Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

SPLITTING FOR RARE-EVENT SIMULATION

Pierre L'Ecuyer
Valérie Demers

Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville
Montréal (Québec), H3C 3J7, CANADA

Bruno Tuffin

IRISA-INRIA, Campus Universitaire de Beaulieu
35042 Rennes Cedex, FRANCE

ABSTRACT

Splitting and importance sampling are the two primary techniques to make important rare events happen more frequently in a simulation, and obtain an unbiased estimator with much smaller variance than the standard Monte Carlo estimator. Importance sampling has been discussed and studied in several articles presented at the Winter Simulation Conference in the past. A smaller number of WSC articles have examined splitting. In this paper, we review the splitting technique and discuss some of its strengths and limitations from the practical viewpoint. We also introduce improvements in the implementation of the multilevel splitting technique. This is done in a setting where we want to estimate the probability of reaching B before reaching (or returning to) A when starting from a fixed state $x_0 \notin B$, where A and B are two disjoint subsets of the state space and B is very rarely attained. This problem has several practical applications.

1 SETTING

We consider a discrete-time Markov chain $\{X_j, j \geq 0\}$ with state space $\mathscr{X}$. Let A and B be two disjoint subsets of $\mathscr{X}$ and let $x_0 \in \mathscr{X} \setminus B$ be the initial state. The chain starts in state $X_0 = x_0$, leaves the set A if $x_0 \in A$, and then eventually reaches B or A. If $x_0 \in A$, time 0 is when the chain first exits from A. Let $\tau_A = \inf\{j > 0 : X_j \in A\}$, the first time when the chain hits A (or returns to A after leaving it), and $\tau_B = \inf\{j > 0 : X_j \in B\}$, the first time when the chain reaches the set B. The goal is to estimate $\gamma = \mathbb{P}[\tau_B < \tau_A]$, the probability that the chain reaches B before A. This particular form of rare-event problem, where γ is small, occurs in many practical situations (Shahabuddin 1994, Heidelberger 1995).

The *standard Monte Carlo* method estimates γ by running n independent copies of the chain up to the stopping time $\tau = \min(\tau_A, \tau_B)$, and counting the proportion of runs for which the event $\{\tau_B < \tau_A\}$ occurs. The resulting estimator $\hat{\gamma}_n$ has *relative error*

$$\mathrm{RE}[\hat{\gamma}_n] = \frac{(\mathrm{Var}[\hat{\gamma}_n])^{1/2}}{\gamma} = \frac{(\gamma(1-\gamma))^{1/2}}{n\gamma} \approx \frac{\gamma^{-1/2}}{n},$$

which increases to infinity when $\gamma \to 0$. This naive estimator is thus highly unreliable when γ is small.

An alternative unbiased estimator of γ, say $\tilde{\gamma}_n$, is said to have *bounded relative error* if $\lim_{\gamma \to 0^+} \mathrm{RE}[\tilde{\gamma}_n] < \infty$. This implies that

$$\lim_{\gamma \to 0^+} \frac{\log(\mathbb{E}[\tilde{\gamma}_n^2])}{\log \gamma} = 2. \tag{1}$$

When the latter (weaker) condition holds, the estimator $\tilde{\gamma}_n$ is said to be *asymptotically efficient* (Heidelberger 1995, Bucklew 2004). To take into account the computing cost of the estimator, it is common practice to consider the *efficiency* of an estimator $\tilde{\gamma}_n$ of γ, defined as $\mathrm{Eff}[\tilde{\gamma}_n] = 1/(\mathrm{Var}[\tilde{\gamma}_n]C(\tilde{\gamma}_n))$ where $C(\tilde{\gamma}_n)$ is the expected time to compute $\tilde{\gamma}_n$. *Efficiency improvement* means finding an unbiased estimator with larger efficiency than the one previously available. The estimator $\tilde{\gamma}_n$ has *bounded work-normalized relative error*, or *relative efficiency bounded away from zero*, if $\lim_{\gamma \to 0^+} \gamma^2 \mathrm{Eff}[\tilde{\gamma}_n] > 0$. It is *work-normalized asymptotically efficient* (a weaker condition) if $\lim_{\gamma \to 0^+} \log(C(\tilde{\gamma}_n)\mathbb{E}[\tilde{\gamma}_n^2])/\log \gamma = 2$. A sufficient condition for this is that (1) holds and $\lim_{\gamma \to 0^+} \log C(\tilde{\gamma}_n)/\log \gamma = 0$.

Splitting and importance sampling are the two major approaches to deal with rare-event simulation. *Importance sampling* increases the probability of the rare event by changing the probability laws that drive the evolution of the system. It then multiplies the estimator by an appropriate likelihood ratio to recover the correct expectation (i.e., so that the estimator remains unbiased for γ in the above setting). The main difficulty in general is to find a good way to change the probability laws. For the details, we refer the reader to Glynn and Iglehart (1989), Heidelberger (1995), Bucklew (2004), and many other references given there.

In the *splitting method*, the probability laws remain unchanged, but an artificial drift toward the rare event is created by terminating with some probability the trajectories that seem to go away from it and by *splitting* (cloning) those

that are going in the right direction. In general, an unbiased estimator is recovered by multiplying the original estimator by an appropriate factor (in some settings, this factor is 1). The method can be traced back to Kahn and Harris (1951) and has been studied (sometimes under different names) by several authors, including Booth and Hendricks (1984), Villén-Altamirano and Villén-Altamirano (1994), Melas (1997), Garvels and Kroese (1998), Glasserman et al. (1998), Glasserman et al. (1999), Fox (1999), Garvels (2000), Del Moral (2004), Cérou, LeGland, Del Moral, and Lezaud (2005), Villén-Altamirano and Villén-Altamirano (2006), and other references cited there.

The splitting methodology was invented to improve the efficiency of simulations of particle transport in nuclear physics; it is used to estimate the intensity of radiation that penetrates a shield of absorbing material, for example (Hammersley and Handscomb 1964, Spanier and Gelbard 1969, Booth and Hendricks 1984, Booth 1985, Booth and Pederson 1992, Pederson, Forster, and Booth 1997). This remains its primary area of application. It is also used to estimate delay time distributions and losses in ATM and TCP/IP telecommunication networks (Akin and Townsend 2001, Gorg and Fuss 1999). In a recent real-life application, splitting is used to estimate the probability that two airplanes get closer than a nominal separation distance, or even hit each other, in a stochastic dynamical model of air traffic where aircrafts are responsible for self-separation with each other (Blom et al. 2005).

In Section 2, we review the theory and practice of splitting in a setting where we want to estimate $\gamma = \mathbb{P}[\tau_B < \tau_A]$. We start with multilevel splitting and then discuss more general alternatives. For multilevel splitting, we propose new variants, more efficient than the standard implementations. Numerical illustrations are given in Section 3. L'Ecuyer, Demers, and Tuffin (2006) contains an expanded version of the present overview article. It also studies the combination of splitting and importance sampling with two types of randomized quasi-Monte Carlo methods: the "classical" one (e.g., Owen 1998, L'Ecuyer and Lemieux 2000) and the *array-RQMC* method for Markov chains proposed by L'Ecuyer, Lécot, and Tuffin (2005).

