads to difficulties. Since $\{M_n, n \geq k\}$ is a moving sum process of the k most recent values, it follows that when α is small $N-(k-1)$ will approximately have a geometric distribution. As a result, $\mathrm{Var}(N) \approx (E[N])^2$, which implies that the use of a straightforward simulation approach to estimate $E[N]$ will be inefficient, both because each run will require an average of $E[N]$ random variables and because a large number of runs will be needed to obtain a precise estimator.

In Section 2, we presume that the process is continually simulated until the moving sum exceeds c and show how to use results from Ross [7] to obtain a simulation estimator of $E[N]$. In Section 3, we use another result of [7] to obtain reasonably precise, though not easily computable, bounds for $E[N]$.

A recursive method of determining these bounds when F is a Bernoulli distribution is given in Section 4. In general, however, a simulation is needed, and, in Section 5, we present a simulation procedure that can be utilized for certain distributions to approximate efficiently these bounds. In subsections 5.1 and 5.2, we utilize this procedure when F is a normal and an exponential distribution; in subsection 5.3, we consider the case of an arbitrary distribution and present a Markov chain monte carlo technique. In Section 6, we consider two-sided control charts where one is interested in the mean time until M_n is either sufficiently large or sufficiently small.

2. SIMULATING $E[N]$

Suppose that N data values are simulated, and suppose that

$$(X_{N-k+2}, \ldots, X_N) = (y_2, \ldots, y_k).$$

Letting $W_1, \ldots, W_{k-1}$ be independent random variables having distribution F that is independent of the X_i, it follows from results of [7] that

$$\mathcal{E} \equiv \frac{1 + \sum_{i=2}^{k} P\left\{\sum_{j=1}^{i-1} W_j \geq c - \sum_{j=i}^{k} y_j\right\}}{\alpha}$$

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

is the value of an unbiased estimator of $E[N]$. Thus, each simulation run produces two unbiased estimators N and $\mathcal{E}$. If we perform r simulation runs, then we can use an estimator of the type

$$a\bar{N} + (1-a)\bar{\mathcal{E}},$$

where the best value of a can be determined by the simulation (see Ross [5, pp. 148–149]). We would expect in most cases that the optimal a value is small.

Remark: If the value of $\mathcal{E}$ cannot be easily computed because of the difficulties of calculating some of the probabilities, then these probabilities can themselves be estimated by a simulation.

3. THE BOUNDS

Let A_n be the event that $M_n \geq c$. It follows from results given in [7] that

$$\frac{1-(k-1)\alpha}{\omega} \leq E[N] \leq \frac{1}{\omega}, \tag{1}$$

where

$$\omega = P(A_k A_{k+1}^c \cdots A_{2k-1}^c).$$

In the following sections we show how ω can be efficiently obtained when F is Bernoulli, normal, or exponential. (The importance of ω was first noted in Hwang and Trietsch [3], where $1/\omega$ was suggested as an approximation for $E[N]$.)

4. EVALUATING THE BOUNDS FOR A BERNOULLI DISTRIBUTION

Let $X_1, X_2, \ldots$ be Bernoulli random variables with parameter p; that is, $P\{X_i = 1\} = p, P\{X_i = 0\} = q = 1-p$. Assume that c is a positive integer, so that N is the first time that there is a run of k values whose sum is at least c. We now develop a recursive scheme to evaluate the probability ω:

$$\begin{aligned}
\omega &= P\left\{\sum_{i=1}^{k} X_i \geq c, \qquad \sum_{i=j}^{k} X_i + \sum_{i=1}^{j-1} X_{k+i} < c, \quad \text{for all } j = 2,\ldots,k\right\} \\
&= P\left\{\sum_{i=1}^{k} X_i = c, \qquad \sum_{i=j}^{k} X_i + \sum_{i=1}^{j-1} X_{k+i} < c, \quad \text{for all } j = 2,\ldots,k\right\} \\
&= P\left\{\sum_{i=1}^{k} X_i = c, \qquad \sum_{i=1}^{j-1} X_i > \sum_{i=1}^{j-1} X_{k+i}, \quad \text{for all } j = 2,\ldots,k\right\} \\
&= \binom{k-1}{c-1} p^c q^{k-c+1} \\
&\quad \times P\left\{\sum_{i=2}^{j-1} X_i \geq \sum_{i=2}^{j-1} X_{k+i}, \qquad j = 3,\ldots,k \,\middle|\, \sum_{i=1}^{k} X_i = c, X_1 = 1, X_{k+1} = 0\right\}.
\end{aligned}$$

