Rare-event simulation of non-Markovian queueing networks using a state-dependent change of measure determined using cross-entropy

Pieter-Tjerk de Boer*

Note: this paper is an extended abstract for RESIM2004 of a full paper that has been accepted for publication in the Annals of Operations Research special issue on the Cross-Entropy method.

Abstract

A method is described for the efficient estimation of small overflow probabilities in non-Markovian queueing network models. The method uses importance sampling with a state-dependent change of measure, which is determined adaptively using the cross-entropy method, thus avoiding the need for a detailed mathematical analysis of the model. Experiments show that the method can be used to estimate overflow probabilities in a two-node tandem queue network model for which simulation using a state-*in*dependent change of measure does not work well.

1 Introduction

This paper is about the simulation of rare overflow events in queueing models, representing for example telecommunications networks, using the technique of *importance sampling*. In this technique, the underlying probability distributions of the model are modified (called a *change of measure* or *tilting*), such that the rare event occurs more frequently in the simulation; by keeping track of the likelihood ratio between a sample path in this tilted system and in the original system, these biased observations can be used to estimate the true overflow probability.

Previous work [PW89, GK95, dB04] on simulating rare events in Markovian queueing models has shown that a state-independent change of measure (e.g., simply replacing the arrival and service rates by other values, without letting them change as the number of customers in the queues changes) may result in an asymptotically efficient simulation in some cases, but also often fails, depending on the model and the model's parameters. Furthermore, [dB00, dBN02] have shown experimentally that using a state-dependent change of measure in such cases does lead to an asymptotically efficient estimator, and that such a state-dependent change of measure can be determined adaptively using the cross-entropy method.

The present paper focuses on generalizing the state-dependent cross-entropy method to non-Markovian queuing models. In order to do so, we start by considering the idea of a state-dependent change of measure for non-Markovian models in more detail, and the concept of rescheduling in particular. Next, we derive the basic equations for determining the change of measure adaptively, based on the cross-entropy method. Finally, we demonstrate the effectiveness of the method experimentally.

In the rest of this extended abstract, it is assumed the reader is familiar with importance sampling, and event-based simulation techniques. More details can be found in the full paper.

2 Simulation with a state-dependent tilt for non-Markovian queueing models

Typically, queueing models are simulated with an event-based simulator: events such as arrivals and service completions are put on a list in their chronological order, and they are performed in their order on the list. The random variables are typically sampled at the moment an event is put onto the list: an interarrival or service time is sampled to determine when the next arrival or service completion will happen. One can also think of each of the events on the event list as associated with a “clock”: a conceptual device that will expire at some future time, and then cause an event to happen. Such a clock may also be non-active, if the corresponding event is not scheduled (e.g., service completion at an idle server).

A trivial way to do state-dependent tilting would be to simply replace the distributions used when the samples are taken, by other distributions, different according to the current state at the moment of sampling. However, it turns out (see the full paper for details) that this does not work well; it gives little simulation speedup. This happens because the system's state may change considerably between the moment the sample is drawn, and the moment the corresponding event is performed.

*Department of Electrical Engineering, Mathematics and Computer Science, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands; e-mail: ptdeboer@cs.utwente.nl

One approach for handling the change in simulation distribution between the moment an event is scheduled and the time at which it occurs, is *rescheduling*; see [NNHG93]. The idea of rescheduling is that the expiration time of a clock can be changed between the moment it is initially set and the moment it expires. The new expiration time can then be calculated based on sampling from a distribution that is tilted differently (namely, appropriately for the system's new state) from the initial sampling, and this sampling should be done *conditionally* on the elapsed time since the event was originally scheduled. Furthermore, one needs to carefully update the likelihood ratio.

In order to be able to reschedule a clock based on a different distribution, we need to store for each clock i its starting time, denoted by s_i. Then after every event (system state change), we reschedule each clock i by taking a sample from the appropriately tilted distribution, conditional on being larger than $t - s_i$ (with t denoting the current simulation time), and finding out which of the clocks expires earliest.

The entire state-dependent tilting is specified by a (large) vector θ, which has a component for each clock in each state.

