Splitting techniques

Pierre L'Ecuyer, François Le Gland, Pascal Lezaud, Bruno Tuffin

September 7, 2008

Contents

Chapter 1

Splitting techniques

1.1 Introduction

As already explained in previous chapters, rare event simulation requires acceleration techniques to speed up the occurrence of the rare events under consideration, otherwise it may take unacceptably large sample sizes to get enough positive realizations, or even a single one, on average. On the other hand, accelerating too much can be counter-productive and even lead to a variance explosion and/or an increase in the computational time. Therefore, an appropriate balance must be achieved, and this is not always easy. This difficulty was highlighted in the previous chapter when discussing the *importance sampling* (IS) technique, whose idea is to change the probability laws driving the model in order to make the events of interest more likely, and to correct the bias by multiplying the estimator by the suitable likelihood ratio.

In this chapter, we review an alternative technique called *splitting*, to accelerate the occurrence rate of the rare events of interest. Here, we do not change the probability laws driving the model. Instead, we use a selection mechanism to favor the trajectories deemed likely to lead to those rare events. The main idea is to decompose the paths to the rare events of interest into shorter subpaths whose probability is not so small, encourage the realizations that take these subpaths (leading to the events of interest) by giving them a chance to reproduce (a bit like in selective evolution), and discourage the realizations that go in the wrong direction by killing them with some positive probability. The subpaths are usually delimited by levels, much like the level curves on a map. Starting from a given level, the realizations of the process

(which we also call *trajectories* or *chains* or *particles*) that do not reach the next level will not reach the rare event, but those that do are split (cloned) into multiple copies when they reach the next level, and each copy pursues its evolution from then on. This creates an artificial drift toward the rare event by favoring the trajectories that go in the right direction. In the end, an unbiased estimator can be recovered by multiplying the contribution of each trajectory by the appropriate weight. The procedure just described in known as *multilevel splitting*.

If we assume for instance that we are simulating a stochastic process (usually a Markov chain) and that the rare event of interest occurs when we reach a given subset of states before coming back to the initial state, then the levels can be defined by a decreasing (embedded) sequence of state sets that all contain the rare set of interest. In general, these levels are defined via an *importance function* whose aim is to represent how close a state is from this rare set. Several strategies have been designed to determine the levels, to decide the number of splits at each level, and to handle the trajectories that tend to go in the wrong direction (away from the rare event of interest). The amount of splitting when reaching a new level is an important issue; with too much splitting, the population of chains will explode, while with too little splitting, too few trajectories are likely to reach the rare event.

There is also the possibility to do away with the levels, by following a strategy that can either split the trajectory of kill it at any given step. One applies splitting (sometimes with some probability) if the weighted importance function is significantly larger at the current (new) state than at the previous state, and we apply Russian roulette (we kill the chain with some probability), when the weighted importance function becomes smaller instead. Russian roulette can also be viewed as splitting the chain into zero copies. The expected number of clones after the split (which is less than 1 in the case of Russian roulette) is usually taken as the ratio of the importance function value at the new state and that at the old state [13, 22].

The most important difficulty in general is to find an appropriate importance function. This function defines the levels (or the amount of splitting if we get rid of levels), and a poor choice can easily lead to bad results. In this sense, its role is analogous to the importance measure whose choice is critical in IS (see the previous chapter).

A question that naturally comes to mind is: What are the advantages and disadvantages of splitting compared with IS? One important advantage of splitting is that there is no need to modify the probability laws that drive

the system. This means (among other things) that the computer program that implements the simulation model can just be a black box, as long as there are facilities to make copies (clones) of the model, and to maintain weights and obtain the current value of the importance function for each of those copies. It is also interesting to observe that for splitting implementations where all chains always have the same weight at any given level, the empirical distribution of the states of the chains when they hit a given level provides an unbiased estimate of the theoretical entrance distribution of the chain at that level (the distribution of the state when it hits that level for the first time) under the original probabilities. With splitting implementations where chains may have different weights, and with IS, this is true only for the weighted (and rescaled) empirical distributions, where each observation keeps its weight when we define the distribution. There are also situations where it is simpler and easier to construct a good importance function for splitting than for IS, because IS can be more sensitive to the behavior of the importance function near the boundaries of the state space, as explained in [9, 12] (see also Section 1.2.3). One limitation of splitting with respect to IS is the requirement to decompose the state space in subsets (or layers) determined by the levels of some importance function, such that the probability of reaching the next level starting from the current one is not so small. When such a decomposition can be found, splitting can be efficiently applied. However, there are situations where the most probable paths that lead to the rare event have very few steps (or transitions), and where rarity comes from the fact that each of these steps has a very low probability. For example, in a reliability setting, suppose that the rare event is a system failure and that the most likely way that this failure occurs is by a failure of two components of the same type, which happens from two transitions of the Markov chain, where each transition has a very small probability. In such a situation, splitting cannot be effectively applied, at least not directly. It would require a trick to separate the rare transitions into several phases. IS, on the other hand, can handle this easily by increasing the probability of occurrence of these rare transitions. It is also important to recognize that in the case of large models (such as a large queueing system with many state variables), the state-cloning operations can easily induce a significant overhead in CPU time.

This chapter is organized as follows. Section 1.2 describes the general principles of splitting techniques and the main versions (or implementations) found in the literature. Section 1.3 provides an asymptotic analysis of the

method in a simplified setting that consists in assuming that reaching the next level from the current one can be modeled by a Bernoulli random variable independent of the current state (given that we have just reached the current level). This is equivalent to assuming that there is a single entrance state at each level. We then discuss how much we should split and how many levels we should define to minimize the variance, or its work-normalized version (the variance multiplied by the expected computing time), in an asymptotic setting. In Section 1.4, we provide an analysis based on interacting particle systems, following the general framework of [10]. This permits us to obtain a central limit theorem in a general setting, in an asymptotic regime where the number of initial trajectories (or particles) increases to infinity. While previous results were focusing on a specific case of splitting where the number of trajectories at each level is fixed, we additionally provide versions of the central limit theorem for other splitting implementations. Section 1.5 applies different versions of the splitting technique to a simple example of a tandem queue, used earlier by several authors. It illustrates the effectiveness of the method, and also the difficulties and the critical issue of finding an appropriate importance function.

Note that both IS and the splitting techniques have been introduced and investigated with the Monte Carlo method as early as in the mid 1940's, in Los Alamos [21, 22, 29]. The main relevant issues, such as an analysis of the optimal splitting strategies and the definition of the importance function, were already identified at that time.

1.2 Principles and implementations

1.2.1 Mathematical setting

Assume that the dynamics of the system under consideration is described by a strong Markov process $X = \{X(t)\,,\, t \geq 0\}$ with state space E, where the time index t can be either continuous (on the real line) or discrete (over the non-negative integers, i.e., $t = 0, 1, 2, \dots$). In the continuous-time case, we assume that all the trajectories are right-continuous with left-hand limits (càdlàg). Let $B \subset E$ be some closed critical region, in which the system could enter with a positive but very small probability, for example 10^{-10} or less. Our objective is to compute the probability of the critical event, i.e.,

$$\gamma = \mathbb{P}[T_B \leq T], \qquad \text{where} \qquad T_B = \inf\{t \geq 0 \,:\, X(t) \in B\}$$

denotes the entrance time into the critical region B, and where T is an almost surely finite stopping time.

