Proceedings of the 2008 Winter Simulation Conference
S. J. Mason, R. R. Hill, L. Mönch, O. Rose, T. Jefferson, J. W. Fowler eds.

IMPLEMENTABLE MSE-OPTIMAL DYNAMIC PARTIAL-OVERLAPPING BATCH MEANS ESTIMATORS FOR STEADY-STATE SIMULATIONS

Wheyming Tina Song
Mingchang Chih

Department of Industrial Engineering
National Tsing Hua University
Hsinchu, Taiwan, 300, R.O.C

ABSTRACT

Estimating the variance of the sample mean from a stochastic process is essential in assessing the quality of using the sample mean to estimate the population mean which is the fundamental question in simulation experiments. Most existing studies for estimating the variance of the sample mean from simulation output assume simulation run length is known in advance. This paper proposes an implementable batch-size selection procedure for estimating the variance of the sample mean without requiring that the sample size or simulation run length *a priori*.
Key Words: Simulation, Variance of the Sample Mean, Mean-Squared-Error

1 Introduction

How good is using the sample mean to estimate the population mean from a stochastic process? This is a fundamental question not only in simulation but also in statistical experiments. Estimating the variance of the sample mean from a stochastic process is essential in assessing the quality of using the sample mean to estimate the population mean. In addition, estimating the variance of the sample mean is also crucial in calculating the confidence and prediction intervals of the population mean and the probability of selecting from alternatives correctly.

Consider a sequence, $\{Y_1, Y_2, \cdots, Y_n\}$, representing the output of a simulation from a covariance-stationary stochastic process $\{Y_i\}_{i=1}^n$, with an unknown mean $\mu = \mathrm{E}(Y)$ and unknown positive variance $R_0 = \mathrm{var}(Y)$. For example, Y_i could be the delay time for the i-th packet at some node in a communication network. Let μ be the performance measure we are interested in, $\overline{\overline{Y}}_n = \Sigma_{i=1}^n Y_i/n$ be the point estimator of μ, and $\mathrm{var}(\overline{\overline{Y}}_n)$ be the quality measure of using $\overline{\overline{Y}}_n$ to estimate μ. The goal of this paper is to estimate $\mathrm{var}(\overline{\overline{Y}}_n)$ with limited memory space, reasonable computation time, and good statistical properties such as mse, but does not require knowledge of the simulation run length *a priori*.

Most existing papers in simulation output assume that either the simulation run length or the sample size n is known in advance. Moreover, many of them view data as being stored in infinite computer memory space. For example, direct (Moran 1975), regenerative (Crane and Iglehart 1975; Glynn and Iglehart 1986), spectral (Priestley 1981; Heidelberger and Welch 1981), non-overlapping batch means (NBM) (Conway 1963, Fishman 1978, Law and Carson 1979, and Schmeiser 1982), overlapping batch means (OBM) (Meketon and Schmeiser 1984), partial-overlapping batch means (PBM) (Welch 1987), standardized time series (Schruben 1983; Glynn and Iglehart 1990), and its variation. The memory space for these estimators is proportional to the sample size n; these estimators require $O(n)$ space when the sample size is not known in advance. Fishman and Yarberry (1997) proposed Labtach.2 non-overlapping batch means requiring $O(log_2 n)$ space. Although $O(log_2 n)$ algorithms require much small memory space than $O(n)$-memory algorithms, Fishman and Yarberry (1997) required the knowledge of *n a priori*. Song (1996) and Pedrosa's (1994) proposed mse-optimal batch size algorithms for estimating the optimal batch size for batch means estimators including NBM and OBM. Both mse-optimal batch size algorithms also required the knowledge of the sample size *a priori*.

We explain in additional detail the two following scenarios with examples to motivate the need for algorithms without assuming the knowledge of the sample size *a priori* within the context of simulation output analysis. (1) Sample size is unknown *a priori*. Consider creating a simulation model of pandemic influenza to evaluate the effectiveness of different decision policies on disease spread and other performance measures (Ferguson *et al.* 2006 and Jenvald *et al.* 2007). The simulation run length (in terms of the number of patients) is random, and so n is not known in advance. (2) Sample size is extremely large. One example is that of using simulation to test data stream algorithms

to estimate entropy (a measure of the rate of transfer of information in a network) from input data that come at a very high rate — a rate so high that it places stress on a limited computing infrastructure. Recent work on data streaming algorithms can be found in Lall *et al.* (2006) and Zhao *et al.* (2007).

To the best of our knowledge, the dynamic non-overlapping batch means (DNBM) estimator proposed by Yeh and Schmeiser (2000) and the dynamic partial-overlapping batch means (DPBM) estimator proposed by Song (2007) are the only existing algorithms for estimating the variance of the sample mean without requiring the knowledge of the simulation run length *a priori*. The common drawback of DNBM and DPBM is that they lack the control of choosing the value of the batch size which is the performance-parameter of the DNBM and DPBM.

In this paper, we first take the DPBM as a base-line, then we estimate the optimal batch size to adjust the DPBM to either increase the number of batches or the batch size as the simulation run-length increases, without storing observations individually.

2 Background

This section reviews traditional batching methods, DPBM estimators (Song 2007), and asymptotic results that are useful for estimating the optimal batch size for the batching estimators for the variance of the sample mean. The section does not have a review on DNBM (Yeh and Schmeiser 2000) in particular, because DNBM is a special case of DPBM.

2.1 Batch Means Estimators

The batching method is a classic methodology in estimating the variance of the sample mean from a stochastic process. Conway (1963) was the first to introduce the idea of the batching method in digital simulation. The method is based on dividing the observations $Y_1, Y_2, \cdots, Y_n$ into b batches, with each batch size being m. In other words, the method groups observations into batches and uses these batches as the basic data for analysis. Batch means estimators with batch size m and shift s are defined as

$$\hat{V}(m,s) = \frac{\sum_{i=1}^{b} (\overline{Y}_{s(i-1)+1} - \overline{\overline{Y}}_n)^2}{d_b}, \tag{1}$$

where $1 \le m \le n,\ 1 \le s \le n-m,\ d_b = b(n/m-1),\ b = \lfloor (n-m+s)/s \rfloor$ is the number of batches (where $\lfloor x \rfloor$ is the greatest integer smaller than or equal to x),

$$\overline{Y}_{s(i-1)+1,m} = \sum_{j=1}^{m} Y_{s(i-1)+j}/m \tag{2}$$

is the i-th batch mean, and $Y_{s(i-1)+j,m}$ is the j-th observation in the i-th batch. For simplicity, we sometimes use $\overline{Y}_{s(i-1)+1}$ instead of $\overline{Y}_{s(i-1)+1,m}$ by suppressing the subscript m.