Then the simulation procedure is as follows:

Algorithm 1: Importance sampling simulation with rescheduling

1. Bring the system in its initial state, marking the appropriate clocks as active and initializing their s_i. Set $t = 0$. Set $L = 1$.
2. For all active clocks i, take a sample x_i from the conditional distribution $f_i(x_i \mid x_i > t - s_i \ , \ \theta_{(i,Y)})$, where $\theta_{(i,Y)}$ denotes the component of θ associated with clock i and state Y.
3. Set $j = \arg\min_i(s_i + x_i)$: clock j will expire next.
4. Set $u = x_j + s_j$; then u is the time at which the next clock expires.
5. Update the likelihood ratio as follows:
$$L := L \cdot \frac{\frac{f_j(u-s_j,0)}{\bar{F}_j(t-s_j,0)} \cdot \prod_{i\neq j} \frac{\bar{F}_i(u-s_i,0)}{\bar{F}_i(t-s_i,0)}}{\frac{f_j(u-s_j,\theta_{(j,Y)})}{\bar{F}_j(t-s_j,\theta_{(j,Y)})} \cdot \prod_{i\neq j} \frac{\bar{F}_i(u-s_i,\theta_{(i,Y)})}{\bar{F}_i(t-s_i,\theta_{(i,Y)})}}, \tag{1}$$
where $\bar{F}_i(\cdot, \vartheta)$ denotes the complementary distribution function for clock i, tilted by ϑ; i.e., $\bar{F}_i(x, \vartheta) = \int_x^\infty f_i(t, \vartheta) dt$.
6. Set $t := u$.
7. Perform the event corresponding to the jth clock: this may entail state changes, and activating or deactivating one or more clocks. If any clocks i are activated, set their $s_i = t$.
8. Repeat from line 2, unless a terminating state (possibly the target state) has been reached.

Note that one "step" of the simulation (i.e., the processing of one event) corresponds to going once through the lines 2–8 of this procedure.

Note further that in the above, the state (vector) variable Y is supposed to contain only the discrete state information (like "how many customers in each queue"), which only changes at the occurrence of events. However, there are also continuous state variables, namely the age of each clock. Allowing the importance sampling change of measure to also depend on these would lead to several extra complexities; therefore, it is not considered in the present work.

3 Cross-entropy for non-Markovian models

In this section, we give the tilting-parameter update equations based on the cross-entropy method. Remember that the purpose of these equations is to use simulation results collected during one simulation run to choose the tilting parameters for the next simulation run; iterating this several times should converge to a set of tilting parameters that provides a good estimator of the quantity of interest (i.e., the overflow probability).

Typically (e.g., for Markovian queueing models, or for state-independent tilting in non-Markovian queueing models), using the cross-entropy method to determine the tilting parameters leads to relatively simple, *explicit* expressions for those tilting parameters in terms of sample averages. In fact, this is the primary reason for using the cross-entropy method, rather than basing the tilting parameters on direct minimization of the variance (see [Rub97, dB00, dBKR04]). As we will see below, in the case of non-Markovian state-dependent tilting, the resulting expressions for the tilting parameters are not explicit, although an approximation can be made that is explicit. However, use of the cross-entropy method rather than direct variance minimization still has the advantage that the expressions for the tilting parameters are *independent* of each other.

A careful derivation of the cross-entropy equations for the the rescheduling simulation algorithm given above, and assuming exponential tilting, results in the following:

$$\mathbb{E}' \sum_{\ell \in S_+} \big(u_\ell - s_\ell\big) - \mathbb{E}' \sum_{\ell \in S_+ \cup S_-} G_{\theta^*}(t_\ell - s_\ell) + \mathbb{E}' \sum_{\ell \in S_-} G_{\theta^*}(u_\ell - s_\ell) = 0, \tag{2}$$

