

Importance Sampling Simulations of Markovian Reliability Systems Using Cross Entropy

Ad Ridder

*Department of Econometrics and Operations Research,
Vrije Universiteit Amsterdam,
de Boelelaan 1105, 1081 HV Amsterdam, Netherlands*

Abstract

This paper reports simulation experiments, applying the cross entropy method such as the importance sampling algorithm for efficient estimation of rare event probabilities in Markovian reliability systems. The method is compared to various failure biasing schemes that have been proved to give estimators with bounded relative errors. The results from the experiments indicate a considerable improvement of the performance of the importance sampling estimators, where performance is measured by the relative error of the estimate, by the relative error of the estimator, and by the gain of the importance sampling simulation to the normal simulation.

1 Introduction

Reliability systems occur frequently in many technological environments such as computer systems, communication networks, air traffic control, electrical power systems, etc. Failure of such a system should be avoided and therefore one should be able to predict the reliability of the system, specifically during the design stage, to ensure that the actual system performs at an acceptable level.

In various studies during the 80's a generic Markovian dependable model was developed to describe and analyse such reliability systems, from whom we mention Goyal et al.[5], Goyal and Lavenberg[6], and Goyal et al.[7]. Later, during the 90's and more recently, importance sampling simulations were applied to these systems in, e.g., Goyal et al.[8], Shahabuddin[17, 18], Juneja and Shahabuddin[12, 13]). The importance sampling algorithms in these papers are based on biasing techniques and they may become rather intricate when certain 'unpleasant' properties are present in the system that may lead to the occurrence of high-probability cycles, making the simple biasing schemes to fail,

see [12, 13]. Our paper is a report of experimental results of an importance sampling algorithm that uses the cross entropy method for finding the new measure.

The cross entropy method, introduced by Rubinstein in [15] for rare event simulation and in [16] for combinatorial optimization, attracts nowadays attention in several research areas, inspiring to contributions in queueing ([3, 4]), reliability ([11, 14]) and optimization ([1]). The reliability system that was studied in Hui et al.[11] is a stochastic network consisting of links that may fail. In such a network the problem is to estimate the probability that certain nodes in the network are connected. The complementary probability should be typically very small, and hence estimation by simulation needs a variance reduction technique such as importance sampling; [11] developed an adaptive importance sampling algorithm based on cross entropy. Lieber et al.[14] considered a reliability model that represents a transmission system and their problem was to estimate the expected unserved energy.

We investigate the Markovian reliability systems that we mentioned earlier, in which failed components are repaired and return to the system being fully operational. The problem is to estimate the probability of a system failure before the system reaches the perfect state in which all components work (see also [12, 13, 17, 18]). In its generality, one can model this as an absorption probability in a finite state Markov chain. Another example of such a problem is the buffer overflow probability in queueing models. Being an important issue in communication networks this problem has been studied widely, also in connection with rare event simulation, recently by de Boer et al.[3, 4] who apply the cross entropy method. Importance sampling means that one implements some other matrix of transition probabilities for drawing sample paths of the Markov chain and that by calculating the likelihood ratios one obtains an unbiased estimator. The remaining problem is finding a matrix that gives an optimal, suboptimal or asymptotically optimal estimator (in terms of the variance), and this problem may be formulated as an optimization program with as many variables as the number of transitions of the Markov chain. Usually, the Markov chain in the queueing models is nicely structured, for instance a phase-homogeneous random walk with only a few phases, and then the dimension of the optimization program is reduced highly. Also, the matrix of transition probabilities of the Markov chain describing the reliability system has a structure that is determined by the failure and repair mechanisms of the system components. The bias heuristics for constructing the new importance sampling matrix exploit these structures, see for instance [12, 13, 17, 18].

Contrary to these heuristics, we consider the full matrix optimization program, applying the iterative procedure of the cross entropy method for finding the new importance sampling matrix. The advantage is that we do not require the knowledge of the structure of

the Markov chain, and this means that the implementation of the importance sampling algorithm is much easier than the intricate biasing schemes. A disadvantage may be that we need to store large matrices. Our results show that we may get excellent performances of the importance sampling estimators, in most cases outperforming the biasing schemes. However, we emphasize that we have no theoretical justification of these results, we cannot say whether the method yields always estimators with bounded relative error.

The paper is organised as follows. In Section 2 we present a continuous time Markov chain, modelling the reliability system, and we phrase the rare event problem. Section 3 describes the associated discrete time Markov chain that is used in the simulations. In Section 4 we summarize briefly some importance sampling issues, and we review some biasing schemes developed specifically for the reliability problem. The generic cross entropy method is described in Section 5, whereas Section 6 gives details of the algorithm for the reliability problem. In Section 7 we present simulation results of three examples, and we conclude in Section 8 with a summary and further directions.

2 The model

Consider a technological system, made up of d different component types, where type i contains K_i similar components. Each component fails, independently of other events, after an exponentially distributed failure time, with failure rate λ_i for type i components. Specifically, failures of components are not correlated, which we formulate explicitly in the following assumption.

Assumption 2.1. *There is no failure propagation.*

Failed components may be repaired by one or more service men, where we allow service priority, and group service with minimal or maximal group sizes. The repair times are exponentially distributed with repair rates μ_i for type i . Consequently, the system may be modelled by a continuous-time Markov chain

$$\mathbf{X}(t) = (X_1(t), X_2(t), \dots, X_d(t)),$$

where $X_i(t)$ denotes the number of failed components of type i at time t . The statespace of the chain is the cartesian product

$$S = \prod_{i=1}^d \{0, 1, \dots, K_i\},$$

with size $|S| = \prod_{i=1}^d (K_i + 1)$. States are denoted as d -dimensional vectors $\mathbf{x} = (x_1, x_2, \dots, x_d)$. Unit vectors are denoted $\mathbf{e}_i := (0, \dots, 0, 1, 0, \dots, 0)$ with the 1 at the i -th coordinate. The perfect state is $\mathbf{0} = (0, 0, \dots, 0)$ when all components are working.