2 SPLITTING

2.1 Multilevel Splitting

We define the splitting algorithm via an *importance function* $h : \mathscr{X} \to \mathbb{R}$ that assigns a *importance value* to each state of the chain (Garvels, Kroese, and Van Ommeren 2002). We assume that $A = \{x \in \mathscr{X} : h(x) \le 0\}$ and $B = \{x \in \mathscr{X} : h(x) \ge \ell\}$ for some constant $\ell > 0$. In the *multilevel splitting* method, we partition the interval $[0, \ell)$ in m subintervals with boundaries $0 = \ell_0 < \ell_1 < \cdots < \ell_m = \ell$. For $k = 1, \ldots, m$, let $T_k = \inf\{j > 0 : h(X_j) \ge \ell_k\}$, let $D_k = \{T_k < \tau_A\}$ denote the event that $h(X_j)$ reaches level ℓ_k before reaching level 0, and define the conditional probabilities $p_k = \mathbb{P}[D_k \mid D_{k-1}]$ for $k > 1$, and $p_1 = \mathbb{P}[D_1]$. Since $D_m \subset D_{m-1} \subset \cdots \subset D_1$, we have

$$\gamma = \mathbb{P}[D_m] = \prod_{k=1}^{m} p_k.$$

The intuitive idea of multilevel splitting is to estimate each probability p_k "separately", by starting a large number of chains in states that are generated from the distribution of $X_{T_{k-1}}$ conditional on the event D_{k-1}. This conditional distribution, denoted by G_{k-1}, is called the (first-time) *entrance distribution at threshold* ℓ_{k-1}, for $k = 1, \ldots, m+1$ (G_0 is degenerate at x_0). Conceptually, the estimation is done in successive *stages*, as follows.

In the first stage, we start N_0 independent chains from the initial state x_0 and simulate each of them until time $\min(\tau_A, T_1)$. Let R_1 be the number of those chains for which D_1 occurs. Then $\hat{p}_1 = R_1/N_0$ is an obvious unbiased estimator of p_1. The empirical distribution $\hat{G}_1$ of these R_1 entrance states X_{T_1} can be viewed as an estimate of the conditional distribution G_1.

In stage k, for $k \ge 2$, ideally we would like to generate N_{k-1} states independently from the entrance distribution G_{k-1}. Or even better, to generate a stratified sample from G_{k-1}. But we usually cannot do that, because G_{k-1} is unknown. Instead, we pick N_{k-1} states out of the R_{k-1} that are available (by cloning if necessary), simulate independently from these states up to time $\min(\tau_A, T_k)$, and estimate p_k by $\hat{p}_k = R_k/N_{k-1}$ where R_k is the number of chains for which D_k occurs. The initial state of each of the N_{k-1} chains at the beginning of stage k has distribution G_{k-1}. Thus, for each of these chains, the event D_k has probability p_k and the entrance state at the next level if D_k occurs has distribution G_k.

Even though the $\hat{p}_k$'s are not independent, we can prove by induction on k that the product $\hat{p}_1 \cdots \hat{p}_m = (R_1/N_0)(R_2/N_1)\cdots(R_m/N_{m-1})$ is an unbiased estimator of γ (Garvels 2000, page 17): If we assume that $\mathbb{E}[\hat{p}_1 \cdots \hat{p}_{k-1}] = p_1 \cdots p_{k-1}$, then

$$\begin{aligned}
\mathbb{E}[\hat{p}_1 \cdots \hat{p}_k] &= \mathbb{E}[\hat{p}_1 \cdots \hat{p}_{k-1} \mathbb{E}[\hat{p}_k \mid N_0, R_1, \ldots, N_{k-1}]] \\
&= \mathbb{E}[\hat{p}_1 \cdots \hat{p}_{k-1} (N_{k-1} p_k)/N_{k-1}] \\
&= p_1 \cdots p_k.
\end{aligned}$$

Combining this with the fact that $\mathbb{E}[\hat{p}_1] = p_1$, the result follows.

2.2 Fixed Splitting vs Fixed Effort

There are many ways of doing the splitting (Garvels 2000). For example, we may clone each of the R_k chains that reached level k in c_k copies, for a fixed positive integer c_k. Then, each $N_k = c_k R_k$ is random. This is *fixed splitting*. If we want the *expected* number of splits of each chain to be

c_k, where $c_k = \lfloor c_k \rfloor + \delta$ and $0 \le \delta < 1$, then we assume that the actual number of splits is $\lfloor c_k \rfloor + 1$ with probability δ and $\lfloor c_k \rfloor$ with probability $1-\delta$.

In the *fixed effort* method, we fix each N_k a priori and make just the right amount of splitting to reach this target value. This can be achieved by *random assignment*: draw the N_k starting states at random, with replacement, from the R_k available states. This is equivalent to sampling N_k states from the empirical distribution $\hat{G}_k$ of these R_k states. In a *fixed assignment*, on the other hand, we split each of the R_k states approximately the same number of times as follows. Let $c_k = \lfloor N_k/R_k \rfloor$ and $d_k = N_k \bmod R_k$. Select d_k of the R_k states at random, without replacement. Each selected state is split c_k+1 times and the other states are split c_k times. The fixed assignment gives a smaller variance than the random assignment because it corresponds to stratification over the empirical distribution $\hat{G}_k$ at level k.

These variants are all unbiased, but they differ in terms of variance. Garvels and Kroese (1998) conclude from their analysis and empirical experiments that fixed effort performs better, mainly because it reduces the variance of the number of chains that are simulated at each stage. It turns out that with optimal splitting factors, this is not always true (see the next subsection).

The fixed effort implementation with random assignment fits the framework of interacting particle systems studied by Del Moral (2004) to approximate Feynman-Kac distributions. In this type of system, particles that did not reach the threshold are killed and replaced by clones of randomly selected particles among those that have succeeded. This redistributes the effort on most promising particles while keeping the total number constant. Cérou, LeGland, Del Moral, and Lezaud (2005) derive limit theorems for the corresponding estimators.

2.3 Variance Analysis for a Simplified Setting

We outline a very crude variance analysis in an idealized fixed-effort setting where

$$N_0 = N_1 = \cdots = N_{m-1} = n$$

and where the $\hat{p}_i$'s are independent binomial random variables with parameters n and $p = \gamma^{1/m}$. Then, for $m > 1$, we have (Garvels 2000, L'Ecuyer, Demers, and Tuffin 2006):

$$\begin{aligned}
&\mathrm{Var}[\hat{p}_1 \cdots \hat{p}_m] \\
&= \prod_{i=1}^{m} \mathbb{E}[\hat{p}_i^2] - \gamma^2 \\
&= \left(p^2 + \frac{p(1-p)}{n}\right)^m - p^{2m} \\
&= \frac{mp^{2m-1}(1-p)}{n} + \frac{m(m-1)p^{2m-2}(1-p)^2}{2n^2} \\
&\quad + \cdots + \frac{(p(1-p))^m}{n^m}.
\end{aligned}$$

If we assume that

$$n \gg (m-1)(1-p)/p, \tag{2}$$

the first term $mp^{2m-1}(1-p)/n \approx m\gamma^{2-1/m}/n$ dominates in the last expression. The standard Monte Carlo variance, on the other hand, is $\gamma(1-\gamma)/n \approx \gamma/n$. To illustrate the huge potential variance reduction, suppose $\gamma = 10^{-20}$, $m = 20$, $p = 1/10$, and $n = 1000$. Then the MC variance is 10^{-23} whereas $mp^{2m-1}(1-p)/n \approx 1.8 \times 10^{-41}$. This oversimplified setting is not realistic, because the $\hat{p}_i$ are generally not independent and it is difficult to have $p_i = \gamma^{1/m}$ for all i, but it gives an idea of the order of magnitude of potential variance reduction.

