

## PERMUTED WEIGHTED AREA ESTIMATORS

James M. Calvin  
Marvin K. Nakayama

Computer Science Department  
New Jersey Institute of Technology  
Newark, NJ 07102, U.S.A.

### ABSTRACT

Calvin and Nakayama previously introduced permuting as a way of improving existing standardized time series methods. The basic idea is to split a simulated sample path into non-overlapping segments, permute the segments to construct a new sample path, and apply a standardized time series scaling function to the new path. Averaging over all permuted paths yields the permuted estimator. This paper discusses applying permutations to the weighted area estimator of Goldsman and Schruben. Empirical results seem to indicate that compared to not permuting, permuting can reduce the length and variability of the resulting confidence interval half widths but with additional computational overhead and some degradation in coverage; however, the decrease in coverage is not as bad as with batching.

### 1 INTRODUCTION

The goal of many steady-state simulations is to construct a confidence interval for the steady-state mean of the simulated process. An approach for doing this is to apply one of Schruben's (1983) standardized time series (STS) methods. Under the assumption that the simulated process satisfies a functional central limit theorem (e.g., Billingsley 1999), the class of methods approximates a centered and scaled version of the simulated stochastic process by a Brownian motion and applies a "scaling function" to it. This yields an estimate of the process's "scale," allowing one to build confidence intervals. Schruben (1983) introduced several scaling functions, including the standardized maximum and the area functions; other scaling functions include the weighted area function of Goldsman and Schruben (1990) and the Cramér-von Mises functions of Goldsman, Kang, and Seila (1999).

One often desires output-analysis methods that yield confidence intervals having good coverage and widths with small expectation or variance, and an approach for trying to improve STS methods is to apply batching. Also introduced

by Schruben (1983), the idea is rather than approximate the entire simulated sample path by a single Brownian motion, break up the sample path into non-overlapping batches, and approximate each batch by a Brownian motion. The scaling function is then applied to each batch and the resulting values are combined to form the batched estimator.

Calvin and Nakayama (2001) proposed an alternative way of improving STS methods based on permuting. The basic idea is to split the original sample path into non-overlapping segments, permute the segments to construct a new sample path, and apply an STS scaling function to the entire new path. Averaging over all permuted paths yields the permuted estimator. One apparent advantage of permuting over batching appears to be that with permuting, the scaling function is applied to each entire permuted path, whereas with batching, the scaling function is applied to each batch separately. Thus, batching seems to require the Brownian approximation to start approximately holding for shorter run lengths, compared to permuting, and Calvin and Nakayama (2001) showed empirically that the small-sample performance for batching is not as good as for permuting.

In this paper we describe applying permuting to the weighted area estimator, which was introduced by Goldsman and Schruben (1990) and is a generalization of Schruben's (1983) area estimator. The weighted area scaling function computes the absolute value of the integral over the unit interval of the product of the centered and normalized process and a weighting function. Goldsman, Meketon, and Schruben (1990) showed that for appropriate choices of the weighting function, the weighted area estimator has better small-sample behavior than the area estimator. The goal of our paper is to study the effect of permuting on the weighted area method.

The rest of our paper has the following organization. In Section 2 we provide background on STS methods, including batching and permuting. Section 3 describes the permuted weighted area estimator, and we discuss its asymptotic distribution in Section 4. We then in Section 5 specialize to a specific weighting function, and Section 6

presents some empirical results. Some concluding remarks are in Section 7. Proofs of the results in this paper can be found in Calvin and Nakayama (2004).

## 2 BACKGROUND ON STANDARDIZED TIME SERIES

Suppose that  $Y = [Y(t) : t \geq 0]$  is a continuous-time stochastic process on  $D[0, \infty)$ , the space of right-continuous real-valued functions with left limits on  $[0, \infty)$ . This framework can accommodate discrete-time processes  $V = [V_n : n = 0, 1, 2, \dots]$  by taking  $Y(t) = V_{\lfloor t \rfloor}$ , where  $\lfloor \cdot \rfloor$  denotes the floor function. Let  $C[0, 1]$  denote the space of continuous real-valued functions on  $[0, 1]$ . Also, for  $n = 1, 2, 3, \dots$ , let  $Y_n = [Y_n(t) : 0 \leq t \leq 1] \in C[0, 1]$  with