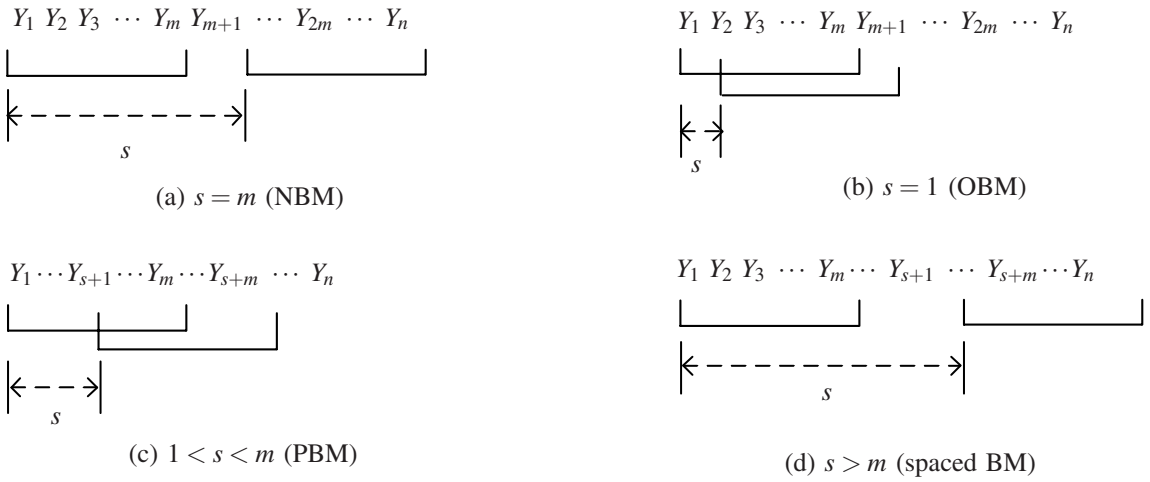
The NBM estimator with batch size m is the special case obtained when $s = m$, and is denoted by $\hat{V}^{\mathrm{N}}(m)$ (see Figure 1(a)). The OBM estimator with batch size m is the special case obtained when $s = 1$, and is denoted by $\hat{V}^{\mathrm{O}}(m)$ (see Figure 1(b)). The PBM estimator with batch size m is the special case obtained when $1 < s < m$, and is denoted by $\hat{V}^{\mathrm{P}}(m)$ (see Figure 1(c)). If $s = \lfloor \alpha m \rfloor$ for $0 < \alpha < 1$, we have a $100(1-\alpha)\%$ OBM estimator, e.g., 50%OBM and 75%OBM estimators for $s = \lfloor m/2 \rfloor$ and $s = \lfloor m/4 \rfloor$, respectively. The spaced batch means estimator with batch size m is the special case obtained when $s > m$ (see Figure 1(d)).

The asymptotic properties of batch means estimators are discussed in several studies. The asymptotic relative bias results, discussed in Meketon and Schmeiser (1984) and Song and Schmeiser (1995), show that all of these batch means estimators have essentially the same bias. Asymptotic relative variance results (Welch 1987; Meketon and Schmeiser 1984) show that $\hat{V}^{\mathrm{N}}(m)$ has 50% more variance than $\hat{V}^{\mathrm{O}}(m)$, while $\hat{V}^{\mathrm{P}}(m)$ with $s = m/2$ (i.e., 50%OBM) has 12% more variance than $\hat{V}^{\mathrm{O}}(m)$, and $\hat{V}^{\mathrm{P}}(m)$ with $s = m/4$ (i.e., 75%OBM) has just 3% more variance than $\hat{V}^{\mathrm{O}}(m)$.

2.2 Dynamic Partial-Overlapping Batch Means

This section reviews the dynamic partial-overlapping batch means (DPBM) proposed by Song (2007). Song (2007) proved that the DPBM is a finite-memory algorithm for implementing 75%OBM(m) in $O(n)$ time with $O(1)$-memory space. The key to developing DPBM in a fixed storage space is by dimensionality reduction using "vector collapsing", which was originated from Fishman (1978)'s idea of doubling batch size for NBM and also adopted in Yeh and Schmeiser (2000) to form the DNBM estimator. The idea of using collapsing to form the DPBM estimator will be illustrated in Figure 2.

Before explaining Figure 2, we need to define notation used in developing DPBM. Let n be the total number of observations, which is not known in advance. Let l be the pre-specified memory size, which is also the size of the vector where all observations are stored. Let k be the total number of times that collapsing has occurred, $k = 0, 1, 2, \ldots, \lceil \log_2 n/g \rceil - 1$, where $g = l/8$. Let $\underline{L}$ be the vector of size l where DPBM stores batch sums and $\underline{L}$ is divided into four vectors: $\underline{A}, \underline{B}, \underline{C}$, and $\underline{D}$ by concatenation, i.e., $\underline{L} = (\underline{A}, \underline{B}, \underline{C}, \underline{D})$. Let $A_k(i)$, $B_k(i)$, $C_k(i)$, and $D_k(i)$ be the numerical values (batch sums) stored in the i-th cell of vectors $\underline{A}$, $\underline{B}$, $\underline{C}$, and $\underline{D}$ in the k-th iteration of collapsing, respectively, where $i = 1, 2, \ldots, 2g$. We sometimes suppress subscripts for convenience; for example, $A_k(1)$ is replaced by $A(1)$. Let r_A, r_B, r_C, and r_D be the cell-locations



(a) $s = m$ (NBM)

(b) $s = 1$ (OBM)

(c) $1 < s < m$ (PBM)

(d) $s > m$ (spaced BM)

Figure 1: Batch Means Estimators: m is the batch size and s is the shift. (All dash-lines indicate distance)

used to store the latest observations in $\underline{A}$, $\underline{B}$, $\underline{C}$, and $\underline{D}$, respectively; where $r_A, r_B, r_C, r_D = 1, \ldots, 2g$. Let m_A, m_B, m_C, and m_D be the associated numbers of observations stored in $A(r_A), B(r_B), C(r_C)$, and $D(r_D)$, respectively, where $m_A, m_B, m_C, m_D = 1, \ldots, m_k$, and $m_k = 2^k$ is the maximum batch size for each cell at the k-th collapsing step. If a cell (batch) at step k is a sum of $m_k = 2^k$ observations, we call the batch a "full batch"; otherwise, we call the batch a "partial batch".