The amount of work (or CPU time, or number of steps simulated) at each stage is proportional to n, so the total work is proportional to nm. Most of this work is to simulate the n chains down to level 0 at each stage. Thus, the efficiency of the splitting estimator under the simplified setting is approximately proportional to $n/[\gamma^{2-1/m} nm^2] = \gamma^{-2+1/m}/m^2$ when (2) holds. By differentiating with respect to m, we find that this expression is maximized by taking $m = -\ln(\gamma)/2$ (we neglect the fact that m must be an integer). This gives $p^m = \gamma = e^{-2m}$, so $p = e^{-2}$. Garvels and Kroese (1998) have obtained this result. The squared relative error in this case is (approximately) $\gamma^{2-1/m}(m/n)\gamma^{-2} = e^2 m/n = -e^2 \ln(\gamma)/(2n)$ and the relative efficiency is proportional to $\gamma^2 \gamma^{-2+1/m}/m^2 = (em)^{-2} = [(e/2)\ln(\gamma)]^{-2}$, again under the condition (2).

When $\gamma \to 0$ for fixed p, we have $m \to \infty$, so (2) does not hold. Then, the relative error increases toward infinity and the relative efficiency converges to zero, at a logarithmic rate in both cases. This agrees with Garvels (2000), page 20. With $\tilde{\gamma}_n = \hat{p}_1 \cdots \hat{p}_m$, the limit in (1) is

$$\begin{aligned}
&\lim_{\gamma \to 0^+} \frac{\log(p^2 + p(1-p)/n)^m}{\log \gamma} \\
&= \lim_{\gamma \to 0^+} \frac{-\log(p^2 + p(1-p)/n)}{-\log p} < 2.
\end{aligned}$$

Thus, this splitting estimator is not quite asymptotically efficient, but almost (when n is very large).

Consider now a fixed-splitting setting, assuming that $N_0 = n$, $p_k = p = \gamma^{1/m}$ for all k, and that the constant splitting factor at each stage is $c = 1/p$; i.e., $N_k = R_k/p$. Then, $\{N_k, k \ge 1\}$ is a *branching process* and the estimator becomes

$$\hat{p}_1 \cdots \hat{p}_m = \frac{R_1}{N_0}\frac{R_2}{N_1} \cdots \frac{R_m}{N_{m-1}} = \frac{R_m p^{m-1}}{n}.$$

From standard branching process theory (Harris 1963), we have that

$$\mathrm{Var}[\hat{p}_1 \cdots \hat{p}_m] = m(1-p)p^{2m-1}/n.$$

If p is fixed and $m \to \infty$, then the squared relative error $m(1-p)/(np)$ is unbounded here as well. However, the limit in (1) becomes

$$\lim_{\gamma \to 0^+} \frac{\log(m(1-p)\gamma^2/(np)+\gamma^2)}{\log \gamma} = \lim_{\gamma \to 0^+} \frac{-2m\log p - \log(1+m(1-p)/(np))}{-m\log p} = 2,$$

so the splitting estimator is asymptotically efficient (Glasserman et al. 1999). This implies that fixed splitting is asymptotically better in this case.

Glasserman et al. (1999) study the *fixed splitting* framework with splitting factor $c_k \equiv c$, for a countable-state space Markov chain. They assume that the probability transition matrix $\mathbf{P}_k$ for the first-entrance state at level k given the first-entrance state at level $k-1$ converges to a matrix $\mathbf{P}$ with spectral radius $\rho < 1$. This implies that $p_k \to \rho$ when $k \to \infty$. Then they use branching process theory to prove that the multilevel splitting estimator (in their setting) is work-normalized asymptotically efficient if and only if $c = 1/\rho$. Glasserman et al. (1998) show that the condition $c = 1/\rho$ is not sufficient for asymptotic efficiency and provide additional necessary conditions in a general multidimensional setting. Their results highlight the crucial importance of choosing a good importance function h.

Even though fixed splitting is asymptotically better under ideal conditions, its efficiency is extremely sensitive to the choice of splitting factors. If the splitting factors are too high, the number of chains (and the amount of work) explodes, whereas if they are too low, the variance is very large because very few chains reach B. Since the optimal splitting factors are unknown in real-life applications, the more robust fixed-effort approach is usually preferable.

2.4 Implementation

The fixed-effort approach has the disadvantage of requiring more memory than fixed splitting, because it must use a *breadth-first* implementation: at each stage k all the chains must be simulated until they reach either A or level ℓ_k before we know the splitting factor at that level. The states of all the chains that reach ℓ_k must be saved; this may require too much memory when the N_k's are large. With fixed splitting, we can adopt a *depth-first* strategy, where each chain is simulated entirely until it hits ℓ or A, then its most recent clones (created at the highest level that it has reached) are simulated entirely, then those at the next highest level, and so on. This procedure is applied recursively. At most one state per level need to be memorized with this approach. This is feasible because the amount of splitting at each level is fixed a priori.

As a second issue, an important part of the work in multilevel splitting is due to the fact that all the chains considered in stage k (from level ℓ_{k-1}) and which do not reach ℓ_k must be simulated until they get down to A. When ℓ_{k-1} is large, this can take significant time. Because of this, the expected amount of work increases with the number of thresholds. One heuristic that reduces this work in exchange for a small bias truncates the chains that reach level $\ell_{k-\beta}$ downward after they have reached ℓ_{k-1}, where $\beta \geq 2$ is a fixed integer large enough so that a chain starting at level $\ell_{k-\beta}$ has a very small probability of getting back up to ℓ_k. We discuss unbiased alternatives in Section 2.7.

2.5 The RESTART Algorithm

The *RESTART* method (Villén-Altamirano and Villén-Altamirano 1994, Villén-Altamirano and Villén-Altamirano 2006) is a variant of splitting where any chain is split by a fixed factor when it hits a level upward, and one of the copies is tagged as the *original* for that level. When any of those copies hits that same level downward, if it is the original it just continues its path, otherwise it is killed immediately. This rule applies recursively, and the method is implemented in a depth-first fashion, as follows: whenever there is a split, all the non-original copies are simulated completely, one after the other, then simulation continues for the original chain. Unbiasedness is proved by Garvels (2000) and Villén-Altamirano and Villén-Altamirano (2002). The reason for killing most of the paths that go downward is to reduce the work. The number of paths that are simulated down to A never exceeds N_0. On the other hand, the number of chains that reach a given level is more variable with this method than with the fixed-effort and fixed-assignment multilevel splitting algorithm described previously. As a result, the final estimator of γ has a larger variance (Garvels 2000). Another source of additional variance is that the resplits tend to share a longer common history and to be more positively correlated. This source of variance can be important when the probability of reaching B from a given level varies significantly with the entrance state at that level (Garvels 2000). In terms of overall efficiency, none of the two methods is universally better; RESTART wins in some cases and splitting wins in other cases. Villén-Altamirano and Villén-Altamirano (2002) provide a detailed variance analysis of RESTART.

2.6 Choice of the Importance Function and Optimal Parameters

Key issues in multilevel splitting are the choices of the importance function h, levels ℓ_k, and splitting factors. To discuss this, we introduce some more notation. Let $\mathscr{X}_k \subset \mathscr{X}$ be the support of the entrance distribution G_k, i.e., the states in which the chain can possibly be when hitting level ℓ_k for the first time. Let $\gamma(x) = \mathbb{P}[\tau_B < \tau_A \mid \tau > j, X_j = x]$, the probability of reaching B before A if the chain is currently in state x, and $p_k(x) = \mathbb{P}[D_k \mid D_{k-1}, X_{T_{k-1}} = x]$, the probability of reaching level k before hitting A if the chain has

just entered level $k-1$ in state x, for $x \in \mathscr{X}_{k-1}$. Note that $p_k = \int_{x\in\mathscr{X}_{k-1}} p_k(x)dG_{k-1}(x)$ and $\gamma = \gamma(x_0)$.