$$Y_n(t) = \frac{1}{n} \int_0^{nt} Y(s) ds,$$

and note that  $Y_n(1)$  is the sample mean of the process over the first  $n$  time units. Suppose that there is a real number  $\mu$  and positive number  $\sigma$  such that if we define processes  $X_n = [X_n(t) : 0 \leq t \leq 1] \in C[0, 1]$  for  $n = 1, 2, \dots$ , with

$$X_n(t) = \sqrt{n}(Y_n(t) - \mu t),$$

then

$$X_n \xrightarrow{\mathcal{D}} \sigma W \quad (1)$$

as  $n \rightarrow \infty$ , where  $\xrightarrow{\mathcal{D}}$  denotes convergence in distribution in  $C[0, 1]$  with uniform topology and  $W$  is a standard Brownian motion. Known as a functional central limit theorem, (1) has been shown to hold under a wide spectrum of assumptions, such as mixing conditions, which specify a form of asymptotic independence of events far apart in time; e.g., see Billingsley (1999).

We now describe Schruben's (1983) STS methods. Let  $\Omega_0 = \{x \in C[0, 1] : x(t) = 0\}$  and  $\Omega_{00} = \{x \in \Omega_0 : x(1) = 0\}$ . Let  $\mathfrak{R}^+$  denote the non-negative real numbers, and define the class  $\mathcal{M}$  of continuous functions  $b : \Omega_{00} \rightarrow \mathfrak{R}^+$  satisfying the following conditions:

1.  $b(\alpha x) = \alpha b(x)$  for  $\alpha \in \mathfrak{R}^+$  and  $x \in \Omega_{00}$ ;
2.  $P\{b(B) > 0\} = 1$ ,

where  $B$  is a standard Brownian bridge. We call  $b \in \mathcal{M}$  a scaling function, and each such  $b$  gives rise to an STS method. For example, taking  $b$  as  $b(x) = |\int_0^1 x(t) dt|$  yields Schruben's (1983) area estimator. Glynn and Iglehart (1990) give examples of other scaling functions, including one for the well-known method of batch means.

Define the function  $\Gamma : \Omega_0 \rightarrow \Omega_{00}$  as  $\Gamma(x)(t) = x(t) - tx(1)$  for  $0 \leq t \leq 1$  and  $x \in \Omega_0$ . It is straightfor-

ward to show that  $\Gamma$  is a continuous mapping and  $B \equiv \Gamma(W)$  is a standard Brownian bridge; e.g., see Billingsley (1999) for details. Also, when (1) holds and for  $b \in \mathcal{M}$ , the continuous-mapping theorem (e.g., Billingsley 1999) implies  $(\Gamma(X_n), X_n(1)) \xrightarrow{\mathcal{D}} (\sigma B, \sigma W(1))$  as  $n \rightarrow \infty$ , where  $B$  and  $W(1)$  are independent. Also, the properties of  $\mathcal{M}$  ensure

$$\begin{aligned} \frac{Y_n(1) - \mu}{b(\Gamma(Y_n))} &= \frac{X_n(1)}{\sqrt{n} b(\Gamma(Y_n))} = \frac{X_n(1)}{b(\Gamma(X_n))} \\ &\xrightarrow{\mathcal{D}} \frac{W(1)}{b(B)} \end{aligned} \quad (2)$$

as  $n \rightarrow \infty$ .

Now let  $H$  denote the distribution function of  $W(1)/b(B)$ , and (2) then implies that

$$P\left(\frac{Y_n(1) - \mu}{b(\Gamma(Y_n))} \leq a\right) \rightarrow H(a)$$

as  $n \rightarrow \infty$ . Thus, choosing the constant  $a_{b,\alpha}$  such that  $H(a_{b,\alpha}) = 1 - \alpha/2$  for  $0 < \alpha < 1$ , we have that

$$[Y_n(1) - a_{b,\alpha} b(\Gamma(Y_n)), Y_n(1) + a_{b,\alpha} b(\Gamma(Y_n))] \quad (3)$$

is an asymptotically valid  $100(1 - \alpha)\%$  confidence interval for  $\mu$  from our simulation run of length  $n$ .