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

That is,

$$\omega = \binom{k-1}{c-1} p^c q^{k-c+1} P\left\{ \sum_{i=2}^{j-1} X_i \geq \sum_{i=2}^{j-1} X_{k+i}, \qquad j = 3, \ldots, k \,\middle|\, \sum_{i=2}^{k} X_i = c-1 \right\}. \tag{2}$$

To evaluate ω, consider two urns: Urn A contains s balls, of which t are worth 1 point and the other $s - t$ are worth 0 points; whereas each ball selected from urn B, independent of previous selections, has value 1 with probability p and value 0 otherwise. Suppose that each round consists of randomly choosing a ball from each urn, noting their values and then discarding the chosen balls. Urn A's score after each round is the sum of the values of all its balls chosen by that time, and similarly for urn B's score. If we let $P(s, t, i)$ denote the probability that urn A is never behind by more than i points at any time during the first $s - 1$ rounds, then it follows from the preceding set of equations that

$$\omega = \binom{k-1}{c-1} p^c q^{k-c+1} P(k-1, c-1, 0). \tag{3}$$

Conditioning on the outcome of the first round gives the following recursion:

$$P(s,t,i) = \frac{t}{s} pP(s-1, t-1, i) + \frac{s-t}{s} qP(s-1, t, i) + \frac{t}{s} qP(s-1, t-1, i+1)$$
$$+ \frac{s-t}{s} pP(s-1, t, i-1), \qquad i \geq 0. \tag{4}$$

Using the boundary condition

$$P(s,t,i) = 0, \qquad i < 0$$
$$P(1,t,i) = 1, \qquad i \geq 0,\ t = 0,1$$

now enables us numerically to compute $P(k-1, c-1, 0)$ and, using Eq. (3), ω.

Remark: In the Bernoulli case, we are sometimes interested in $E[T]$, where

$$T = \min\{j \geq c : M_j \geq c\}.$$

That is, T is the first time that there have been c data values equal to 1 within a run of k or fewer consecutive values. Noting that

$$N = \max(T, k),$$

we can determine the relation between $E[T]$ and $E[N]$ by conditioning on S, the first time that the sum of all data values is equal to c.

$$E[N - T] = \sum_{i=c}^{k-1} E[N - T | S = i] P\{S = i\}$$
$$= \sum_{i=c}^{k-1} (k-i) \binom{i-1}{c-1} p^c q^{i-c}.$$

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

That is,

$$E[T] = E[N] - \sum_{i=c}^{k-1} (k-i)\binom{i-1}{c-1} p^c q^{i-c}. \tag{5}$$

When F is not Bernoulli, a simulation is needed to approximate ω. Our general simulation approach is indicated in the next section.

5. A SIMULATION APPROACH FOR EVALUATING ω

Recall that with A_n defined as the event that $M_n \geq c$, we have

$$\begin{aligned} \omega &= P(A_k A_{k+1}^c \cdots A_{2k-1}^c) \\ &= P(A_k) P(A_{k+1}^c \cdots A_{2k-1}^c | A_k) \\ &= \alpha\beta \end{aligned}$$

where

$$\beta = P(A_{k+1}^c \cdots A_{2k-1}^c | A_k).$$

Assuming that α can be computed, we propose to use simulation to estimate β. Our approach will be first to generate the values of $X_2, \ldots, X_k$ conditional on the event that $\sum_{i=1}^k X_i \geq c$. If $X_{k+1}, \ldots, X_{2k-1}$ are independent random variables having distribution F, then $P(D|X_2, \ldots, X_k)$ is an unbiased estimator of β, where

$$D = \left\{ \sum_{i=j}^{k} X_i + \sum_{i=1}^{j-1} X_{k+i} < c, \quad \text{for all } j = 2, \ldots, k \right\}.$$

As the preceding probability cannot be directly computed, we need to continue with our simulation. Now, whereas we could generate the values of $X_{k+1}, \ldots, X_{2k-1}$, and then use the indicator function of D as the estimator, we instead propose to use the conditional probability of D, given all the variables except X_{k+1} as our estimator. Whereas such a conditional expectation estimator will always have a smaller variance than the indicator function, the key role of X_{k+1} leads us to believe that the reduction in variance might be substantial.

