Rare event simulation for a static distribution

Frédéric Cérou, Pierre del Moral, Teddy Furon, Arnaud Guyader

Rare event simulation for a static distribution[*]

F. Cérou[*][†]   P. Del Moral[‡]   T. Furon[†]   A. Guyader[§]

September 5, 2008

1   Introduction and motivation

The goal of this work is to deal with rare events for a fixed probability law. Unlike many other works concerning rare event estimation and simulation, we are simply concerned here with events of the type $\{X \in A\}$ for some random element X, with $\mathbb{P}(X \in A) \ll 1$, and with no dynamical model for X (i.e. X is not a process indexed by the time). In order to use the framework developped for Markov processes (see [2, 5]), we construct a family of Markov transition kernels whose invariant measures are the law of X restricted on smaller and smaller sets, the smallest being A. As usual when using a splitting technique in rare event simulation, we decompose the rare event in not so rare nested events, with the product of probabilities being the probability of the rare event.

Our motivation for this framework comes from problems occurring in watermarking of digital contents. Here the term watermarking refers to a set of techniques for imbedding/hiding information in a digital file (typically audio or video), such that the change is not noticed, and very hard to remove. See [11] for details. In order to be used in an application, a watermarking technique must be reliable. Here are two application scenarii where a wrong estimation of the probability of error could lead to a disaster.

Copy protection.   Assume commercial contents are encrypted and watermarked and that future consumer electronics storage devices have a watermark detector. These devices refuse to record a watermarked content. The probability of false alarm is the probability that the detector considers an original piece of content (which has not been watermarked) as protected. The movie that a user shot during his holidays could be rejected by his storage device. This absolutely non user-friendly behavior really scares consumer electronics manufacturers. In the past, the Copy Protection Working Group of the DVD forum evaluated that at most one false alarm should happen in 400 hours of video [11]. As the detection rate was one decision per ten seconds, this implies a probability of false alarm in the order of 10^{-5}. An accurate experimental assessment of such a low probability of false alarm would demand to feed a real-time watermarking detector with non-watermarked content during 40,000 hours, *i.e.* more than 4 years! Proposals in response of the CPTWG's call were, at that time, never able to guarantee this level of reliability.

Fingerprinting.   In this application, users' identifiers are embedded in purchased content. When content is found in an illegal place (*e.g.* a P2P network), the right holders decode the hidden message, find a serial number, and thus they can trace the traitor, *i.e.* the customer who has illegally broadcast their copy. However, the task is not that simple because dishonest users might collude. For security reason, anti-collusion

codes have to be employed. Yet, these solutions (also called weak traceability codes [1]) have a non-zero probability of error (defined as the probability of accusing an innocent). This probability should be, of course, extremely low, but it is also a very sensitive parameter: anti-collusion codes get longer (in terms of the number of bits to be hidden in content) as the probability of error decreases. Fingerprint designers have to strike a trade-off, which is hard to conceive when only rough estimation of the probability of error is known. The major issue for fingerprinting algorithms is the fact that embedding large sequences implies also assessing reliability on a huge amount of data which may be practically unachievable without using rare event analysis.

2   Assumptions and ingredients

We assume that X is a random element on $\mathbb{R}^d$ for some $d > 0$, and denote its probabity law by μ. We denote by A the rare set of interest, and we assume that $A = \{x \in \mathbb{R}^d \text{ s.t. } \Phi(x) \geq L\}$ for some continuous function $\Phi : \mathbb{R}^d \mapsto \mathbb{R}$ and some real number L. We also assume that we know how to draw i.i.d. samples from μ.
Now to construct the algorithm, we will need to choose the following ingredients. First we need to choose an increasing sequence in $\bar{\mathbb{R}}$ $\{L_0, \ldots, L_n\}$, with $L_0 = -\infty$ and $L_n = L$. If we denote $A_j = \{x \in \mathbb{R}^d, \ \Phi(x) > L_j\}$, we get a family of nested sets $\mathbb{R}^d = A_0 \supset A_1 \supset \cdots \supset A_n = A$ such that $\mathbb{P}(X \in A_k | X \in A_{k-1})$ is not too small. For indices $m > n$, we assume that $L_m = L_n$ and $A_m = A_n$. We also need to choose a Markov transition kernel K on $\mathbb{R}^d$ having μ as an invariant measure, and which is μ-symmetric, that is