Rather than apply the scale function to the entire centered and scaled process, batching (Schruben 1983) divides a sample path into  $m \geq 2$  batches, applies the scale function to each batch, and combines the results using an  $l_p$  norm for some  $p \geq 1$ . Specifically, for  $i = 1, 2, \dots, m$ , define  $\Lambda_{i,m} : \Omega_0 \rightarrow \Omega_0$  as

$$\Lambda_{i,m}(x)(t) = \sqrt{m} \left[ x\left(\frac{t+i-1}{m}\right) - x\left(\frac{i-1}{m}\right) \right]$$

for  $0 \leq t \leq 1$ , so  $W_i \equiv \Lambda_{i,m}(W)$ ,  $i = 1, 2, \dots, m$ , are independent standard Brownian motions. The batched STS scaling function corresponding to  $b \in \mathcal{M}$  is then

$$b_{m,p}^*(x) = \left( \frac{1}{m} \sum_{i=1}^m (b \circ \Gamma \circ \Lambda_{i,m}(x))^p \right)^{1/p},$$

where  $f \circ g(x) = f(g(x))$  for functions  $f$  and  $g$ , and it can be shown that  $\Gamma \circ \Lambda_{i,m} = \Gamma \circ \Lambda_{i,m} \circ \Gamma$ .

We now review how to apply the permuting method of Calvin and Nakayama (2001) to construct new STS methods from existing ones. The basic idea is to split the sample path into  $m$  non-overlapping equal-length segments. Permute the  $m$  segments to generate another sample path, and apply an STS scaling function  $b$  to the entire permuted

path. Averaging over all permutations gives the permuted estimator.

More precisely, fix  $m \geq 2$  and let  $S_m$  denote the set of permutations of  $\{1, 2, \dots, m\}$ . For  $\pi = (\pi(1), \pi(2), \dots, \pi(m)) \in S_m$ , define  $T_\pi : \Omega_0 \rightarrow \Omega_0$  so that  $T_\pi(x)$  is the function obtained by gluing together the  $m$  segments of  $x$  according to the permutation  $\pi$ , i.e., for  $1 \leq i \leq m$ , let  $\Delta_i : \Omega_0 \rightarrow \Omega_0$  with

$$\Delta_i(x)(s) = x\left(\frac{i-1}{m} + s\right) - x\left(\frac{i-1}{m}\right), \quad 0 \leq s \leq \frac{1}{m},$$

and set

$$T_\pi(x)(s) = \sum_{j=1}^{i-1} \Delta_{\pi(j)}(x)\left(\frac{1}{m}\right) + \Delta_{\pi(i)}(x)\left(s - \frac{i-1}{m}\right)$$

for  $(i-1)/m \leq s < i/m$  and  $1 \leq i \leq m$ . For each  $\pi \in S_m$ ,  $T_\pi(B) \stackrel{\mathcal{D}}{=} B$ , where  $\stackrel{\mathcal{D}}{=}$  denotes equality in distribution, so  $b \circ T_\pi$  is a STS scaling function whenever  $b$  is. All  $m!$  scaling function values obtained this way are then combined by applying an  $l_p$  norm, so we get

$$\tilde{b}_{m,p}(x) = \left( \frac{1}{m!} \sum_{\pi \in S_m} (b \circ T_\pi(x))^p \right)^{1/p}, \quad (4)$$

which is the *permuted scaling function* with  $m$  segments using scale function  $b$  and the  $l_p$  norm. Calvin and Nakayama (2003) show that this idea can be applied to the maximum estimator, which they introduce, and to the (unweighted) area estimator of Schruben (1983).

### 3 WEIGHTED AREA ESTIMATORS

Goldsmann and Schruben (1990) introduced the weighted area estimator, which we now describe. For  $q = [q(t) : 0 \leq t \leq 1]$  a continuous weighting function, the weighted area scaling function is defined as

$$b_{\text{wa}}(x) = \left| \int_{t=0}^1 q(t) x(t) dt \right|.$$

Examples of weighting functions  $q$  are given in Goldsmann, Meketon, and Schruben (1990), Goldsmann, Kang, and Seila (1999), and Foley and Goldsmann (1999). When  $q(x) \equiv 1$ , then  $b_{\text{wa}}(x)$  is just Schruben's area estimator.

