FILTERED GIBBS SAMPLER FOR ESTIMATING BLOCKING IN PRODUCT FORM NETWORKS

Lachlan L. H. Andrew[*] and Felisa J. Vázquez-Abad[†]
The ARC Special Research Centre for Ultra-Broadband Information Networks (CUBIN)
Department of Electrical and Electronic Engineering, The University of Melbourne, Victoria 3010, Australia
Email: l.andrew@ee.mu.oz.au, fva@ee.mu.oz.au

Abstract

The filtered Gibbs sampler (FGS) is a very simple yet efficient method of performance evaluation for a class of communication networks. This paper confirms that the FGS is considerably more efficient than the standard Gibbs sampler. It also shows that estimates of the accuracy of the results can be excessively optimistic, and presents a more conservative estimate.

Keywords— Simulation techniques; maximum packing channel allocation

1 Introduction

Product form stationary distributions arise in many models for telecommunications systems, notably truncated multiclass $M/G/\infty$ queues. They model traditional circuit switched networks with fixed routing, cellular networks with frequency-reuse constraints [1–3], packet networks with fixed routing using effective bandwidth admission control [4,5] or with marking-based admission control [6], or intelligent networks in which connections require a particular set of services for the duration of the call [7]. Many other applications are listed in [8].

The importance of product form networks has led to many techniques for analysing them [9]. The primary performance measure for circuit switched models is the blocking probability. This may be calculated from the normalising constant (G of (1)); G may be calculated by convolutional methods [10,11], numerical inversion of generating functions [12] or by Monte Carlo integration [3, 13].

This paper investigates the performance of Markov chain Monte Carlo simulation as an alternative means of estimating blocking probabilities in product form networks [14,15] (see also [16]). In addition to blocking analysis, these algorithms can generate actual samples from the state distribution, which can be used as is done in [17].

The next section defines the task of estimating blocking probabilities, and Section 3 introduces the filtered Gibbs sampler (FGS). The FGS is numerically tested on the example networks described in 4, with the results presented in Sections 5 and 6. Section 7 shows that confidence intervals based on standard estimates of variance can be significantly too small, and proposes a more conservative variance estimator.

2 Network Model

A truncated multiclass $M/G/\infty$ network has an R-dimensional state space, $\mathcal{S} \subset \mathbb{N}^R$, with a stationary state distribution

$$\pi(n) = \frac{1}{G} \prod_{i=1}^{R} \left(\frac{\rho_i^{n_i}}{n_i!} \right), \quad n = (n_1, \ldots, n_R) \in \mathcal{S} \qquad (1)$$

where $\rho_i = \lambda_i/\mu$ is the load offered to class i, λ_i is the arrival rate for class i, $1/\mu$ is the common mean service time, and G normalises π so that $\sum_n \pi(n) = 1$.

In circuit switched networks, the R classes are distinct routes through the network and n_i is the number of calls currently using route i. If the network can support a particular combination of calls, then it can also support any subset of those calls. Thus for any feasible $n = (n_1, \ldots, n_R) \in \mathcal{S}$, we have $\{n' : n'_i \le n_i\} \subseteq \mathcal{S}$.

The form of the feasible region, $\mathcal{S}$ is often

$$\mathcal{S} = \{n \in \mathbb{N}^R : An \le \mathbf{C}\} \qquad (2)$$

(but [7, 18] give exceptions). Here $A = [a_{ji}] \in \{0,1\}^{L\times R}$ (or more generally $\mathbb{N}^{L\times R}$) specifies the number of channels required by route i on link j, and $\mathbf{C} = (C_i) \in \mathbb{N}^L$ is a vector of the number of channels available on each link.

Let B be the network blocking probability. A feasible state, n, is a blocking state for route i if one more call on route i would lead to an infeasible state. The set of blocking states for route i, $i = 1, \ldots, R$, is

$$\mathcal{B}_i = \{n \in \mathcal{S} : \exists j, a_{ji} + (An)_j > C_j\}. \qquad (3)$$

Let $B_i = \mathsf{P}(n \in \mathcal{B}_i)$ be the blocking probability of route i. Writing $\lambda = \sum_{i=1}^{R} \lambda_i$ for the total arrival rate gives

$$B = \sum_{i=1}^{R} \left(\frac{\lambda_i}{\lambda} \right) B_i. \qquad (4)$$

[*] Supported by the Australian Research Council (ARC).