We thus propose the following approach to estimating β.

1. Generate $X_2, \ldots, X_k$ conditional on the event that $\sum_{i=1}^k X_i \geq c$, and let x_i, $i = 2, \ldots, k$ be these generated values.
2. Generate independent random variables $X_{k+2}, \ldots, X_{2k-1}$ having distribution F, and let $y_{k+i}, i = 2, \ldots, k-1$ be the generated values.
3. Let $m = \max_{j=2,\ldots,k} (\sum_{i=j}^k x_i + \sum_{i=2}^{j-1} x_{k+i})$.
4. Obtain the estimator $H(c-m) = \lim_{h \to 0} F(c-m-h)$.

The preceding should be repeated many times and the average of the values of $H(c-m)$ obtained, call it $\hat{\beta}$, is the estimator of β.

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

Remark: The preceding assumes that $H(x)$ can be efficiently computed. If this is not the case, then in Step 2 we should also generate the value of X_{k+1}, say it is x_{k+1}, and then use the indicator of the event $\{x_{k+1} + m < c\}$ as the estimator of β for that iteration.

In order to utilize the preceding approach to estimate β, and thus obtain bounds for $E[N]$, it is necessary to be able to generate $X_2, \ldots, X_k$ conditional on the event that $\sum_{i=1}^{k} X_i \geq c$. In Sections 5 and 6 we show how this can be accomplished for the normal and the exponential distributions F. For the normal, we first generate the random variable $\sum_{i=1}^{k} X_i$ conditional on its value exceeding c, and then generate X_i, $i = 2, \ldots, k$ conditional on the generated value of the sum; for the exponential, we directly generate $X_i, i = 2, \ldots, k$ conditional on the event that $\sum_{i=1}^{k} X_i \geq c$, without first generating the value of the sum.

5.1. The Case of a Normal Distribution

If F is a normal distribution with mean μ and standard deviation σ, then by considering the random variables $(X_i - \mu)/\sigma$ the problem reduces to one in which the distribution is a standard normal. So suppose that F is the standard normal distribution. Since M_k will then be normal with mean 0 and variance k, our first step in the simulation is to generate a standard normal random variable $Z = M_k/\sqrt{k}$ conditional on the event that Z exceeds $b = c/\sqrt{k}$, when $\alpha = P\{Z \geq b\}$ is small. That is, we want to generate a random variable having density function

$$f(x) = \frac{1}{\bar{\Phi}(b)\sqrt{2\pi}}\, e^{-x^2/2}, \qquad x \geq b,$$

where $\bar{\Phi}(x) = P\{Z \geq x\}$ is the standard normal tail distribution function. To accomplish this, we will use the acceptance–rejection algorithm (see [5]) with an exponential random variable that is conditioned to be larger than b; that is, its density function is

$$g(x) = \lambda e^{-\lambda(x-b)}, \qquad x \geq b.$$

Since

$$\frac{f(x)}{g(x)} = \frac{e^{-\lambda b} e^{\lambda x - x^2/2}}{\lambda \bar{\Phi}(b)\sqrt{2\pi}}$$

it easily follows that

$$C(\lambda) \equiv \max_{x \geq b} \frac{f(x)}{g(x)} = \frac{e^{\lambda^2/2 - \lambda b}}{\lambda \bar{\Phi}(b)\sqrt{2\pi}}. \tag{6}$$

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

Simple calculus now shows that $C(\lambda)$ is minimized when

$$\lambda = \frac{b + \sqrt{b^2 + 4}}{2} \tag{7}$$

with λ taken as equal to the value specified in Eq. (6).

The acceptance–rejection algorithm for generating a random variable distributed according to f by using ones generated according to g, is as follows:

1. Generate a random number U_1 and set $X = b - \frac{1}{\lambda}\log(U_1)$.
2. Generate a random number U_2.
3. If $U_2 \leq \exp\{-X^2/2 + \lambda X - \lambda^2/2\}$, stop; otherwise, return to 1.