This is the generic Markovian dependable model of [5, 6, 7, 8, 12, 13, 17, 18], where also the following structural property is assumed.

Assumption 2.2. *The Markov chain $\{\mathbf{X}(t) : t \geq 0\}$ is irreducible over its statespace.*

Let $F \subset S$ be a specific set of states, called the failure set, or the set of down states. Once in F the system is unavailable, and only after one or more repairs have brought the system back to a state in $S \setminus F$, the system is running again. Typical examples of failure sets are

- $F = \{K_1, K_2, \dots, K_d\}$: all components have failed.
- $F = \{\mathbf{x} \in S : x_i = K_i \text{ for at least one } i\}$: all components of at least one type have failed.

Associated with the failure set F we can recognize three system performance measures:

- Mean time to failure (MTTF), the mean time to reach the failure set from an arbitrary system state $\mathbf{x} \in S \setminus F$.
- System unavailability, or the long run fraction of time the system is down:

$$\lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t 1\{\mathbf{X}(s) \in F\} ds,$$

where $1\{\cdot\}$ is the usual indicator function.

- Failure entrance probabilities (FEP) $\gamma(\mathbf{x})$. Let T_0 be the time to reach the perfect state, and T_F the time to failure:

$$T_0 := \inf\{t > 0 : \mathbf{X}(t) = \mathbf{0}\}, \quad T_F := \inf\{t > 0 : \mathbf{X}(t) \in F\}.$$

Then we define

$$\gamma(\mathbf{x}) := P_{\mathbf{x}}(T_F < T_0), \tag{1}$$

i.e., the probability of reaching the failure set before state $\mathbf{0}$ starting from state $\mathbf{x} \in S \setminus F$. (When the starting state is $\mathbf{0}$ we have to adjust T_0 to mean that first there must have been a jump to some other state before returning to $\mathbf{0}$.)

In a highly reliable system the MTTF is large and the other two performance measures are very small—meaning of the order of 10^{-6} or smaller—, because reaching the failure set is supposed to be a rare event when the system starts in the perfect state $\mathbf{0}$ or in a state \mathbf{x} consisting of a few failed components. This may be accomplished by assuming that the failure rates are several orders of magnitudes smaller than the repair rates, $\lambda_i \ll \mu_i$ for all types i . For instance, in Shahabuddin[17] it is assumed that all the repair rates satisfy $\mu_i \geq \mu_{\min} = 1$ and all the failure rates $\lambda_i \leq \lambda_{\max} = \epsilon$ where $\epsilon \rightarrow 0$

becomes small. This ϵ is called the rarity parameter of the system. In that case, it is necessary to assume also that any ‘path’ leading to the failure set must pass through states that have repair transitions. In the next section we shall formulate the precise assumption making the system highly reliable.

Applying regenerative properties of Markov chains one can show (see, e.g., Goyal et al. [8]) that the MTTF and the system unavailability can be rewritten into expressions that consist of FEP and other factors, where these other factors are not small (or large). Thus when one applies simulation for estimation of the performance measures, it suffices to implement an efficient algorithm for the failure entrance probabilities. Furthermore, we restrict to finding the FEP from state $\mathbf{0}$ only.

Remark 2.3. In this paper we restrict to calculating or estimating the failure entrance probability $\gamma := \gamma(\mathbf{0})$.

3 The Discrete-time Markov Chain

Let $\{\mathbf{X}_n : n = 0, 1, \dots\}$ be the discrete-time Markov chain (DTMC) that results by embedding the continuous-time process $\{\mathbf{X}(t) : t \geq 0\}$ at the jump epochs. In fact, when we would apply discrete event simulation of the continuous-time chain, this would lead automatically to the DTMC, and, moreover, the failure entrance probability $\gamma = P_0(T_F < T_0)$ is the same in both models; (slightly abusing notations, we let T_0 and T_F be the first entrance times now for the DTMC). The DTMC has one-step transition probabilities $p_{\mathbf{x}\mathbf{y}}$, defined by $p_{\mathbf{x}\mathbf{y}} = \nu_{\mathbf{x}\mathbf{y}}/\nu_{\mathbf{x}}$, where $\nu_{\mathbf{x}\mathbf{y}}$ is the rate of transition $\mathbf{x} \rightarrow \mathbf{y}$ of the continuous-time Markov chain, and where $\nu_{\mathbf{x}} = \sum_{\mathbf{y}} \nu_{\mathbf{x}\mathbf{y}}$.

The transition probabilities depend on a rarity parameter ϵ , that is, they are functions $p_{\mathbf{x}\mathbf{y}}(\epsilon)$ of a positive parameter ϵ that becomes very small. For instance, consider the model of [17]: the failure rates are functions of ϵ such as $\lambda_i = \epsilon^{r_i}$ for some $r_i \geq 1$, and the repair rates are constants μ_i ; and suppose that \mathbf{x} is a state from where transitions occur due to failures of types 1 and 2, or due to a repair of a single component of type 1. Then the transition probability to state $\mathbf{y} = \mathbf{x} + \mathbf{e}_1$ due to failure of a type 1 component is

$$p_{\mathbf{x}\mathbf{y}} = \frac{(K_1 - x_1)\epsilon^{r_1}}{(K_1 - x_1)\epsilon^{r_1} + (K_2 - x_2)\epsilon^{r_2} + \mu_1},$$

and to $\mathbf{y} = \mathbf{x} - \mathbf{e}_1$:

$$p_{\mathbf{x}\mathbf{y}} = \frac{\mu_1}{(K_1 - x_1)\epsilon^{r_1} + (K_2 - x_2)\epsilon^{r_2} + \mu_1}.$$

Let \mathcal{I} be the set of internal states, i.e., not being $\mathbf{0}$ or lying in F :

$$\mathcal{I} := S \setminus (\{\mathbf{0}\} \cup F).$$

Under the following assumption reaching the failure set is a rare event, see Juneja and Shahabuddin[12].