$$\forall (x, y) \in \mathbb{R}^{2d}, \ K(x, dy)\mu(dx) = K(y, dx)\mu(dy).$$

As we will see in the sequel, the choice of the L_j's can be made adaptive and is thus not an issue. But the choice of the kernel K is crucial. Even if any μ-symmetric kernel would eventually do the job, we need to carefully choose it to make the algorithm efficient. We will discuss this point later on.
Now we can consider the following Markov chain: $X_0 \sim \mu$ and the inhomogeneous transitions given by $P(X_n \in dy | X_{n-1} = x) = M_n^K(x, dy)$, with

$$M_k^K(x, dy) = K(x, dy)\mathbb{1}_{A_k}(y) + K(x, A_k^c)\,\delta_x(dy).$$

For $k \in \{0, \ldots, n\}$, let us denote $\mu_k(dx) = \frac{1}{\mu(A_k)}\,\mathbb{1}_{A_k}(x)\,\mu(dx)$ the normalized restriction of μ on A_k.

Proposition 1. *The measure μ_k is invariant by the transition kernel M_k^K.*

Proof

$$
\begin{aligned}
\int_x \mu_k(dx) M_k^K(x, dy) &= \int_x \mu_k(dx)(K(x, dy)\mathbb{1}_{A_k}(y) + K(x, A_k^c)\,\delta_x(dy)) \\[2mm]
&= \int_x \int_z \mu_k(dx)K(x, dz)(\mathbb{1}_{A_k}(z)\delta_z(dy) + \mathbb{1}_{A_k^c}(z)\delta_x(dy)) \\[2mm]
&= \int_x \frac{1}{\mu(A_k)}\mathbb{1}_{A_k}(x)\mu(dx)K(x, dy)\mathbb{1}_{A_k}(y) + \int_z \frac{1}{\mu(A_k)}\mathbb{1}_{A_k}(y)\mu(dy)K(y, dz)\mathbb{1}_{A_k^c}(z) \\[2mm]
&= \int_x \frac{1}{\mu(A_k)}\mathbb{1}_{A_k}(x)\mu(dx)K(x, dy)\mathbb{1}_{A_k}(y) + \int_z \frac{1}{\mu(A_k)}\mathbb{1}_{A_k}(y)\mu(dz)K(z, dy)\mathbb{1}_{A_k^c}(z) \\[2mm]
&= \mu_k(dy).
\end{aligned}
$$
∎

Proposition 2. *For every test function φ, for $k \in \{0, \ldots, n\}$, we have the following Feynman-Kac representation*

$$\mu_{k+1}(\varphi) = \frac{\mathbb{E}[\varphi(X_k) \prod_{j=0}^{k} \mathbb{1}_{A_{j+1}}(X_j)]}{\mathbb{E}[\prod_{j=0}^{k} \mathbb{1}_{A_{j+1}}(X_j)]}.$$

proof   We use induction to show that

$$\mathbb{E}[\varphi(X_k)\prod_{j=0}^{k}\mathbb{1}_{A_{j+1}}(X_j)] = \mu(A_{k+1})\,\mu_{k+1}(\varphi).$$

The case $k = 0$ is obvious. Then assume the property true for $k - 1$. We write, using the Markov property and proposition 1,

$$
\begin{aligned}
\mathbb{E}[\varphi(X_k)\prod_{j=0}^{k}\mathbb{1}_{A_{j+1}}(X_j)] &= \mathbb{E}[\mathbb{E}[\varphi(X_k)\mathbb{1}_{A_{k+1}}(X_k)|X_0,\ldots,X_{k-1}]\prod_{j=0}^{k-1}\mathbb{1}_{A_{j+1}}(X_j)] \\[2mm]
&= \mathbb{E}[M_k^K(\varphi\mathbb{1}_{A_{k+1}})(X_{k-1})\prod_{j=0}^{k-1}\mathbb{1}_{A_{j+1}}(X_j)] \\[2mm]
&= \mu(A_k)\,\mu_k(M_k^K(\varphi\mathbb{1}_{A_{k+1}})) \\[2mm]
&= \mu(A_k)\,\mu_k(\varphi\mathbb{1}_{A_{k+1}}).
\end{aligned}
$$

Now it is obvious that

$$\mu_k(\varphi\mathbb{1}_{A_{k+1}}) = \mu_{k+1}(\varphi)\frac{\mu(A_{k+1})}{\mu(A_k)}.$$

Then taking the case $\varphi = \mathbb{1}$ we have

$$\mathbb{E}[\prod_{j=0}^{k-1}\mathbb{1}_{A_{j+1}}(X_j)] = \mu(A_k), \tag{1}$$

which concludes the proof.