We now present an expression for the permuted weighted area scaling function for  $m \geq 2$  segments and  $p = 2$ . For

$x \in \Omega_{00}$  and weighting function  $q$ , define

$$C_{i,j}(x) = \int_0^{1/m} q\left(t + \frac{i-1}{m}\right) \times \left( x\left(t + \frac{j-1}{m}\right) - x\left(\frac{j-1}{m}\right) \right) dt$$

for  $i, j = 1, 2, \dots, m$ . Also, let

$$Q_i = \int_{(i-1)/m}^{i/m} q(t) dt$$

for  $i = 1, 2, \dots, m$ , and define  $Z_i(x) = x(i/m) - x((i-1)/m)$  for  $i = 1, 2, \dots, m$ . Then the following holds:

**Theorem 1** *The permuted weighted area scaling function  $\tilde{b}_{\text{wa},m,2}$  with  $m \geq 1$  segments and  $p = 2$  is*

$$\begin{aligned} & \tilde{b}_{\text{wa},m,2}(x) \\ &= \left[ \frac{1}{m} \sum_{i=1}^m \sum_{l=1}^m C_{i,l}(x)^2 \right. \\ &+ \frac{1}{m(m-1)} \sum_{\substack{i,j=1 \\ i \neq j}}^m \sum_{\substack{k,l=1 \\ k \neq l}}^m C_{i,k}(x) C_{j,l}(x) \\ &+ \frac{1}{m} \sum_{i=2}^m (i-1) Q_i^2 \sum_{j=1}^m Z_j(x)^2 \\ &+ \frac{1}{m(m-1)} \sum_{i=2}^m (i-1)(i-2) Q_i^2 \sum_{\substack{r,s=1 \\ r \neq s}}^m Z_r(x) Z_s(x) \\ &+ \frac{1}{m} \sum_{\substack{i,j=2 \\ i \neq j}}^m ((i \wedge j) - 1) Q_i Q_j \sum_{r=1}^m Z_r(x)^2 \\ &+ \frac{1}{m(m-1)} \sum_{\substack{i,j=2 \\ i \neq j}}^m [(i-1)(j-1) - ((i-1) \wedge (j-1))] \\ &\quad \times Q_i Q_j \sum_{\substack{r,s=1 \\ r \neq s}}^m Z_r(x) Z_s(x) \\ &+ \frac{2}{m} \sum_{i=1}^m \sum_{l=1}^m C_{i,l}(x) Z_l(x) \sum_{j=i+1}^m Q_j \\ &+ \frac{2}{m(m-1)} \sum_{i=1}^m \sum_{l=1}^m C_{i,l}(x) \sum_{\substack{r=1 \\ r \neq l}}^m Z_r(x) \\ &\quad \left. \times \left( \sum_{j=2}^i (j-1) Q_j + \sum_{j=i+1}^m (j-2) Q_j \right) \right]^{1/2} \quad (5) \end{aligned}$$

for  $x \in \Omega_{00}$ .

We now examine the complexity of computing the permuted weighted area scaling function  $\tilde{b}_{wa,m,2}(x)$ . Suppose that  $x \in \Omega_{00}$  is a path of length  $n$ . We can compute each  $C_{i,j}(x)$  in  $\Theta(n/m)$  operations, so computing all the  $C_{i,j}(x)$  for  $i, j = 1, 2, \dots, m$ , requires  $\Theta(mn)$  operations. Given  $x$ , calculating the  $Z_r(x)$ ,  $r = 1, 2, \dots, m$ , requires  $\Theta(m)$  work. Also, computing each  $Q_i$  requires  $\Theta(1)$  work, so computing all the  $Q_i$  for  $i = 1, 2, \dots, m$ , requires  $\Theta(m)$  work. Once all of the  $C_{i,j}(x)$  and  $Q_i$  are calculated, we then compute  $\tilde{b}_{wa,m,2}(x)$  as in (5). For the second term in (5), observe that