[†] Supported by NSERC-Canada grant # WFA0184198 and ARC.

3 Markov Chain Monte Carlo Simulation

Evaluating blocking probabilities using (1) directly is a difficult numerical problem for realistic sized networks. Moreover, in many cases, it is not sufficient to know the blocking probability, and it is desirable to sample from the distribution itself (see for example [17]). Monte Carlo techniques, such as the FGS, bridge the gap between exact algorithms [10–12] and approximations [8,19,20]. They allow a quantifiable tradeoff between computational time and accuracy, while being conceptually simple.

This section presents the construction of a "surrogate" Markov chain $\{X_k : k = 1, 2, \ldots\}$ with state space $\mathcal{S}$ whose steady state probabilities are given exactly by π in (1), that is:

$$\forall\, n \in \mathcal{S} \quad \lim_{k\to\infty} \mathsf{P}(X_k = n) = \pi(n). \tag{5}$$

Such methods are called Markov chain Monte Carlo (MCMC) methods. Then B can be estimated as $\hat{Y}(S) = (1/S)\sum_{i=1}^{S} y(x_i)$ for any function $y(\cdot)$ with $\mathsf{E}[y(X)] = B$.

A fixed relative accuracy, $\mathsf{Var}[\hat{Y}(S)]/B^2$, can be obtained faster by either decreasing the CPU time required to evaluate $y(X)$ or by using an estimator of B with reduced variance. This tradeoff is quantified by the *relative efficiency* defined by

$$\mathcal{E}_r(\hat{Y}) = \lim_{S\to\infty} \frac{B^2}{\text{CPU}[\hat{Y}(S)]\mathsf{Var}[\hat{Y}(S)]},$$

where $\text{CPU}[\hat{Y}(S)]$ denotes the average CPU time of the simulation that produces the S samples.

Note that it is not necessary for the S replications to be independent. However, if there is significant correlation between them, then $\mathsf{Var}[\hat{Y}(S)]$ may be very much larger than $\mathsf{Var}[\hat{Y}(1)]/S$, which would have resulted from independent samples. Thus, in addition to having the desired steady state distribution, a good surrogate process will have a lower correlation between successive states than the simple arrival/departure process. This can reduce the variance of the final estimate of the blocking probability by orders of magnitude.

One good MCMC method is the Gibbs sampler [21]. After describing the standard Gibbs sampler, this section presents an enhancement called the *filtered* Gibbs sampler.

3.1 The Standard Gibbs Sampler

The Gibbs sampler applies to multi-dimensional state spaces. The key principle is that each transition in the surrogate Markov chain updates only one component, but the transition probabilities are proportional to the (known) stationary conditional probabilities for that component given the current values of all other components. This is clearly ideally suited to product form distributions, where these conditional probabilities have a very simple form. It is the ability to make large changes to each component which reduces the correlation between samples generated by a Gibbs sampler, and the use of the stationary probabilities in the updates which results in a high overall efficiency.

In the following, the algorithms for generating state X_{k+1} from X_k require the following notation. For $X \in \mathbb{N}^R$, define:

$$X^{-j} = (X(1), \ldots, X(j-1), X(j+1), \ldots, X(R)),$$

which is a vector in $\mathbb{N}^{R-1}$, missing component j. Given any $x \in \mathcal{S}$ and an index $1 \le j \le R$, the notation $\pi(\cdot|x^{-j})$ is used for the conditional probability of the jth component given all the others:

$$\pi(y|x^{-j}) = \mathsf{P}(X(j) = y|X^{-j}\!= x^{-j}) = \frac{\pi(x_y)}{\sum_{x(j)=0}^{C_j(x)} \pi(x)},$$

where $x_y(i) = x(i)$ for $i \neq j$ and $x_y(j) = y$, and $C_j(x)$ is the state dependent bound such that all states in the sum in the denominator lie in $\mathcal{S}$.

Definition 1 *A* Gibbs Update *is a rule for generating X_{k+1} from X_k of the form:*
1. Select $\sigma_k \in \{1, \ldots, R\}$, independent of the history of the process.
2. Set $X_{k+1}(\sigma_k) \sim \pi(\cdot|X_{k+1}^{-\sigma_k})$ and leave all other components unchanged.

The key property of Gibbs updates is that if X_k is distributed according to π (denoted $X_k \sim \pi$) then $X_{k+1} \sim \pi$.