∎

3   The algorithm

From proposition 2 we see that we are in the framework of Feynman-Kac formulae, and thus we can construct an approximation of the associated measures using an interacting particle method as the one studied in [4]. Basically, at each iteration k, it consists in propagating the particles according to the transitions given by M_k^K, and then a selection of the particles according to the potential, here $\mathbb{1}_{A_{k+1}}$ (i.e. a zero-one valued function in this case).
Also note that moving a particle according to M_k^K is twofold: first we propose a new transition according to K, and accept the transition only if it stays in A_k, keeping the old position otherwise.
Concerning the approximation of the rare event probability, we just consider the following obvious property

$$\mathbb{P}(X \in A_n) = \prod_{j=0}^{n}\mathbb{P}(X \in A_{j+1}|X \in A_j) = \mathbb{E}[\prod_{j=0}^{n}\mathbb{1}_{A_{j+1}}(X_j)] = \prod_{j=0}^{n}\frac{\mathbb{E}[\mathbb{1}_{A_{j+1}}(X_j)\prod_{m=0}^{j-1}\mathbb{1}_{A_{m+1}}(X_m)]}{\mathbb{E}[\prod_{m=0}^{j-1}\mathbb{1}_{A_{m+1}}]}.$$

We see here that we can approximate at each stage $\mathbb{P}(X \in A_{j+1}|X \in A_j)$ by the proportion of the particles already in the next set, and the total probability is estimated as the product of those.
This gives the algorithm 1, which can also be considered in the rare event simulation framework as a kind of importance splitting method, introduced for example by [7] in the context of particle physics, or in the telecommunication area by [14].

Algorithm 1

<div align="center">

Parameters

</div>

N the number of particles, the sequence $\{L_0, \ldots, L_n\}$ of levels.

<div align="center">

Initialization

</div>

Draw an i.i.d. N-sample $\xi_0^j, j = 1, \ldots, N$ of the law μ.

<div align="center">

Iterations

</div>

for $k = 0$ to n /* level number */

    Let $I_0 = \{j \,/\, \xi_k^j \in A_{k+1}\}$.

    for $j \in I_0$, let $\tilde{\xi}_{k+1}^j = \xi_k^j$, and for $j \notin I_0$, let $\tilde{\xi}_{k+1}^j$ be a copy of ξ_k^ℓ where ℓ is chosen randomly in I_0 with uniform probabilities.

    Let $p_k = \frac{|I_0|}{N}$.

    From each sample $\tilde{\xi}_{k+1}^j, j = 1 \ldots N$, draw a new sample $\hat{\xi}_{k+1}^j \sim K(\tilde{\xi}_{k+1}^j, .)$.

    If $\hat{\xi}_{k+1}^j \in A_{k+1}$ then let $\xi_{k+1}^j = \hat{\xi}_{k+1}^j$, and $\xi_{k+1}^j = \tilde{\xi}_{k+1}^j$ otherwise.

endfor

<div align="center">

Output

</div>

Estimate the probability of the rare event by $\hat{P}_A = \prod_{h=1}^{n} p_h$.

The last set of particles is a (non independent) identically distributed sample of the law of the rare event μ_n.

The asymptotic behavior as the number of particles $N \to \infty$ of the interacting particle model we have constructed has been extensively studied in [4]. For example Proposition 9.4.1 and remark 9.4.1 give that $\sqrt{N}(\hat{P}_A - P(X \in A))$ converges, as $N \to +\infty$ in distribution to a centered Gaussian with variance σ^2. Unfortunately, this asymptotic variance is often not explicit, and depends on the kernel K in a complicated way.

4 Tuning the algorithm

4.1 Choice of the kernel K

The choice of the transition kernel K is of course critical, and in practice will depend on the application, so that we cannot give a completely general way of finding it. But in the case of a Gibbs measure given by a bounded potential, we can use the Metropolis algorithm, as first proposed by [9], or a variant later proposed by Hastings [6].