Note that this can always be transformed into a model where the stopping time T is defined as the first hitting time of some set Δ by the process X, i.e.,

$$T = \inf\{t \geq 0 \,:\, X(t) \in \Delta\}.$$

For this, it suffices to put enough information in the state of the Markov process X so that T and every statistic that we want to compute are measurable with respect to the filtration generated by X up to time T. From now on, we assume that T is a stopping time of that form. As an important special case, this covers the situation where T is a deterministic finite time horizon: It suffices to include either the current clock time, or the time that remains on the clock before the time horizon is reached, in the definition of the state $X(t)$. For example, if we are interested in the probability that some Markov process $\{Y(t),\, t \geq 0\}$ hits some set C before some deterministic time t_1, then we can define $X(t) = (t, Y(t))$ for all t, $B = (0, t_1) \times C$, and $\Delta = B \cup ([t_1, \infty) \times E)$. Here, T_B is the first time when Y hits C if this happens before time t_1, $T_B = \infty$ otherwise, and $T = \min(t_1, T_B)$. Alternatively, it may be more convenient to define $X(t) = (t_1 - t, Y(t))$, where $t_1 - t$ is the time that remains on the clock before reaching the horizon t_1, B is the same as before, and $\Delta = B \cup ((-\infty, 0] \times E)$. For situations of this type, we will assume (when needed) that the state $X(t)$ always contains the clock time t, and that the sets B and Δ depend on the time horizon t_1. More generally, we could also have one of more clocks with random time-varying speeds.

Our results could be generalized to situations where the objective is to compute the entrance distribution in the critical region, or the probability distribution of critical trajectories, i.e.,

$$\mathbb{E}[\phi(X(T_B)) \mid T_B \leq T] \qquad \text{or} \qquad \mathbb{E}[f(X(t)\,,\, 0 \leq t \leq T_B) \mid T_B \leq T],$$

respectively, for some measurable functions ϕ and f. For simplicity, we focus our development here on the problem of estimating γ, which suffices to illustrate the main issues and tools.

The fundamental idea of splitting is based on the assumption that there exist some well identifiable intermediate subsets of states that are visited much more often than the rare set B, and that must be crossed by sample paths on their way to B. In splitting, the step-by-step evolution of the system follows the original probability measure. Entering the intermediate

states, which is usually characterized by crossing a threshold determined by a control parameter, triggers the splitting of the trajectory. This control is generally defined via the a so-called *importance function* h [16] which should satisfy $B = \{x \in E : h(x) \geq L\}$ for some level L.

Multilevel splitting uses an increasing sequence of values $L_0 \leq \cdots \leq L_k \leq \cdots \leq L_n$ with $L_n = L$, and defines the decreasing sequence of sets

$$E \supset B_0 \supset \cdots \supset B_k \supset \cdots \supset B_n = B,$$

with

$$B_k = \{x \in E : h(x) \geq L_k\},$$

for any $k = 0, 1, \cdots, n$. Note that in the case of a deterministic time horizon, $h(x)$ will usually depend on the current time, which is contained in the state x. Similarly, we can define the entrance time

$$T_k = \inf\{t \geq 0 : X(t) \in B_k\},$$

into the intermediate region B_k, and the event $A_k = \{T_k \leq T\}$, for $k = 0, 1, \cdots, n$. Again, these events form a decreasing sequence

$$A_0 \supset \cdots \supset A_k \supset \cdots \supset A_n = \{T_B \leq T\},$$

and the product formula

$$\begin{aligned} \mathbb{P}[T_B \leq T] &= \mathbb{P}(A_n) = \mathbb{P}(A_n \cap \cdots \cap A_k \cap \cdots \cap A_0) \\ &= \mathbb{P}(A_n \mid A_{n-1}) \cdots \mathbb{P}(A_k \mid A_{k-1}) \cdots \mathbb{P}(A_1 \mid A_0)\, \mathbb{P}(A_0), \end{aligned} \tag{1.1}$$

clearly holds, where ideally each conditional probability on the right side of (1.1) is "not small". The idea is to estimate each of these conditional probabilities somehow separately, although not completely independently, according to a branching splitting technique.

Suppose for now that all the chains have the same weight at any given level. A population of N_0 independent trajectories of the Markov process is created (their initial states can be either deterministic or generated independently from some initial distribution), and each trajectory is simulated until it enters the first intermediate region B_0 or until time T is reached, whichever occurs first. Let R_0 be the number of trajectories that have managed to enter the first intermediate region B_0 before time T, The fraction $\widehat{p}_0 = R_0/N_0$ is an

unbiased estimate of $\mathbb{P}(A_0) = \mathbb{P}[T_0 \le T]$. At the next stage, N_1 replicas (or offspring) of these R_0 successful trajectories are created, so as to maintain a sufficiently large population; this is done by cloning some states if $N_1 > R_0$ or choosing them randomly otherwise. Each new trajectory is simulated until it enters the second intermediate region B_1 or until time T is reached, whichever occurs first. Again, the fraction $\widehat{p}_1 = R_1/N_1$ of the R_1 successful trajectories that have managed to enter the second intermediate region B_1 before time T is a natural estimate of $\mathbb{P}(A_1 \mid A_0) = \mathbb{P}[T_1 \le T \mid T_0 \le T]$. The procedure is repeated again until the last step, in which each trajectory is simulated until it enters the last (and critical) region $B_n = B$ or until time T is reached, whichever occurs first. The fraction of the successful trajectories that have managed to enter the last (and critical) region $B_n = B$ before time T is a natural estimate of $\mathbb{P}(A_n \mid A_{n-1}) = \mathbb{P}[T_n \le T \mid T_{n-1} \le T]$. In other words, the probability of the rare event is estimated as the product of estimates of the transition probabilities from one intermediate region to the next intermediate region, where the transition probability at level k is estimated as the fraction $\widehat{p}_k = R_k/N_k$ of the number R_k of successful trajectories that have managed to enter the next intermediate region before time T over the number N_k of trials. In case $R_k = 0$ at any given stage k, we define $\hat{p}_{k'} = 0$ for all $k' > k$.

It is worth noting that the resulting estimator is unbiased, although the successive estimates are dependent because the result at level $k+1$ depends on the entrance states in region B_k [15, 26]. Indeed, by induction, assuming that $\mathbb{E}[\widehat{p}_0 \cdots \widehat{p}_{k-1}] = p_1 \cdots p_{k-1}$ with $p_k = \mathbb{P}(A_k \mid A_{k-1})$, we have

$$\begin{aligned}
\mathbb{E}[\widehat{p}_0 \cdots \widehat{p}_k] &= \mathbb{E}[\widehat{p}_0 \cdots \widehat{p}_{k-1}\, \mathbb{E}[\widehat{p}_k \mid N_0, \cdots, N_{k-1}, R_0, \cdots, R_{k-1}]\,] \\
&= \mathbb{E}[\widehat{p}_0 \cdots \widehat{p}_{k-1}\, (N_{k-1}\, p_k)/N_{k-1}] \qquad (1.2) \\
&= p_0 \cdots p_k \;=\; \gamma.
\end{aligned}$$

All the implementations described below are also unbiased [27].

The *entrance distribution* to B_k is the probability distribution μ_k of $X(T_k)$, the first entrance state into B_k, conditional on $T_k \le T$. An important observation is that each of the R_k trajectories that hits B_k before T hits it for the first time at a state having distribution μ_k. Using the same conditioning argument as in (1.2), one can see that for any measurable set $C \subseteq B_k$, the proportion of these R_k trajectories that hit B_k for the first time in C is a random variable (actually a ratio of two random variables)

with expectation $\mu_k(C)$. That is, the empirical distribution $\widehat{\mu}_k^N$ of these R_k entrance states into B_k is an unbiased estimators of μ_k. Then, the N_{k+1} states obtained after the splitting are essentially a bootstrap sample from this empirical distribution. However, the R_k entrance states into B_k are not independent (in fact, they can be strongly dependent in some cases, especially when k is large), and this complicates the convergence analysis of this empirical distribution. We will return to this in Section 1.4. We recognize that the empirical distribution is undefined when $R_k = 0$. This is rarely a problem in practice and we neglect this possibility here.