where

- $\mathbb{E}'$ denotes (untilted) expectation conditional on reaching the target (overflow) state;
- $G_\theta(y) = \mathbb{E}_\theta(X \mid X > y)$; basically, this is the expectation of the random variable associated with the jth clock, conditional on being larger than y, and exponentially tilted with parameter θ;
- θ^* is the CE-optimal tilting parameter for clock j in state k;
- S_+ is the set of all ℓ such that in the ℓth step on the sample path the system is in state k and clock j expires first;
- $S_-(j,k)$ is the set of all ℓ such that in the ℓth step on the sample path the system is in state k and clock j is active but another clock expires first;
- t_ℓ is the time stamp associated with the state information for the ℓth step, i.e., the time of the previous clock expiration;
- u_ℓ is the time which is found in the ℓth step as the earliest time at which a clock will expire; therefore, $u_\ell = t_{\ell+1}$;
- s_ℓ is the time at which the jth clock was last started before t_ℓ.

Note that the indices j and k, indicating the state and clock under consideration, have been omitted in most of this notation; (2) must be solved once for each possible pair of clock and state. Furthermore, note that the values of S_+, S_-, s_ℓ, t_ℓ and u_ℓ are properties of the random sample path, and thus random variables.

So, in order to determine the optimal tilting parameters using the cross-entropy method, one would solve (2) for θ^*, once for every clock in every state; typically, $\mathbb{E}'$ would be estimated using sample averages obtained in a previous iteration. (Note that it is not obvious that (2) has a unique solution; in the full paper, it is established that it has at least one solution, while its unicity has only been formally established for a limited class of probability distributions.)

Now the essential difference with the Markovian case can be appreciated: it is not possible to rewrite (2) into an explicit expression for θ^* in terms of sample path expectations $\mathbb{E}'$. This is because of the presence of terms which depend both on the samples (through t_ℓ, s_ℓ and u_ℓ), and on θ^* (as a parameter to $G_{\theta^*}(\cdot)$). In the Markovian case, the tail expectations $G_\theta(y)$ do not depend on the argument y (and thus on the sample paths) because of the memoryless property.

So how can we solve (2) in practice? One approach is to use an iterative solver, e.g. the bisection method, regula falsi, etc. (see, e.g., [PFTV88]); this requires that sufficient detail from all sample paths be stored to calculate the left-hand side of (2) for a large number of values of θ^*.

An alternative approach is to choose two (or a few more) values for θ *before* the previous simulation is run. Then one can calculate the left-hand side of (2) for those θ values by keeping track of sample sums during the simulation, without needing to store sample path details. For the final estimate of θ^* (i.e., the zero-crossing of the left-hand side) a (linear) approximation can then be used, like

$$\hat{\theta}^* = \theta_1 - \frac{\theta_2 - \theta_1}{A(\theta_2) - A(\theta_1)} A(\theta_1), \tag{3}$$

where $A(\theta)$ denotes the left-hand side of (2), and θ_1 and θ_2 are the values chosen in advance; in practice, $A(\cdot)$ would be approximated using sample averages. Clearly, this does not give an exact calculation of θ^*. However, in practice that may not be a problem: approximating $\mathbb{E}'$ by simulation results will never give an exact result anyway, even if an exact solution method is used; and a small error in the tilting parameters does not make the resulting estimator (in the next iteration) biased, it just makes its variance a bit larger. Furthermore, by properly choosing θ_1 and θ_2 at every iteration, even this approximate algorithm can converge to the true value.

Note: for models with a large state space, techniques similar to those from the state-dependent Markovian method can (and need to) be used to take states together in order to get more accurate estimates of the tilting parameters in regions of the state space that are rarely visited. For details, the reader is referred to the full paper, and/or [dB00, dBN02].

4 Experimental results: a tandem queueing network

We consider a queueing network consisting of two queues in tandem. The arrival process to the first queue has a uniform distribution on [0, 2/3]. Both servers have the same service time distribution, namely hyper-exponential