Assumption 3.1. *For all states $\mathbf{x} \in \mathcal{I}$ there exists at least one direct path from \mathbf{x} to $\mathbf{0}$ containing no states of F with $\Theta(1)$ probability as the rarity parameter $\epsilon \rightarrow 0$; and there do not exist paths from $\mathbf{0}$ to F with $\Theta(1)$ probability.*

Here we used the notation Θ for comparing functions: let $f(\epsilon), g(\epsilon)$ be functions of $\epsilon > 0$. Then $f(\epsilon) = \Theta(g(\epsilon))$ (as $\epsilon \rightarrow 0$) if there exist positive constants c_1, c_2 such that $c_1|g(\epsilon)| \leq |f(\epsilon)| \leq c_2|g(\epsilon)|$ for all ϵ sufficiently small. The two transition probabilities given in the example above are $\Theta(\epsilon^{r_1})$ and $\Theta(1)$ probabilities, respectively. When Assumption 3.1 holds, [12] showed that $\gamma = \Theta(\epsilon^r)$ for some positive r .

Notice that it is elementary Markov chain theory to set up a system of linear equations to hold for the probabilities $(\gamma(\mathbf{x}))_{\mathbf{x} \in \mathcal{I}}$, and then find γ by

$$\gamma = \sum_{\mathbf{x} \in \mathcal{I}} p_{\mathbf{0}\mathbf{x}} \gamma(\mathbf{x}).$$

The system of equations may be solved numerically by a Gauss-Seidel iteration method since the matrix of transition probabilities is a sparse matrix, typically in most reliability applications: transitions due to failure have the form $\mathbf{x} \rightarrow \mathbf{x} + \mathbf{e}_i$ when a component of type i fails, and transitions due to repair are either $\mathbf{x} \rightarrow \mathbf{x} - \mathbf{e}_i$ for single repair, or $\mathbf{x} \rightarrow \mathbf{x} - g_i \mathbf{e}_i$ for group repair. Thus the transition probabilities $p_{\mathbf{x}\mathbf{y}}$ are nonzero only for a ‘few’ neighboring states. However, the number of states can become very large making even the sparse matrix implementation too ‘big’. In that case, simulation is an alternative method.

4 Importance Sampling Simulations

Let (Ω, \mathcal{F}) be the coordinate sample space associated with the discrete-time process $\{\mathbf{X}_n : n = 0, 1, \dots\}$. It is also the sample space for discrete time simulations. Each sample $\omega \in \Omega$ is an infinite sample path $\omega = (\omega_0, \omega_1, \dots)$ with $\mathbf{X}_n(\omega) = \omega_n \in S$. Let \mathcal{Q} be a class of probability measures on (Ω, \mathcal{F}) such that the process $\{\mathbf{X}_n : n = 0, 1, \dots\}$ is a Markov chain and such that the chain is irreducible over its state space S . Let P be the original probability measure (induced by the failure and repair rates of the model), conditioned on the event that the process starts at $\mathbf{0}$. It is clear from Assumption 2.2 that $P \in \mathcal{Q}$. We are interested in calculating or estimating the P -probability $\gamma := P(T_F < T_0)$ that the process reaches a failed state (in F) before returning to the perfect state $\mathbf{0}$. Let $A \subset \Omega$ be the collection of all sample paths that reach F before returning to $\mathbf{0}$:

$$A := \{\omega \in \Omega : T_F(\omega) < T_0(\omega)\}.$$

Let Z be the output of a simulation experiment, i.e., the random variable $1\{A\}$. Hence,

$$\gamma = P(A) = E_P[1\{A\}] = E_P[Z],$$

where we indicate explicitly the expectation w.r.t. P . (And recall that P is a conditional probability, conditioned on a start in state $\mathbf{0}$.) Under Assumption 3.1 A is a rare event, meaning that $P(A)$ decays to zero as a function of the rarity parameter ϵ .

In the simulations we will deal with sample paths of finite length rather than of infinite length. In fact, for any probability measure $Q \in \mathcal{Q}$, due to the irreducibility,

$$Q(T_F < \infty) = 1 \quad \text{and} \quad Q(T_0 < \infty) = 1. \quad (2)$$

Thus in a simulation experiment where we have implemented Q , we will not generate infinite sample paths ω , but we stop the simulation of a realisation as soon as either F or $\mathbf{0}$ has been reached. We denote by Ω_0 the subset of all finite sample paths ending in either F or $\mathbf{0}$. We denote conveniently finite length paths again as ω :

$$\Omega_0 := \{\omega = (\omega_0, \dots, \omega_n) : n = 1, 2, \dots; \omega_1, \dots, \omega_{n-1} \in \mathcal{I}; \omega_n \in \{\mathbf{0}\} \cup F\}.$$

When we simulate the process—using P —we generate i.i.d. copies Z_1, Z_2, \dots, Z_N of Z and construct the estimator

$$\hat{\gamma} := \frac{1}{N} \sum_{i=1}^N Z_i.$$

Commonly, one calls this the standard or direct simulation method, or the crude Monte Carlo method (Asmussen and Rubinstein [2]), and it provides easily a variance analysis. For instance to obtain a relative error $\text{RE}[\hat{\gamma}]$ of 10%—meaning that the 90%-confidence interval half-width is about 16% relative to γ —one needs a sample size $N \approx 100/\gamma$ which becomes too large for the simulation to be efficient and reliable when γ is very small, say less than 10^{-6} . Notice that we define RE as the ratio of the standard deviation to the expectation of the estimator.

Furthermore, it is customary to analyse the variance or the relative error (RE) of the estimator with respect to the rarity parameter. Considering the standard estimator, a moment reflection shows that its RE is unbounded when $\gamma = \Theta(\epsilon^r)$ (where $r > 0$):

$$\text{RE}[Z] = \frac{\sqrt{\text{Var}[Z]}}{E[Z]} = \frac{\sqrt{\gamma - \gamma^2}}{\gamma} = \frac{1}{\sqrt{\gamma}} \sqrt{1 - \gamma} \geq \frac{1}{2} \frac{1}{\sqrt{\gamma}} \geq \frac{1}{2} \frac{1}{\sqrt{c_2 \epsilon^r}},$$

for all ϵ sufficiently small, and thus $\text{RE}[Z] \rightarrow \infty$ when $\epsilon \rightarrow 0$.