One-dimensional case: Selecting the levels. If the Markov chain has a *one-dimensional state space* $\mathscr{X} \subset \mathbb{R}$, $\gamma(x)$ is increasing in x, and if $A = (-\infty, 0]$ and $B = [\ell, \infty)$ for some constant ℓ, then we could simply choose $h(x) = x$ (or any strictly increasing function). In this case, the kth level is attained when the *state* reaches the value ℓ_k. This value need not be reached exactly: in general, the chain can jump directly from a smaller value to a value larger than ℓ_k, perhaps even larger than ℓ_{k+1}. So even in the one-dimensional case, the entrance state x at a given level is not unique in general and the probability $p_k(x)$ of reaching the next level depends on this (random) entrance state. It remains to choose the levels ℓ_k.

We saw earlier that in a fixed effort setting and under simplifying assumptions, it is optimal to have $p_k \equiv p = e^{-2}$ for all k. This gives $m = -\ln(\gamma)/2$ levels. To obtain equal p_k's, it is typically necessary to take unequal distances between the successive levels ℓ_k, i.e., $\ell_k - \ell_{k-1}$ must depend on k.

Suppose now that we use fixed splitting with $c_k = 1/p_k = e^2$ for each k. If we assume (crudely) that each chain is split by a factor of e^2 at each stage, the total number of copies of a single initial chain that have a chance to reach B is

$$e^{2m-2} = e^{-\ln(\gamma)-2} = e^{-2}\gamma^{-1}. \tag{3}$$

Since each one reaches B with probability γ, this crude argument indicates that the expected number of chains that reach B is approximately equal to $p = e^{-2}$ times the initial number of chains at stage 0, exactly as in the fixed-effort case. However, the *variance* generally differs.

For RESTART, Villén-Altamirano and Villén-Altamirano (1994) concluded from a crude analysis that $p_k \approx e^{-2}$ was approximately optimal. However, their more careful analysis in Villén-Altamirano and Villén-Altamirano (2002) indicates that the p_k's should be as small as possible. Since the splitting factor at each level must be an integer, they recommend $p_k = 1/2$ and a splitting factor of $c_k = 2$.

Cérou and Guyader (2005) determine the thresholds adaptively for the splitting with fixed effort in dimension 1. They first simulate n chains (trajectories) until these chains reach A or B. Then they sort the chains according to the maximum value of the importance function h that each chain has reached. The k trajectories with the largest values are kept, while the $n-k$ others are re-simulated, starting from the state at which the highest value of the importance function was obtained for the $(n-k)$-th largest one. They proceed like this until $n-k$ trajectories have reached B. Their estimator is proved to be consistent, but is biased.

Multidimensional case: Defining the importance function. In the case of a multidimensional state space, the choice of h is much more difficult. Note that h and the ℓ_k's jointly determine the probabilities $p_k(x)$ and p_k. Based on large deviation theory, Glasserman et al. (1998) shows that the levels need to be chosen in a way consistent with the most likely path to a rare set. Garvels, Kroese, and Van Ommeren (2002) show by induction on k that for any fixed $p_1, \ldots, p_m$, h should be defined so that $p_k(x) = p_k$ (independent of x) for all $x \in \mathscr{X}_{k-1}$ and all k. This rule minimizes the residual variance of the estimator from stage k onward. With an h that satisfies this condition, the optimal levels and splitting factors are the same as in the one-dimensional case: $m = -(1/2)\ln\gamma$ levels, $p_k \approx e^{-2}$ and $\mathbb{E}[N_k] = N_0$ for each k. A simple choice of h and ℓ_k's that satisfies these conditions is

$$h(x) = h^*(x) \stackrel{\text{def}}{=} \gamma(x) \quad \text{and} \quad \ell_k = e^{-2(m-k)} = \gamma e^{2k}.$$

Garvels, Kroese, and Van Ommeren (2002) gave the following (equivalent) alternative choice: $\ell_k = k$ for each k and

$$h(x) \quad = \quad h^{**}(x) \stackrel{\text{def}}{=} \frac{\ln(\gamma(x)/\gamma)}{2} = m + \frac{\ln(\gamma(x))}{2}$$

for all $x \in \mathscr{X}$. However, these levels are optimal only if we assume that the chain can reach ℓ_k only on the set $\{x : \gamma(x) = e^{-2(m-k)}\}$, an optimistic assumption that rarely holds in practice, especially in the multidimensional case.

Garvels, Kroese, and Van Ommeren (2002) also show how to get a first estimate of $\gamma(x)$ beforehand, in simple situations where the Markov chain has a finite state space, by simulating the chain backward in time. They construct an approximation of h^{**} from this estimate and then use it in their splitting algorithm. They apply their method to a tandem queue with two or three nodes and obtain good results. However, this method appears to have limited applicability for large and complicated models.

Booth and Hendricks (1984) propose adaptive methods that *learn* the importance function as follows. In their setting, the state space is partitioned in a finite number of regions and the importance function h is assumed to be constant in each region. This importance function is used to determine the expected splitting factors and Russian roulette probabilities (see Section 2.8) when a chain jumps from one region to another. They estimate the "average" value of $\gamma(x)$ in region j by the fraction of chains that reach B among those that have entered region j. These estimates are taken as importance functions in further simulations used to improve the estimates, and so on.

Constructing the functions h^* or h^{**} essentially requires the knowledge of the probability $\gamma(x)$ for all x. But if we knew these probabilities, there would be no need for simulation! This is very similar (and related) to the issue of constructing the optimal change of measure in importance sampling (Glasserman et al. 1998). In general, finding an optimal h, or an h for which $p_k(x)$ is independent of x, can

be extremely difficult or even impossible. When $p_k(x)$ depends on x, selecting the thresholds so that $p_k \approx e^{-2}$ is not necessarily optimal. More importantly, with a bad choice of h, splitting may *increase* the variance, as illustrated by the next example.

Example 1 This example was used by Parekh and Walrand (1989), Glasserman et al. (1998), Glasserman et al. (1999), and Garvels (2000), among others. Consider an open tandem Jackson network with two queues, arrival rate 1, and service rate μ_j at queue j for $j = 1,2$. Let $X_j = (X_{1,j}, X_{2,j})$ denote the number of customers at each of the two queues immediately after the jth event (arrival or end of service). We have $A = \{(0,0)\}$ and $B = \{(x_1,x_2) : x_2 \geq \ell^*\}$ for some large integer ℓ^*. A naive choice of importance function here would be $h(x_1,x_2) = x_2$. This seems natural at first sight because the set B is defined in terms of x_2 only. With this choice, the entrance distribution at level k turns out to be concentrated on pairs (x_1,x_2) with small values of x_1. To see why, suppose that $x_2 = \ell_{k'} > 0$ for some integer k' and that we are in state (x_1,x_2-1) where $x_1 > 0$ is small. The possible transitions are to states (x_1+1,x_2-1), (x_1,x_2-2), and (x_1-1,x_2), with probabilities proportional to 1, μ_2, and μ_1, respectively. But the chains that go to state (x_1-1,x_2) are cloned whereas the other ones are not, and this tends to increase the population of chains with a small x_1.

Suppose now that $\mu_1 < \mu_2$ (the first queue is the bottleneck). In this case, the most likely paths to overflow are those where the first queue builds up to a large level and then the second queue builds up from the transfer of customers from the first queue (Heidelberger 1995). The importance function $h(x_1,x_2) = x_2$ does not favor these types of paths; it rather favors the paths where x_1 remains small and these paths have a very high likelihood of returning to $(0,0)$ before overflow. As a result, splitting with this h may give an even larger variance than no splitting at all. For this particular example, h^{**} increases in both x_1 and x_2 (Garvels, Kroese, and Van Ommeren 2002).