4.2 Adaptive level sets

As we may not have a great insight about the law μ, the choice of the levels $L_1, \ldots, L_n$ might prove to be quite tricky. For example, we want to avoid the particle system to completely die by choosing two consecutive level too far appart. As proposed in [3], the level sets can be chosen adaptively, ensuring that the particle system will not die, and that the levels are distributed in a way to minimize the asymptotic variance of the estimate of the rare event probability.

The method is very simple. We choose a prescribed success rate p between two consecutive levels. In practice, $0.75 \le p \le 0.8$ works very well. After each application of the kernel M_k^K, we sort the particles ξ_{k+1}^j according to their scores $\Phi(\xi_{k+1}^j)$. Then we choose as the next level the quantile $l_{k+1} = \Phi(\xi_{k+1}^{j_0})$ such that a proportion p of the particles are above it. From this level l_{k+1}, one defines A_{k+1}, and the rest of the algorithm is unchanged. We end up with a variant where the levels are evenly spaced in terms of probability of succes, which, as mentioned in [8] and [2], gives a minimal asymptotic variance.

The algorithm then stops when some $l_{n_0} \ge L$, and the probability is estimated by $\hat{P}_A = p^{n_0-1} \ r$, with r being the actual number of particles in the last iteration being above level L. Note that the number n_0 of steps is in theory random, but with N reasonably large, it is fixed by the ratio of the logarithms

$$\left\lceil \frac{\log \mathbb{P}(X \in A)}{\log p} \right\rceil \tag{2}$$

with a probability very close to 1.

The cost of adaptive levels in term of complexity is just a $\log N$ factor (from the quick sort), and in term of accuracy it introduces a bias which is asymtotically zero (actually, on numerical computations, it is decreasing in N^{-1}, and thus is negligeable compared to the standard deviation term).

4.3 Less dependent sample

Unlike all importance sampling based methods, our algorithm gives a sample distributed according to the real law of the rare event μ_n, but not independent. This may lead to poor variance behavior in some cases. The samples are not independent because of the splitting of successful particles. But the effect of M_k^K is towards more independence among particles. Thus we can think of iterating the kernel a fixed number of times, or until a fixed number of particles, say 90 or 95%, have actually moved from their first position (i.e. at least one proposed K transition has been accepted).

If we consider all the particles together, each application of the kernel can be seen as applying a kernel $(M_k^K)^{\otimes N}$. It is obvious from proposition 1 that $(\mu_k)^{\otimes N}$ (the joint law of an i.i.d. sample) is an invariant measure for $(M_k^K)^{\otimes N}$. Then from [10], Proposition 13.3.2, we get that the total variation norm between the law of the sample as we iterate the kernel, and $(\mu_k)^{\otimes N}$, is non increasing. So even if it is not very helpful, these iterations at least do not harm.

When the chosen kernel K is of Metropolis-Hastings type, so is M_k^K (with a potential that can be infinite). Then, using [13], we can say a bit more, provided (which is generally the case) that the kernel used for the proposed transitions is aperiodic and irreducible. By Corollary 2 in [13], the Metropolis is also Harris recurrent, and then by Theorem 13.3.3 in [10], we have for all initial distribution λ

$$\| \int \lambda(dx)(M_k^K)^m(x, .) - \mu_k \| \to 0 \text{ when } m \to +\infty,$$

where the norm is in total variation. Then we have for any initial cloud of particles $\Xi = (\xi^1, \ldots, \xi^N)$, and any test functions $(\phi^1, \ldots, \phi^N)$,

$$|\delta_\Xi((M_k^K)^{\otimes N})^m(\bigotimes_{j=1}^N \phi_j) - \prod_{j=1}^N \mu_k(\phi_j)|$$