Suppose that one can simulate the process by generating sample paths under some other probability measure Q . When the original measure P is absolutely continuous with respect to Q ($P \ll Q$), there exists a likelihood ratio $L = dP/dQ$, and we obtain

$$E_Q[LZ] = E_P[Z] = \gamma,$$

hence, we might as well estimate γ by drawing—using Q —i.i.d. copies $L_1Z_1, L_2Z_2, \dots, L_NZ_N$ of LZ and construct the importance sampling (IS) estimator

$$\hat{\gamma}_Q := \frac{1}{N} \sum_{i=1}^N L_i Z_i.$$

Clearly, the problem is to find a measure Q for which

$$\text{Var}_Q[\hat{\gamma}_Q] < \text{Var}_P[\hat{\gamma}_P],$$

and, preferably, for which the lefthand side is as small as possible, because this would mean that we can reduce the sample size N_Q of the importance sampling simulation to obtain the same performance as the direct simulation method.

Various failure biasing schemes or heuristics have been developed to construct the new transition matrix $\mathbf{Q} = (q_{\mathbf{x}\mathbf{y}})$, see e.g. Shahabuddin[17], Juneja and Shahabuddin[12, 13]. The basic idea is to increase the probability of the direct paths to the failure set. It seems that there are two structural properties that have great influence on the implementation of a successful biasing scheme.

Property 4.1. *From all internal states $\mathbf{x} \in \mathcal{I}$ a repair transition is possible.*

Property 4.2. *The failure rates satisfy $\lambda_i = \Theta(\epsilon^s)$ with the same $s \geq 1$ for all types.*

In the latter case we say that the failure rates are balanced, otherwise they are unbalanced. When Properties 4.1 and 4.2 hold, Shahabuddin[17] proved that the importance sampling estimator has a bounded relative error for any new measure Q under which the transition probabilities do not depend on ϵ , i.e., $q_{\mathbf{x}\mathbf{y}} = \Theta(1)$ (whenever $p_{\mathbf{x}\mathbf{y}} > 0$). A simple failure biasing scheme (SFB) with parameter θ suffices: increase the total failure-transition probability at any state to a constant θ (typically 0.5) while keeping their original mutual proportionalities; similarly, the repair-transition probabilities are reduced to $1 - \theta$. In case of unbalanced failure rates (but with Property 4.1) [17] introduced a balanced failure biasing scheme (BFB) and showed bounded relative error. A BFB scheme with parameter θ is similar to SFB(θ), however it distributes the total probability θ equally to the failure-transitions. Difficulties arise when Property 4.1 does not hold, typically in case of group repair. Juneja and Shahabuddin[12, 13] show that SFB and BFB may fail (unbounded relative error), and they construct various biasing schemes (IGBS, SB-RBS) that give bounded relative error. However, these schemes require more knowledge of the structures of transition probabilities, making these algorithms rather intricate. An IGBS (implementable general biasing scheme) has two parameters, p and δ , where p is like the parameter θ of SFB and BFB, and δ is much smaller. Furthermore, the transitions of the Markov chain are said to be either of high-probability or of low-probability. When a high-probability transition occurs during a simulation run, the next state is chosen by

applying SFB (with p) when the failure transitions are balanced, or BFB(p) otherwise; in case of a low-probability transition the same rule applies with δ (in stead of p). Finally, the splitting based regular biasing scheme (SB-RBS) goes into much more detail of the transition structure of the DTMC to identify high-probability cycles, we refer to [13] for further details.

An optimal measure Q^* would lead to a zero-variance estimator, but this measure does not satisfy the condition that $P \ll Q^*$, because one shows easily that $Q^*(\cdot) = P(\cdot|A)$, that is, the optimal probability measure for estimating $P(A)$ is the conditional probability measure conditioned on the event A (see, e.g., Heidelberger[9]). Thus, the optimal measure satisfies

$$dQ^*(\omega) = \frac{dP(\omega)Z(\omega)}{\gamma}, \quad (3)$$

for all $\omega \in \Omega$. Clearly, $Q^* \notin \mathcal{Q}$, because the process $\{\mathbf{X}_n : n = 0, 1, \dots\}$ is not an irreducible Markov chain under the optimal measure Q^* . However, it suggests that a measure Q that approximates Q^* (in some sense) and that satisfies the absolute continuous property $P \ll Q$, would be a ‘good’ candidate for the importance sampling algorithm. Indeed, this is what we shall show in the forthcoming sections, where we restrict to measures $Q \in \mathcal{Q}$ for the following two reasons.

- When the Markov chain is not irreducible under Q , there exist events $B \subset \Omega$ ($B \in \mathcal{F}$) such that $Q(B) = 0$ while $P(B) > 0$: e.g., take two states in two different Q -closed sets and construct a cycle between these states with positive P -probability. Then $P \ll Q$ does not hold and importance sampling is not possible (formally).
- Irreducible chains give finite (with probability one) samples in the simulations, see (2)

5 Cross Entropy

The cross entropy method has been introduced recently by Rubinstein in [15] for application to rare event simulation and in [16] for combinatorial optimization. The basic idea of the cross entropy method is to consider the Kullback-Leibler distance or divergence as a distance between probability measures. In our model we define for any two probability measures μ, ν on (Ω, \mathcal{F}) this distance to be

$$d(\mu, \nu) := E_\mu \left[\log \frac{d\mu}{d\nu} \right],$$

if $\mu \ll \nu$, otherwise $d(\mu, \nu) = \infty$. Though not being a metric it suits our purposes to approximate the optimal measure Q^* for estimating γ by an importance sampling

measure $Q \in \mathcal{Q}$, namely by trying to minimize the Kullback-Leibler distance on \mathcal{Q} :

$$\min_{Q \in \mathcal{Q}} d(Q^*, Q). \quad (4)$$

Using the expression (3) for Q^* we rewrite $d(Q^*, Q)$ after some manipulations into

$$d(Q^*, Q) = E_P \left[\frac{Z}{\gamma} \log \frac{Z dP}{\gamma} \right] - \frac{1}{\gamma} E_P[Z \log dQ].$$