In a more general setting where the chains can have different weights, we also define the *weight* of a trajectory as follows. A starting trajectory has weight 1. Each time it is split, its weight is divided by the number of offspring (or its expected value when this number is random). As a consequence, the above estimator of γ is just the sum of weights of the successful trajectories, divided by the number of trajectories that were originally started. If Russian roulette is applied, the weights can also increase when a chain survives the roulette. In that case, an unbiased estimator of γ is the sum of (final) weights of the chains that reach B at time $T_B \leq T$, and an unbiased estimator of the entrance distribution to B_k is the weighted empirical distribution of the states of the chains that hit k, at the first step when they hit it.

1.2.2 Implementations

There are many different ways of implementing the splitting idea. First, various types of strategies can be used to determine the number of retrials (i.e., clones) of a chain at each level, including the following ones:

- In a *fixed splitting* implementation, each trajectory that has managed to reach the intermediate region B_{k-1} before time T receives the same deterministic number O_{k-1} of offspring. Then, $N_k = R_{k-1}\, O_{k-1}$ is a random variable. One advantage is that this can be implemented in a depth-first fashion, recursively: at level k, each chain is simulated until $\min(T, T_k)$. If A_k occurs, each clone is completely simulated by looking at all its offspring, before going to the next clone. Thus, it suffices to store a single entrance state at each level.

- In a *fixed effort* implementation, a fixed and predetermined number N_k of offspring is allocated to the collection of successful trajectories that have managed to reach the intermediate region B_{k-1} before time

T. To determine the starting point of the offspring, all the entrance states must be known, which means that the algorithm must be applied sequentially, level by level. Several strategies are then possible to assign the offspring to a successful trajectory. In the *random assignment*, the N_k starting states are selected at random, with replacement, from the R_{k-1} available states. In the *fixed assignment*, each successful trajectory is split approximately the same number of times, resulting in a smaller variance [1]. This is applied by first assigning $\lfloor N_k/R_{k-1} \rfloor$ offspring (or splits) to each state, and then assigning the remaining N_k mod R_{k-1} offspring to distinct trajectories chosen at random (without replacement), so these chosen trajectories would have $\lfloor N_k/R_{k-1} \rfloor+1$ offspring assigned to them [25, 26].

- In a *fixed success* implementation [24], a different perspective is considered. The idea is to create and simulate sufficiently many offspring, from time T_{k-1} onward, so that a fixed and predetermined number H_k of trajectories actually manage to reach the intermediate region B_k before time T. The issue here is to control the computational effort, because the number N_k of replicas needed to achieve exactly H_k successes is random. On the positive side, this implementation sorts out the extinction problem automatically, by construction, i.e., the simulation will always run until it reaches the rare event a sufficient number of times. On the other hand, the computing effort may have a large variance.

- In a *fixed probability of success* implementation proposed in [8], the level sets are constructed recursively such that the probability to reach one level from the previous level is approximately q, where q a fixed constant such that $0 < q < 1$. In this variant, assuming $\mathbb{E}[T] < \infty$, each of the $N = N_0$ chains is simulated until it reaches the recurrent set Δ. Let us denote by $X^i(\cdot)$ the trajectory of the ith chain, T^i its stopping time, and $S_{N,i} = \sup_{0 \le t \le T^i} h(X^i(t))$ the maximum value of the importance function over its entire trajectory. Sort in increasing order the values $(S_{N,1}, \cdots, S_{N,N})$, to obtain $S_{N,(1)} \ge \cdots \ge S_{N,(N)}$. The $K = \lfloor N\,q \rfloor$ chains yielding the largest values $S_{N,(N-K+1)}, \cdots, S_{N,(N)}$ are kept, and in order to maintain a population of N chains, $(N-K)$ new trajectories are simulated with initial state the state at which the value $S_{N,(N-K)}$ was recorded, and until they reach Δ. Combining

the maximum value of the importance function of these $(N - K)$ new trajectories with the K values recorded previously, we obtain a new sample of N values that we sort again in increasing order. We repeat the procedure while $S_{N,(N-K)} \leq L$, i.e., while at least $N - K$ chains have not reached B. The number n of iterations of the algorithm, and the number R of chains reaching B when the algorithm stops, are random variables, and the estimator of the probability of the rare event is $(K/N)^n (R/N)$. This estimator is biased, but consistent. It also achieves the same asymptotic variance as $N \to \infty$ than the fixed effort algorithm, with a probability q of going from any given level to the next one.

Another important issue, from a practical viewpoint, is the computational effort required at each level. If T is the return time to a given set A of "initial" states, for example, then the average time before either reaching the next level or going back to A is likely to increase significantly when k increases. Several techniques can be designed to alleviate this problem. A simple heuristic is to pick a positive integer β and just kill (truncate) the chains that go down by β levels or more below the current level L_{k-1}, based on the idea that they are very unlikely to come up again and reach level k. This reduces the computational time, but on the other hand introduces a bias. One way to deal with this bias is to apply the Russian roulette principle [22] and modify accordingly the weight of the chain. Several versions of this are proposed in [25, 26], including the following ones (where we also select a positive integer β and we assume that the chain tries to reach level k):

- *Probabilistic truncation* applies the Russian roulette each time a trajectory crosses a level $k - 1 - j$ downward, for any $j \geq \beta$. We select real numbers $r_{k-1,j} \in [1, \infty)$ for $j = \beta, \ldots, k - 1$. Whenever a chain crosses level $k - 1 - j$ downward from level $k - 1$, for $j \geq \beta$, it is killed with probability $1 - 1/r_{k-1,j}$. If it survives, its weight is multiplied by $r_{k-1,j}$. When a chain of weight $w > 1$ reaches level k, it is cloned into $w - 1$ additional copies and each copy is given weight 1 (if w is not an integer, we make $\lfloor w \rfloor$ additional copies with probability $\delta = w - \lfloor w \rfloor$ and $\lfloor w - 1 \rfloor$ additional copies with probability $1 - \delta$). The latter is done to reduce the variance introduced by the weights.

- *Periodic truncation* [26] reduces the variability due to the Russian roulette in probabilistic truncation by adopting a more systematic se-

lection of the chains that we keep. Otherwise it works similarly to probabilistic truncation and also uses positive integers $r_{k-1,j}$. It also uses a random integer $D_{k-1,j}$ generated uniformly in $\{1, \ldots, r_{k-1,j}\}$, for each k and $j \geq \beta$. When a chain crosses a level $k-1-j$ downward, if it is the $(i\, r_{k-1,j} + D_{k-1,j})$-th chain that does that for some integer i, it is retained and its weight is multiplied by $r_{k-1,j}$, otherwise it is killed.

- *Tag-based truncation* [26] fixes beforehand the level at which a chain would be killed. Each chain is *tagged* to level $k-1-j$ with probability $q_{k-1,j} = (r_{k-1,j}-1)/(r_{k-1,\beta} \cdots r_{k-1,j})$ for $j = \beta, \ldots, k-1$, and it is killed if it reaches that level. With the remaining probability, it is never killed. By properly choosing integers $r_{k-1,j}$, the proportion of chains tagged to level L_{k-1-j} can be *exactly* $q_{k-1,j}$, while the probability of receiving a given tag is the same for all chains.

To get rid of the weights, which carry additional variance, we can let the chain resplits when it up-crosses some levels after it went down and its weight was increased. The idea is to keep the weights close to 1. The above truncation schemes can be adapted to fit that framework.

One of the best-known versions of splitting is the RESTART method [31, 32, 33]. Here, when a chain hits a level upward, fixed splitting is used (i.e., the chain is split by a fixed factor), but one of the copies is tagged as the *original* for that level. Truncation is used to reduce the work: When a non-original copy hits its creation level downward, it is killed. Only the original chain continues its path (to avoid starvation). The weight of the original chain accounts (in some sense) for those that are killed, to keep the estimator unbiased. This rule applies recursively, and the method is implemented in a depth-first fashion, as follows: whenever there is a split, all the copies are simulated completely, one after the other, then simulation continues for the original chain. The gain in work-reduction is counter-balanced by the loss in terms of a higher variance in the number of chains, and a stronger positive correlation between the chains due to resplits [15].