Let b_1, b_2, b_3, and b_4 be the "numbers of full batches" in vectors $\underline{A}$, $\underline{B}$, $\underline{C}$, and $\underline{D}$, respectively, where $b_1 = r_A - 1 + \lfloor \frac{m_A}{m_k} \rfloor$, $b_2 = r_B - 1 + \lfloor \frac{m_B}{m_k} \rfloor$, $b_3 = r_C - 1 + \lfloor \frac{m_C}{m_k} \rfloor$, and $b_4 = r_D - 1 + \lfloor \frac{m_D}{m_k} \rfloor$.

Figure 2 shows how data in a DPBM algorithm are collapsed and stored in finite space. In Figure 2, y_1 is stored in $A_0(1)$, y_2 is stored in $A_0(2)$, $\cdots$, and y_{2g} is stored in $A_0(2g)$. When a new observation y_{2g+1} appears, the DPBM algorithm starts its first collapsing to store data in $\underline{B}$, then update data in vector $\underline{A}$. Specifically, We store the sum of y_2 and y_3 into $B_1(1)$, i.e., $B_1(1) = A_0(2) + A_0(3)$. Similarly, $B_1(2) = A_0(4) + A_0(5), \cdots, B_1(g) = A_0(2g) + A_0(2g+1)$. Then, we update vector $\underline{A}$ in that $A_1(1) = A_0(1) + A_0(2)$, $A_1(2) = A_0(3) + A_0(4), \cdots, A_1(g) = A_0(2g-1) + A_0(2g)$, $A_1(g+1) = A_0(2g+1)$. When another new observation y_{2g+1} appears, we update vectors $\underline{B}$ and $\underline{A}$ in that $B_1(g+1) = A_0(2g+2)$ $A_1(g+1) = A_0(2g+1) + A_0(2g+2)$ and $B_1(g+1) = A_0(2g+2)$. When the data stored in vector $\underline{A}$ is full again, i.e., $A_1(2g) = y(4g-1) + y(4g)$, the DPBM algorithm collapses again. The order for updating vectors is D, C, B, A. The numbers 1, 2, 3, and 4 listed on the arrows in Figure 2 indicate the updating order in each iteration.

In general, the DPBM algorithm starts to collapse data when the data in vector $\underline{A}$ is full. Specifically, a new observation is added in the current cell if the number of observations contained in the current cell, m_A, is less than the full batch size $m_k = 2^k$, indicating that vector $\underline{A}$ is not full. As long as the r_A-th cell contains the same number of observations as the full batch size and the vector is full, we collapse these $2g$ cells into g cells. After collapsing the vector, the full batch size is updated by doubling the previous value. The logic used to collapse data into vectors $\underline{B}$, $\underline{C}$, and $\underline{D}$ is the same as that for vector $\underline{A}$, but the data used for collapsing into $\underline{B}$, $\underline{C}$, and $\underline{D}$ differ. The order of collapsing in DPBM is $\underline{D}$ and $\underline{C}$ first, and then $\underline{B}$, and finally $\underline{A}$. In summary, we develop DPBM by collapsing four vectors $\underline{A}, \underline{B}, \underline{C}$, and $\underline{D}$. We use four vectors for the DPBM because the 75%OBM has 4 times more batches than the NBM estimator.

The DPBM estimator for estimating the variance of the sample mean proposed in Song(2007) at step k is

$$\hat{V}_{\mathrm{DPBM}}(m_k) = \frac{1}{d_b}\Big[\sum_{i=1}^{b_1}\Big(\frac{A_k(i)}{m_k} - \overline{\overline{Y}}_n\Big)^2 + \sum_{i=1}^{b_2}\Big(\frac{B_k(i)}{m_k} - \overline{\overline{Y}}_n\Big)^2 + \sum_{i=1}^{b_3}\Big(\frac{C_k(i)}{m_k} - \overline{\overline{Y}}_n\Big)^2 + \sum_{i=1}^{b_4}\Big(\frac{D_k(i)}{m_k} - \overline{\overline{Y}}_n\Big)^2\Big], \tag{3}$$

where $m_k = 2^k$ is the batch size, $s = \lfloor m_k/4 \rfloor$ is the shift, $b = \lfloor (n - m_k + s)/s \rfloor$ is the total number of batches, $d_b = b((n/m_k) - 1)$ is the denominator, $b_i, i = 1, 2, 3, 4$ are the numbers of full batches, and the batch means are

$$\frac{A_k(i)}{m_k} = \overline{Y}_{m_k(i-1)+1,m_k}, \tag{4}$$

$$\frac{B_k(i)}{m_k} = \overline{Y}_{m_k(i-1)+2s+1,m_k}, \tag{5}$$

$$\frac{C_k(i)}{m_k} = \overline{Y}_{m_k(i-1)+s+1,m_k}, \tag{6}$$

$$\frac{D_k(i)}{m_k} = \overline{Y}_{m_k(i-1)+3s+1,m_k}. \tag{7}$$

The DNBM is a special case of DPBM and is defined as

$$\hat{V}_{\mathrm{DNBM}}(m_k) = \frac{1}{d_b}\left[\sum_{i=1}^{b_1}(\frac{A_k(i)}{m_k} - \overline{\overline{Y}}_n)^2\right]. \tag{8}$$

It is noted that the memory space required for the DNBM is one quarter of that for the DPBM because the DNBM stores data only in vector $\underline{A}$.

2.3 Asymptotic Results

This section reviews existing asymptotic results that are useful for studying the optimal batch size. The proof of these existing asymptotic results is given in Song (1988) and Song and Schmeiser (1995).