Under some technical assumptions, a Markov chain formed from Gibbs updates is called a Gibbs Sampler. For example, if σ_k are i.i.d. random variables then $\{X_k\}$ forms a Markov chain, while if $\sigma_k = k(\text{mod } R)$, then $\{(X_k, \sigma_k)\}$ forms a Markov chain, as does every Nth sample, $\{X_{Nk}\}$.

For (1), $\pi(\cdot|X_{k+1}^{-\sigma_k})$ is a one dimensional Poisson distribution truncated by (2). For each $1 \le j \le R$, let

$$P_j(m) = \sum_{n=0}^{m} \frac{\rho_j^n}{n!} \quad m = 1, \ldots, C, \tag{6}$$

where $C = \max_i C_i$. Let $Z_i(X) = C_i - \sum_{c\in L_i} a_{ic}X(c)$ be the number of free channels on link i in state X, where $L_i = \{j : a_{ij} \neq 0\}$ is the set of all routes using the ith link. At every step k, let $j = \sigma_k$ and let

$$C_j(X_k) = \min_{i:j\in L_i} (Z_i(X_k)/a_{ij} + X_k(j)) \tag{7}$$

be the maximum allowable number of connections using route j given X_k^{-j}. Then the required conditional probability satisfies $\mathsf{P}(X_{k+1}(j) \le m) = P_j(m)/P_j(C_j(X_k)), m = 0, \ldots, C_j(X_k)$.

Since $X_k \sim \pi$, it is possible to estimate B_i by $(1/S)\sum_{k=1}^{S} \mathbf{1}_{\{X_k\in\mathcal{B}_i\}}$, where $\mathbf{1}_{\{A\}} = 1$ if A is true, 0 otherwise. However since updates to component j only change $\mathbf{1}_{\{X_k\in\mathcal{B}_i\}}$ when i and j share a link, this involves significant unnecessary computation. Having evaluated $C_j(X_k)$ and X_{k+1}, it is easy to calculate $\mathbf{1}_{\{X_{k+1}\in\mathcal{B}_j\}} = \mathbf{1}_{\{X_k(j)=C_j(X_k)\}}$ for the component j which is updated at iteration k. Thus B_i be estimated by

$$Y_i(S) = \frac{1}{S(i)}\sum_{k=1}^{S} y_i(X_k)\mathbf{1}_{\{\sigma_k=i\}} \tag{8}$$

where $y_i(X_k) = \mathbf{1}_{\{X_{k+1}\in\mathcal{B}_i\}}$, and $S(i) = \sum \mathbf{1}_{\{\sigma_k=i\}}$ counts the number of iterations where $\sigma_k = i$. These *local estimates* converge to B_i at rate $\mathcal{O}(S^{-1/2})$ as S increases.

3.2 Filtered Gibbs Sampler

Consider a Markov chain $\{X_k\}$ and an estimator of the form:

$$\bar{X}_S = \frac{1}{S}\sum_{k=1}^{S} f(X_k).$$

The method of *filtered Monte Carlo* is based on conditioning at each stage [21]:

$$\bar{X}'_S = \frac{1}{S}\sum_{k=1}^{S} \mathsf{E}[f(X_{k+1})|X_k].$$

This is closely related to "inverse convolution" of [22].

Applying this method to the Gibbs samplers requires evaluation of the conditional probabilities:

$$\begin{aligned}\mathsf{P}(X_{k+1}\in\mathcal{B}_j|X_k) &= \frac{P_j(C_j(X_k)) - P_j(C_j(X_k)-1))}{P_j(C_j(X_k))}\\ &\equiv g(C_j(X_k);\rho_j)\end{aligned} \tag{9}$$

where $P_j(\cdot)$ are given in (6) and $C_j(X_k)$ is given in (7). When it is feasible to pre-compute $g(\cdot;\cdot)$, calculation of the probabilities is as simple as reading a table. This is the case when there is a small number of distinct loads, ρ_j, in the network.

The FGS combines the filtering with the distribution of the estimation via the local estimates as follows.

Definition 2 *The* Filtered Gibbs Sampler *(FGS) is constructed from the chain $\{X_k\}$ with Gibbs updates using $\sigma_k = k \operatorname{mod}(R) + 1$, by evaluating the sample average:*

$$\hat{Y}(S) = \frac{R}{S}\sum_{k=1}^{S}\left(\frac{\lambda_{\sigma_k}}{\lambda}\right) y_{\sigma_k,F}(X_k), \tag{10}$$

where $y_{i,F}(x) = g(C_i(x);\rho_i) = \mathsf{P}(X_{k+1}\in\mathcal{B}_i|X_k = x)$.

