On the importance function in splitting simulation

MARNIX J.J. GARVELS, JAN-KEES C.W. VAN OMMEREN
Department of Mathematical Sciences, University of Twente, The Netherlands
garvels@cs.utwente.nl, j.c.w.vanommeren@math.utwente.nl

DIRK P. KROESE
Department of Mathematics, University of Queensland, Australia
kroese@maths.uq.edu.au

Abstract. The *splitting method* is a simulation technique for the estimation of very small probabilities. In this technique, the sample paths are split into multiple copies, at various stages in the simulation. Of vital importance to the efficiency of the method is the *Importance Function* (IF). This function governs the placement of the thresholds or surfaces at which the paths are split. We derive a characterisation of the optimal IF and show that for multi-dimensional models the "natural" choice for the IF is usually not optimal. We also show how nearly optimal splitting surfaces can be derived or simulated using reverse time analysis. Our numerical experiments illustrate that by using the optimal IF, one can obtain a significant improvement in simulation efficiency.

1 INTRODUCTION

The splitting method is a simple simulation method for the estimation of rare event probabilities. The method is based on the idea to restart the simulation in certain system states, in order to generate more occurrences of the rare event. For general references on the splitting method and the closely related RESTART method we refer to [1, 2, 3, 4] and [5, 6, 7].

During recent years many numerical and theoretical investigations have been made on the *efficiency* of the splitting/RESTART method. In various papers it was indicated that the method could yield asymptotically optimal estimators, and even yield estimators with bounded relative error [3, 4, 1, 8]. In other papers the efficiency of the method was put into question [9, 2]; in particular, when dealing with *multi-dimensional* state spaces. However, it was pointed out in [10] that the reduction in efficiency for the multi-dimensional could be remedied by the correct choice the *Importance Function*, which governs the placements of the splitting surfaces. Indeed, this was already implicitly suggested in [5, 6]. Another approach to efficiently deal with multi-dimensional state spaces is to use Direct Probability Redistribution as in [11, 12].

The purpose of this paper is to further develop the concept of Importance Function (IF). In particular, we derive a characterisation of the optimal IF and show how nearly optimal "splitting surfaces" can be derived by reverse time analysis/simulation.

The rest of the paper is organised as follows. We start with a brief review of the splitting method in section 2. In section 3 we characterise the optimal IF and discuss its relevance to splitting simulation. In section 4 we show how the optimal IF can be implemented; and in particular, how the "optimal" splitting surfaces can be estimated by reverse time analysis. In section 5 we investigate empirically the usefulness of the IF in splitting simulation, focusing on the 2- and 3-node tandem queue. Conclusions are given in section 6.

2 THE SPLITTING METHOD

We briefly review the splitting method in the setting of [3]. Consider a Markov process $X := (X_t, t \geq 0)$ with state space E. We are interested in the probability that X enters some set $A \subset E$ before it enters (or returns to) another set $B \subset E$. Often the sets A and B can be characterised as $A = \{x \in E \,|\, h(x) \geq L\}$ and $B = \{x \in E \,|\, h(x) \leq 0\}$ for some real function h and positive number L. From now on, we will assume this is the case, unless specified otherwise.

Define the real-valued process $Z = (Z_t)$, by $Z_t := h(X_t)$, for all $t \geq 0$. For any *level* (or *threshold*) $\ell > 0$, let T_ℓ denote the first time that the process Z hits or upcrosses level ℓ. Using the characterisation above we are thus interested in estimating the *overflow probability*, γ say, of the event $D := \{T_L < T_0\}$. Note that γ depends on the

initial distribution of X. With a slight abuse of notation we will sometimes use T_A for T_L and T_B for T_0, to indicate that these times correspond to X entering the sets A and B, respectively.

The splitting method works as follows, see figure 1 for an illustration. First, we partition the interval $[0, L)$ into m subintervals $[L_0, L_1), [L_1, L_2), \ldots, [L_{m-1}, L_m)$, with $0 =: L_0 < L_1 < \ldots < L_m := L$. We assume for simplicity that Z actually *hits* all thresholds $L_1, \ldots, L_m$ if event D occurs. Let D_k denote the event that process Z reaches level L_k before returning to 0. It is clear that this is equal to defining $D_k := \{T_k < T_0\}$. Then $D_1, D_2, \ldots, D_m$ is a nested sequence of events, decreasing to $D_m = D$. And, with $p_1 := \mathbb{P}(D_1), p_2 := \mathbb{P}(D_2 \,|\, D_1), \ldots$, we have