Result 1.

$$n\mathrm{var}(\overline{\overline{Y}}_n) = \gamma_0 R_0 - \frac{\gamma_1 R_0}{n} + o(n^{-1}), \tag{9}$$

where n is the simulation run length,

$$R_0 = \mathrm{var}(Y) \tag{10}$$

is the variance of the data Y,

$$\gamma_0 = 1 + 2\sum_{h=1}^{\infty}\rho_h \tag{11}$$

is the sum of all correlations,

$$\gamma_1 = 2\sum_{h=1}^{\infty} h\rho_h \tag{12}$$

is the sum of all weighted correlations, and $\rho_h = \mathrm{corr}(Y_i, Y_{i+h})$ *is the lag h correlation of* Y_i *and* Y_{i+h}*, which satisfies* $\rho_h = \sigma^2 O(\delta^h)$ *for* $h = 1,2,\ldots,\delta \in (0,1)$ *to reflect a general correlation structure for a wide range of stochastic processes, including waiting times in steady state M/M/1 queueing systems which discussed in Aktaran-Kalayc et al. (2007).*

The notation $o(g(n))$ represents the little o function in that $o(g(n))/g(n) \to 0$ as $n \to \infty$, and the notation $O(g(n))$ represents the big o function in that $|O(g(n))/g(n)| \to c$ as $n \to \infty$ where c is a real value.

Result 2. *For NBM, 50%OBM, 75%OBM, and OBM, if* γ_1 *converges absolutely to a finite limit, then*

$$\lim_{\substack{n\to\infty \\ m\to\infty}} nmbias(\hat{V}) = -\gamma_1 R_0. \tag{13}$$

Result 3. *For NBM, 50%OBM, 75%OBM, and OBM, if* γ_0 *converges absolutely to a finite limit, then*

$$\lim_{\substack{n\to\infty \\ m\to\infty}} \frac{n^3}{m} var(\hat{V}) = -c_v(\gamma_0 R_0)^2, \tag{14}$$

where c_v is called a variance constant. The constants c_v for NBM, 50%*OBM*, 75%*OBM*, and OBM are 1.5, 1.12, 1.03, and 1.

Result 4. *For NBM, 50%OBM, 75%OBM, and OBM, if n and m are large, and* γ_1 *and* γ_0 *converge absolutely to a finite limit, then the optimal batch size*

$$m^* \approx \left[2n(1/c_v)^2(\gamma_1/\gamma_0)^2\right]^{1/3}. \tag{15}$$

In Equation (4), the growth rate of m^* with $n^{1/3}$ is proportional to the cube root of the square of the ratio γ_1/γ_0. This ratio can be written as

$$\frac{\gamma_1}{\gamma_0} = \sum_{h=1}^{\infty}\left(\frac{\rho_h^*}{\sum_{k=0}^{\infty}\rho_k^*}\right), \text{ where} \tag{16}$$

$$\rho_h^* = \begin{cases} 0.5, & h = 0 \\ \rho_h, & h = 1,2,\ldots \end{cases}$$

Song and Schmesier(1995) interpreted Equation (16) as the "balance point" of the absolute lags $|h|$, with forces represented by the correlations ρ_h. Pedrosa (1994) interpreted Equation (16) as the "center of gravity" of a stochastic process, which is similar to the term "center of gravity" in physics. In this paper, we refer to the fraction γ_1/γ_0 as the center of gravity.

3 The Mse-Optimal DPBM Algorithm

The DNBM and DPBM are parameterized by the batch size, which takes the value $m = 2^{\lceil \log_2 n/g \rceil - 1}$ (Yeh and Schmeiser 2000, Song 2007), where g is the pre-specified memory size and n is the total number of observations stored in the memory space. That is, the batch size used in DNBM and DPBM are completely determined by g and n. Therefore,

the value of the batch size used in DNBM and DPBM does not reflect the correlation structure of data.

To overcome the drawback common to both DNBM and DPBM, i.e., their inability to reflect the correlation structure of the data, we propose the mse-optimal DPBM, which first takes the value of DPBM(m_k) as the base line and then adjusts its value to reflect the correlation structure of the data when the batch size m_k is "far smaller" or "far larger" than m^*, where m^* is the minimal-mse batch size of DPBM satisfying

$$\text{mse}[\hat{V}_{\text{DPBM}}(m^*)] \leq \text{mse}[\hat{V}_{\text{DPBM}}(m)] \tag{17}$$

for any real value m.

The mse-optimal DPBM algorithm adjusts and improves the DPBM(m) via estimating the optimal batch and thereafter determining to either increase batch size or increase the numbers of batches. The estimation of the optimal batch requires two major steps: (1) to form another estimator which is called the B-DPBM, which will be introduced in Subsection 3.1, and (2) to estimate the center of gravity γ_1/γ_0.

3.1 The B-DPBM Estimator

In this section, we explain how we obtain the B-DPBM($m_k/2$) given DPBM(m_k). We will show in Theorem 5 that the B-DPBM($m_k/2$) is algebraically equivalent to $\hat{V}(m_k/2, s = m_k/2)$, the 50%OBM with batch size $m_k/2$. We need $8g$ storage to store batch means to implement the DPBM, but we need no more than 4 cells to obtain the B-DPBM.

As reviewed in Section 2.2, the data used to construct DPBM are batch means stored into four vectors $\underline{A}$, $\underline{B}$, $\underline{C}$, and $\underline{D}$. Specifically, assuming that the current step of the DPBM estimators is k, each cell of $\underline{A}$, $\underline{B}$, $\underline{C}$ and $\underline{D}$ contains one batch mean with the batch size $m_k = 2^k, k = 0, 1, 2, \ldots$. Our problem is how we can revisit batch means with the batch size $m_k/2$ given that the batch means at the previous step $k-1$ with the batch size $m_k/2$ are already overwritten at sept k.

In addition to the notation defined earlier, below we define more notation used to form B-DPBM.

- $\underline{A}^{'}, \underline{B}^{'}$: the virtual vectors used to store data used for the B-DPBM.
- $A^{'}_{k-1}(i), i = 1, 2, \ldots, \lceil \frac{n}{m_{k-1}} \rceil$: the batch sum stored in the i-th cell of the virtual vector $\underline{A}^{'}$.
- $B^{'}_{k-1}(i), i = 1, 2, \cdots, \lceil \frac{n-(m_{k-1}/2)}{m_{k-1}} \rceil$: the batch sum stored in the i-th cell of the virtual vector $\underline{B}^{'}$.