Each of the periodic Gibbs samplers embedded in the computation of (10) is dedicated to estimating $\mathcal{B}_{\sigma_k}$. Since $S(\sigma_k)/S \to 1/R$ as $S\to\infty$, it follows that under the FGS, $\hat{Y}(S)\to B$ [15]. Filtering can be applied analogously for other sequences, σ_k.

Unlike most exact techniques whose complexity is $\mathcal{O}(C)$, the complexity per iteration of the FGS is $\mathcal{O}(1)$ as the capacity per link increases. However, its primary strength is that it is $\mathcal{O}(R\max_i |L_i|)$ as the number of nodes and links increases. The complexity of all known exact methods is exponential in the number of links.

4 Test networks

The FGS was tested on two network topologies:

(a) Mesh-torus: a rectangular grid with each node connected to four neighbours, wrapping at the edges. Components of the state vector n are the numbers of current calls on a route. In the experiments, the load on all routes was equal. Static shortest path routing ensured a constant number of routes used each link.

(b) Cellular: Spatial reuse constraints in cellular networks with dynamic channel assignment produce "cliques" of cells with a maximum aggregate number of calls [1]. These cliques are analogous to links, while cells correspond to routes. The networks considered here employ a hexagonal grid of cells, and cliques consist of groups of three mutually adjacent cells.

5 Correlation

For a single random variable, $\mathsf{Var}[Y] = \mathsf{Var}[\mathsf{E}[Y|Z]] + \mathsf{E}[\mathsf{Var}[Y|Z]]$, and conditioning always entails a variance reduction. However, it is not always the case for Markov chains that $\mathsf{Var}[\hat{Y}'_S] \le \mathsf{Var}[\hat{Y}_S]$, due to the correlation structure [21]. Explicitly,

$$\begin{aligned}\mathsf{Var}[\hat{Y}_i(S)] &= \frac{1}{S}\mathsf{Var}[y_i(X_1)]\\ &\quad + \frac{2}{S^2}\sum_{j=1}^{S-1}\sum_{k=1}^{S-k}\mathsf{Cov}[y_i(X_j), y_i(X_{j+k})],\end{aligned}$$

and an increase in the second term may exceed the decrease in the first term.

The variance $\mathsf{Var}[\hat{Y}_i(S)]$ can be estimated using batch means (grouping runs of K samples to obtain approximately independent estimates [23]). The impact of the correlation can be quantified by the ratio of $\mathsf{Var}[\hat{Y}_i(S)]$ to $\mathsf{Var}[y_i(X_1)]/S$, the variance estimated by treating individual samples as independent.

Figure 1 shows the results of using batches of size $K = 3\times 10^6$ (10000 for each of the 300 routes) in a 5×5 mesh-torus, for both the FGS and the standard Gibbs sampler. (Note that these only show the impact of correlation, and do not compare the actual variances of FGS and the standard Gibbs sampler.) These results show that the covariance term has minimal impact except when blocking is very high. This justifies ignoring its effect in arguing that filtering should reduce the variance of the estimated blocking probability. However, when blocking is high, the variance of the final blocking estimator using FGS is up to an order of magnitude higher than would be predicted by treating samples as independent. Since this does not occur without filtering, the benefit due to filtering would be overestimated in the case of high blocking if batch means were not used. This effect is greatest for networks with many channels per link, as they have a higher occupancy per channel for a given blocking probability, due to increased trunking efficiency.

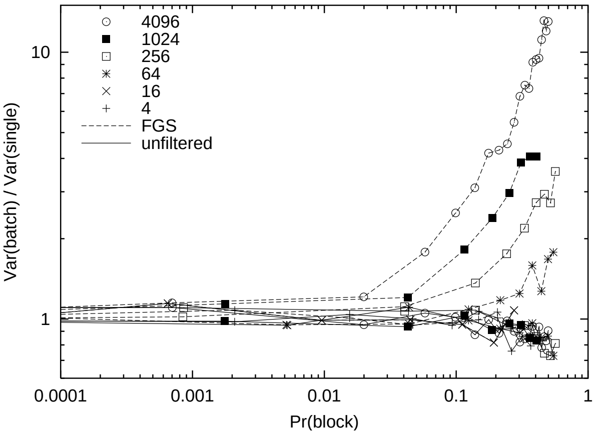

Figure 1. Ratio of variance of a 4-hop route estimated by batch means, $K = 3 \times 10^6$, divided by $\mathsf{Var}[y_i(X_1)]/S$. 5×5 mesh-torus network, 4 to 4096 channels per link.