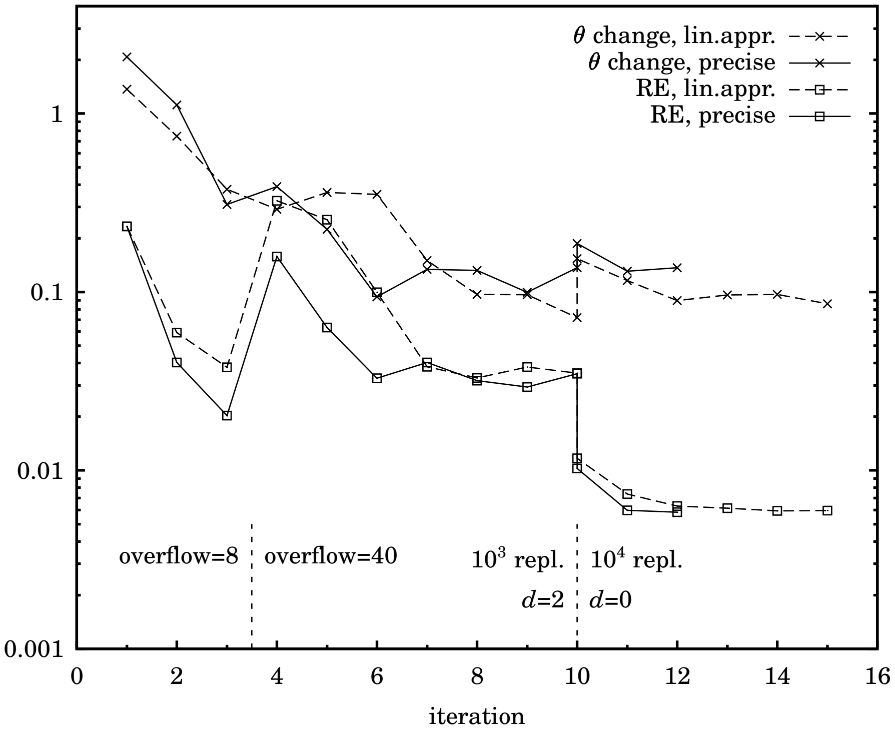

Figure 1: Convergence process for the tandem model at overflow level 40; see text for explanation

with equally probable rates 4 and 12. The initial state is the state immediately after arrival of the first customer to the network; we are interested in the probability of overflow of the total network population within a busy cycle.

This simple queueing model addresses the main features of the method described in this paper: non-Markovianness and the need for state-dependent tilting. The latter is illustrated by trying state-independent tilting: at overflow level 20 it still works reasonably well, giving 3.4% relative error at 10^5 replications, but at overflow level 40 already 10^7 replications are needed for similar accuracy. As we will see below, the state-dependent method gives better accuracy, even with just 10^4 replications and at much higher overflow levels. Note that this tandem model has two equally but lightly loaded servers, like the typical Markovian examples from [GK95] for which state-independent tilting also does not work well.

4.1 Convergence behaviour

Figure 1 illustrates the convergence process at overflow level 40. The horizontal axis counts the iterations. The vertical axis shows the relative error at each iteration (lower two curves, marked by squares), and the average absolute change in the tilting parameters θ (upper two curves, marked by crosses). Solving (2) "precisely" and using the linear approximation are compared (solid and dashed lines, respectively).

The first iteration was started with all tilting parameters set to 0, i.e., the original probability distributions were used. Consequently, the target event itself is much too rare to be observed, so the overflow level was lowered to 8 (which makes the overflow probability about 0.014). Two more iterations were done at this overflow level. Then the overflow level was set to the value of interest, namely 40. After a total of 6 (precise) or 7 (linear approximation) iterations, the system apparently has converged, as shown by the fact that the relative error does not decrease further, and the changes in θ do not become smaller.

Up to the 10th iteration, 10^3 replications were used per iteration, and local averaging was used, meaning results from states whose queue contents differ by not more than 2 were taken together. The 10th iteration was then repeated with 10^4 replications, and at the same time local averaging was switched off. Clearly, increasing the number of replications by a factor of 10 immediately reduces the relative error by a factor of about $\sqrt{10}$, as is to be expected. In further iterations, the relative error decreases further; this is caused by the improved estimation of the tilting parameters, due both to the larger number of replications, and the more localized estimation of the tilting parameters due to not using local averaging anymore.