  The relationship between $A^{'}_{k-1}(i), B^{'}_{k-1}(i)$ and $A_k(i), B_k(i), C_k(i), D_k(i)$ is given in Equations (18) to (21).

$$A^{'}_{k-1}(2i-1) = \sum_{j=i}^{r_A} A_k(j) - \sum_{j=i}^{r_B} B_k(j), \tag{18}$$
$$i = 1, \cdots, \lceil \frac{n}{m_k} \rceil,$$
$$A^{'}_{k-1}(2i) = A_k(i) - A^{'}_{k-1}(2i-1), \tag{19}$$
$$i = 1, \cdots, \lfloor \frac{n}{m_k} + 0.5 \rfloor,$$

$$B^{'}_{k-1}(2i-1) = \sum_{j=i}^{r_C} C_k(j) - \sum_{j=i}^{r_D} D_k(j), \tag{20}$$
$$i = 1, \cdots, \lceil \frac{n-(m_k/4)}{m_k} \rceil,$$
$$B^{'}_{k-1}(2i) = C_k(i) - B^{'}_{k-1}(2i-1), \tag{21}$$
$$i = 1, \cdots, \lfloor \frac{n-(m_k/4)}{m_k} + 0.5 \rfloor,$$

- $b_{A'}, b_{B'}$: the full number of batches in the virtual vectors $\underline{A}^{'}$ and $\underline{B}^{'}$ in one-step backward DPBM. Specifically, $b_{A'} = \lfloor \frac{n}{m_{k-1}} \rfloor$ and $b_{B'} = \lfloor \frac{n-s}{m_{k-1}} \rfloor = \lfloor \frac{n-(m_{k-1}/2)}{m_{k-1}} \rfloor$.

We name $\underline{A}^{'}$ and $\underline{B}^{'}$ as virtual vectors because they are not actually stored in memory. All cells in $\underline{A}^{'}$ and $\underline{B}^{'}$ are just mathematical expressions, and as such can be obtained as functions of the current cells in vectors $\underline{A}, \underline{B}, \underline{C}$, and $\underline{D}$.

The total number of cells in $\underline{A}^{'}$ and $\underline{A}$ differ. The $\underline{A}^{'}$ contains $\lceil n/m_{k-1} \rceil$ cells, while $\underline{A}$ contains $2g$ cells, and $\lceil n/m_{k-1} \rceil \geq 2g$. For example, let $2g = 4, n = 12, k = 2$. $\underline{A}^{'}$ has $12/2 = 6$ cells, while $\underline{A}$ has only four cells. The values of $A^{'}_{k-1}(i)$ and $A_{k-1}(i)$ are identical for $i = 1, 2, \ldots, 2g$. A similar argument can be made for $B^{'}_{k-1}(i)$ and $B_{k-1}(i)$. The $\underline{B}^{'}$ contains $\lceil (n-(m_{k-1}/2))/m_{k-1} \rceil$ cells, while $\underline{B}$ contains $2g$ cells, and $\lceil (n-(m_{k-1}/2))/m_{k-1} \rceil \geq 2g$. The values of $B^{'}_{k-1}(i)$ and $B_{k-1}(i)$ are identical for $i = 1, 2, \ldots, 2g-1$.

The B-DPBM estimator as a function of $\underline{A}^{'}$ and $\underline{B}^{'}$ is defined as:

$$\hat{V}_{\text{B-DPBM}}(m_k/2) = \frac{1}{d_b}[\sum_{i=1}^{b_{A'}} (\frac{A^{'}_{k-1}(i)}{m_{k-1}} - \overline{\overline{Y}}_n)^2 + \sum_{i=1}^{b_{B'}} (\frac{B^{'}_{k-1}(i)}{m_{k-1}} - \overline{\overline{Y}}_n)^2], \tag{22}$$

where $m_{k-1} = m_k/2$, $d_b = b[(\frac{n}{m_{k-1}}) - 1]$; $b = \lfloor (n - m_{k-1} + s)/s \rfloor$; $A^{'}_{k-1}(i)$ and $B^{'}_{k-1}(i)$ are defined in equations (18) to (21).

Result 5. $\hat{V}_{\text{B-DPBM}}(m_k/2)$ *is algebraically equivalent to 50%OBM*$(m_k/2)$*, where* m_k *is the batch size at step* k *in implementing DPBM.*

Theorem 5 can be proved easily by plugging Equations (18) to (21) into the right hand side of Equation (22). We must note that $\hat{V}_{\text{B}-\text{DPBM}}(m_k/2)$ differs from $\hat{V}_{\text{DPBM}}(m_k/2)$ because $\hat{V}_{\text{B}-\text{DPBM}}(m_k/2)$ is 50%OBM and $\hat{V}_{\text{DPBM}}(m_k/2)$ is 75%OBM.

Obtaining the B-DPBM takes computation time $O(l^2)$ because it takes $O(l)$ to compute each $A'_{k-1}(2i-1), A'_{k-1}(2i), B'_{k-1}(2i-1)$, and $B'_{k-1}(2i)$, listed in Equations (18) and (21). It is noted that we need no more than 4 cells to obtain the B-DPBM, and these 4 cells are used to store the value computed in Equations (18) and (21).

3.2 Estimating the Optimal Batch size

This subsection discusses how to estimate the center of gravity for the stochastic process, and the optimal batch size for the DPBM estimator.

Result 6. *For large n and m, and assuming that γ_0 converge to a finite limit,*

$$\gamma_0 \approx n\text{E}(\hat{V}_{\text{B}}(m/2))/R_0 \tag{23}$$

Result 6 follows from Result 1 and simple algebra. Based on Result 6, a natural estimator of γ_0 is

$$\hat{\gamma}_0 \approx n\hat{V}_{\text{B}}(m/2)/\hat{R}_0, \tag{24}$$

where $\hat{R}_0 = \dfrac{(\sum_{i=1}^{n} Y_i^2 - n\overline{\overline{Y}})^2}{n-1}$, which can be computed using 2 storage space by maintaining a sum and sum of squares.

Result 7. *For large n and m, and assuming that γ_1 and γ_0 converge to a finite limit,*

$$\text{E}(\hat{V}_{\text{D}}(m)) \approx R_0\left(\frac{\gamma_0}{n} - \frac{\gamma_1}{mn}\right), \tag{25}$$

$$\text{E}(\hat{V}_{\text{B}}(m/2)) \approx R_0\left(\frac{\gamma_0}{n} - \frac{2\gamma_1}{mn}\right). \tag{26}$$

Result 7 is a direct consequence of Results 1 and 2.