$$= \ |\prod_{j=1}^N (M_k^K)^m(\phi_j)(\xi_j) - \prod_{j=1}^N \mu_k(\phi_j)| \to 0 \text{ when } m \to +\infty.$$
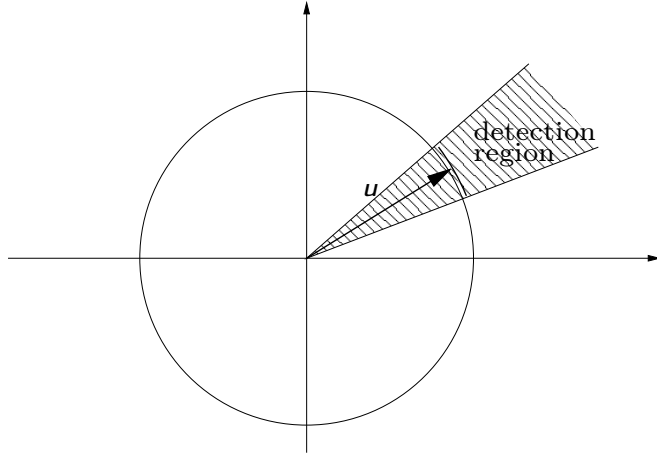
Figure 1: Detection region

By a standard density argument, we get that for all test function ϕ on $(\mathbb{R}^d)^N$,

$$|\delta_\Xi((M_k^K)^{\otimes N})^m(\phi) - \mu_k^{\otimes N}(\phi)| \to 0 \text{ when } m \to +\infty.$$

This means that the more iteration we do with the kernel, the closer we get from an independent sample.

5 Applications

5.1 Zero bit watermarking

The zero bit watermarkig is a toy example where X is a Gaussian vector in $\mathbb{R}^d$, with zero mean and identity covariance matrix, $\Phi(X) = \frac{\langle X, u \rangle}{\|X\|}$ and u is a fixed normalized vector. Then the region $A = \{x \in \mathbb{R}^d \ s.t. \ \Phi(x) > L\}$ is a cone as shown on figure 1. For a Gaussian distribution, the obvious choice for kernel is the following. If we start from any point x, then the new position is distributed like

$$x' = \frac{x + \alpha W}{\sqrt{1 + \alpha^2}},$$

where W is a $\mathcal{N}(0, Id)$ $\mathbb{R}^d$ valued random vector and α a positive number.
Note that here we can compare our estimates of the rare event probability with the result of a numerical integration. We have run our algorithm in this case with adaptive levels and iterations of the kernel until 90% of the particles have moved at each step.
For several numbers of particles, we have done the complete algorithm 200 times in order to estimate the bias and variance. Figure 2 shows the rate of convergence in N^{-1} for the relative bias, and figure 3 shows the convergence of the (normalized by the rare event probability) standard deviation to minimum achievable, which is that of i.i.d. samples at each stage, that is $\sqrt{N}((n_0 - 1)\frac{p}{1-p} + r_0)$ where n_0 is given by equation 2, and r_0 such that $\mathbb{P}(X \in A) = p^{n_0-1}r_0$ (see [3]).

5.2 Tardos probabilistic codes

We are interested here in embedding an identifier in each copy of a purchased content. Then a copy, maybe the result of several manipulations, or even a collusion, is found on the web, and we want to decide wether
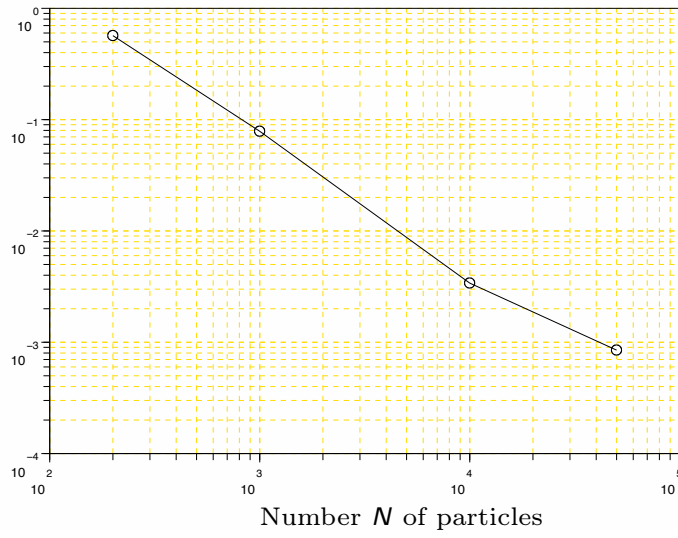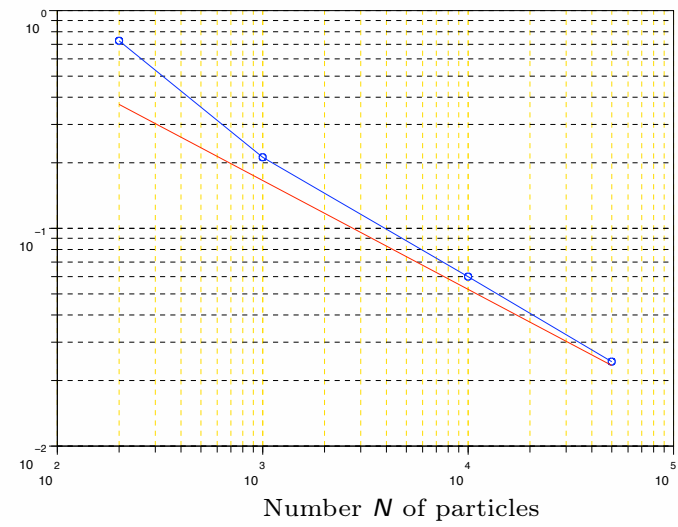
Figure 2: Relative bias.