In the discrete-time situation, another implementation does not make use of levels, but applies *splitting and Russian roulette* at each step of the simulation [3, 5, 13, 28]. The number of splits and the killing probabilities are determined in terms of the importance function h. Define $\alpha = \alpha(x, y) = h(y)/h(x)$ the ratio of importance values for a transition from x to y. If $\alpha \geq 1$, the chain is split in C copies where $\mathbb{E}[C] = \alpha$, whereas if $\alpha < 1$ it is killed with probability $1 - \alpha$ (this is Russian roulette). A weight is

again associated to each chain to keep the estimator unbiased: whenever a chain of weight w is split in C copies, the weight of all the copies is set to $w/\mathbb{E}[C]$. When Russian roulette is applied, the weight of a surviving chain is multiplied by $1/(1-\alpha)$.

Yet another version, again in the discrete-time situation, mixes splitting and Russian roulette with IS. The *weight* of a chain is redefined as the weight due to splitting and Russian roulette (just like above) times the *likelihood ratio* accumulated so far (see the previous chapter on IS). To reduce the variance of the weights, the idea of *weight windows* was introduced in [2], and further studied in [4, 14, 27]. The goal is to keep the weights of chain inside a given predefined window, with the aim of reducing the variance. This is done by controlling the *weighted importance* of each chain, defined as the product of its weight w and the value of the importance function $h(x)$ at its current state, so that it remains close to $\gamma = \mathbb{P}[T_B \leq T]$ for the trajectories for which $T_B \leq T$. If these windows are selected correctly (this requires a good prior approximation of γ), the main source of variance will then be the random *number* of chains that reach B [4]. To proceed, we select three real numbers $0 < a_{\min} < a < a_{\max}$. Whenever the weighted importance $\omega = w\,h(x)$ of a chain falls below $a_{\min}$, Russian roulette is applied, killing the chain with probability $1 - \omega/a$. If the chain survives, its weight is set (increased) 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 (decreased) weight w/c to each copy. If $a = (a_{\min} + a_{\max})/2 \approx \mathbb{P}[T_B \leq T]$, this number has expectation N_0 (approximately), the initial number of chains.

1.2.3 Major issues to address

The general principles and some known versions of splitting having been described, we now discuss several key issues that need to be addressed for an efficient implementation of splitting.

- *How to define the importance function h?* This is definitely the most important and most difficult question to address. For the multilevel splitting, in the simple case where the state space is one-dimensional and included in $\mathbb{R}$, the final time is an almost surely finite stopping time, and the critical region has the form $B = [b, \infty)$, then all strictly increasing functions h are equivalent if we assume that we have the freedom to select the levels (it suffices to move the levels to obtain

the same subsets B_k). So we can just take $h(x) = x$, for instance. Otherwise, especially if the state space is multidimensional, the question is much more complicated. Indeed, the importance function is a one-dimensional projection of the state space. Under simplifying assumptions, it is shown in [16] and later in this paper that ideally, to minimize the residual variance of the estimator from the current stage onward, the probability of reaching the next level should be the same at each possible entrance state to the current level. This is equivalent to having $h(x)$ proportional to $\mathbb{P}[T_B \leq T \mid X(0) = x]$. But if we knew these probabilities, we would know the exact solution and there would be no need for simulation. In this sense, it is a similar issue to that of the optimal (zero variance) change of measure in importance sampling. The idea is then to use an approximation of $\mathbb{P}[T_B \leq T \mid X(0) = x]$, or an adaptive (learning) technique. One way to learn the importance function was proposed in [4]: the state space is partitioned in a finite number of regions and the importance function h is assumed to be constant in each region. The "average" value of $\mathbb{P}[T_B \leq T \mid X(0) = x]$ in each region is estimated by the fraction of chains that reach B among those that have entered this region. These estimates are combined to define the importance function for further simulations, which are used in turn to improve the estimates, and so on. We will see in Section 1.5, on a simple tandem queue, that the choice of the importance function is really a critical issue; an intuitively appealing (but otherwise poor) selection can lead to high inefficiency.

It is important to emphasize that the above analysis considers only the variance and not the computing time (the work). If we take the work into account (which we should normally do) then taking $h(x)$ proportional to $\mathbb{P}[T_B \leq T \mid X(0) = x]$ is not necessarily optimal, because the expected work to reach B may depend substantially on the current state x.

In a rare event setting, it is important to understand how a proposed importance function would behave asymptotically as a function of the rare event probability γ when $\gamma \to 0$, i.e., in a *rare-event asymptotic* regime. This type of analysis is pursued in [9], in a framework where γ is assumed to be well approximated by a large deviation asymptotic, for which the rate of decay is described by the solution of the Hamilton-Jacobi-Bellman (HJB) nonlinear partial differential equations associ-

ated with some control problem. The authors show that a good importance function must be a viscosity subsolution of the HJB equations, multiplied by an appropriate scalar selected so that the probability of reaching a given level k from the previous level $k-1$, is $1/O_{k-1}$ when $L_k = k-1$. In the context of fixed splitting, this condition is necessary and sufficient for the expected total number of particles not to grow exponentially with $-\log\gamma$. Moreover, if the subsolution also has its maximal possible value at a certain point, then the splitting scheme is asymptotically optimal, in the sense that the relative variance grows slower than exponentially in $-\log\gamma$.

- *How to choose the number of offspring?* In fixed splitting, the question is how to select the number O_k of offspring at each level. If we do not split enough, reaching the next level (and the rare event) becomes unlikely. On the other hand, if we split too much, the number of trajectories will explode exponentially with the number of levels, which will result in computational problems. The proper amount has to be found out. In the next subsection, we investigate this issue in a simplified setting. In fixed effort splitting, no explosion is possible, as a fixed total number N_k of offspring is allocated at level k to the collection of successful trajectories that have managed to reach B_k. Nonetheless, deciding how many offspring to create, as well as the number of successful trajectories in the case of a fixed-performance implementation, are important issues.

- *Given the importance function h, how many intermediate regions should be introduced and how to define the increasing sequence of thresholds ?* The next subsection investigates this point. However, the exact optimal strategy depends on the implementation considered. There is also the option to learn the levels, as done in the fixed-probability-of-success method of [8].

1.3 Analysis in a simplified setting: a coin-flipping model

Suppose we have already selected an importance function and one of the splitting implementations discussed in the previous section. For a given total

computing budget, we would like to find the number and the locations of the thresholds, or equivalently the numbers n, $p_0, \cdots, p_n$, that minimize the variance of the estimator. We are also interested in convergence results for the variance and the work-normalized variance, under various asymptotic regimes, such as when $N \to \infty$ while n and $p_0, \cdots, p_n$ are fixed, or when $\gamma \to 0$ and $n \to \infty$, for example. Here we study these questions and provide partial answers under a very simplified (but tractable) model, for the fixed-effort and fixed-splitting strategies. The main focus is on the asymptotic behavior when $N \to \infty$. Our simplified setting is a coin-flipping model uniquely characterized by the initial probability $p_0 = \mathbb{P}(A_0)$ (i.e., the occurrence of the event A_0 depends only on the outcome of a $\{0, 1\}$ Bernoulli trial with parameter p_0), and by the transition probabilities $p_k = \mathbb{P}(A_k \mid A_{k-1})$ (i.e., the occurrence of A_k, conditional on A_{k-1}, depends only on the outcome of a $\{0, 1\}$ Bernoulli trial with parameter p_k), for $k = 1, \cdots, n$. This model is equivalent to assuming that there is only a single entrance state at each level.