Result 8. *For large n and m, and assuming that γ_1 and γ_0 converge to a finite limit,*

$$\gamma_1 \approx nm\left[\text{E}(\hat{V}_{\text{D}}(m)) - \text{E}(\hat{V}_{\text{B}}(m/2))\right]/R_0 \tag{27}$$

Result 8 follows from Result 7 and simple algebra. Based on Result 8, a natural estimator of γ_1 is

$$\hat{\gamma}_1 \approx nm\left[\hat{V}_{\text{D}}(m) - \hat{V}_{\text{B}}(m/2)\right]/\hat{R}_0. \tag{28}$$

Based on Results 4, 6 and 8, a natural estimator of the optimal batch size m^* is

$$\hat{m^*} \approx \left(1.12n(\frac{\hat{\gamma}_1}{\hat{\gamma}_0})^2\right)^{1/3} + 1. \tag{29}$$

3.3 Algorithm of Mse-Optimal DPBM

Having completed the discussion of the DPBM and B-DPBM, and the estimator of the optimal batch size $\hat{m}^*$, we now ready to develop an algorithm, called mse-optimal DPBM, for estimating the variance of the sample mean via estimating the optimal batch size of the 75%OBM, without knowing the sample size in advance.

Mse-optimal DPBM Algorithm:

Step 0 (Initialization). $n = 1; k = 0; m = 2^k = 1; L(i) = 0, i = 1, \ldots, 8g$;
$m_A = m_B = m_C = m_D = 1$; $r_A = r_B = r_C = r_D = 0$.

Step 1. If the r_A-th cell in $\underline{A}$ has room ($m_A < m$), then set $m_A \leftarrow m_A + 1$ and go to Step 5.

Step 2. If $\underline{A}$ has room (i.e., $r_A < 2g$), then go to (2.1); else go to (2.2).

(2.1) Initialize m_A and increment the current cell for $\underline{A}$, i.e., $m_A = 1$ and set $r_A \leftarrow r_A + 1$. Go to Step 4.

(2.2) Check whether this is the first collapse. If $k = 0$ go to (2.2.1); else if $k = 1$ go to (2.3); else (2.2.2).

(2.2.1) Collapse the vector to update first $\underline{B}$ and then $\underline{A}$. Initialize the values of m_A, m_B, r_A, and r_B.

- $B(i) = A(2i) + A(2i+1)$, $i = 1, \ldots, g-1$; $B(g) = A(2g)$.
- $A(i) = A(2i-1) + A(2i)$, $i = 1, 2, \ldots, g$.
- $m_A = 1, m_B = 2^k, r_A = g+1, r_B = g$.

Go to Step 3.

(2.2.2) To form the B-DPBM estimator and estimate the optimal batch size

- Compute $\hat{V}_B(m/2)$ (refer to Equation (22)).
- Compute $\hat{m}^* = \left(1.12n(\frac{\hat{\gamma}_1}{\hat{\gamma}_0})^2\right)^{1/3} + 1$. (refer to Equation (29)).
- Determine whether to increase the batch size or the number of batches: If $m \le \hat{m}^*$, go to Step (2.3); else let $g = g+1$ and go to Step 2.

(2.3) Collapse the vector to update first $\underline{D}$, then $\underline{C}$, then $\underline{B}$, then $\underline{A}$.
Initialize the values of m_A, m_B, m_C and m_D.

- $D(i) = B(2i) + B(2i+1)$, $i = 1, \ldots, g-1$; $D(g) = B(2g)$.
- $C(i) = B(2i-1) + B(2i)$, $i = 1, \ldots, g$.
- $B(i) = A(2i) + A(2i+1)$, $i = 1, \ldots, g-1$; $B(g) = A(2g)$.
- $A(i) = A(2i-1) + A(2i)$, $i = 1, \ldots, g$.
- $m_A = 1, m_B = 2^k, m_C = 2^k + 2^{k-1}, m_D = 2^{k-1}$.
- $r_A = g+1, r_B = g, r_C = g, r_D = g$.

Step 3. Update the total number of collapses, and the batch size. $k = k+1$, $m = 2^k$.

Step 4. Initialize the sum stored in the current cell $A(r_A)$, i.e., $A(r_A) = 0$.

Step 5. Add the new observation y_n in the current cell in $\underline{A}$, i.e., $A(r_A) = A(r_A) + y_n$.

Step 6. Add the new observation y_n in the current cell in vector $\underline{J}$, where $J = B$ or C or D.

(6.1) If the r_J-th cell in vector $\underline{J}$ has room ($m_J < m$), then set $m_J \leftarrow m_J + 1$ and go to (6.3).

(6.2) Initialize the value of m_J and the sum stored in the current cell in $J(r_J)$.
$m_J = 1, r_J = r_J + 1, J(r_J) = 0$.

(6.3) $J(r_J) = J(r_J) + y_n$.

Step 7. If there is no new observation, compute the variance estimator $\hat{V}_{\mathrm{B}}(m_B)$, where m_B is defined as

$$m_B = m/2; \tag{30}$$

else update the sample size (i.e., $n = n+1$) and return to Step 1.

It is noted that in Step (2.2.2), we increase g by 1 each time when $m > \hat{m}^*$. It might save computational effort to increase g more than 1, for example, doubling the value g. In future research, one could investigate how to increase the storage space efficiently when the algorithm determines to increase the number of batches instead of increasing the batch size.

4 The Performance of Mse-Optimal DPBM Procedure

We evaluate the $\hat{m}^*$ on the first-order autoregressive (AR(1)) processes with $\phi = 0.82, 0.96$ and 0.98. The AR(1) process is defined as $Y_i = \mu + \phi(Y_{i-1} - \mu) + \varepsilon_i; i = 1, 2, \ldots, n$, where the ε_i's are identically independent distributed normal random variables with mean 0 and variance $(1-\phi^2)\mathrm{var}(Y)$. The correlation of the AR(1) at lag h is $\rho_h = \rho^{|h|}$, where for simplicity we use $\rho \equiv \rho_1$. The relationships for $\rho, \gamma_0, \mathrm{var}(Y), \mathrm{var}(\overline{\overline{Y}}_n)$, and n are

$$\rho = (\gamma_0 - 1)/(\gamma_0 + 1),$$

$$\mathrm{var}(Y) = \frac{n\mathrm{var}(\overline{\overline{Y}}_n)}{1 + 2\sum_{h=1}^{n-1}(1 - h/n)\rho_h} = \frac{n\mathrm{var}(\overline{\overline{Y}}_n)}{\frac{1+\rho}{1-\rho} - \frac{2\rho(1-\rho^n)}{n(1-\rho)^2}}.$$

The corresponding parameters for the AR(1) processes are listed in Table 1.