Figure 3: Standard deviation.

or not it can be originating from a certain user. The rare event will be to accuse an innocent user to be guilty for this.

The embedded message consists of bits $X = (X_1, \ldots, X_m)$, where each X_i is independent from the others, and drawn from a Bernouilli $B(p_i)$. The p_i's are themselves random, drawn from a given distribution with density f on $[0,1]$. Then we find a copy with fingerprint $y = (y_1, \ldots, y_m) \in \{0,1\}^m$. We conclude that a user is guilty if the score $S(X) = \sum_{i=1}^m y_i g_i(X_i)$ is larger than some value L, for some given functions g_i's. This approach was proposed by Tardos in [12], where he derives good choices for f and the g_i's.

To apply our algorithm, we need to choose the kernel K. As the X_i are independent, we choose randomly r indices $\{j_1, \ldots, j_r\} \in \{1, \ldots, m\}$, with r being a fixed parameter. Then for each j_l, we draw a new X'_{j_l} independently from $B(p_i)$.

References

[1] A. Barg, G. R. Blakley, and G. A. Kabatiansky. Digital fingerprinting codes: problem statements, constructions, identification of traitors. *IEEE Trans. on Signal Processing*, 51(4):960–980, April 2003.

[2] F. Cérou, P. Del Moral, F. Le Gland, and P. Lézaud. Genetic genealogical models in rare event analysis. *Latin American Journal of Probability and Mathematical Statistics*, 1, 2006.

[3] Frédéric Cérou and Arnaud Guyader. Adaptive multilevel splitting for rare event analysis. *Stoch. Anal. Appl.*, 25(2):417–443, 2007.

[4] P. Del Moral. *Feynman-Kac formulae, Genealogical and interacting particle systems with applications.* Probability and its Applications. Springer-Verlag, New York, 2004.

[5] Pierre Del Moral and Pascal Lezaud. Branching and interacting particle interpretation of rare event probabilities. In Henk Blom and John Lygeros, editors, *Stochastic Hybrid Systems : Theory and Safety Critical Applications*, number 337 in Lecture Notes in Control and Information Sciences, pages 277–323. Springer–Verlag, Berlin, 2006.

[6] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970.

[7] H. Kahn and T.E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards Appl. Math. Series*, 12:27–30, 1951.

[8] Agnès Lagnoux. Rare event simulation. *Probability in the Engineering and Informational Sciences*, 20(1):45–66, 2006.

[9] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[10] S. P. Meyn and R. L. Tweedie. *Markov chains and stochastic stability.* Communications and Control Engineering Series. Springer-Verlag London Ltd., London, 1993.

[11] Copy protection technical working group. `www.cptwg.org`.

[12] G. Tardos. Optimal probabilistic fingerprint codes. In *Proc. of the 35th annual ACM symposium on theory of computing*, pages 116–125, San Diego, 2003. ACM.

[13] Luke Tierney. Markov chains for exploring posterior distributions. *Ann. Statist.*, 22(4):1701–1762, 1994. With discussion and a rejoinder by the author.

[14] M. Villén-Altamirano and J. Villén-Altamirano. RESTART : a straightforward method for fast simulation of rare events. In Jeffrey D. Tew, Mani S. Manivannan, Deborah A. Sadowski, and Andrew F. Seila, editors, *Proceedings of the 1994 Winter Simulation Conference, Orlando 1994*, pages 282–289, December 1994.