Figure 1 suggests that, for high blocking, the true variance of the standard Gibbs sampler is actually less than would be predicted by treating samples as independent. This indicates a negative correlation between samples, but the reason for this is unclear.

6 Improvement due to filtering

Consider a single link of C channels, used by N routes of load ρ each, and assume that the blocking probability, B, is low. As was demonstrated in Section 5, for low blocking the variance of FGS is dominated by the variance of each update, rather than the covariance introduced by the Markov structure. We will now quantify the variance reduction due to the conditioning. Let

$$D_A = \sum_{j=0}^{C} A^j/j!$$

and note that for $B << 1$ (small A or large C), $D_A \approx e^A$. Denote the Erlang loss function by

$$E_k(\rho) = \frac{\rho^k/k!}{\sum_{j=0}^{k} \rho^j/j!}.$$

The blocking probability of the link is $B = E_C(N\rho)$, and the variance of the Gibbs sampler estimator is $B - B^2$.

For low B, the occupancy of the N routes is well approximated by independent Poisson variables. Each FGS update will see the link filled with the aggregate of the $N-1$ other routes, which is Poisson with rate $(N-1)\rho$. Thus with probability

$$\frac{((N-1)\rho)^{C-j}/(C-j)!}{D_{(N-1)\rho}},$$

the FGS estimate is $E_j(\rho)$. Thus

$$\begin{aligned}&\mathsf{Var}[FGS] + B^2 \\ &\quad = \sum_{j=0}^{C} \frac{((N-1)\rho)^{C-j}/(C-j)!}{D_{(N-1)\rho}} \left(\frac{\rho^j/j!}{\sum_{k=0}^{j} \rho^k/k!}\right)^2,\end{aligned}$$

and

$$\begin{aligned}&\frac{\mathsf{Var}[FGS] + B^2}{\mathsf{Var}[GS] + B^2} \\ &\quad = \frac{D_{N\rho}}{D_{(N-1)\rho}} \sum_{j=0}^{C} \frac{C! \quad (N-1)^{C-j}}{(C-j)!j! \quad N^C} \frac{\rho^j/j!}{(\sum_{k=0}^{j} \rho^k/k!)^2} \\ &\quad = e^\rho \left(\frac{N-1}{N}\right)^C \sum_{j=0}^{C} \binom{C}{j} \left(\frac{1}{N-1}\right)^j \frac{E_j(\rho)}{\sum\limits_{k=0}^{j} \rho^k/k!} \qquad (11) \\ &\quad \to \left(\frac{N-1}{N}\right)^C \text{ as } \rho \to 0. \qquad (12)\end{aligned}$$

This analysis extends easily to unequal loads.

Figure 2 shows the increase in the relative efficiency of the FGS compared to a standard Gibbs sampler for 5×5 and 200×200 cellular networks ($N = 3$) and 5×5 and 7×7 mesh-torus networks ($N = 15, 42$). The results are very similar for both cellular networks, while the results differ for the two mesh-torus networks. This is because cellular networks have $N = 3$ cells per clique, while the values of N differ greatly for the mesh-tori.

As indicated by (11), the gain in relative efficiency due to conditioning increases as the capacity of the of the links increases. It is a minimum in the range of blocking probabilities which are of greatest interest, around 10^{-2} to 10^{-3}. However, even in this range the gains are substantial for networks with many channels per link.

7 Confidence Intervals

In contrast to the Bernoulli outcomes of standard Monte Carlo, FGS produces samples from an unknown and highly skewed distribution. This makes it possible to underestimate the variance of the estimator by orders of magnitude if insufficient samples are taken. Figure 3 shows the estimate of blocking after each iteration, and also the value ("traditional upper") which is usually used as the upper limit of a confidence interval, i.e., the estimated mean plus twice the estimated standard deviation. After a small number of samples, this "2σ" upper limit is below the true value for much more than 2.5% of the time (which it would be in the Gaussian case), and is ineffective as a confidence bound.