2.7 Unbiased Truncation

We pointed out earlier that a large fraction of the work in multilevel splitting is to simulate the chains down to level zero at each stage. Truncating the chains whenever they fall below some level $\ell_{k-\beta}$ in stage k reduces the work but introduces a bias. A large β may give negligible bias, but also a small work reduction. In what follows, we describe unbiased truncation techniques based on the *Russian roulette* principle (Kahn and Harris 1951, Hammersley and Handscomb 1964).

Probabilistic truncation. The idea here is to kill the chains at random, with some probability, independently of each other. The survivors act as *representatives* of the killed chains. For stage k, we select real numbers $r_{k,2},\ldots,r_{k,k-1}$ in $[1,\infty)$. The first time a chain reaches level ℓ_{k-j} from above during that stage, for $j \geq 2$, it is killed with probability $1 - 1/r_{k,j}$. If it survives, its *weight* is multiplied by $r_{k,j}$. (This is a version of Russian roulette.) When a chain of weight $w > 1$ reaches level ℓ_k, it is cloned into $\lfloor w \rfloor$ additional copies with probability $\delta = w - \lfloor w \rfloor$ and $\lfloor w-1 \rfloor$ additional copies with probability $1-\delta$. Each copy is given weight 1. Now, the number of representatives retained at any given stage is random. Note that we may have $r_{k,j} = 1$ for some values of j.

Periodic truncation. To reduce the variability of the number of selected representatives at each level ℓ_{k-j}, we may decide to retain every $r_{k,j}$-th chain that down-crosses that level and multiply its weight by $r_{k,j}$; e.g., if $r_{k,j} = 3$, we keep the third, sixth, ninth, etc. This would generally give a biased estimator, because the probability that a chain is killed would then depend on its sample path up to the time when it crosses the level (for instance, the first chain that down-crosses the level would always be killed if $r_{k,j} > 1$). As simple trick to remove that bias is to modify the method as follows: generate a random integer $D_{k,j}$ uniformly in $\{1,\ldots,r_{k,j}\}$, retain the $(i\,r_{k,j} + D_{k,j})$-th chain that down-crosses level ℓ_{k-j} for $i = 0,1,2,\ldots$, and kill the other ones. We assume that the random variables $D_{k,2},\ldots,D_{k,k-1}$ are independent. Then, any chain that down-crosses the level has the same probability $1 - 1/r_{k,j}$ of being killed, independently of its trajectory above that level. This is true for any positive integer $r_{k,j}$. Moreover, the proportion of chains that survive has less variance than for the probabilistic truncation (the killing indicators are no longer independent across the chains). The chains that reach ℓ_k are cloned in proportion to their weight, exactly as in the probabilistic truncation.

Tag-based truncation. In the periodic truncation method, the level at which a chain is killed is determined only when the chain reaches that level. An alternative is to fix all these levels right at the beginning of the stage. We first select positive integers $r_{k,2},\ldots,r_{k,k-1}$. Then each chain is *tagged* to the level ℓ_{k-j} with probability $q_{k,j} = (r_{k,j}-1)/(r_{k,2}\cdots r_{k,j})$ for $j = 2,\ldots,k-1$, and to level ℓ_0 with probability $1 - q_{k,k-1} - \cdots - q_{k,2} = 1/(r_{k,2}\cdots r_{k,k-1})$. Thus, all the chains have the same probability of receiving any given level and the probability of receiving level zero is positive. If the tags are assigned randomly and independently across the chains, then this method is equivalent to probabilistic truncation. But if the integers $r_{k,2},\ldots,r_{k,k-1}$ are chosen so that their product divides (or equals) N_k, the number of chains at the beginning of stage k, then the tags can also be assigned so that the proportion of chains tagged to level ℓ_{k-j} is *exactly* $q_{k,j}$, while the probability of receiving a given tag is the same for all chains. The reader can verify that the following scheme gives one way of achieving this: Put the N_k

chains in a list (in any order), generate a random integer D uniformly in $\{0,\ldots,N_k-1\}$, and assign the tag $k-j^*(i,D)$ to the i-th chain in the list, for all i, where $j^*(i,D)$ is the smallest integer j in $\{2,\ldots,k\}$ such that $r_{k,2}\cdots r_{k,j}$ *does not* divides $(D+i)$ mod N_k (when $(D+i)$ mod $N_k=0$, we put $j^*(i,D)=k$). . After the tags are assigned, the chains can be simulated one by one for that stage. Whenever a chain down-crosses for the first time (in this stage) a level ℓ_{k-j} higher than its tag, its weight is multiplied by $r_{k,j}$. If it down-crosses the level of its tag, it is killed immediately. The chains that reach ℓ_k are cloned in proportion to their weight, as before.

Unbiasedness. L'Ecuyer, Demers, and Tuffin (2006) show that all the above truncation methods are unbiased by proving the next proposition and then showing that each truncation method satisfies the assumptions of the proposition.

Proposition 1 *Suppose there are real numbers $r_{k,2},\ldots,r_{k,k-1}$ in $[1,\infty)$ such that for $j=2,\ldots,k-1$, each chain has a probability $1-1/r_{k,j}$ of being killed at its first down-crossing of level ℓ_{k-j}, independently of its sample path up to that moment, and its weight is multiplied by $r_{k,j}$ if it survives. Then the truncated estimator remains unbiased.*

Getting rid of the weights. In the unbiased truncation methods discussed so far, the surviving chains have different weights. The variance of these weights may contribute significantly to the variance of the final estimator. For example, if k is large, the event that a chain reaches ℓ_k (from ℓ_{k-1}) after going down to ℓ_1 is usually a rare event, and when it occurs the corresponding chain has a large weight, so this may have a non-negligible impact on the variance. This can be addressed by resplitting the chains within the stage when they up-cross some levels, instead of increasing their weights at down-crossings. We explain how the probabilistic and tag-based truncation methods can be modified to incorporate this idea. In these methods, the weights of all chains are always 1, and whenever a chain down-crosses ℓ_{k-j} (not only the first time), for $j\geq 2$, it can get killed.

Probabilistic truncation and resplitting within each stage. The probabilistic truncation method can be modified as follows. During stage k, whenever a chain reaches a level ℓ_{k-j} from below, it is split in $r_{k,j}$ identical copies that start evolving independently from that point onward (if $r_{k,j}$ is not an integer, we split the chain in $\lfloor r_{k,j}+1\rfloor$ copies with probability $\delta=r_{k,j}-\lfloor r_{k,j}\rfloor$ and in $\lfloor r_{k,j}\rfloor$ copies with probability $1-\delta$). Whenever a chain down-crosses ℓ_{k-j} (not only the first time), for $j\geq 2$, it is killed with probability $1-1/r_{k,j}$. All chains always have weight 1.

Tag-based truncation with resplits. This method is equivalent to applying RESTART separately within each stage of the multistage splitting algorithm. It modifies the tag-based truncation as follows: Whenever a chain up-crosses level ℓ_{k-j} for $j\geq 2$, it is split in $r_{k,j}$ copies. One of these $r_{k,j}$ copies is identified as the original and keeps its current tag, while the other $r_{k,j}-1$ copies are tagged to the level ℓ_{k-j} where the split occurs. As before, a chain is killed when it down-crosses the level of its tag.

Unbiasedness. L'Ecuyer, Demers, and Tuffin (2006) prove the following proposition and show that the two truncation methods with resplits that we just described satisfy its assumptions.