$$\begin{aligned} & \sum_{\substack{i,j=1 \\ i \neq j}}^m \sum_{\substack{k,l=1 \\ k \neq l}}^m C_{i,k}(x)C_{j,l}(x) \\ &= \left( \sum_{i,k=1}^m C_{i,k}(x) \right)^2 - \sum_{i=1}^m \left( \sum_{j=1}^m C_{i,j}(x) \right)^2 \\ & \quad - \sum_{k=1}^m \left( \sum_{i=1}^m C_{i,k}(x) \right)^2 + \sum_{i,j=1}^m C_{i,j}(x)^2, \end{aligned}$$

which can be computed with work  $\Theta(m^2)$ . To compute the last term in (5), define

$$\begin{aligned} S_Z &= \sum_{r=1}^m Z_r(x), \\ S_{Q,1,i} &= \sum_{j=1}^i (j-1)Q_j, \quad i = 2, 3, \dots, m, \\ S_{Q,2,i} &= \sum_{j=i+1}^m (j-2)Q_j, \quad i = 1, 2, \dots, m-1, \end{aligned}$$

which requires a total of  $\Theta(m^2)$  calculations. Therefore,

$$\begin{aligned} & \sum_{i=1}^m \sum_{l=1}^m C_{i,l}(x) \sum_{\substack{r=1 \\ r \neq l}}^m Z_r(x) \\ & \times \left( \sum_{j=2}^i (j-1)Q_j + \sum_{j=i+1}^m (j-2)Q_j \right) \\ &= \sum_{i=1}^m \sum_{l=1}^m C_{i,l}(x) (S_Z - Z_l(x)) (S_{Q,1,i} + S_{Q,2,i}), \end{aligned}$$

so calculating the last term in (5) takes  $\Theta(m^2)$  work. It is clear that all of the other terms in (5) require  $O(m^2)$  work. Hence, the total work to compute  $\tilde{b}_{wa,m,2}(x)$  for a simulation of length  $n$  and  $m$  segments is  $\Theta(mn + m^2)$ .

Generating a simulation of length  $n$  requires  $\Theta(n)$  work, but the leading constant here may be large compared to the

leading constants in the  $\Theta(mn + m^2)$  work needed to compute the permuted weighted area estimator, but nevertheless, this suggests that  $m$  should not be taken to be too large.

#### 4 ESTIMATING CRITICAL POINTS

The complicated expression for the permuted weighted area estimator in (5) seems to preclude exact calculation of critical points of the distribution of  $W(1)/\tilde{b}_{wa,m,2}(B)$ , which are needed to construct confidence intervals as in (3). Instead, we can estimate the critical points by sampling. To do this, let  $Z_i \equiv Z_i(B)$  and  $C_{ij} \equiv C_{ij}(B)$  be normally distributed with mean 0, and we need to calculate  $\text{cov}(Z_i, Z_j)$ ,  $\text{cov}(C_{ij}, Z_k)$  and  $\text{cov}(C_{ij}, C_{kl})$  for all  $i, j, k, l = 1, 2, \dots, m$ .

Once all of the covariances have been calculated, we can estimate the critical points of the distribution  $H$  of  $W(1)/\tilde{b}_{wa,m,2}(B)$  by first generating  $Z_i$  and  $C_{ij}$ ,  $i, j = 1, 2, \dots, m$ , having a joint normal distribution with mean 0 and the given covariance structure. (Recall that a multivariate normal  $V \sim N(0, \Sigma)$  with mean vector 0 and covariance matrix  $\Sigma$  can be obtained by setting  $V = AY$ , where  $Y \sim N(0, I)$  and  $\Sigma = AA'$ . The matrix  $A$  can be calculated using a singular value decomposition; e.g., see Golub and Van Loan 1989). The generated  $Z_i$  and  $C_{ij}$  are then put into (5) to obtain a sample of  $\tilde{b}_{wa,m,2}(B)$ . We generate a sample of  $W(1) \sim N(0, 1)$ , independent of the  $Z_i$  and  $C_{ij}$ , and putting these together yield a sample of  $W(1)/\tilde{b}_{wa,m,2}(B)$ . By repeating this process many times, we can obtain the resulting empirical distribution function as an estimate of the true distribution function  $H$ .