The sample size n (Row 4) guarantees a 95% confidence interval on the mean μ with a half length equal to 0.05μ (Wilson, 1979). The parameters of the three processes are selected such that $\mathrm{E}(Y) = 0$ and $\mathrm{var}(\overline{\overline{Y}}_n) = 1$. The ϕ for all three processes is greater than 0.82, which indicates that the correlation of the simulation outputs is moderately large.

The actual optimal batch size m^* (Row 5), defined in Equation (17) for DPBM (75%OBM) is obtained via the simulation experiments. In practice, m^* can not be computed because the parameters of the data process are unknown.

Table 1: Performance for AR(1) Processes

Parameters				
ϕ	0.82	0.96	0.98	1
γ_0	10.0	50	100	2
γ_1	49.5	1250	5000	3
n	500	2500	5000	4
m^*	24	125	242	5
$\mathrm{mse}(\hat{V}_{\mathrm{D}}(m^*))$	0.1	0.1	0.1	6
$\mathrm{mse}(\hat{V}_{\mathrm{B}}((m_{\mathrm{B}})))$	0.12	0.12	0.12	7
increase ratio	20%	20%	20%	8

Note that the last column indicates the row numbers.

We apply leading-digit rules (Song and Schmeiser 2008) to report the estimates in the last three rows. That is, we report the point estimate through the leading digit of the standard error. The estimated mse of the B-DPBM (Row 7) does not differ much (about 20%) (Row 8) from the ideal mse value for the DPBM (Row 6). The three AR(1) processes studied here encourage future research for the evaluation of more-general processes.

5 Conclusion and Future Research

This paper proposes a mse-optimal batching algorithm that requires limited memory space and reasonable computation time, and has good statistical properties such as small mean-squared-error (mse), but does not require knowledge of the simulation run length *a priori*. From a theoretical point of view, this paper does not significant extend previous research, since the key formula used in this paper is Song and Schmeiser's optimal batch size approximation and Song's DPBM algorithm. In terms of the practical application, however this paper represents an advance in that it present

an implementable batch-size selection procedure without requiring that the sample size *a priori*.

There are several issues that merit future research. For instance, an area that deserves further exploration is the possible advantage of using partial batches into the DPBM in order to achieve better statistical property such as mse. Alternatively, one could investigate how to increase the storage space efficiently when the algorithm determines to increase the number of batches instead of increasing the batch size. Moreover, the three AR(1) processes studied in this paper encourage future research for the evaluation of more-general processes.

Acknowledgment

This work is supported by the National Science Council of the Republic of China under grant NSC-95-2221-E-007-175. The authors also thank NCHC (National Center for High-performance Computing) for providing the high speed computer to run the simulation experiments.

References

Aktaran-Kalaycı, T., Alexopoulos, C., Argon, N.T., Goldsman, D. and Wilson, J.R. (2007) Exact expected values of variance estimators for simulation. *Naval Research Logistics,* **54**, 397–410.

Conway, R.W. (1963) Some tactical problems in digital simulation. *Management Science,* **6**, 92–110.

Crane, M.A. and Iglehart, D.L. (1975) Simulating stable stochastic systems, III: regenerative process and discrete-event simulations. *Operations Research,* **23**, 33–45.

Ferguson, N.M., Cummings, D.A., Fraser, C., Cajka, J.C., Cooley, P.C. and Burke, D.S. (2006) Strategies for mitigating an influenza pandemic, *Nature*, **442**, 448–452.

Fishman, G.S. (1978) Grouping observations in digital simulation, *Management Science*, **24**, 510–521.

Fishman, G.S., and Yarberry, L.S. (1997) An implementation of the batch means method, *INFORMS Journal on Computing*, **9**, 296–310.

Foley, R.D. and Goldsman, D. (1999) Confidence Intervals using orthogonally weighted standardized time series, *ACM Transactions on Modeling and Computer Simulation*, **9**, 297–325.

Glynn, P.W. and Iglehart, D.L. (1986) Estimation of steady-state central moments by the regenerative method of simulation. *Operations Research Letters,* **5**, 271–276.

Glynn, P.W. and Iglehart, D.L. (1990) Simulation output analysis using standardized time series. *Mathematics of Operations Research,* **15**, 1–16.

Heidelberger, P., and Welch, P.D. (1981) A spectral method for confidence interval generation and run length control in simulation. *Communication of the ACM,* **24**, 233–245.

Jenvald, J., Morin, M., Timpka, T. and Eriksson, H. (2007) Simulation as decision support in pandemic influenza preparedness and response. *Proceedings of ISCRAM,* 295–304.

Lall, A., Sekar, V., Ogihara, M., Xu, J., and Zhang, U. (2006) Data streaming algorithms for estimating entropy of network traffic. ACM SIGMETRICS/Performance Conference, 145–156.

Law, A. M., and J. S. Carson. (1979). A sequential procedure for determining the length of a steady-state simulation. *Operations Research*, 27, 1011–1025.

Meketon, M.S. and Schmeiser, B.W. (1984) Overlapping batch means: something for nothing? *Proceedings of the Winter Simulation Conference*, 227–230.

Moran, M.S. (1975) The estimation of standard errors in Monte Carlo simulation experiments. *Biometrika,* **62**, 1–4.

Pedrosa, A. and Schmeiser, B.W. (1994) Estimating the variance of the sample mean: Optimal batch-size estimation and 1-2-1 overlapping batch means. Technical Report SMS94-3, School of Industrial Engineering, Purdue University, West Lafayette, IN.

Priestley, M.B. (1981) *Spectral Analysis and Time Series*. Academic Press, London.

Schmeiser, B.W. (1982) Batch size effects in the analysis of simulation output. *Operations Research,* **30**, 556–568.

Schruben, L.W. (1983) Confidence interval estimation using standardized time series. *Operations Research*, **31**, 1090–1108.

Song, W.-M. T. (1988) Estimators of the variance of the sample mean: quadratic forms, optimal batch sizes, and linear combinations. Ph.D. Dissertation, School of Industrial Engineering, Purdue University, W. Lafayette, ind, USA.

Song, W.-M. T. (1996) On the estimation of optimal batch sizes in the analysis of simulation output. *European Journal of Operational Research*, **88**, 304–319.

Song, W.-M. T. (2007) A finite-memory algorithm for estimating the variance of the sample mean. *IIE Transactions*, **39**, 703–711.