It should be noted that the preceding used that $-(1/\lambda)\log(U)$ is exponential with rate λ, and also that the conditional distribution of an exponential conditioned to exceed a constant is distributed as the constant plus the (unconditional) exponential. The preceding can be made more efficient by noting that the inequality in 3 can be written as

$$-\log(U_2) \geq (X - \lambda)^2/2,$$

and then using (as in [5, pp. 70–71]) the fact that if an exponential exceeds an independent random variable Y, then the amount by which it exceeds Y is exponential and is independent of Y. Using this, we can rewrite the preceding algorithm as follows:

1. Generate W and Y, independent exponential random variables with rate 1.
2. Set $X = b + W/\lambda$.
3. Set $E = Y - (X - \lambda)^2/2$.
4. If $E \geq 0$, stop; otherwise, return to 1.

The values of X and E obtained from the algorithm are that of independent random variables, with X being distributed according to f and E being exponential with rate 1. Thus not only do we obtain by this algorithm a standard normal random variable conditioned to exceed b, but also an independent exponential random variable which can then be used for the next generation of a conditioned standard normal. The number of iterations needed by the algorithm is a geometric random variable with mean $C(\lambda)$ given by Eqs. (6) and (7). For values of b that are of interest, $C(\lambda)$ is quite small. For instance, if $b = 3$, then (to two decimal places) $\lambda = 3.3$ and $C(\lambda) = 1.04$. (Thus, the simulation of a sequence of independent standard normals that are all conditioned to exceed 3 requires that for each one we generate an average of 1.08 exponentials with rate 1 and perform an average of 2.08 multiplications.) Also, since

$$C(\lambda) \leq C(b) = \frac{e^{-b^2/2}}{b\bar{\Phi}(b)\sqrt{2\pi}},$$

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

it follows from the inequality (see Ross [6])

$$\bar{\Phi}(b) > \frac{e^{-b^2/2}}{\sqrt{2\pi}} \frac{\sqrt{b^2+4}-b}{2}$$

that

$$C(\lambda) < \frac{2}{b(\sqrt{b^2+4}-b)},$$

which again indicates the efficiency of the algorithm. (The value of the right-hand side of the preceding is, for instance, 1.207 when $b = 2$, 1.101 when $b = 3$, and 1.059 when $b = 4$.)

Remark: If we only wanted to generate a single standard normal conditioned to exceed the positive constant b, then another algorithm to consider is acceptance–rejection using

$$g(x) = e^{b^2/2} x e^{-x^2/2}, \qquad x \geq b.$$

Noting that the distribution function corresponding to this density is

$$G(x) = 1 - e^{-(x^2-b^2)/2}, \qquad x \geq b,$$

it follows from the inverse transform method [5] that we can generate a random variable X having density g by generating a uniform (0,1) random variable U and then using $X = \sqrt{b^2 - 2\log(1-U)}$. In fact, since $1 - U$ is also uniform over (0,1), a slight savings is obtained by replacing $1 - U$ by U in the preceding. Also, since for $x \geq b$

$$\frac{f(x)}{g(x)} = \frac{e^{-b^2/2}}{\bar{\Phi}(b)\sqrt{2\pi}x},$$

it follows that

$$C \equiv \max_{x \geq b} \frac{f(x)}{g(x)} = \frac{e^{-b^2/2}}{\bar{\Phi}(b)\sqrt{2\pi}b}.$$

Hence, using the acceptance–rejection technique, we can generate Z conditional on $Z > b$ as follows:

1. Generate random numbers U_1 and U_2.
2. Set $X = \sqrt{b^2 - 2\log(U_1)}$.
3. If $U_2 \leq b/X$, stop; otherwise, return to Step 1.

The value of X is then that of a standard normal conditioned to exceed b, and the number of iterations of Step 1 that are needed is a geometric random variable with mean C. Although not as efficient as the previous algorithm presented when a sequence of conditioned standard normals is required, it might be competitive when

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

only a single one is needed. For example, when $b = 3$, it requires on average 1.094 iterations, with each iteration needing 2 random numbers, and 1 log, 1 square root, and 1 multiplication calculation.

Let us now suppose that we have simulated the value of $M_k = \sum_{i=1}^{k} X_i$ conditional on the event that it is larger than c; let its simulated value be y. We now need to generate $X_2, \ldots, X_k$ conditional on the event that $M_k = y$. To accomplish this, we make use of the following lemma, whose proof is straightforward.