We now present the desired covariances, starting with  $\text{cov}(Z_i, Z_j)$ . Recall  $Z_i = B(i/m) - B((i-1)/m)$ , where  $B$  is a standard Brownian bridge. Then since  $E[Z_k] = 0$  and  $\text{cov}(B(s), B(t)) = (s \wedge t) - st$ , it can be shown that

$$\text{cov}(Z_i, Z_j) = \begin{cases} -1/m^2 & \text{when } i \neq j, \\ (m-1)/m^2 & \text{when } i = j. \end{cases}$$

Since  $E[Z_k] = 0$ , we can compute  $\text{cov}(C_{ij}, Z_k)$  as

$$\begin{aligned} & \text{cov}(C_{ij}, Z_k) \\ &= \begin{cases} -\int_0^{1/m} q\left(t + \frac{i-1}{m}\right) \frac{t}{m} dt & \text{when } j \neq k, \\ \int_0^{1/m} q\left(t + \frac{i-1}{m}\right) \left(t - \frac{t}{m}\right) dt & \text{when } j = k. \end{cases} \end{aligned}$$

Also, we can show that since  $E[C_{ij}] = 0$ ,

$$\begin{aligned} & \text{cov}(C_{ij}, C_{kl}) \\ &= -\left( \int_{s=0}^{1/m} q\left(s + \frac{i-1}{m}\right) s ds \right) \\ & \quad \times \left( \int_{t=0}^{1/m} q\left(t + \frac{k-1}{m}\right) t dt \right) \end{aligned}$$

when  $j \neq l$ , and

$$\begin{aligned} & \text{cov}(C_{ij}, C_{kl}) \\ &= \int_{s=0}^{1/m} q\left(s + \frac{i-1}{m}\right) \int_{t=0}^s q\left(t + \frac{k-1}{m}\right) t \, dt \, ds \\ &+ \int_{s=0}^{1/m} q\left(s + \frac{i-1}{m}\right) s \int_{t=s}^{1/m} q\left(t + \frac{k-1}{m}\right) dt \, ds \\ &- \left( \int_{s=0}^{1/m} q\left(s + \frac{i-1}{m}\right) s \, ds \right) \\ &\times \left( \int_{t=0}^{1/m} q\left(t + \frac{k-1}{m}\right) t \, dt \right) \end{aligned}$$

when  $j = l$ .

## 5 A SPECIFIC WEIGHTING FUNCTION

Goldsman and Schruben (1990) and Foley and Goldsman (1999) consider the weighting function

$$q(t) = \sqrt{840}(3t^2 - 3t + 1/2), \quad (6)$$

in which case

$$Q_i = \frac{\sqrt{840}}{2m^3} (6i^2 - 6i + 2 - 6im + 3m + m^2).$$

Also, using the formulae from Section 4, we get

$$\begin{aligned} & \text{cov}(C_{ij}, Z_k) \\ &= -\frac{\sqrt{840}}{m} \left[ \frac{6i^2 - 4i + 1}{4m^4} - \frac{3i - 1}{2m^3} + \frac{1}{4m^2} \right] \end{aligned}$$

when  $j \neq k$ , and

$$\begin{aligned} & \text{cov}(C_{ij}, Z_k) \\ &= \sqrt{840} \left( 1 - \frac{1}{m} \right) \left[ \frac{6i^2 - 4i + 1}{4m^4} - \frac{3i - 1}{2m^3} + \frac{1}{4m^2} \right] \end{aligned}$$

when  $j = k$ . Moreover,

$$\begin{aligned} & \text{cov}(C_{ij}, C_{kl}) \\ &= -840 \left[ \frac{6i^2 - 4i + 1}{4m^4} - \frac{3i - 1}{2m^3} + \frac{1}{4m^2} \right] \\ &\times \left[ \frac{6k^2 - 4k + 1}{4m^4} - \frac{3k - 1}{2m^3} + \frac{1}{4m^2} \right] \end{aligned}$$