From the graphs it is clear that the linear approximation converges a bit slower than the precise calculation,

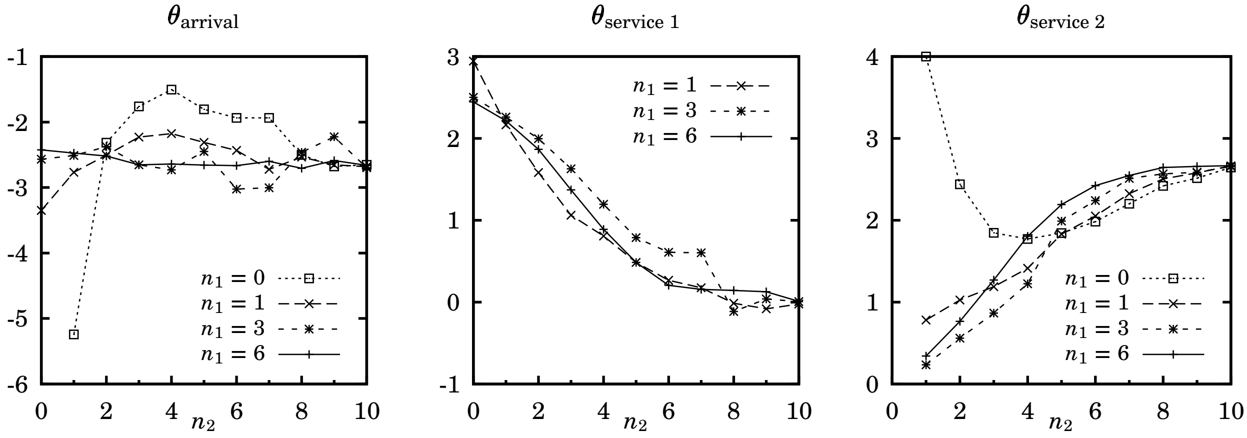

Figure 2: State-dependent tilting parameters for the tandem model

level	estimate	relative error
20	$3.972 \cdot 10^{-6}$	0.0043
40	$2.044 \cdot 10^{-12}$	0.0065
80	$2.640 \cdot 10^{-25}$	0.0069
160	$2.230 \cdot 10^{-51}$	0.0074

Table 1: Asymptotic behaviour of the tandem model

as was to be expected. Actually, the difference is not large, 1 to 3 iterations, although the figure may be a bit flattering due to the problem of choosing θ_1 and θ_2; in these experiments, their difference was set intuitively by the experimenter, and a different choice might give worse results.

4.2 State-dependent tilting parameters

Figure 2 shows how the tilting parameters depend on the state, for overflow level 40, after the process has converged; these are the tilting parameters used in the last iteration of Figure 1. Each of these plots shows the value of one of the three tilting parameters as a function of the second queue length (n_2), for several values of the first queue length (n_1). Note that n_1 and n_2 are limited to the ranges 0...6 and 0...10, respectively, due to the use of boundary layers (see the full paper or [dB00, dBN02]).

Note that some of the tilting parameters are meaningless in some states and therefore missing in the graphs; namely $\theta_{\text{service 1}}$ if $n_1 = 0$ and $\theta_{\text{service 2}}$ if $n_2 = 0$ because the corresponding clock is not active, and all tilting parameters for $n_1 = n_2 = 0$ because this is a terminating state. Furthermore, at $n_1 = 0, n_2 = 1$, a service completion at the second server would immediately terminate the busy cycle, so this sample path cannot reach the target; hence, the tilting is effectively made infinite by setting $\theta_{\text{service 2}} = 4$.

From the plots it is clear that both service tilting parameters have a rather strong dependence on the second queue length. Dependence on the first queue length is much weaker; actually it seems like only the cases $n_1 = 0$ and $n_1 > 0$ would need to be distinguished, so fewer boundary layers would have been sufficient. However, one generally does not know this in advance. The arrival tilting seems to be mostly state-independent, except for low values of both n_1 and n_2. Note that part of any state-dependence suggested by the graphs may simply be noise, since the tilting parameters themselves are simulation results.