For the work-normalized analysis, we need to make some assumptions on how much work it takes, on average, to run a trajectory from a given level $k-1$ until it reaches either the next level or the set $A = \Delta \setminus B$ (i.e., the stopping time T without reaching B). In case where there is a natural drift toward A, it appears reasonable to assume that the chains will reach A is $O(1)$ expected time, independently of n, if A and B (and therefore γ) are fixed. If we use truncation and/or Russian roulette, we still have $O(1)$ expected time. Then, the total expected work for all stages is proportional to $\sum_{k=0}^{n} \mathbb{E}[N_k]$. This is the assumption we will make everywhere in this section, unless stated otherwise. If $\mathbb{E}[N_k] = N$ for all k, then this sum is $N(n+1)$. For simplicity, we will further assume that the constant of proportionality (in the $O(1)$ expected time mentioned above) is 1.

In a different asymptotic regime, where $\gamma \to 0$ and $n \to \infty$ jointly, and if truncation and/or Russian roulette are not applied, the average time to reach A should increase when $\gamma \to 0$, typically as $O(-\ln \gamma)$, in which case the total work will be proportional to $(-\ln \gamma)(n+1)\sum_{k=0}^{n} N_k$. If we further assume that $p_0, \cdots, p_n$ are all equal to a fixed constant p, then $\gamma = p^{(n+1)}$, so $-\ln \gamma = -(n+1)\ln p$ and the total work is proportional to $(-\ln p)(n+1)^2 \sum_{k=0}^{n} N_k$. As it turns out, the extra linear factor $(n+1)$ has a negligible role in the asymptotic behavior [18, 20, 26].

1.3.1 Fixed effort

Several analytical studies have been performed for the fixed-effort model. In [25], an asymptotic analysis is performed for the case where $N_k = N$ and $p_k = p = \gamma^{1/(n+1)}$ for all k, in the simplified setting adopted here. In this setting, $R_0, R_1, \cdots, R_n$ are independent binomial random variables with parameters n and p. Then, we have [15, 25, 26]:

$$\begin{aligned}\mathrm{var}(\widehat{p}_0 \cdots \widehat{p}_n) &= \prod_{k=0}^{n} \mathbb{E}(\widehat{p}_k^2) - \gamma^2 \\ &= (p^2 + p\,(1-p)/N)^{n+1} - p^{2(n+1)} \\ &= \frac{(n+1)\,p^{2n+1}\,(1-p)}{N} + \frac{n\,(n+1)\,p^{2n}\,(1-p)^2}{2N^2} \\ &\qquad + \cdots + \frac{(p\,(1-p))^{n+1}}{N^{n+1}}.\end{aligned}$$

If we assume that $N \gg n(1-p)/p$, the first term

$$(n+1)\,p^{2n+1}\,(1-p)/N \approx (n+1)\,\gamma^{2-1/(n+1)}/N$$

dominates this variance expression. Given that the expected work is $N(n+1)$, the *work-normalized variance* is proportional to $[(n+1)\,\gamma^{2-1/(n+1)}/N]\,N\,(n+1) = (n+1)^2\,\gamma^{2-1/(n+1)}$, asymptotically, when $N \to \infty$. Minimizing w.r.t. n yields a minimum value at $(n+1) = -\frac{1}{2}\ln\gamma$, which corresponds to $p = e^{-2}$. If we assume that the constant of proportionality is 1, as we said earlier, then the resulting work-normalized relative variance is $(\ln\gamma)^2 e^2/4$.

For the asymptotic regime where $\gamma \to 0$ while p and N are fixed (so $n \to \infty$), the first term no longer dominates the variance expression, because the assumption $N \gg n(1-p)/p$ is no longer valid. In this case, the relative error and it work-normalized version both increase to infinity at a logarithmic rate [26].

1.3.2 Fixed splitting

Under a fixed-splitting setting, the algorithm is equivalent to a simple Galton-Watson branching processes, where each successful trial for which the event A_k occurs receives the same (deterministic) number O_k of offspring, for $k = 0, \cdots, n-1$. Each p_k is estimated by $\widehat{p}_k = R_k/N_k$. An unbiased estimator

of $\gamma = \mathbb{P}(A_n) = p_0\, p_1 \cdots p_n$ is then given by

$$\widehat{p}_0 \cdots \widehat{p}_n = \frac{R_0}{N_0}\frac{R_1}{N_1} \cdots \frac{R_n}{N_n} = \frac{R_n}{N_0\, O_0 \cdots O_{n-1}} = \gamma\, \frac{R_n}{N\, m_n}\ ,$$

where $N = N_0$ and $m_k = m_{k-1}\ O_{k-1}\ p_k = p_0\ O_0\ p_1 \cdots p_{k-1}\ O_{k-1}\ p_k$ for $k = 1, \cdots, n$, with $m_0 = p_0$ by definition. The second equality in the display follows from the relation $N_k = R_{k-1}\ O_{k-1}$, which holds for $k = 1, \cdots, n$, and means that the probability of the rare event is equivalently estimated as the fraction of the number R_n of successful trials, for which the rare event A_n occurs, over the maximum possible number of trials, $N_0\ O_0 \cdots O_{n-1}$.

The relative variance of this estimator is

$$\frac{\mathrm{var}(\widehat{p}_0 \cdots \widehat{p}_n)}{\gamma^2} = \frac{1}{N}\ \sum_{k=0}^{n} \frac{1-p_k}{m_k}\ .$$

Moreover, by the strong law of large numbers, $R_k/N \to m_k$ almost surely for any $k = 0, 1, \cdots, n$ when $N \to \infty$, and in particular $\widehat{p}_0 \cdots \widehat{p}_n \to \gamma$ almost surely when $N \to \infty$. We also have the central limit theorem:

$$\sqrt{N}\ \left(\frac{\widehat{p}_0 \cdots \widehat{p}_n}{\gamma} - 1\right) \Longrightarrow \mathcal{N}\left(0, \sum_{k=0}^{n} \frac{1-p_k}{m_k}\right)$$

in distribution as $N \to \infty$, where $\mathcal{N}(\mu, \sigma^2)$ denotes a normal random variable with mean μ and variance σ^2.

The following performance analysis follows [23]. Under our assumptions, the total work is approximately $C = \sum_{k=0}^{n} N_k$, which satisfies

$$\frac{C}{N} = \sum_{k=0}^{n} \frac{N_k}{N} = \sum_{k=0}^{n} \frac{N_k}{R_k}\ \frac{R_k}{N} \to \sum_{k=0}^{n} \frac{m_k}{p_k},$$

almost surely when $N \to \infty$.

Suppose that we are allowed a fixed *expected* total computing budget c, i.e., we have the constraint $\mathbb{E}[C] \le c$. What is the optimal way of selecting N, n, $p_0, \cdots, p_n$ and $O_0, \cdots, O_{n-1}$, to minimize the variance given this fixed budget? Assuming that we use all the budget and that N is large enough so we can approximate C/N by its almost sure limit as $N \to \infty$, and neglecting the fact that n and the O_k must be integers, this optimization problem can be formulated as:

$$\min\ \ \frac{1}{N}\sum_{k=0}^{n} \frac{1-p_k}{m_k} \qquad \text{subject to} \qquad N\sum_{k=0}^{n} \frac{m_k}{p_k} = c.$$

Solving this in terms of N and $O_0, \cdots, O_n$, with the other variables fixed, yields

$$N = c\,\frac{(1/p_0 - 1)^{1/2}}{\sum_{k=0}^{n}(1/p_k - 1)^{1/2}} \quad \text{and} \quad O_k = \left(\frac{p_{k+1}(1-p_{k+1})}{p_k(1-p_k)}\right)^{1/2} \frac{1}{p_{k+1}},$$

for $k = 0, \cdots, n-1$. This gives the relative variance

$$\frac{\mathrm{var}(\widehat{p}_0 \cdots \widehat{p}_n)}{\gamma^2} = \frac{1}{c}\,\left(\sum_{k=0}^{n}(1/p_k - 1)^{1/2}\right)^2.$$