when  $j \neq l$ , and

$$\begin{aligned} & \text{cov}(C_{ij}, C_{kl}) \\ &= \frac{1}{m^7} [120 + 420m - 2520imk^2 - 2520i^2mk \\ &- 420k - 420i + 546m^2 + 3780imk + 2520im^2k \\ &+ 70m^4 + 315m^3 - 1260mk - 1260im \\ &+ 1512ik + 504k^2 + 504i^2 - 1260im^2 + 945mk^2 \\ &- 1260m^2k - 1890ik^2 + 945i^2m - 1890i^2k \\ &+ 2520i^2k^2 + 420i^2m^2 + 420m^2k^2 \\ &- 420m^3k - 420im^3] \\ &- 840 \left[ \frac{6i^2 - 4i + 1}{4m^4} - \frac{3i - 1}{2m^3} + \frac{1}{4m^2} \right] \\ &\times \left[ \frac{6k^2 - 4k + 1}{4m^4} - \frac{3k - 1}{2m^3} + \frac{1}{4m^2} \right] \end{aligned}$$

when  $j = l$ , which was computed using Matlab (2002).

We estimated quantile points for the permuted weighted area estimator by running simulations as described in the beginning of Section 4. For various values of  $m$ , the estimates for the quantiles based on  $10^6$  independent replications are given in Table 1, which also has the same level quantiles for the batched weighted area estimator. The second row of the table gives the quantile levels and the rows below give the corresponding quantile points; e.g.,  $P(W(1)/b_{\text{wa},3,2}(B) \leq 2.276) \approx 0.950$ . Goldsman and Schruben (1990) show that if the scaling function  $b$  corresponds to the batched weighted area estimator with  $m$  batches for our specific weighting function  $q$  in (6), then  $W(1)/b(B)$  has a Student's- $t$  distribution with  $m$  degrees of freedom. Thus, we did not need to estimate the quantiles for batching. The row for  $m = 1$  corresponds to the critical points for the unpermuted and unbatched weighted area estimator in both sets of columns.

For  $m = 2$  the quantiles for permuting are significantly larger than the corresponding quantiles for batching, while for other values of  $m$ , the quantiles for permuting are always smaller. It appears that the reason for this is that  $Q_1 = Q_2 = 0$  when  $m = 2$  for our choice of weighting function  $q$  in (6), so in (5), all of the terms drop out except for the first two. When  $m > 2$ , this does not occur, so it seems that the case of  $m = 2$  is an anomaly, which is reflected in the quantiles.

## 6 EMPIRICAL RESULTS

We now present results from some simulation experiments. The model simulated is the embedded discrete-time Markov chain (DTMC) of the number of customers in an M/M/1 queue with traffic intensity  $\rho = 0.8$  on a truncated state space  $\{0, 1, \dots, 100\}$ . We ran  $10^5$  independent replications, with each replication consisting of 2000 simulated transitions. In

Table 1: Estimated Critical Points

$m$	Permuting			Batching		
	0.950	0.975	0.995	0.950	0.975	0.995
1	6.314	12.71	65.66	6.314	12.71	65.66
2	4.827	7.800	19.52	2.920	4.303	9.925
3	2.276	2.993	5.048	2.353	3.182	5.841
4	2.071	2.644	4.108	2.132	2.776	4.604
5	1.983	2.508	3.767	2.015	2.571	4.032

each replication we constructed a 90% confidence interval for  $\mu$ , the steady-state average number of customers in the system, using the regular (i.e., no permuting nor batching), permuted and batched weighted area estimators with the weighting function in (6). Table 2 presents the results of the observed coverages (in the column labelled ‘‘Cov’’), average half width of the confidence intervals (‘‘Avg HW’’), and variance of the half widths (‘‘Var HW’’) for various values of  $m$ . The row for  $m = 1$  gives the results for the regular weighted area estimator, and the rows for  $m \geq 2$  are for the permuted and batched estimators.

As  $m$  increases, both batching and permuting lead to smaller and less-variable half widths than the regular estimator but at a cost of worse coverage. The degradation in coverage is worse for batching than for permuting. For each value of  $m$ , batching has smaller average half width and variance than permuting, but also worse coverage. If we compare permuting and batching for different values of  $m$  so that permuting and batching have about the same coverage (e.g., comparing  $m = 5$  for permuting and  $m = 3$  for batching), then permuting has smaller and less variable confidence intervals.