Proposition 2 *Suppose there are positive real numbers $r_{k,2},\ldots,r_{k,k-1}$ such that for $j=2,\ldots,k-1$, each chain is killed with probability $1-1/r_{k,j}$ whenever it down-crosses level ℓ_{k-j}, independently of its sample path up to the time when it reached that level, and that this chain is split into C chains when it up-crosses that same level, where C is a random variable with mean $r_{k,j}$, independent of the history so far. Then the estimator with probabilistic truncation and resplits (without weights) is unbiased for γ.*

Effectiveness and Implementation. The resplit versions of the truncation methods are expected to give a smaller variance but require more work. So there is no universal winner if we think of maximizing the efficiency. One disadvantage of the resplit versions is that the *number* of chains alive at any given time during stage k has more variance and *may* exceed N_{k-1}. In a *worst-case* situation, a chain may go down and up many times across several levels without being killed, giving rise to a flurry of siblings along the way. Fortunately, this type of bad behavior has an extremely small probability and poses no problem when the splitting parameters are well chosen. In all our experiments, the number of chains alive simultaneously during any given stage k has rarely exceeded N_{k-1}. If we want to insist that the number of chains never exceeds N_{k-1}, we can use weights instead of splitting, but just for the splits that would have made the number of chains too large. We may want to do that if the chains are stored in an array of size $n=N_{k-1}$ and we do not want their number to exceed n. This type of implementation is needed when we combine splitting with the array-RQMC method (L'Ecuyer, Demers, and Tuffin 2006).

We have a lot of freedom for the choice of the truncation and resplit parameters $r_{k,j}$. We can select different sets of values at the different stages of the multilevel splitting algorithm. It appears sensible to take $r_{k,j}=1/\hat{p}_{k-j}=N_{k-j-1}/R_{k-j}$, the actual splitting factor used at level ℓ_{k-j} of the splitting algorithm, for $j\geq 2$. In our experiments, this has always worked well.

2.8 Getting Rid of the Levels

In some versions of the *splitting and Russian roulette* technique, there are no levels (or thresholds), but only an importance function (some authors call it *branching function*). For instance, Ermakov and Melas (1995) and Melas (1997) study a general setting where a chain can be split or killed at any transition. If the transition is from x to y and if $\alpha = h(y)/h(x) \geq 1$, then the chain is split in a random number C of copies where $\mathbb{E}[C] = \alpha$, whereas if $\alpha < 1$ it is killed with probability $1-\alpha$ (this is Russian roulette). In case of a split, the $C-1$ new copies are started from state x and new transitions are generated (independently) for those chains. Their method is developed to estimate the average cost per unit of time in a regenerative process, where a state-dependent cost is incurred at each step. In the simulation, each cost incurred in a given state x is divided by $h(x)$. We may view $1/h(x)$ as the *weight* of the chain at that point. At the end of a regenerative cycle, the total weighted cost accumulated by the chain over its cycle is the *observation* associated with this cycle. The expected cost per cycle is estimated by averaging the observations over all simulated cycles. The expected length of a cycle is estimated in the same way, just replacing costs by lengths. The authors show that their method is consistent and propose an adaptive algorithm that estimates the optimal h.

This method can be applied to a finite-horizon simulation as well. In our setting, it suffices to replace the regeneration time by the time when the chain reaches A or B, and forget about the length of the cycle. When a chain reaches B, it contributes its weight $1/h(X_{\tau_B})$ to the estimator. For a very crude analysis, suppose we take $h(x) = \gamma(x)$ and that there is a split in two every time the function h doubles its value. Here, $h(y)/h(x) = \gamma(y)/\gamma(x)$, so a chain that reaches the set B would have split in two approximately $-\log_2 \gamma$ times. This gives a "potential" of $2^{-\log_2 \gamma} = 1/\gamma$ copies that can possibly reach B for each initial chain at level 0, the same number as for the multilevel splitting and RESTART; see Equation (3). This argument suggests that an optimal h in this case should be proportional to $\gamma(x)$.

In general, splitting and Russian roulette can be implemented by maintaining a weight for each chain. Initially, each chain has weight 1. Whenever a chain of weight w is split in C copies, the weight of all the copies is set to either w/C or $w/\mathbb{E}[C]$. Booth (1985) shows that using $w/\mathbb{E}[C]$ is usually better. When Russian roulette is applied, the chain is killed with some probability $\alpha < 1$; if it survives, its weight is multiplied by $1/(1-\alpha)$. The values of C and α at each step can be deterministic or random, and may depend on the past history of the chain. Whenever a cost is incurred, it must be multiplied by the weight of the chain. Unbiasedness for this general setting is proved (under mild conditions) by Booth and Pederson (1992), for example.

2.9 Weight Windows

Particle transport simulations in nuclear physics often combine splitting and Russian roulette with importance sampling. Then, the weight of each chain must be multiplied by the *likelihood ratio* accumulated so far. The *weight* is redefined as this product. In the context of rare events, it is frequently the case that the final weight of a chain is occasionally large and usually very small. This gives rise to a large variance and a highly-skewed distribution, for which variance estimation is difficult.

To reduce the variance of the weights, Booth (1982) introduced the idea of *weight windows*, which we define as follows (see also Booth and Hendricks (1984) and Fox (1999)). Define the *weighted importance* of a chain as the product of its weight w and the value of the importance function $h(x)$ at its current state. Select three real numbers $0 < a_{\min} < a < a_{\max}$. Whenever the weighted importance $\omega = w h(x)$ of a chain falls below $a_{\min}$, we apply Russian roulette, killing the chain with probability $1-\omega/a$. If the chain survives, its weight is set to $a/h(x)$. If the weighted importance ω rises above $a_{\max}$, we split the chain in $c = \lceil \omega/a_{\max} \rceil$ copies and give weight w/c to each copy. The estimator of $\gamma = \mathbb{P}[\tau_B < \tau_A]$ is the sum of weights of all the chains that reach the set B before reaching A. The importance function $h^*(x) = \gamma(x)$ should be approximately optimal in this case. The basic motivation is simple: if the weight window is reasonably narrow, all the chains that reach B would have approximately the same weight, so the only significant source of variance would be the *number* of chains that reach B (Booth and Hendricks 1984). If we take $a = (a_{\min} + a_{\max})/2 \approx \gamma$, then this number has expectation n (approximately), where n is the initial number of chains.

In the original proposal of Booth (1982) and Booth and Hendricks (1984), the windows are on the weights, not on the weighted importance. The state space is partitioned in a finite number of regions (say, up to 100 regions), the importance function is assumed constant in each region, and each region has a different weight window, inversely proportional to the value of the importance function in that region. Such weight windows are used extensively in the Los Alamos particle transport simulation programs. Our formulation is essentially equivalent, except that we do not assume a finite partition of the state space.

Fox (1999), Chapter 10) discusses the use of weight windows for splitting and Russian roulette, but does not mention the use of an importance function. Weight windows without an importance function could be fine when a good change of measure (importance sampling) is already applied to drive the system toward the set B. Then, the role of splitting and Russian roulette is only to "equalize" the contributions of the chains that reach B and kill most of those whose anticipated contribution is deemed negligible, to save work. This type of splitting, based only on weights and without an importance

function, gives no special encouragement to the chains that go toward B. If we use it alone, the event $\{\tau_B < \tau_A\}$ will remain a rare event.

If there is no importance sampling, the multilevel splitting techniques described earlier (except those with truncation and no resplits, in Section 2.7) have the advantage of not requiring explicit (random) weights. All the chains that reach level ℓ_k have the same weight when they reach that level for the first time. So there is no need for weight windows in that context.