LEMMA 1: *If $Z_1, \ldots, Z_n$ are independent standard normal random variables, then the conditional distribution of Z_1 given that $\sum_{i=1}^{n} Z_i = a$ is the normal distribution with mean a/n and variance $(n-1)/n$.*

Using the lemma, we can generate $X_2, \ldots, X_k$ in sequence, as follows. First generate X_2 by generating a normal random variable with mean y/k and variance $(k-1)/k$. Now, suppose that for $2 \le i < k$, $X_2, \ldots, X_i$ have been generated and that their values are $x_2, \ldots, x_i$. Then, generate X_{i+1} by generating the value of a normal random variable with mean $(y - \sum_{j=2}^{i} x_j)/(k - i + 1)$ and variance $(k - i)/(k - i + 1)$. Continue on until all of $X_2, \ldots, X_k$ have been generated. Then follow the remaining steps outlined in Section 5 to complete the simulation.

Suppose that F is the standard normal distribution and $c = 3\sqrt{k}$. As N is the first time that a moving sum of k normal random variables differs from its mean by more than 3 times its standard deviation, it follows in this case that $\alpha = 0.00135$. The following table gives simulated estimates of β, and the resulting estimates of the lower bounds $LB = (1 - (k-1)\alpha)/\omega$ and the upper bounds $UB = 1/\omega = 1/(\alpha\beta)$, for values of k between 3 and 8.

k	$\hat{\beta}$	$\hat{LB}$	$\hat{UB}$
3	0.849012	870.12	872.47
4	0.765767	963.40	967.32
5	0.695602	1059.14	1064.89
6	0.636687	1155.58	1163.43
7	0.586904	1251.89	1262.12
8	0.544332	1347.97	1360.83

The preceding table indicates the precision of the bounds. For instance, we see that for $k = 5$, the estimate of 1062.02 is within 2.87 of $E[N]$. (Indeed, a simulation gave that $E[N] \approx 1062.27$.)

5.2. The Case of an Exponential Distribution

Suppose now that F is an exponential distribution with rate 1. That is,

$$F(m) = 1 - e^{-m} \quad \text{if } m \ge 0.$$

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

(If F is exponential with rate μ, then μX is exponential with rate 1 and so we can, without loss of generality, assume that the exponential rate is 1.) Let $Y_1,\ldots,Y_k$ be exponential with rate 1, and suppose that c is such that α is small, where

$$\alpha = P\left\{\sum_{i=1}^{k} Y_i > c\right\}.$$

To simulate the Y_i conditional on their sum exceeding c, interpret them as the interarrival times of a Poisson process $\{N(t)\}$ having rate 1. Since $\sum_{i=1}^{k} Y_i > c$ is equivalent to the event that $N(c) \leq k-1$, we start by simulating $N(c)$, a Poisson random variable with mean c, conditional on it being less than or equal to $k-1$. To accomplish this, starting with $j = k-1$, recursively compute

$$a_j = e^{-c}c^j/j!, \qquad j = k-1,\ldots,0,$$

and then set

$$p_j = a_j \Big/ \sum_{j=0}^{k-1} a_j, \qquad j = k-1,\ldots,0.$$

To simulate $N(c)$ conditional on it being less than or equal to $k-1$, use the discrete inverse transform algorithm starting with $k-1$. That is, use the following algorithm:

1. $j = k-1$
2. $v = p_j$
3. Generate a random number U
4. If $U \leq v$, set $N(c) = j$ and stop
5. $j = j-1$
6. $v = v + p_j$
7. Go to 4.

Suppose that the generated value of $N(c)$ is $N(c) = j$. Now generate random numbers $U_1,\ldots,U_{k-1}$. Order the values $U_1,\ldots,U_j$ and let the ordered values be $U_{(i)}, i = 1,\ldots,j$. Then, let $Y_1,\ldots,Y_{k-1}$ be as follows:

$$\sum_{i=1}^{r} Y_i = cU_{(r)}, \qquad r = 1,\ldots,j$$

$$\sum_{i=1}^{j+1} Y_i = c - \log(U_{j+1})$$

$$\sum_{i=1}^{r} Y_i = \sum_{i=1}^{r-1} Y_i - \log(U_r), \qquad r = j+2,\ldots,k-1.$$

The validity of the preceding is based on the result that, given the number of events of a Poisson process by time c, the unordered event times are independent uniform $(0,c)$ random variables.