Next, minimizing w.r.t. $p_0, p_1, \cdots, p_n$ for a given n gives that the transition probabilities should all be equal to the same value, $p_k = p = \gamma^{1/(n+1)}$ for all k. This implies that the branching rates should all be the same, $O_k = O = 1/p$, which corresponds to the critical regime of the Galton–Watson branching process, for which $O\ p = 1$. It also implies that the initial population size should be equal to $N = c/(n+1)$. In this optimal case, the work-normalized relative variance becomes

$$c\,\frac{\mathrm{var}(\widehat{p}_0 \cdots \widehat{p}_n)}{\gamma^2} = (n+1)^2(\gamma^{-1/(n+1)} - 1) = (n+1)^2\frac{(1-p)}{p}.$$

Finally, minimizing w.r.t. n gives

$$n = \frac{-\ln\gamma}{\ln(1+u^*)} - 1 \approx -0.6275\,\ln\gamma - 1,$$

where $u^* \approx 3.9214$ is the unique positive minimum of the mapping $u \mapsto u/(\ln(1+u))^2$. Thus, the transition probabilities should all be equal to $p = 1/(1+u^*)$. The resulting work-normalized variance is

$$c\,\frac{\mathrm{var}(\widehat{p}_0 \cdots \widehat{p}_n)}{\gamma^2} = \frac{u^*\,(\ln\gamma)^2}{(\ln(1+u^*))^2} \approx 1.5449(\ln\gamma)^2,$$

which is slightly smaller than the value $(\ln\gamma)^2 e^2/4 \approx 1.8473(\ln\gamma)^2$ obtained in the fixed–effort case.

Consider now an asymptotic regime where $p_k = p$ is fixed and $n \to \infty$, so that $\gamma = p^{n+1} \to 0$. Suppose that $O_k = 1/p$, i.e., $N_{k+1} = R_k/p$, for $k = 0, \cdots, n$. Then the relative variance

$$\frac{\mathrm{var}(\widehat{p}_0 \cdots \widehat{p}_n)}{\gamma^2} = \frac{1}{N}\,(n+1)\,(1-p)/p$$

is unbounded when $n \to \infty$. However, the *asymptotic logarithmic relative variance* (see the chapter on robustness issues for rare event estimators) is

$$\lim_{n\to\infty} \frac{\ln[\mathrm{var}(\widehat{p}_0 \cdots \widehat{p}_n)/\gamma^2]}{\ln \gamma} = \lim_{n\to\infty} \frac{\ln[1 + (1/N)\,(n+1)\,(1-p)/p]}{(n+1)\ln p} = 0,$$

which means that the splitting estimator is asymptotically efficient under the assumptions made. Asymptotic results of this type were shown in [18, 20] in a more general setting where the probability transition matrix for the first-entrance state at level k, given the first-entrance state at level $k-1$, converges to a matrix with spectral radius $\rho < 1$, which implies that $p_k \to \rho$ when $k \to \infty$. In [20], the authors also show that in their setting, the multilevel splitting estimator is also work-normalized asymptotically efficient if and only if $O_k = 1/\rho$ for all k. This results holds if the expected computing time at level k is proportional to N_k, and it still holds if this expected time increases polynomially in k.

It is important to emphasize that for practical applications, γ and p are unknown, so the condition $O_k = 1/p$ (exactly) for all k cannot really be satisfied. Then, the population of chains is likely to either decrease too much and perhaps extinguish (so no chain will reach B) or explode (so the amount of work will also explode). This suggests that when γ is very small, fixed splitting is likely to lead to a large relative variance of the estimator and also a huge variance in the computing costs. For this reason, the more robust fixed-effort approach is usually preferable.

1.4 Analysis and central limit theorem in a more general setting

We will now relax the "coin-flipping" assumption of the previous section, so that the probability of hitting the next level L_{k+1} may now depend on the entrance state into the current set B_k. This is certainly more realistic. For example, the are situations where we might enter B_k and B_{k+1} simultaneously, in which case this probability is 1.

We study the performance of some of the splitting implementations introduced in Section 1.2.2 in the framework of multilevel Feynman–Kac distributions and their approximation in terms of interacting particle systems [6, 7, 10, 11]. We state a central limit theorem and provide expressions for

the asymptotic variance for the various implementations, in the large-sample asymptotic regime in which γ is fixed and $N = N_0 \to \infty$ (assuming that this implies $\mathbb{E}[N_k] \to \infty$ for all k). This provides some insight on the issues raised in Section 1.2.3. The results are stated here without proofs; most of the proofs can be found in the references cited above. We emphasize that this analysis is not for a rare-event asymptotic regime, for which $\gamma \to 0$; for this, we refer the reader to [9, 20]. For simplicity, all along this section, we make the assumption that at any level, all the chains have the same weight (so no Russian roulette is allowed, for example).

1.4.1 Empirical entrance distributions

As we pointed out earlier, splitting can be used to estimate expectations of more general functions of the sample paths than just the probability γ. In particular, we argued in Section 1.2.1 that when all the particles have the same weight, then the entrance distribution at any level does not depend on the choice of importance function, is the same as for the original chain, and can be estimated in an unbiased way by the empirical entrance distribution at that level, which we shall denote by $\hat{\mu}_k^N$. This empirical distribution is already available at no extra cost when running the simulation.

More specifically, recall that N_k particles are simulated in stage k, and R_k of them hit B_k at the end of that stage. Let $\{\xi_k^i,\, i = 1, \ldots, N_k\}$ be the states of the N_k chains at the end of stage k, and let $I_k = \{i : \xi_k^i \in B_k\}$ be the subset of those states that have successfully hit B_k by their stopping time T. Note that I_k has cardinality R_k. We have

$$\hat{\mu}_k^N = \frac{1}{R_k} \sum_{i \in I_k} \delta_{\xi_k^i}, \tag{1.3}$$

where δ_x represents the Dirac mass at x.

Proposition 1.4.1 *For any measurable set $C \subseteq B_k$, $\mathbb{E}[\hat{\mu}_k^N(C)] = \mu_k(C)$. This implies that for any measurable function ϕ,*

$$\mathbb{E}[\mathbb{E}[\phi(X(T_k)) \mid T_k \leq T]] = \mathbb{E}\left[\frac{1}{R_k} \sum_{i \in I_k} \phi(\xi_k^i)\right].$$

By taking $\phi(x)$ equal to the indicator that $x \in B_k$ in this proposition, we recover the unbiasedness result in (1.2).

1.4.2 Large sample asymptotics

We saw that the empirical entrance distribution $\hat{\mu}_k^N$ provides an unbiased estimate of μ_k, but what about the convergence (and convergence speed) of $\hat{\mu}_k^N$ to μ_k when $N \to \infty$? The next proposition answers this question by providing a central limit theorem, which can be proved using the technology developed in [10].

Proposition 1.4.2 *Let $\phi : E \to \mathbb{R}$ be a bounded and continuous function and $0 \leq k \leq n$. Then there is a constant $v_k(\phi)$, that depends on ϕ and on the splitting implementation, such that*

$$\sqrt{N}\ \left(\frac{1}{R_k}\ \sum_{i \in I_k} \phi(\xi_k^i) - \mathbb{E}[\phi(X(T_k)) \mid T_k \leq T] \right) \Longrightarrow \mathcal{N}(0, v_k(\phi))\ ,$$

in distribution when $N \to \infty$, where $\mathcal{N}(0, \sigma^2)$ is a normal random variable with mean 0 and variance σ^2. The result also extends to unbounded functions ϕ under appropriate uniform integrability conditions.

We also have a central limit theorem for the probability of reaching level k before T. When $k = n$, this gives a central limit theorem for the estimator of γ, the probability of the rare event.

Proposition 1.4.3 *For $0 \leq k \leq n$, there is a constant V_k that depends on the splitting implementation, such that*

$$\sqrt{N}\ \left(\frac{\widehat{p}_0 \cdots \widehat{p}_k}{p_0 \cdots p_k} - 1 \right) \Longrightarrow \mathcal{N}(0, V_k)$$

in distribution when $N \to \infty$.