4.3 Asymptotic behaviour

The above results were all obtained at an overflow level of 40. Simulations have also been performed at levels of 20, 80 and 160. The simulations at 80 and 160 were not started "from scratch" with a zero tilting, but from the tilting obtained at level 20. This helps the convergence. Table 1 shows the results, all at 10^4 replications. Clearly, the relative error grows slowly with increasing overflow level. However, the growth seems to be slowing down, suggesting that the relative error actually is bounded.

For verification, a normal (i.e., without importance sampling) simulation was done with 10^9 replications, for overflow level 20. This resulted in a probability estimate of $3.94 \cdot 10^{-6}$ with a relative error of 0.016. The importance sampling estimate agrees with this.

5 Discussion and conclusions

In this paper, a novel method has been described for estimating small overflow probabilities in non-Markovian queueing networks through simulation, using a state-dependent change of measure that is determined adaptively using the cross-entropy method. We have given the main equations, pointed out some difficulties compared to the Markovian case, and proposed a solution. Experiments have demonstrated that a state-dependent change of measure combined with rescheduling gives a significant speedup for simulation of overflows in a single queue, and enables efficient simulation of overflows in a two-node tandem queue for which state-independent tilting is not efficient. Furthermore, promising preliminary results have been obtained for a three-node model; however, an observed but unexplained fluctuation in the relative error is still a cause for concern in such examples.

The most important limitation of the method in its present form is the problem of handling larger models, e.g., four or more queues. Firstly, with a larger state space, more states will be visited rarely on typical paths to the target event, which causes the estimates of the tilting parameters for those states to be inaccurate unless many more replications are simulated. Secondly, with a more complicated model typically more clocks will be active simultaneously, causing the amount of computation per step of the simulation to increase (much more than in simulation without rescheduling). Thirdly, it has been observed experimentally in three and four-queue networks that the estimate of the relative error tends to fluctuate in consecutive iterations, something that is not observed in the two-queue tandem model; more research is needed to understand this.

A second area that needs further research is the convergence process. At present, this process is poorly understood. In fact, it is not clear whether the equations that need to be solved have a unique solution in all cases, and completing a proof of this (or of the contrary) would be of much interest. In practical experiments, convergence has turned out to be not too difficult, but there are several parameters that need to be tuned by the human operator. Finding algorithms to automate this would be very useful for implementation in a practical tool. Furthermore, a linearization has been proposed to simplify the calculations, at the cost of more iterations. Empirical evidence suggests that this extra cost need not be large, but more experience needs to be collected. Possibly, an improved scheme can be devised.

References

[dB00] Pieter-Tjerk de Boer. *Analysis and efficient simulation of queueing models of telecommunication systems*. PhD thesis, University of Twente, 2000.

[dB04] Pieter-Tjerk de Boer. Analysis of state-independent is measures for the two-node tandem queue. In *Fifth Workshop on Rare Event Simulation, RESIM2004*, 2004.

[dBKR04] P. T. de Boer, D. P. Kroese, and R. Y. Rubinstein. A fast cross-entropy method for estimating buffer overflows in queueing networks. *Management Science*, 50(7):883–895, 2004.

[dBN02] Pieter-Tjerk de Boer and Victor F. Nicola. Adaptive state-dependent importance sampling simulation of markovian queueing networks. *European Transactions on Telecommunications*, 13(4):303–315, 2002.

[GK95] Paul Glasserman and Shing-Gang Kou. Analysis of an importance sampling estimator for tandem queues. *ACM Transactions on Modeling and Computer Simulation*, 5(1):22–42, January 1995.

[NNHG93] Victor F. Nicola, Marvin K. Nakayama, Philip Heidelberger, and Ambuj Goyal. Fast simulation of highly dependable systems with general failure and repair processes. *IEEE Transactions on Computers*, 42(12):1440–1452, 1993.

[PFTV88] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[PW89] S. Parekh and J. Walrand. A quick simulation method for excessive backlogs in networks of queues. *IEEE Transactions on Automatic Control*, 34:54–66, 1989.

[Rub97] Reuven Y. Rubinstein. Optimization of computer simulation models with rare events. *European Journal of Operations Research*, 99:89–112, 1997.