Thus the optimization problem for approximating the optimal measure Q^* becomes

$$\max_{Q \in \mathcal{Q}} E_P[Z \log dQ].$$

Suppose that we would use simulation to solve this optimization problem—we would solve the stochastic counterpart—, then we notice that under the original measure P the event A is rare, thus we would require many samples ω before we would have sufficiently many observations of $Z = 1$. Therefore, let $Q_0 \in \mathcal{Q}$ arbitrary but such that A is ‘less rare’ and use that

$$E_P[Z \log dQ] = E_{Q_0} \left[\frac{dP}{dQ_0} Z \log dQ \right].$$

Our optimization problem has become

$$\max_{Q \in \mathcal{Q}} E_{Q_0} \left[\frac{dP}{dQ_0} Z \log dQ \right]. \quad (5)$$

In the cross entropy method we consider the stochastic counterpart of the optimization problem (5) and solve it via an iterative scheme. More precisely, we construct a sequence of probability measures $(Q_j)_{j=0,1,\dots} \subset \mathcal{Q}$ on (Ω, \mathcal{F}) according to

$$Q_{j+1} = \arg \max_{Q \in \mathcal{Q}} \frac{1}{N_j} \sum_{i=1}^{N_j} \frac{dP}{dQ_j}(\omega^{(i)}) Z(\omega^{(i)}) \log dQ(\omega^{(i)}), \quad (6)$$

where $\omega^{(i)}, i = 1, 2, \dots, N_j$ are N_j sample paths from Ω generated randomly using measure Q_j . The iteration starts off at the chosen Q_0 . There are several issues that arise immediately:

- Can we implement an algorithm based on the iteration (6)?
- Does the iteration converge (and in what sense)?
- How large should the iteration sample size N_j be?

The first issue is problem dependent, and we shall present an algorithm in the next section. The second question has not been answered yet in full generality. Homem-de-Mello and Rubinstein[10] and de Boer et al.[3] address this problem in the context of the estimation of tail probabilities (and buffer overflow probabilities in some queueing models); they

restrict the class of \mathcal{Q} to measures that are parameterized by finite dimensional vectors. Convergence means here convergence of vectors, e.g. ℓ_1 -norm, and [10] showed this in a slightly adapted version of the iteration. The third issue—the sample size—is also problem dependent and one has to experiment with it. Typically, the sample size N_j in the j -th iteration is small, much smaller than the sample size N for estimating finally the probability γ . However, it should be large enough so that the generated sample paths may solve the optimization program in the iteration (6).

6 A Cross Entropy Algorithm for the Failure Entrance Probability

Let $Q \in \mathcal{Q}$ and denote $q_{\mathbf{x}\mathbf{y}}$ for the transition probabilities of the Markov chain $\{\mathbf{X}_n : n = 0, 1, \dots\}$ when Q is implemented. For $\omega \in \Omega_0$ (finite sample path ending in either the failure set F or in state $\mathbf{0}$) let $U_{\mathbf{x}\mathbf{y}}(\omega)$ be the number of transitions from state \mathbf{x} to \mathbf{y} . Because the process is a Markov chain, we have

$$dQ(\omega) = \prod_{(\mathbf{x}, \mathbf{y}) \in S \times S} (q_{\mathbf{x}\mathbf{y}})^{U_{\mathbf{x}\mathbf{y}}(\omega)},$$

and

$$\frac{dP}{dQ_j}(\omega) = \prod_{(\mathbf{x}, \mathbf{y}) \in S \times S} \left(\frac{p_{\mathbf{x}\mathbf{y}}}{q_{\mathbf{x}\mathbf{y}}^{(j)}} \right)^{U_{\mathbf{x}\mathbf{y}}(\omega)}.$$

We substitute these expressions in the cross entropy iteration (6) while denoting

$$z_i := Z(\omega^{(i)}), \quad \text{and} \quad u_i(\mathbf{x}, \mathbf{y}) := U_{\mathbf{x}\mathbf{y}}(\omega^{(i)}),$$

to obtain

$$\begin{aligned} \max_Q \sum_{i=1}^{N_j} \prod_{(\mathbf{x}, \mathbf{y})} \left(\frac{p_{\mathbf{x}\mathbf{y}}}{q_{\mathbf{x}\mathbf{y}}^{(j)}} \right)^{u_i(\mathbf{x}, \mathbf{y})} z_i \log \prod_{(\mathbf{x}, \mathbf{y})} (q_{\mathbf{x}\mathbf{y}})^{u_i(\mathbf{x}, \mathbf{y})} \\ = \max_Q \sum_{i=1}^{N_j} \sum_{(\mathbf{x}, \mathbf{y})} \ell_i z_i u_i(\mathbf{x}, \mathbf{y}) \log q_{\mathbf{x}\mathbf{y}}, \end{aligned} \quad (7)$$

with likelihood ratios

$$\ell_i := \prod_{(\mathbf{x}, \mathbf{y})} \left(\frac{p_{\mathbf{x}\mathbf{y}}}{q_{\mathbf{x}\mathbf{y}}^{(j)}} \right)^{u_i(\mathbf{x}, \mathbf{y})}.$$

Notice that in problem (7) $\ell_i, z_i, u_i(\mathbf{x}, \mathbf{y})$ are numbers returned from the simulation of N_j sample paths, and that the optimization variables are $(q_{\mathbf{x}\mathbf{y}})_{(\mathbf{x}, \mathbf{y}) \in S \times S}$. In order to be transition probabilities, they satisfy the conditions $q_{\mathbf{x}\mathbf{y}} \geq 0$ for all $\mathbf{x}, \mathbf{y} \in S$, and

$$\sum_{\mathbf{y} \in S} q_{\mathbf{x}\mathbf{y}} = 1, \quad \text{for all } \mathbf{x} \in S. \quad (8)$$

We construct the Lagrange function associated with optimization (7) under the restriction (8), and solve the first order condition by differentiating partially with respect to $q_{\mathbf{x}\mathbf{y}}$. After some straightforward calculus we get the solution

$$q_{\mathbf{x}\mathbf{y}}^{(j+1)} = \frac{\sum_{i=1}^{N_j} \ell_i z_i u_i(\mathbf{x}, \mathbf{y})}{\sum_{i=1}^{N_j} \ell_i z_i \sum_{\mathbf{y}} u_i(\mathbf{x}, \mathbf{y})}. \quad (9)$$

It says that the new transition probability from \mathbf{x} to \mathbf{y} is the fraction of the $\mathbf{x} \rightarrow \mathbf{y}$ transitions made by only those generated sample paths that reach the failure set ($z_i = 1$) and compensated by the likelihood ratios (ℓ_i).