By combining these two propositions, we also obtain a central limit result for the unconditional average cost, when a cost is incurred when we hit B_k:

$$\frac{\sqrt{N}}{p_0 \cdots p_k}\ \left(\frac{\widehat{p}_0 \cdots \widehat{p}_k}{R_k}\ \sum_{i \in I_k} \phi(\xi_k^i) - \mathbb{E}[\phi(X(\min(T, T_k)))] \right) \Longrightarrow \mathcal{N}(0, v_k(\phi))\ ,$$

in distribution when $N \to \infty$, if we assume that $\phi(x) = 0$ when $x \notin B_k$. By taking $\phi(x)$ equal to the indicator that $x \in B_k$, and $v_k(\phi) = V_k$, we recover the result of the second proposition.

An intuitive argument to justify these central limit theorems is that although the particles have dependent trajectories to a certain extent, the amount of dependence remains bounded, in some sense, when $N \to \infty$. The idea (roughly) is that the trajectories that start from the same initial state in stage 0 have some dependence, but those that start from different initial states are essentially independent (in fixed splitting they are totally independent whereas in fixed effort they are almost independent when N is large). When $N \to \infty$ while everything else is fixed, the number of initial states giving rise to one or more successful trajectories eventually increases approximately linearly with N, while the average number of successful trajectories per successful initial state converges to a constant. So the amount of independence increases (asymptotically) linearly with N, and this explains why these central limit theorems hold.

In what follows, we derive expressions for the asymptotic variance V_n, for selected splitting implementations. Straightforward modifications can provide expressions for V_k, for $0 \leq k < n$. One may also rightfully argue that instead of normalizing by $\sqrt{N}$ in the central-limit theorem, we should normalize by $C_N = \sum_{k=0}^{n} N_k$, the total number of particle-levels simulated, which could be seen as the total amount of computational work if we assume that simulating one particle for one level represents one unit of work. This makes sense if we assume that the expected work is the same at each level. If $C_N/N \to C$ in probability as $N \to \infty$, which is typically the case (in particular, $C_N/N = C = n + 1$ exactly in the fixed effort implementations), then using Slutsty's lemma yields

$$\sqrt{C_N}\ (\widehat{p}_0 \cdots \widehat{p}_n/\gamma - 1) \Longrightarrow \mathcal{N}(0, C\, V_n)$$

in distribution as $N \to \infty$. So normalizing by C_N instead of N only changes the variance by the constant factor C.

Define the function h_B by

$$h_B(x) = \mathbb{P}[T_B \leq T \mid X(t) = x],$$

for $x \in E$. This function turns out to be an optimal choice of importance function when we want to estimate γ. We also define

$$\nu_k = \frac{\mathrm{var}[h_B(X(T_k)) \mid T_k \leq T]}{\mathbb{E}^2[\phi(X(T_k)) \mid T_k \leq T]} = \frac{\int_E h_B^2(x) d\mu_k(x)}{\left(\int_E h_B(x) d\mu_k(x)\right)^2} - 1,$$

the relative variance of the random variable $h_B(X(T_k))$ conditional on $T_k \leq T$ (i.e., when $X(T_k)$ is generated from μ_k). These ν_k depend only on the original model, and not on the splitting implementation.

In the *fixed splitting* implementation, we have

$$V_n = \sum_{k=0}^{n} \frac{1-p_k}{m_k} + \sum_{k=0}^{n-1} \frac{\nu_k}{m_k} \left(1 - \frac{1}{O_k}\right),$$

where m_k is defined recursively by $m_0 = p_0$ and $m_k = m_{k-1}\, O_{k-1}\, p_k$ for $k = 1, \cdots, n$. This coincides for $n = 1$ with equation (2.21) in [15]. We also have

$$\frac{C_N}{N} = \sum_{k=0}^{n} \frac{N_k}{N} \longrightarrow C = \sum_{k=0}^{n} \frac{m_k}{p_k},$$

in probability as $N \to \infty$.

In the *fixed effort* implementation with *random assignment* using *multinomial resampling*, it is shown in [7] that

$$V_n = \sum_{k=0}^{n} \left(\frac{1}{p_k} - 1\right) + \sum_{k=0}^{n-1} \frac{\nu_k}{p_k}.$$

In the *fixed effort* implementation with *fixed assignment* using *residual resampling*, if $1/p_k$ is not an integer, for any $k = 0, 1, \cdots, n$, then [1]

$$V_n = \sum_{k=0}^{n} \left(\frac{1}{p_k} - 1\right) + \sum_{k=0}^{n-1} \frac{\nu_k}{p_k} \left(1 - p_k\,(1 - r_k)\right).$$

If $\frac{1}{2} < p_k < 1$ then $r_k = 1 - p_k$ and it is shown in [7] that

$$V_n = \sum_{k=0}^{n} \left(\frac{1}{p_k} - 1\right) + \sum_{k=0}^{n-1} \frac{\nu_k}{p_k} (1 - p_k^2).$$

In each of the three cases considered above, the asymptotic variance splits as the sum of two terms, a first term that depends on the transition probabilities only, i.e., indirectly on the thresholds only, and a second term that depends on the entrance distributions also, i.e., indirectly on the importance

[1] De Pierre: What if it is an integer, for example $p_k = 1/5$?

function h that defines the shape of the intermediate regions. If for any given k the function h_B is constant on the support of the entrance distribution μ_k, then $\nu_k = 0$ and the second term vanishes, so only the first term remains and we obtain

$$C\,V_n = \left(\sum_{k=0}^{n} \frac{m_k}{p_k}\right) \sum_{k=0}^{n} \frac{1-p_k}{m_k} \qquad \text{and} \qquad C\,V_n = (n+1) \sum_{k=0}^{n} \frac{1-p_k}{p_k}\ ,$$

in the fixed splitting case and in the (two different implementations of the) fixed effort case, respectively. Note that if the continuous-time Markov chain has almost surely continuous trajectories, then the support of the entrance distribution μ_k is $\{x \in E : h(x) = L_k\}$, and a sufficient condition for $\nu_k = 0$ is to take $h = h_B$ as importance function. In this special case, the model reduces to the coin-flipping model already studied in Section 1.3.

1.5 A numerical illustration

The example described in this section is simple, but it has been widely used, because it provides a good illustration of the impact of the choice of importance function [20, 15]. We consider an open tandem Jackson queueing network with two queues. The arrival rate at the first queue is $\lambda = 1$ and the mean service time is $\rho_i = 1/\mu_i$ at queue i, for $i = 1, 2$. The corresponding discrete time Markov chain is given by $X = \{X_j,\, j \geq 0\}$, where $X_j = (X_{1,j}, X_{2,j})$ is the number of customers in each of the two queues immediately after the jth event, where an event is an arrival or a service completion at a given queue. Our goal is to estimate the probability of reaching $B = \{(x_1, x_2) : x_2 \geq L\}$, the set of states for which the second queue has length at least L, before reaching $A = \{(0, 0)\}$. The final stopping time is $T = \min(T_A, T_B)$.

To illustrate the impact and difficulty of the choice of the importance function h, some choices are compared in [25, 26] for the case where $\rho_1 < \rho_2$, and in [19, 17, 16] for $\rho_1 > \rho_2$. Consider the three choices

$$\begin{aligned} h_1(x_1, x_2) &= x_2, \\ h_2(x_1, x_2) &= (x_2 + \min(0, x_2 + x_1 - L))/2, \\ h_3(x_1, x_2) &= x_2 + \min(x_1, L - x_2 - 1) \times (1 - x_2/L). \end{aligned}$$

The function h_1 is the simplest choice and is motivated (naively) by the fact 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, because we need at least $L - \min(0, x_2 + x_1 - L)$ arrivals at the first queue and $L - x_2$ transfers to the second queue. The function h_3 is inspired by [30], who uses $h(x_1, x_2) = x_2 + x_1$ when $\rho_1 < \rho_2$. This h was modified as follows. We have $h_3(x) = x_1 + x_2$ when $x_1 + x_2 \leq L - 1$ and $h_3(x) = L$ when $x_2 \geq L$. In between, i.e., in the area where $L - x_1 - 1 \leq x_2 \leq L$, we interpolate linearly in x_2 for any fixed x_1.