We also measured the CPU times to run some of our simulations and found that runs with batching with  $m = 5$  and those with the regular estimator required about the same amount of time, but runs with permuting with  $m = 5$  took about 3 times as long. However, one should keep in mind that we simulated a very simple model (embedded DTMC of an M/M/1 queue), so the work required to generate the sample path is minimal. For a more complicated model, generating the sample path might require much more work, so the run times may not be as different for the various methods.

## 7 CONCLUSIONS

Calvin and Nakayama (2003) showed that permuting improves Schruben’s (1983) (unweighted) area estimator. This paper considered applying permuting to the weighted area estimator. Goldsman, Meketon, and Schruben (1990) previously showed the (unpermuted) weighted area estimator to have better small-sample behavior than the (unpermuted and unweighted) area estimator. Our results here seem to indicate that permuting can also improve the weighted area estimator.

Table 2: Empirical Results

$m$	Method	Cov (%)	Avg HW	Var HW
1	Regular	86.4	5.19	32.31
2	Permuting	85.0	4.01	18.35
2	Batching	81.0	2.45	4.47
3	Permuting	80.4	2.04	2.64
3	Batching	78.0	1.91	1.97
4	Permuting	79.0	1.83	1.81
4	Batching	75.4	1.65	1.12
5	Permuting	78.1	1.72	1.46
5	Batching	72.9	1.47	0.71

## ACKNOWLEDGMENTS

The authors thank David Goldsman for helpful discussions regarding this work, and Alex Gerbessiotis for assistance in writing the simulation programs for the experiments. This material is based upon work supported by the National Science Foundation under Grant No. 9900117. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation

## REFERENCES

- Billingsley, P. 1999. *Convergence of Probability Measures*, Second Edition. New York: John Wiley.
- Calvin, J. M. and M. K. Nakayama. 2001. Improving standardized time series methods by permuting path segments. In *Proceedings of the 2001 Winter Simulation Conference*, ed. B. A. Peters, J. S. Smith, D. J. Medeiros, and M. W. Rohrer, 348–353. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.
- Calvin, J. M. and M. K. Nakayama. 2003. Permuted standardized time series methods. Submitted.
- Calvin, J. M. and M. K. Nakayama. 2004. Permuted weighted area estimators for steady-state simulations. Forthcoming NJIT CS technical report.
- Foley, R. D. and D. Goldsman. 1999. Confidence intervals using orthonormally weighted standardized time series. *ACM Transactions on Modeling and Computer Simulation* 9: 297–325.
- Glynn, P. W. and D. L. Iglehart. 1990. Simulation output analysis using standardized time series. *Mathematics of Operations Research* 14: 1–16.
- Goldsman, D., K. Kang, and A. F. Seila. 1999. Cramer-von Mises variance estimators for simulations. *Operations Research* 47: 299–309.
- Goldsman, D., M. S. Meketon, and L. W. Schruben. 1990. Properties of standardized time series weighted area variance estimators. *Management Science* 36: 602–612.

- Goldman, D. and L. W. Schruben. 1990. New confidence interval estimators using standardized time series. *Management Science* 36: 393–397.
- Golub, G. H. and C. F. Van Loan. 1989. *Matrix Computations*, Second Edition. Baltimore: Johns Hopkins University Press.
- MATLAB 6.5 software package. 2002. Natick, Massachusetts: The MathWorks.
- Schruben, L. W. 1983. Confidence interval estimation using standardized time series. *Operations Research* 31:1090–1108.

## AUTHOR BIOGRAPHIES

**JAMES M. CALVIN** is an associate professor in the Department of Computer Science at the New Jersey Institute of Technology. He received a Ph.D. in operations research from Stanford University and is an associate editor for *ACM Transactions on Modeling and Computer Simulation*. Besides simulation output analysis, his research interests include global optimization and probabilistic analysis of algorithms.

**MARVIN K. NAKAYAMA** is an associate professor in the Department of Computer Science at the New Jersey Institute of Technology. He received a Ph.D. in operations research from Stanford University. He won second prize in the 1992 George E. Nicholson Student Paper Competition sponsored by INFORMS and is a recipient of a CAREER Award from the National Science Foundation. He is the area editor for the Stochastic Modeling Area of *ACM Transactions on Modeling and Computer Simulation* and an associate editor for *Inform Journal on Computing*. His research interests include applied probability, statistics, simulation and modeling.