To see why this occurs, consider the terms in

$$B_i = \sum_{j=0}^{C} g(j; \rho_i) \mathsf{P}(C_i(X) = j). \qquad (13)$$

Without filtering, $y_j(X) = 1$ if $C_i(X) = x_i$ and 0 otherwise. If at least one non-zero sample is generated then the

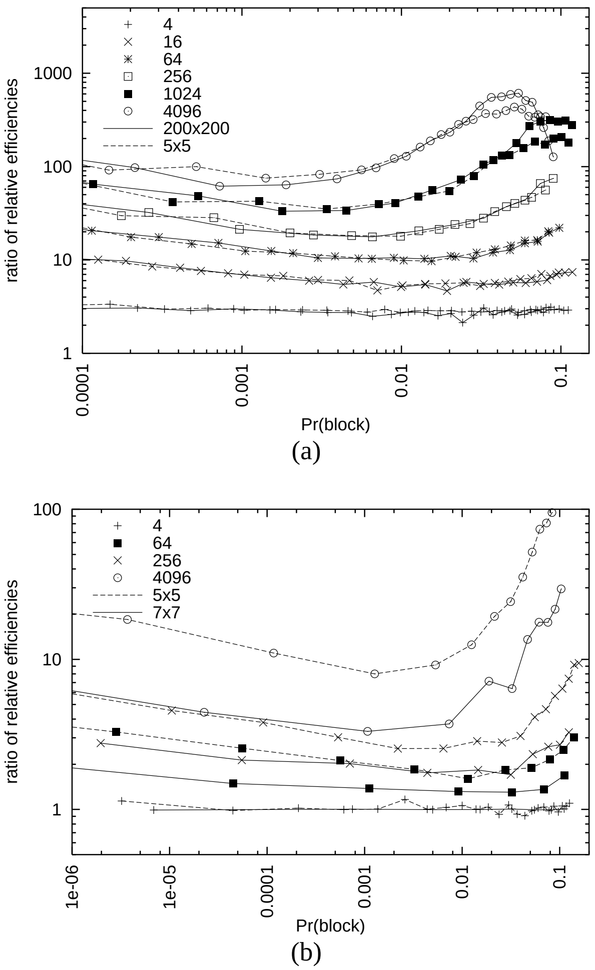

Figure 2. Ratio of efficiency of filtered to standard Gibbs sampler with 4 to 4096 channels per link for (a) 5×5 and 200×200 cellular (b) 5×5 and 7×7 mesh-torus networks.

variance estimator will, with high probability, be of the correct order of magnitude. If all samples are 0, it is clear that the sample variance (zero) is not a true indication of the error. However, this is not the case for highly skewed continuous distributions. There are many non-zero terms in (13) which have a high probability, but make very little contribution to the sum due to small values of $g(j;\rho_i)$. Thus if the sample size is too small, the sample mean and variance can be very much smaller than the ensemble values, without any tell-tale zeros to indicate their unreliability. For the FGS to be of practical value, it is necessary to be able to detect when an estimate is statistically unreliable.

For a better indication of the accuracy of the result, consider the individual terms ("partial expectations") of (13). Figure 4 plots these terms against the cumulative probability for a 37-cell cellular network with 64 channels and 12 Erlangs per cell. (As $\mathsf{P}(C_i < j)$ is monotonic in j, the horizontal axis is simply a non-linear scale for j.)