In [25, 26], the authors compare these functions in the fixed-effort case, with several truncation implementations. For a numerical example with $\rho_1 = 1/4$, $\rho_2 = 1/2$, and $L = 30$, for instance, they estimate the constants V_n and CV_n defined in the previous section, and find that they are much higher for h_1 than for h_2 and h_3. Using h_3 yields just slightly better results than h_2. The truncation and resplit increases the variance slightly, but it also decreases the computational time, and overall it improves the work-normalized variance CV_n roughly by a factor of 3. Detailed results can be found in [26].

When $\rho_1 > \rho_2$, the first queue is the bottleneck of the system, and the most likely sample paths to B are those where the first queue builds up first, and then there is a transfer of customers from the first to the second queue. But h_1 does not favor these types of paths. Instead, it favors the paths where x_1 remains small, because the customers in the first queue are transfered quickly to the second queue. As a result, splitting with h_1, can give a variance that is even larger than with standard Monte Carlo in this case [19]. This problem can be solved by a better choice of h [33].

Variants of this example with $B = \{(x_1, x_2) : x_1 + x_2 \geq L\}$ and $B = \{(x_1, x_2) : \min(x_1, x_2) \geq L\}$ are examined in [9], where the authors design importance functions by finding subsolutions to the HJB equations associated with a control problem. These importance functions perform extremely well when γ is very small.

Bibliography

[1] James E. Baker. Reducing bias and inefficiency in the selection algorithm. In John J. Grefenstette, editor, *Proceedings of the 2nd International Conference on Genetic Algorithms, Cambridge MA 1987*, pages 14–21, Mahwah, NJ, 1987. Lawrence Erlbaum Associates.

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

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

[4] Thomas E. Booth and John S. Hendricks. Importance estimation in forward Monte Carlo calculations. *Nuclear Technology / Fusion*, 5:90–100, January 1984.

[5] Thomas E. Booth and Shane P. Pederson. Unbiased combinations of nonanalog Monte Carlo techniques and fair games. *Nuclear Science and Engineering*, 110:254–261, 1992.

[6] Frédéric Cérou, Pierre Del Moral, François Le Gland, and Pascal Lezaud. Limit theorems for the multilevel splitting algorithm in the simulation of rare events. In *Proceedings of the 2005 Winter Simulation Conference, Orlando 2005*, pages 682–691, December 2005.

[7] Frédéric Cérou, Pierre Del Moral, François Le Gland, and Pascal Lezaud. Genetic genealogical models in rare event analysis. *ALEA, Latin American Journal of Probability and Mathematical Statistics*, 1:181–203, 2006. Paper 01–08.

[8] Frédéric Cérou and Arnaud Guyader. Adaptive multilevel splitting for rare event analysis. *Stochastic Analysis and Applications*, 25(2):417–443, March 2007.

[9] Thomas Dean and Paul Dupuis. Splitting for rare event simulation: A large deviation approach to design and analysis. *Stochastic Processes and their Applications*, 2008. To appear.

[10] Pierre Del Moral. *Feynman–Kac Formulae. Genealogical and Interacting Particle Systems with Applications*. Probability and its Applications. Springer, New York, 2004.

[11] 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, Berlin, 2006.

[12] Paul Dupuis, Ali Devin Sezer, and Hui Wang. Dynamic importance sampling for queueing networks. *Annals of Applied Probability*, 17:1306–1346, 2007.

[13] Sergei M. Ermakov and Viatcheslav B. Melas. *Design and Analysis of Simulation Experiments*, volume 339 of *Mathematics and its Applications*. Kluwer Academic Publishers, Dordrecht, 1995.

[14] Bennett L. Fox. *Strategies for Quasi–Monte Carlo*, volume 22 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, Norwell, MA, 1999.

[15] Marnix J. J. Garvels. *The Splitting Method in Rare Event Simulation*. Ph.D Thesis, Faculty of Mathematical Sciences, University of Twente, Enschede, October 2000.

[16] Marnix J. J. Garvels, Jan-Kees C. W. van Ommeren, and Dirk P. Kroese. On the importance function in splitting simulation. *European Transactions on Telecommunications*, 13(4 (Special issue on Rare Event Simulation)):363–371, July–August 2002.

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

[18] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. Splitting for rare event simulation : Analysis of simple cases. In *Proceedings of the 1996 Winter Simulation Conference, San Diego 1996*, pages 302–308, December 1996.

[19] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. A large deviations perspective on the efficiency of multilevel splitting. *IEEE Transactions on Automatic Control*, AC–43(12):1666–1679, December 1998.

[20] Paul Glasserman, Philip Heidelberger, Perwez Shahabuddin, and Tim Zajic. Multilevel splitting for estimating rare event probabilities. *Operations Research*, 47(4):585–600, July–August 1999.

[21] Herman Kahn. Modifications of the Monte Carlo method. Paper P–132, The Rand Corporation, Santa Monica, CA, 1949.

[22] Herman Kahn and Ted E. Harris. Estimation of particle transmission by random sampling. *National Bureau of Standards Applied Mathematical Series*, 12:27–30, 1951.

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

[24] François Le Gland and Nadia Oudjane. A sequential algorithm that keeps the particle system alive. 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 351–389. Springer, Berlin, 2006.

[25] Pierre L'Ecuyer, Valérie Demers, and Bruno Tuffin. Splitting for rare–event simulation. In *Proceedings of the 2006 Winter Simulation Conference, Monterey 2006*, pages 137–148, December 2006.

[26] Pierre L'Ecuyer, Valérie Demers, and Bruno Tuffin. Rare events, splitting, and quasi–Monte carlo. *ACM Transactions on Modeling and Computer Simulation*, 17(2 (Special issue honoring Perwez Shahabuddin)), April 2007. Article 9.

[27] Pierre L'Ecuyer and Bruno Tuffin. Splitting and weight windows to control the likelihood ratio in importance sampling. In *Proceedings of the 1st International Conference on Performance Evaluation Methodologies and Tools (ValueTools), Pisa 2006*, volume 180 of *ACM International Conference Proceeding Series*. ACM, October 2006. Article 21.

[28] Viatcheslav B. Melas. On the efficiency of the splitting and roulette approach for sensitivity analysis. In *Proceedings of the 1997 Winter Simulation Conference, Atlanta 1997*, pages 269–274, December 1997.

[29] Harry Soodak. Pile kinetics. In Clark Goodman, editor, *The Science and Engineering of Nuclear Power*, volume 2, chapter 8, pages 89–102. Addison–Wesley, Reading, MA, 1949.

[30] José Villén-Altamirano. Rare event RESTART simulation of two–stage networks. *European Journal of Operations Research*, 179(1):148–159, May 2007.

[31] Manuel Villén-Altamirano and José Villén-Altamirano. RESTART : A straightforward method for fast simulation of rare events. In *Proceedings of the 1994 Winter Simulation Conference, Orlando 1994*, pages 282–289, December 1994.

[32] Manuel Villén-Altamirano and José Villén-Altamirano. Analysis of RESTART simulation : Theoretical basis and sensitivity study. *European Transactions on Telecommunications*, 13(4 (Special issue on Rare Event Simulation)):373–385, July–August 2002.

[33] Manuel Villén-Altamirano and José Villén-Altamirano. On the efficiency of RESTART for multidimensional state systems. *ACM Transactions on Modeling and Computer Simulation*, 16(3):251–279, July 2006.