3 EXAMPLES

Example 2 We return to Example 1, an open tandem Jackson queueing network with two queues. The choice of h is crucial for this example, especially if $\mu_1 < \mu_2$ (Glasserman et al. 1998). Here we look at a case where $\mu_1 > \mu_2$. We consider the following choices of h:

$$h_1(x_1,x_2) = x_2; \quad (4)$$

$$h_2(x_1,x_2) = (x_2 + \min(0, x_2 + x_1 - \ell))/2; \quad (5)$$

$$h_3(x_1,x_2) = x_2 + \min(x_1, \ell - x_2 - 1) \times (1 - x_2/\ell). \quad (6)$$

The function h_1 is a naive choice based on the idea that the set B is defined in terms of x_2 only. The second choice, h_2, counts ℓ minus half the minimal number of steps required to reach B from the current state. (To reach B, we need at least $\ell - \min(0, x_2 + x_1 - \ell)$ arrivals at the first queue and $\ell - x_2$ transfers to the second queue.) The third choice, h_3, is adapted from Villén-Altamirano (2006), who recommends $h(x_1,x_2) = x_2 + x_1$ when $\mu_1 > \mu_2$. This h was modified as follows. We define $h_3(x) = x_1 + x_2$ when $x_1 + x_2 \le \ell - 1$ and $h_3(x) = \ell$ when $x_2 \ge \ell$. In between, i.e., in the area where $\ell - x_1 - 1 \le x_2 \le \ell$, we interpolate linearly in x_2 for any fixed x_1. This gives h_3.

We did a numerical experiment with $\mu_1 = 4$, $\mu_2 = 2$, and $\ell = 30$, with our three choices of h. For each h and each truncation method discussed earlier, we computed the *variance per chain*, $V_n = n\mathrm{Var}[\hat{\gamma}_n]$, where n is the (expected) number of chains at each level, and the *work-normalized variance per chain*, $W_n = S_n \mathrm{Var}[\hat{\gamma}_n]$, where S_n is the expected total number of simulated steps of the n Markov chains. If S_n is seen as the computing cost of the estimator, then $1/W_n$ is the usual measure of *efficiency*. For fixed splitting without truncation and resplits, V_n and W_n do not depend on n.

Here we briefly summarize the detailed results given in L'Ecuyer, Demers, and Tuffin (2006). We have $V_n \approx \gamma \approx 1.3 \times 10^{-9}$ with standard Monte Carlo (no splitting) and $V_n \approx 1.1 \times 10^{-16}$ with the multilevel splitting with h_2, using fixed effort and no truncation. This is a huge variance reduction. With h_1, $\hat{V}_n$ and $\hat{W}_n$ were significantly higher than for h_2 and h_3, whereas h_3 was just a bit better than h_2. The truncation and resplit methods improved the efficiency roughly by a factor of 3. There is slightly more variance reduction with the variants that use resplits than with those that do not resplit, but also slightly more work, and the efficiency remains about the same.

Example 3 We consider an Ornstein-Uhlenbeck stochastic process $\{R(t), t \ge 0\}$, which obeys the stochastic differential equation

$$dR(t) = a(b - R(t))dt + \sigma dW(t)$$

where $a > 0$, b, and $\sigma > 0$ are constants, and $\{W(t), t \ge 0\}$ is a standard Brownian motion (Taylor and Karlin 1998). This is the Vasicek model for the evolution of short-term interest rates (Vasicek 1977). In that context, b can be viewed as a long-term interest rate level toward which the process is attracted with strength $a(b - R(t))$.

Suppose the process is observed at times $t_j = j\delta$ for $j = 0, 1, \ldots$ and let $X_j = R(t_j)$. Let $A = (-\infty, b]$, $B = [\ell, \infty)$ for some constant ℓ, and $x_0 \ge b$. We want to estimate the probability that the process exceeds level ℓ at one of the observation times before it returns below b, when started from $R(0) = x_0$. Here we take $b = 0$.

Suppose we take the importance function h equal to the identity. The thresholds ℓ_k should be placed closer to each other as k increases, because the attraction toward $b = 0$ becomes stronger. Preliminary empirical experiments suggest the following rule, which makes the p_k's approximately independent of k: set tentatively $\ell_k = \ell\sqrt{k/m}$ for $k = 1, \ldots, m$, let k^* be the largest k for which $\ell_k < 2$, and reset $\ell_k = \ell_{k^*}(k/k^*)$ for $k = 1, \ldots, k^* - 1$. The latter makes the first thresholds approximately equidistant.

Because of the time discretization, the entrance distribution G_k has positive support over the entire interval $[\ell_k, \infty)$. This means that a chain can cross an arbitrary number of thresholds in a single jump. The simulation starts from a fixed state only at the first level.

We made some experiments with $a = 0.1$, $b = 0$, $\sigma = 0.3$, $x_0 = 0.1$, $\delta = 0.1$, $\ell = 4$, and $m = 14$ levels. With these parameters, we have $V_n \approx \gamma \approx 1.6 \times 10^{-8}$ with standard Monte Carlo (no splitting) and $V_n \approx 1.0 \times 10^{-14}$ with the multilevel splitting without truncation, with either the fixed splitting or fixed effort approach. The truncation and resplit methods improve the work-normalized variance W_n roughly by a factor of 3, as in the previous example. The work is reduced by a factor of 4.3 without the resplits and by a factor of 3.5 with the resplits, but the variance is increased roughly by a factor of 1.4 without the resplits and 1.2 with the resplits.

The benefits of splitting and of truncation increase with ℓ. For $\ell = 6$, for example, we have $V_n \approx \gamma \approx 4.2 \times 10^{-18}$ with standard Monte Carlo and $V_n \approx 5.0 \times 10^{-33}$ with the multilevel splitting without truncation, with $m = 30$ (this gives p_k's of approximately the same size as with $\ell = 4$ and $m = 14$). In this case, the truncation and resplit methods reduce the work-normalized variance approximately by a factor of 8 to 10. Fixed effort and fixed splitting also have comparable efficiencies when no truncation is used.

Example 4 There are situations where the splitting method is not appropriate whereas importance sampling can be made very effective. Consider for example a highly-reliable Markovian multicomponent system (Shahabuddin 1994) for which the failure of a few components (e.g., 2 or 3) may be sufficient for the entire system to fail, and where all the components have a very small failure rate and a high repair rate. If we want to apply splitting, the thresholds must be defined in terms of the vector of failed components (the state of the system). But whenever there are failed components, the next event is a repair with a very high probability. So regardless of how we determine the thresholds, the probabilities p_k of reaching the next threshold from the current one are always very small. For this reason, the splitting method cannot be made efficient in this case. On the other hand, there are effective importance sampling methods for this type of model (Shahabuddin 1994, Cancela, Rubino, and Tuffin 2002).

4 CONCLUSION

Splitting is a valuable but seemingly under-exploited variance reduction technique for rare-event simulation. It certainly deserves further study. In multidimensional settings, finding out an appropriate importance function h can be a difficult task and seems to the the main bottleneck for an effective application of the method. Providing further hints in this direction, and developing adaptive techniques to learn good importance functions, would be of significant interest. Unfortunately, splitting can hardly be applied to problems where rarity comes from the occurrence of a low-probability transition that cannot be decomposed in several higher-probability transitions.

ACKNOWLEDGMENTS

This research has been supported by NSERC-Canada grant No. ODGP0110050 and a Canada Research Chair to the first author, an NSERC-Canada scholarship to the second author, and EuroNGI Network of Excellence, INRIA's cooperative research initiative RARE and "SurePath ACI Security" Project to the third author, and an FQRNT-INRIA Travel Grant to the first and third authors.

REFERENCES

Akin, O., and J. K. Townsend. 2001. Efficient simulation of TCP/IP networks characterized by non-rare events using DPR-based splitting. In *Proceedings of IEEE Globecom*, 1734–1740.

Blom, H. A. P., G. J. Bakker, J. Krystul, M. H. C. Everdij, B. K. Obbink, and M. B. Klompstra. 2005. Sequential Monte Carlo simulation of collision risk in free flight air traffic. Technical report, Project HYBRIDGE IST-2001-32460.

Booth, T. E. 1982. Automatic importance estimation in forward Monte Carlo calculations. *Transactions of the American Nuclear Society* 41:308–309.