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

Remark: Rather than generating and then ordering $U_1, \ldots, U_j$ in the preceding, we could make use of the result that, conditional on the $(j+1)$st event of a Poisson process occurring at time t, the times of the first j events are distributed as the ordered values of a set of j uniform random variables on $(0, t)$. Thus, after we had obtained the generated value $N(c) = j$, we could generate random numbers $V_1, \ldots, V_{j+1}$. If we let

$$A = \frac{1}{\sum_{i=1}^{j+1} \log(V_i)}$$

and then define

$$U_{(1)} = A \log(V_1)$$
$$U_{(r)} = U_{r-1} + A \log(V_r), \qquad r = 2, \ldots, j$$

then $U_{(i)}, i = 1, \ldots, j$ are the ordered values that can be used in the preceding.

If we now let

$$X_j = Y_{j-1}, \qquad j = 2, \ldots, k,$$

then we have generated the exponentials $X_2, \ldots, X_k$ conditional on the event that $\sum_{i=1}^{k} X_i \geq c$. A simulation run is now completed by following steps 2 through 4 of the procedure of Section 5.

5.3. The Case of a General Distribution

In the case of a general distribution F, it might not be possible to utilize the approach outlined in Section 3. First, we might not be able to compute the value of $\alpha = P\{M_k > c\}$, and, even if we could, then we might not be able to efficiently simulate $X_1, \ldots, X_k$ conditional on the event that $M_k > c$.

To obtain an efficient simulation procedure, suppose first that we can compute the value of α, so it remains to approximate

$$\beta = P\{M_{k+1} < c, \ldots, M_{2k-1} < c \mid M_k \geq c\}.$$

To approximate β, start by choosing values $x_1, \ldots, x_k$ that are within the range of F and are such that $\sum_{i=1}^{k} x_i \geq c$. Then generate the values of $k-1$ independent random variables having distribution F, and let $x_{k+1}, \ldots, x_{2k-1}$ denote these generated values. Let

$$\mathbf{X} = (x_1, \ldots, x_{2k-1}).$$

Each iteration of the monte carlo algorithm operates as follows:

1. Choose one of the indices $1, \ldots, 2k-1$ at random by generating a random number U and then letting $I = \text{Int}((2k-1)U) + 1$.

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press

2. If $I \le k$, generate the value of a random variable having distribution F that is conditioned to be at least $c - (\sum_{i=1}^{k} x_i - x_I)$, and let this be the new value of x_I.
3. If $I > k$, generate the value of a random variable having distribution F and let this be the new value of x_I.
4. Let $J = 1$ if $\sum_{i=j}^{k} x_i + \sum_{i=1}^{j-1} x_{k+i} < c$, for all $j = 2, \ldots, k$, and let it equal 0 otherwise.

Many iterations of the preceding should be performed, and the proportion of them that have $J = 1$ is the estimate of β. It can be shown (see Geman and Geman [2]) that this estimator converges to β as the number of iterations of the algorithm goes to ∞.

If α cannot be directly computed, then we suggest that it be approximated by a simulation that uses importance sampling via the tilted distribution. The details of this simulation are given in [5, Ex. 8s]. Once α has been approximated, we can use the procedure given in the preceding.

6. TWO-SIDED CONTROL CHARTS

Suppose now that for given constants $c_2 < c_1$ we want to determine $E[N]$, where

$$N = \min\{n \ge k : M_n \le c_2 \text{ or } M_n \ge c_1\},$$

where, as before, $M_n = \sum_{i=n-k+1}^{n} X_i$. If we now define A_n to be the event that either $M_n \le c_2$ or $M_n \ge c_1$, then we can proceed in an analogous way as before.

References

1. Crowder, S.V. (1987). Run length distribution of EWMA charts. *Technometrics* 401–407.
2. Geman, S. & Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 6: 721–724.
3. Hwang, F. & Trietsch, D. (1996). A simple relationship between the pattern probability and the rate of false signals in control charts. *Probability in the Engineering and Informational Sciences* 11(3): 315–324.
4. Roberts, S.W. (1959). Control chart tests based on geometric moving averages. *Technometrics* 1(3): 239–250.
5. Ross, S.M. (1997). Simulation, 2nd ed. Boston: Academic Press.
6. Ross, S.M. (1998). Using the importance sampling identity to bound tail probabilities. *Probability in the Engineering and Informational Sciences* 12(4): 445–452.
7. Ross, S.M. (1998). The mean waiting time for a pattern. *Probability in the Engineering and Informational Sciences* 13(1): 1–9.

https://doi.org/10.1017/S0269964899132078 Published online by Cambridge University Press