The cross entropy algorithm reads as:

Step 0: Initialize a measure $Q_0 \in \mathcal{Q}$. Set $j = 0$.

Step 1: Generate N_j sample paths randomly using measure Q_j .

Step 2: Apply the update formula (9).

Step 3: Stop when convergence has been obtained, otherwise set $j = j + 1$ and go to step 1.

Because the sample size N_j in the j -th cross entropy iteration is small, there will be (many) transitions $\mathbf{x} \rightarrow \mathbf{y}$ that have positive probability $p_{\mathbf{x}\mathbf{y}} > 0$ originally, but do not occur in the CE-iteration, and would give update $q_{\mathbf{x}\mathbf{y}}^{(j+1)} = 0$. This is highly undesirable, since once a transition gets zero probability, it remains zero, and this leads finally to very unreliable estimates of γ . To overcome this problem we suggest to adapt the update rule, where we consider two possibilities:

- Weighting with the original probabilities

$$q_{\mathbf{x}\mathbf{y}}^{(j+1)} = \alpha p_{\mathbf{x}\mathbf{y}} + (1 - \alpha) \frac{\sum_{i=1}^{N_j} \ell_i z_i u_i(\mathbf{x}, \mathbf{y})}{\sum_{i=1}^{N_j} \ell_i z_i \sum_{\mathbf{y}} u_i(\mathbf{x}, \mathbf{y})}, \quad (10)$$

for some $\alpha \in (0, 1)$.

- Smoothing with the previous iteration

$$q_{\mathbf{x}\mathbf{y}}^{(j+1)} = \alpha q_{\mathbf{x}\mathbf{y}}^{(j)} + (1 - \alpha) \frac{\sum_{i=1}^{N_j} \ell_i z_i u_i(\mathbf{x}, \mathbf{y})}{\sum_{i=1}^{N_j} \ell_i z_i \sum_{\mathbf{y}} u_i(\mathbf{x}, \mathbf{y})},$$

for some $\alpha \in (0, 1)$.

Choice of the Initial Measure Q_0

The initial probability measure Q_0 should be chosen in such a manner that the set A is not rare. There are several possibilities.

- Use a—simple—bias scheme, such as SFB.
- Take the uniform transition matrix: whenever $p_{\mathbf{x}\mathbf{y}} > 0$ let

$$q_{\mathbf{x}\mathbf{y}}^{(0)} = 1/\{\text{number of transitions out of state } \mathbf{x}\}.$$

7 Examples

We give three examples; the first example is small (125 states) but it has two difficulties, group repair and unbalanced failure rates so that neither Property 4.1 nor Property 4.2 hold. The second example is from Juneja and Shahabuddin[12] where the Markov chain has 3125 states with balanced failure rates (Property 4.2) and group repair (so Property 4.1 does not hold). The last example is a bit larger, 40320 states, but now both Properties 4.1 and 4.2 hold.

For each of these examples we present (i) the exact probability γ obtained numerically by the Gauss-Seidel iteration method for linear systems; (ii) the results of the importance sampling estimator using the biasing schemes (except SB-RBS) described in Section 4; (iii) the results of the importance sampling estimator using the cross entropy method. Since we know the exact value, we calculate the numerical relative error of all estimates: ratio of absolute error to the exact value. The performance of the estimators is given through their statistical relative error: ratio of the estimated standard deviation to the estimated mean. And the estimators are compared through the gain: improvement in variance taking into account the longer computing times,

$$\text{gain} = \frac{\text{Var}[\hat{\gamma}] \times \text{CPU time normal simulations}}{\text{Var}_Q[\hat{\gamma}_Q] \times \text{CPU time IS simulations with } Q}.$$

These three performances are summarized in one number that we call the performance number:

$$\text{performance number} := \log \left(\frac{1}{\text{num. rel. error estimate}} \times \frac{1}{\text{stat. rel. error estimator}}, \times \text{gain} \right).$$

Notice that we treat the three performance measures on the same importance level in this formula: a ten times higher gain equals a ten times improvement in one of the relative errors. In our view there is no reason to discriminate between the three measures, because a good importance sampling algorithm should provide speed up of simulation times as well as reliable estimates. When only speed would be the critical issue, the formula is

easily adapted by giving the gain more weight. The logarithm in the formula is only for purposes of reducing the resulting number, any other increasing function may be substituted.

The importance sampling simulations for estimating γ use a constant number of events, where an event is a transition of the Markov chain. In the cross entropy method we first simulate, in the j -th iteration, N_j sample paths ending in state $\mathbf{0}$ or the failure set F , and then simulate the same number of events as in the biasing methods for obtaining a γ estimate. Both parts are counted in the CPU time.

We use the following notation:

SFB(θ) means the simple failure biasing scheme with parameter $\theta \in (0, 1)$, see [17].

BFB(θ) means the balanced failure biasing scheme with parameter $\theta \in (0, 1)$, see [17].

IGBS($p; \delta$) means the implementable general failure biasing scheme with parameter $p, \delta \in (0, 1)$, see [12].

CE($k; n; \alpha$) means the cross entropy method with k iterations each of $N_j = n$ sample paths and with matrix updates using the weighting factor α , see Section 6.

Remark 7.1. In all our cross entropy simulations we use

- initial matrix \mathbf{Q}_0 with uniform transition probabilities;
- updating via weighting (10);
- the same number $N_j = n$ for the iteration sample sizes.