Song, W.-M. T. and Schmeiser, B.W. (1995) Optimal mean-squared-error batch size. *Management Science*, **41**, 110–123.

Song, W.-M.T., and Schmeiser, B.W. (2008) Omitting Meaningless Digits in Point Estimates: The Probability Guarantee of Leading-Digit Rules, *Operations research*, forthcoming.

Song, W.-M.T., and Schmeiser, B.W. (1988) Minimal-mse linear combinations of variance estimator of the sample mean. *Proceedings of the Winter Simulation Conference*, 414–421.

Welch, P.D. (1987) On the relationship between batch means, overlapping batch means and spectral estimation .*Proceedings of the Winter Simulation Conference*, 320–323.

Wilson, J.R. (1979) Variance reduction techniques. Ph.D. Dissertation, School of Industrial Engineering, Purdue University, West Lafayette, IN.

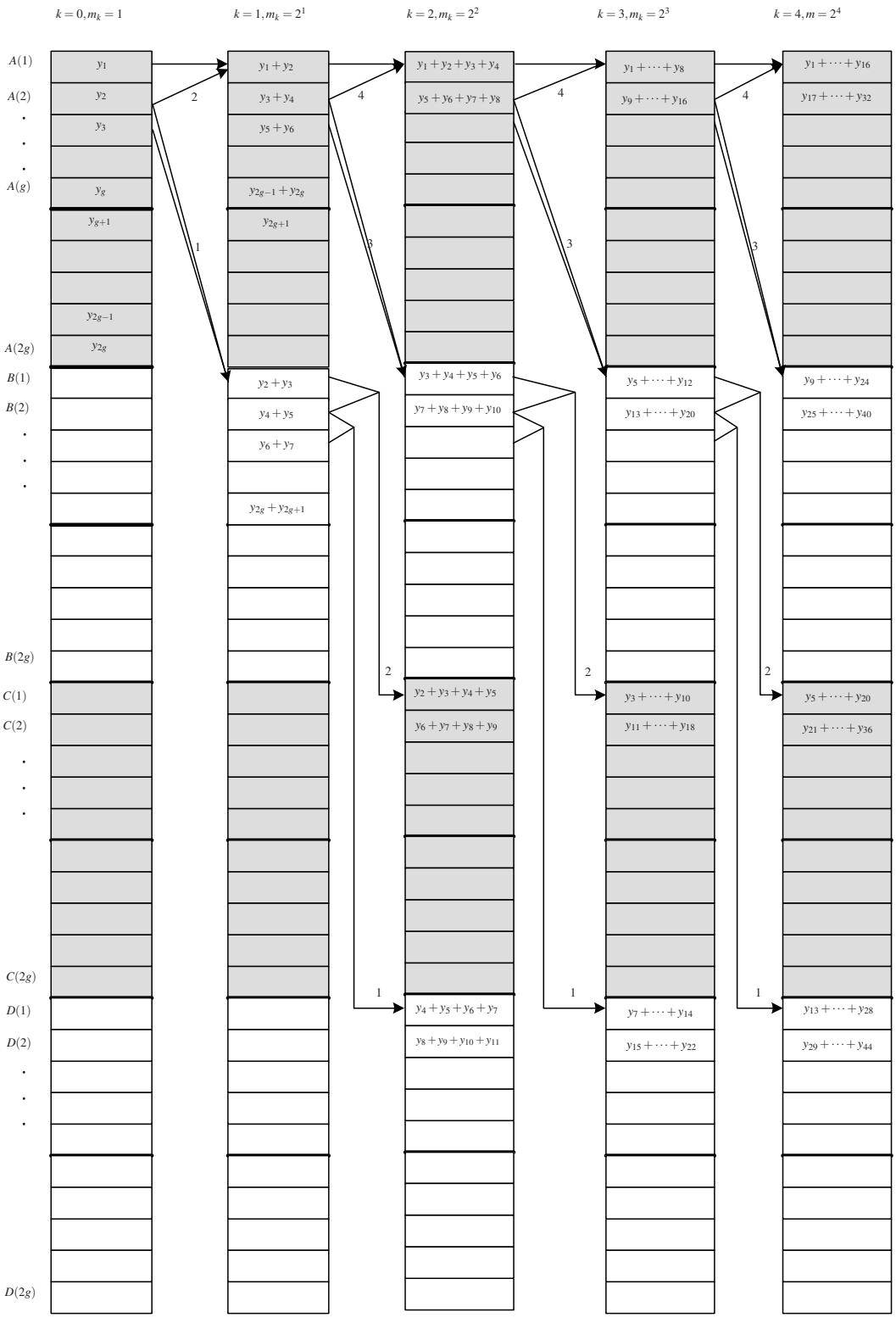
Yeh, Y. and Schmeiser, B.W. (2000) Simulation output analysis via dynamic batch means. *Proceedings of the Winter Simulation Conference*, 637–645.

Zhao, H., A. Lall, M. Ogrihara, O. Spatscheck, J. Wang, and Wu, J. (2007). A data streaming algorithm for estimating entropies of OD flows. *Proceedings of the ACM Internet Measurement Conference*.

AUTHOR BIOGRAPHIES

WHEYMING T. SONG is a professor in the Department of Industrial Engineering at the National Tsing Hua University in Taiwan. She received her undergraduate degree in statistics and masters degree in industrial engineering at Cheng-Kung University in Taiwan in 1979. She then received masters degrees in applied mathematics in 1983 and industrial engineering in 1984, both from the University of Pittsburgh. Dr. Song received her Ph.D. from the School of Industrial Engineering at Purdue University in 1989. She Joined Tsing Hua in 1990 after spending one year as a visiting assistant professor at Purdue IE. Her research interests are applied operations research; probability and statistics; and the statistical aspects of stochastic simulation, including input modeling, output analysis, variance reduction, and ranking and selection. <wheyming@ie.nthu.edu.tw>.

MINGCHANG CHIH received his undergraduate degree in Industrial and Systems Engineering from Chung Yuan Christian University in Taiwan in June 2001. He then received Master's degree in Industrial Management from National Taiwan University of Science and Technology in Taiwan in June 2003. He is now a Ph.D. candidate at National Tsing Hua University. <d927805@oz.nthu.edu.tw>.



Note: The numbers $1, 2, 3$ and 4 listed on the arrows indicate the updating order in each iteration

Figure 2: The idea of collapsing in DPBM