Since $g(j;\rho_i)$ is known, it suffices to estimate

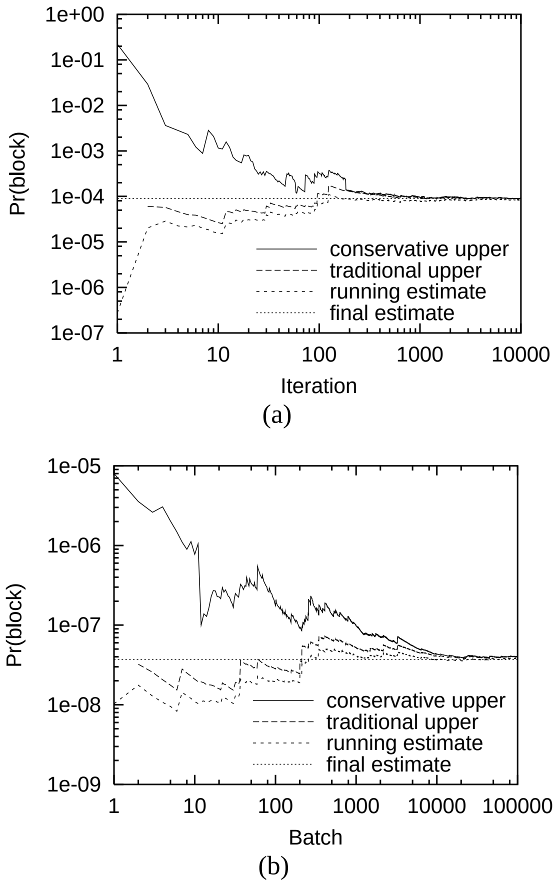

Figure 3. Estimated upper bounds on B: $\hat{B} + 2\sigma$ and the conservative estimator of (14). 3×3 mesh, 64 channels per link. (a) 13 Erlangs per route, simple variance (b) 10 Erlangs per route, batches of 100

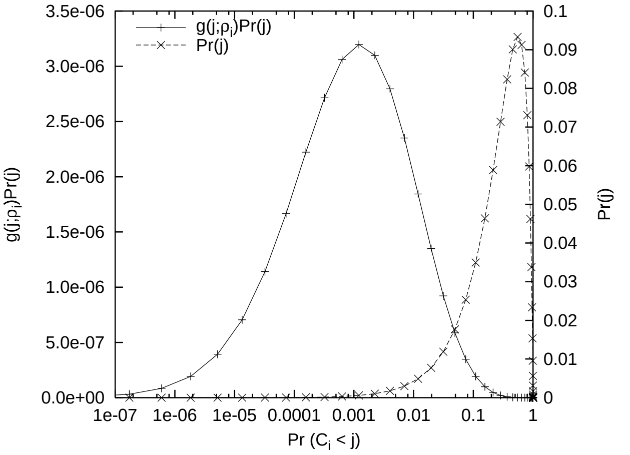

Figure 4. Comparison of partial expectations and state probabilities.

$\mathsf{P}(C_i(X) = j)$, or those for which $g(j;\rho_i)\mathsf{P}(C_i(X) = j)$ is a significant fraction of B. Because these terms decay rapidly for $j < \mathsf{argmax}_j(g(j;\rho_i)\mathsf{P}(j))$, as seen in Figure 4, it is possible to determine by inspection when all "significant" terms have been estimated with sufficient confidence.

To quantify this, assume that the sample contains enough points to capture the peak of the probability distribution, which requires orders of magnitude less data than capturing the peak of the partial expectation. (Note the different scales in Figure 4.) Let m be the smallest value such that $\mathsf{P}(C_i = m)$ can be reliably estimated from the sample, and for $j \geq m$, let p_j be the sample estimate of $\mathsf{P}(C_i = j)$. For $j < m$, conservatively approximate the tail as $\mathsf{P}(C_i = j) \approx p_j \equiv p_m \Delta^{j-m}$, where Δ is fitted to the sample data. In this paper,

$$\Delta = \sqrt[h]{\frac{\sum_{j=0}^{h-1} p_{m+j}}{\sum_{j=0}^{h-1} p_{m+h+j}}},$$

where m is the smallest value of $C_i(X)$ observed more than once in the simulation, and h is such that $m + 2h$ is the fourth smallest such value.

Ignoring correlations (Section 5), the variances of the estimates p_j based on S samples, and $\mathsf{Var}[\hat{Y}_i(S)]$ can then be approximated by

$$\begin{aligned} \hat{V}_i(j) &= p_j(1-p_j)/S, \\ \hat{V}_i &= \sum_{j=0}^{C} (g(j;\rho_i))^2 \hat{V}_i(j). \end{aligned} \qquad (14)$$

The curve "conservative" in Figure 3 plots $\hat{B} + 2\sqrt{\hat{V}}$. It is clearly overly conservative for very small sample sizes, since p_j, $j < m$, are very conservative. However, if the sample is large enough for B to be suitably accurate, then the bound becomes usably tight.

8 Concluding Remarks

The filtered Gibbs sampler not only outperforms the usual Gibbs sampler, but its relative efficiency actually grows with problem size and with increasing load.

The key limitations of the FGS are its relatively poor performance when the load per channel is low, and the difficulty in deriving good confidence intervals for its results.