In the examples below we will present also the results of the importance sampling estimator based on the matrix \mathbf{Q}_0 to illustrate that under the associated initial measure Q_0 of the cross entropy method the event A is indeed less rare.

7.1 Example 1

There are 3 types, each with 4 components, with failure rates $\lambda_1 = \epsilon^2, \lambda_2 = \lambda_3 = \epsilon$ (unbalanced), and repair rates $\mu_i \equiv 1$. There is a single repairman who applies preemptive priority according to $1 > 2 > 3$ (type 1 has highest priority, etc). For types 1 and 2 group repair starts after two of that type have failed: all failed components of that type are repaired simultaneously (when assigned to the repairman) at a rate μ_i : it is a common feature in these models that the repair rate remains μ_i for the group repair. Type 3 components are repaired one by one as soon as one has failed (and type 3 is assigned to the repairman). Finally, the system breaks down as soon as all components of all types have failed.

- $\epsilon = 0.1$.

$\gamma = 1.179 \cdot 10^{-7}$ exact (numerically).

The normal simulation used 2,000,000,000 events.

The IS methods used 10,000,000 events.

All IS results are averages of 5 repetitions.

method	RE estimate	RE estimator	gain	performance number
uniform	0.6155	0.2052	3588	10.25
BFB(0.6)	0.4409	0.2060	4475	10.81
BFB(0.7)	0.5690	0.1375	5544	11.17
BFB(0.8)	0.5935	0.1944	20033	12.03
BFB(0.9)	0.7037	0.1216	9862	11.65
IGBS(0.6;0.1)	0.6002	0.1984	8502	11.18
IGBS(0.7;0.25)	0.5864	0.1133	10893	12.01
IGBS(0.8;0.05)	0.7391	0.1533	16313	11.88
IGBS(0.8;0.25)	0.5988	0.0976	12890	12.30
IGBS(0.9;0.35)	0.6797	0.1301	14239	11.99
CE(3;3000;0.1)	0.0459	0.0147	196309	19.49
CE(5;2500;0.15)	0.0763	0.0408	50502	16.60
CE(5;3000;0.1)	0.0159	0.0299	18267	17.46
CE(7;2000;0.2)	0.0520	0.0440	16066	15.76
CE(7;3000;0.15)	0.0107	0.0162	114716	20.31

- Observations: we experimented with an extensive range of θ 's in the BFB(θ) schemes and of $(p; \delta)$ combinations in IGBS(p, δ) schemes. All estimates are poor (relative error of 50% and worse), though the gain may be considerable. The performance numbers of the BFB schemes and the IGBS schemes are all in the range 8 - 12 showing not much difference between those of BFB and IGBS. On average the gains of the IGBS schemes are slightly higher. The schemes shown in the table are among those with the best performance numbers.

Also, we experimented with various CE($k; n; \alpha$) schemes. Here we found excellent estimates (relative errors of 1% occurring) and huge gains. It suffices to do only a few iterations, 3 - 7, while choosing the sample size of order 2000 - 3000 and the weighting factor $\alpha \approx 0.1 - 0.2$. Some of the best are tabulated, and, clearly, they outperform the best bias schemes.

7.2 Example 2

There are 5 types, each with 4 components, with failure rates $\lambda_i \equiv \epsilon$ (balanced), and repair rates $\mu_i \equiv 1$. There is a single repairman who applies preemptive priority according to $1 > 2 > 3 > 4 > 5$ (type 1 has highest priority, etc). For types 1 and 2, group repair starts after two of that type have failed: all failed components of that type are repaired simultaneously (when assigned to the repair man). Type 3,4 and 5 components are repaired one by one as soon as one has failed (and gets assigned to the repairman). The system breaks down as soon as all components of at least one type have failed.

- $\epsilon = 0.001$.

$\gamma = 1.916 \cdot 10^{-6}$ exact (numerically).

The normal simulation used 50,000,000 events.

The IS methods used 25,000,000 events.

Results are averages of 5 repetitions.

method	RE estimate	RE estimator	gain	performance number
uniform	0.4878	0.2313	59	6.25
SFB(0.4)	0.2368	0.2646	28	6.12
SFB(0.85)	0.6778	0.1975	207	7.34
SFB(0.9)	0.6299	0.2286	187	7.17
IGBS(0.5; 0.09)	0.0385	0.0807	78	10.13
IGBS(0.7; 0.09)	0.0335	0.0741	87	10.47
IGBS(0.8; 0.06)	0.0706	0.0692	115	10.07
CE(3;3000;0.1)	0.0390	0.0064	48812	19.10
CE(3;2500;0.2)	0.0474	0.0083	48553	18.63
CE(5;1500;0.15)	0.0573	0.0050	52148	19.01
CE(5;1500;0.2)	0.0555	0.0060	79603	19.30

- Observations: generally, similar to Example 1. The IGBS schemes show good estimates and small relative errors, though the gain is not that much. Overall the IGBS schemes improve greatly the SFB schemes, however, they are outperformed by the cross entropy schemes. The latter have excellent performances in all three criteria. Again, a few iterations suffices taking a a small weighting factor $\alpha \approx 0.1 - 0.2$. The sample sizes may vary a bit more.

7.3 Example 3

There are 6 types, with 5, 4, 6, 3, 7, 5 components respectively, with balanced (but different) failure rates ($2.5\epsilon, \epsilon, 5\epsilon, 3\epsilon, \epsilon, 5\epsilon$) and different repair rates (1.0, 1.5, 1.0, 2.0, 1.0, 1.5).

There is a single repairman who applies preemptive priority according to $1 > 2 > 3 > 4 > 5 > 6$ (type 1 has highest priority, etc). All types are repaired one by one as soon as one has failed (and gets assigned to the repairman). The system breaks down as soon as all components of at least one type have failed.

- $\epsilon = 0.001$.

$\gamma = 7.488 \cdot 10^{-7}$ exact (numerically).

The normal simulation used 200,000,000 events.

The IS methods used 1,000,000 events.