Booth, T. E. 1985. Monte Carlo variance comparison for expected-value versus sampled splitting. *Nuclear Science and Engineering* 89:305–309.

Booth, T. E., and J. S. Hendricks. 1984. Importance estimation in forward Monte Carlo estimation. *Nuclear Technology/Fusion* 5:90–100.

Booth, T. E., and S. P. Pederson. 1992. Unbiased combinations of nonanalog Monte Carlo techniques and fair games. *Nuclear Science and Engineering* 110:254–261.

Bucklew, J. A. 2004. *Introduction to rare event simulation*. New York: Springer-Verlag.

Cancela, H., G. Rubino, and B. Tuffin. 2002. MTTF estimation by Monte Carlo methods using Markov models. *Monte Carlo Methods and Applications* 8 (4): 312–341.

Cérou, F., and A. Guyader. 2005, October. Adaptive multilevel splitting for rare event analysis. Technical Report 5710, INRIA.

Cérou, F., F. LeGland, P. Del Moral, and P. Lezaud. 2005. Limit theorems for the multilevel splitting algorithm in the simulation of rare events. In *Proceedings of the 2005 Winter Simulation Conference*, ed. F. B. A. M. E. Kuhl, N. M. Steiger and J. A. Joines, 682–691.

Del Moral, P. 2004. *Feynman-Kac formulae. genealogical and interacting particle systems with applications*. Probability and its Applications. New York: Springer.

Ermakov, S. M., and V. B. Melas. 1995. *Design and analysis of simulation experiments*. Dordrecht, The Netherlands: Kluwer Academic.

Fox, B. L. 1999. *Strategies for quasi-Monte Carlo*. Boston, MA: Kluwer Academic.

Garvels, M. J. J. 2000. *The splitting method in rare event simulation*. Ph. D. thesis, Faculty of mathematical Science, University of Twente, The Netherlands.

Garvels, M. J. J., and D. P. Kroese. 1998. A comparison of RESTART implementations. In *Proceedings of the 1998 Winter Simulation Conference*, 601–609: IEEE Press.

Garvels, M. J. J., D. P. Kroese, and J.-K. C. W. Van Ommeren. 2002. On the importance function in splitting simulation. *European Transactions on Telecommunications* 13 (4): 363–371.

Glasserman, P., P. Heidelberger, and P. Shahabuddin. 1999. Asymptotically optimal importance sampling and stratification for pricing path dependent options. *Mathematical Finance* 9 (2): 117–152.

Glasserman, P., P. Heidelberger, P. Shahabuddin, and T. Zajic. 1998. A large deviations perspective on the efficiency of multilevel splitting. *IEEE Transactions on Automatic Control* AC-43 (12): 1666–1679.

Glasserman, P., P. Heidelberger, P. Shahabuddin, and T. Zajic. 1999. Multilevel splitting for estimating rare event probabilities. *Operations Research* 47 (4): 585–600.

Glynn, P. W., and D. L. Iglehart. 1989. Importance sampling for stochastic simulations. *Management Science* 35:1367–1392.

Gorg, C., and O. Fuss. 1999. Simulating rare event details of atm delay time distributions with restart/lre. In *Proceedings of the IEE International Teletrafic Congress, ITC16*, 777–786: Elsevier.

Hammersley, J. M., and D. C. Handscomb. 1964. *Monte carlo methods*. London: Methuen.

Harris, T. 1963. *The theory of branching processes*. New York: Springer-Verlag.

Heidelberger, P. 1995. Fast simulation of rare events in queueing and reliability models. *ACM Transactions on Modeling and Computer Simulation* 5 (1): 43–85.

Kahn, H., and T. E. Harris. 1951. Estimation of particle transmission by random sampling. *National Bureau of Standards Applied Mathematical Series* 12:27–30.

L'Ecuyer, P., V. Demers, and B. Tuffin. 2006. Rare-events, splitting, and quasi-Monte Carlo. submitted.

L'Ecuyer, P., C. Lécot, and B. Tuffin. 2005. A randomized quasi-Monte Carlo simulation method for Markov chains. submitted.

L'Ecuyer, P., and C. Lemieux. 2000. Variance reduction via lattice rules. *Management Science* 46 (9): 1214–1235.

Melas, V. B. 1997. On the efficiency of the splitting and roulette approach for sensitivity analysis. In *Proceedings of the 1997 Winter Simulation Conference*, 269–274. Piscataway, NJ: IEEE Press.

Owen, A. B. 1998. Latin supercube sampling for very high-dimensional simulations. *ACM Transactions on Modeling and Computer Simulation* 8 (1): 71–102.

Parekh, S., and J. Walrand. 1989. A quick simulation method for excessive backlogs in networks of queues. *IEEE Transactions on Automatic Control* AC-34:54–56.

Pederson, S. P., R. A. Forster, and T. E. Booth. 1997. Confidence intervals for Monte Carlo transport simulation. *Nuclear Science and Engineering* 127:54–77.

Shahabuddin, P. 1994. Importance sampling for the simulation of highly reliable markovian systems. *Management Science* 40 (3): 333–352.

Spanier, J., and E. M. Gelbard. 1969. *Monte Carlo principles and neutron transport problems*. Reading, Massachusetts: Addison-Wesley.

Taylor, H. M., and S. Karlin. 1998. *An introduction to stochastic modeling*. third ed. San Diego: Academic Press.

Vasicek, O. 1977. An equilibrium characterization of the term structure. *Journal of Financial Economics* 5:177–188.

Villén-Altamirano, J. 2006. Rare event RESTART simulation of two-stage networks. manuscript.

Villén-Altamirano, M., and J. Villén-Altamirano. 1994. RESTART: A straightforward method for fast simulation of rare events. In *Proceedings of the 1994 Winter Simulation Conference*, 282–289: IEEE Press.

Villén-Altamirano, M., and J. Villén-Altamirano. 2002. Analysis of RESTART simulation: Theoretical basis and sensitivity study. *European Transactions on Telecommunications* 13 (4): 373–386.

Villén-Altamirano, M., and J. Villén-Altamirano. 2006. On the efficiency of RESTART for multidimensional systems. manuscript.

AUTHOR BIOGRAPHIES

PIERRE L'ECUYER is Professor in the Département d'Informatique et de Recherche Opérationnelle, at the Université de Montréal, Canada. He holds the Canada Research Chair in Stochastic Simulation and Optimization. His main research interests are random number generation, quasi-Monte Carlo methods, efficiency improvement via variance reduction, sensitivity analysis and optimization of discrete-event stochastic systems, and discrete-event simulation in general. He is currently Associate/Area Editor for *ACM TOMACS*, *ACM TOMS*, and *Statistical Computing*. He obtained the prestigious *E. W. R. Steacie* fellowship in 1995-97 and a *Killam* fellowship in 2001-03. His recent research articles are available on-line from his web page: <http://www.iro.umontreal.ca/∼lecuyer>.

VALÉRIE DEMERS is a PhD Student in mathematics at the Université de Montréal. Her main research areas are randomized quasi-Monte Carlo methods for Markov chains, rare-event simulation, and variance reduction methods in general. Her e-mail address is <demersv@IRO.UMontreal.CA>.

BRUNO TUFFIN received his PhD degree in applied mathematics from the University of Rennes 1 (France) in 1997. Since then, he has been with INRIA in Rennes. He spent 8 months as a postdoc at Duke University in 1999. His research interests include developing Monte Carlo and quasi-Monte Carlo simulation techniques for the performance evaluation of telecommunication systems, and developing new Internet-pricing schemes. His web page is <www.irisa.fr/armor/lesmembres/Tuffin/Tuffin_en.htm>.