References

[1] D. Everitt and N. W. Macfadyen. Analysis of multicellular mobile radiotelephone systems with loss. *Br. Telecom Technol. J.*, 1(2), 1983, 37–45.

[2] D. L. Pallant and P. G. Taylor. Modeling handovers in cellular mobile networks with dynamic channel allocation. *Operations Research*, 43(1), 1995, 33–42.

[3] Richard J Boucherie and Michel Mandjes. Estimation of performance measures for product form cellular mobile communications networks. *Telecommunication Systems*, 10, 1998, 321–354.

[4] Frank P. Kelly. Effective bandwidths at multi-class queues. *Queue. Syst.*, 9, 1991, 5–16.

[5] A. W. Berger and W. Whitt. Effective bandwidths with priorities. *IEEE/ACM Trans. Networking*, 6(4), 1998, 447–460.

[6] Frank P. Kelly, Peter B. Key, and Stan Zachary. Distributed admission control. *IEEE J. Select. Areas Commun.*, 18(12), 2000, 2617–2628.

[7] Scott Jordan. A continuous state space model of multiple service, multiple resource communication networks. *IEEE Trans. Commun.*, 43(2/3/4), 1995, 477–484.

[8] Debasis Mitra and John A. Morrison. Erlang capacity and uniform approximations for shared unbuffered resources. *IEEE/ACM Trans. Networking*, 2(6), 1994, 558–570.

[9] Keith W. Ross. *Multiservice loss models for broadband telecommunication networks*, (Berlin, Germany: Springer-Verlag, 1995).

[10] J. P. Buzen. Computational algorithms for closed queueing networks with exponential servers. *Comm. ACM*, 16, 1973, 527–531.

[11] J. L. Coleman, W. Henderson, and P. G. Taylor. A convolution algorithm for calculating exact equilibrium distributions in resource allocation problems with moderate user interference. *IEEE Trans. Commun.*, 42(2/3/4), 1994, 1106–1111.

[12] Gagan L. Choudhury, Kin K. Leung, and Ward Whitt. An inversion algorithm to compute blocking probabilities in loss networks with state-dependent rates. *IEEE/ACM Trans. Networking*, 3(5), 1995, 585–601.

[13] Keith W Ross, Danny H. K. Tsang, and Jie Wang. Monte Carlo summation and integration applied to multiclass queuing networks. *J. ACM*, 41(6), 1994, 1110–1135.

[14] P. E. Lassila and J. T. Virtamo. Efficient Monte Carlo simulation of product form systems. *Proc. Nordic Teletraffic Seminar (NTS) 14*, Copenhagen, Denmark, 1998, 355–366.

[15] F. J. Vázquez-Abad and L. Andrew. Filtered Gibbs sampler for estimating blocking probabilities in WDM optical networks. *Proc. 14th European Simulation Multiconference*, Ghent, Belgium, May, 2000, 548–555.

[16] P. E. Lassila and J. T. Virtamo. Variance reduction in Monte Carlo simulation of product form systems. *Electron. Lett.*, 34(12), 1998, 1204–1205.

[17] Adrian E. Conway and David E. O'Brien. Hybrid analysis of response time distributions in queueing networks. *IEEE Trans. Commun.*, 41(7), 1993, 1091–1101.

[18] Jaakob Kind, Thomas Niessen, and Rudolf Mathar. Theory of maximum packing and related channel assignment strategies for cellular radio networks. *Math. Meth. Op. Res.*, 48(1), 1998, 1–16.

[19] C. Knessl and C. Tier. Asymptotic approximations and bottleneck analysis in product form queueing networks with large populations. *Perf. Eval.*, 33(4), 1998, 219–248.

[20] Debasis Mitra, John A. Morrison, and K. G. Ramakrishnan. Optimization and design of network routing using refined asymptotic approximations. *Perf. Eval.*, 36-37, 1999, 267–288.

[21] S. M. Ross. *Simulation*, (Boston, MA: Academic Press, 2nd edition, 1997).

[22] Pasi Lassila and Jorma Virtamo. Nearly optimal importance sampling for Monte Carlo simulation of loss systems. *ACM Trans. Model. Comput. Simul.*, 10(4), 2000, 326–347.

[23] C. Alexopoulos and A. Seila. Output data analysis. *Handbook of Simulation*, chapter 7, 225–272. John Wiley and Sons, New York, NY, 1998.