Results are averages of 5 repetitions.

method	RE estimate	RE estimator	gain	performance number
uniform	0.0358	0.0430	1045	13.43
SFB(0.5)	0.0900	0.2433	38	7.45
SFB(0.7)	0.1742	0.2235	51	7.19
BFB(0.7)	0.0232	0.0509	756	13.37
BFB(0.8)	0.0169	0.0451	975	14.06
IGBS(0.6; 0.2)	0.1211	0.2205	54	7.62
IGBS(0.7; 0.2)	0.1156	0.2455	51	7.50
CE(3;2500;0.1)	0.0271	0.0024	194115	21.82
CE(3;2500;0.15)	0.0277	0.0092	59104	19.26
CE(3;2500;0.2)	0.0134	0.0120	44592	19.43

- Observations: the SFB and IGBS schemes showed the worst performance, though the failure rates are balanced. Surprisingly, the simple uniform rule gives a performance that matches that of the BFB schemes. Again, a few iterations of the cross entropy method outperforms the bias schemes. Experimenting with more iterations (e.g. 5), we found that the gain drops due to the fact that the relative errors remain about the same but the running times increase.

8 Conclusion

We investigated the applicability and usability of the cross entropy method for rare event simulation in Markovian reliability models. Traditional biasing schemes are susceptible to the structure of the underlying Markov chain, implicating that simple biasing schemes may fail, in which cases one needs to implement more complicated biasing algorithms. On the other hand, our three examples—with three different structures—showed that one cross entropy implementation worked for all of them, and that the cross entropy estimators had excellent performance measures, outperforming the biasing estimators.

Furthermore it is quite remarkable that the parameters of our cross entropy algorithm, viz. iteration sample sizes, number of iterations, and weighting factor, might be chosen at about the same values in our three examples, though these are different of size and structure.

In our implementation of the cross entropy method we need to optimize the full matrix of transition probabilities, sometimes this is called a state dependent tilting. Although current computer powers allow storage of huge matrices, there is a limit. An alternative importance sampling method would be to implement a state independent tilting, the biasing schemes SFB and BFB are typical examples of this. In a state independent tilting the transition probabilities are parameterized by a vector of low dimensions, and it would be interesting to investigate whether cross entropy might be applicable to find the optimal values of the parameters. In fact, this corresponds to considering a subclass of the class of probability measures \mathcal{Q} in the optimization program (4).

Acknowledgement

The author likes to thank Sing-Kong Cheung for his master thesis on the same topic, and the referees for their helpful comments.

References

- [1] G. Alon, D. Kroese, T. Raviv and R.Y. Rubinstein, “Application of the cross entropy method for optimal buffer allocation in a simulation based environment”, submitted.
- [2] S. Asmussen and R. Rubinstein, “Steady state rare event simulation in queueing models and its complexity properties”. In J. Dshalalow (ed.), *Advances in queueing theory, theory, methods and open problems*, pp. 429–461, CRC Press, 1995.
- [3] P-T. de Boer, D.P. Kroese and R.Y. Rubinstein, “A fast cross-entropy method for estimating buffer overflows in queueing networks”, submitted, 2002.
- [4] P-T. de Boer, V.F. Nicola and R.Y. Rubinstein, “Adaptive importance sampling simulation of queueing networks”, *Proceedings of the 2000 Winter Simulation Conference*, Orlando, Florida, pp. 646–655, 2000.
- [5] A. Goyal, W.C. Carter, E. de Souza e Silva, S.S. Lavenberg and K.S. Trivedi, “The system availability estimator”, *Proceedings of 16-th Annual International Symposium on Fault Tolerance Computing*, Vienna, Austria, pp. 84–89, 1986.

- [6] A. Goyal and S.S. Lavenberg, “Modeling and analysis of computer system availability”, *IBM Journal of Research and Development* 31, pp. 651–664, 1987.
- [7] A. Goyal, S.S. Lavenberg and K.S. Trivedi, “Probabilistic modeling of computer system availability”, *Annals of Operations Research*, 8, pp. 285–306, 1987.
- [8] A. Goyal, P. Shahabuddin, P. Heidelberger, V.F. Nicola and P.W. Glynn, “A unified framework for simulating Markovian models of highly dependable systems”, *IEEE Transactions on Computers*, 41, pp. 36–51, 1992.
- [9] P. Heidelberger, “Fast simulation of rare events in queueing and reliability models”, *ACM Transactions on Modelling and Computer Simulation* 5, pp. 43–85, 1995.
- [10] T. Homem-de-Mello T. and R.Y. Rubinstein, “Rare event estimation for static models via cross-entropy and importance sampling”, submitted, 2002.
- [11] K-P. Hui, N. Bean, M. Kraetzl and D.P. Kroese, “The cross-entropy method for network reliability”, submitted, 2003.
- [12] S. Juneja and P. Shahabuddin, “Fast simulation of Markov chains with small transition probabilities”, *Management Science* 47, pp. 547–562, 2001.
- [13] S. Juneja and P. Shahabuddin, “Splitting-based importance sampling algorithm for fast simulation of Markov reliability models with general repair policies”, *IEEE Transactions on Reliability* 50, pp. 235–245, 2001.
- [14] D. Lieber, A. Nemirovskii and R.Y. Rubinstein, “A fast Monte Carlo method for evaluation of reliability indices”, *IEEE Transactions on Reliability* 48, pp. 256–261, 1999.
- [15] R.Y. Rubinstein, “Optimization of computer simulation models with rare events”, *European Journal of Operations Research*, 99, pp. 89–112, 1997.
- [16] R.Y. Rubinstein, “The cross-entropy method for combinatorial and continuous optimization”, *Methodology and Computing in Applied Probability*, 1, pp. 127–190, 1999.
- [17] P. Shahabuddin, “Importance sampling for the simulation of highly reliable Markovian systems”, *Management Science* 40, pp. 333–352, 1994.
- [18] P. Shahabuddin, “Fast transient simulation of Markovian models of highly dependable systems”, *Performance Evaluation* 20, pp. 267–286, 1994.