$$\gamma = p_1\, p_2 \cdots p_m .$$

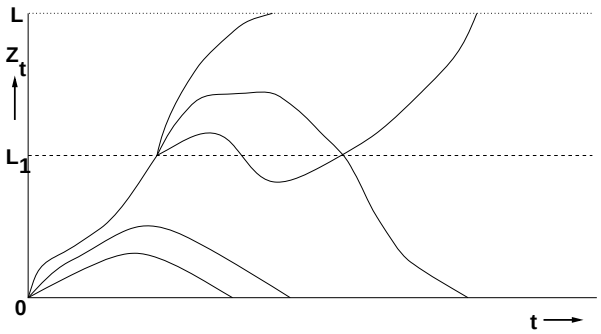

Figure 1: *An illustration of the splitting method*

Instead of estimating γ directly, we estimate the conditional probabilities $\{p_k,\ k = 1, \ldots, m\}$ in m consecutive *stages*. At the first stage we run n_1 (fixed) independent copies of X (and Z). Define $I_j^{(1)}$ as the indicator that the jth copy of Z reaches level L_1 before visiting 0, $j = 1, \ldots, n_1$. Let R_1 be the total number of copies out r_n that reach level L_1. Then an obvious estimate of p_1 is R_1/n_1. We save the *entrance states* (also called *saved states*) of all paths that reach level L_1. Specifically, for every copy of Z which crosses level L_1 we remember the state of the corresponding X at the time of crossing. We then proceed with the second stage. Here we start $R_1 n_2$ new independent copies of Z, n_2 copies from each copy from a certain saved state. Next, we generate Bernoulli variables $I_j^{(2)}, j = 1, \ldots, R_1 n_2$, such that $I_j^{(2)}$ indicates whether the jth copy of Z (Z starting from level L_1 and X from a saved state) reaches level L_2 before 0. This process repeats itself at all the subsequent stages $3, \ldots, m$. We call $r_k := R_{k-1} n_k$ the *simulation effort* at stage k, and R_k the (random) number of starting states or *successes* at stage k, $k = 1, \ldots, m$ ($R_0 := 1$).

In general the indicators $\{I_i^{(k)}\}$ are not independent; the success probability of an indicator depends typically on the state from which X restarts. However, with

$$\hat{p}_k := \frac{R_k}{r_k},\ \ k = 1, \ldots, m,$$

the 'natural' estimator

$$\hat{\gamma} := \prod_{k=1}^{m} \hat{p}_k, \tag{1}$$

is still *unbiased*, see e.g. [3].

For the simplest case where all the indicators $I_1^{(1)}, \ldots, I_{r_m}^{(m)}$ are independent, the variance of $\hat{\gamma}$ is given (see [3]) by

$$\mathbb{V}\mathrm{ar}\, \hat{\gamma} = \gamma^2 \sum_{k=1}^{m} \frac{1 - p_k}{\prod_{j=1}^{k} p_j n_j} = \gamma^2 \sum_{k=1}^{m} \frac{1 - p_k}{p_k\, \mathbb{E} r_k} . \tag{2}$$

The efficiency of the splitting method, determined by the variance of $\hat{\gamma}$ above, depends crucially on the appropriate choice of the number of intermediate levels $L_1, \ldots, L_m$ and the number of copies $r_1, \ldots, r_m$ generated at each threshold, also called the *splitting number*. Detailed investigations of the efficiency of the splitting method in [6, 3], based on (2), indicate that the levels should be chosen such that the probability of crossing a threshold when starting from the previous threshold, i.e., p_k, is roughly equal to $\mathrm{e}^{-2} \approx 0.135$. Moreover, r_k should be chosen approximately equal to $1/p_k \approx \mathrm{e}^2$, and for small γ we should take approximately $-\log(\gamma)/2$ thresholds.

Remark 1 The splitting implementation described above is called the Fixed Splitting (FS) implementation [3], because in every stage k we have a fixed number of re-samples of every saved state equal to r_k. In [3] it is shown that the Fixed Effort (FE) implementation, in which not the number of re-samples but instead the simulation effort is fixed, is much less sensitive to the choice of the simulation parameters, is easier to implement, and above all, gives a better performance, as measured by our performance evaluation ratio RTV, to be discussed in the section 5. The only disadvantages of the FE implementation is that the variance estimation is more complicated and that it requires more computer memory. We have chosen to implement the FS method in the practical section because of memory limitations.

3 THE IMPORTANCE FUNCTION

For a 1-dimensional process $X = Z$ the formulation of the splitting method in the previous section yields a very efficient and robust method for estimating the rare event probability γ. However, when X is a multi-dimensional process, the situation becomes more difficult. An example is the case where X represents the number of customers in a 2-node tandem queue. The efficiency of the splitting

method for this system was questioned in [9, 2]. However, in [11, 10] it was shown that the efficiency could be dramatically increased by correctly chosing the multi-dimensional thresholds.

To understand the splitting procedure for multi-dimensional processes we need to look at the formulation in section 2 in a slightly different way. First, note that by choosing the splitting levels $L_0, L_1, \ldots, L_m$ for the Z process we induce splitting *surfaces* $\mathcal{L}_0, \ldots, \mathcal{L}_m$ for the X process on E, with $\mathcal{L}_k = \{x \in E \,|\, h(x) = L_k\}$. Every time when X hits the splitting surface $\mathcal{L}_k$, r_k copies of the process are restarted from the point of impact. In this formulation T_k can be viewed as the first time that X hits $\mathcal{L}_k$, and D_k is the event that X reaches A (or equivalently $\mathcal{L}_m$) before B (or equivalently $\mathcal{L}_0$).

But, it is not at all clear whether this is the best way to choose the splitting surfaces. In fact, any choice $\tilde{\mathcal{L}}_k = \{x \in E \,|\, f(x) = L_k\}, k = 0, \ldots, m$ for some real function f could define subsequent splitting surfaces, provided that $\tilde{\mathcal{L}}_0 = \mathcal{L}_0$ and $\tilde{\mathcal{L}}_m = \mathcal{L}_m$. The function f that governs the choice of splitting surfaces is called the *importance function* (IF). It measures, in some sense, how close we are to set A. In this section we have a closer look at how to choose the IF.

3.1 One-dimensional State Space

In the one-dimensional case the selection of an IF is not an issue, since the splitting levels are basically fixed. To see this, consider (for simplicity) the case where $X = Z$ is a real-valued Markov process which is *skip-free* to the right. In particular, the process must hit each intermediate level in order to reach level L. Define the random variable M as the maximum state reached before X enters set $B = (-\infty, 0]$. The overflow probability, starting from $x \in (0, L)$, is easily seen to be equal to

$$\begin{aligned} \gamma_x &:= \mathbb{P}(T_A < T_B \,|\, X_0 = x) \\ &= \mathbb{P}(M \geq L \,|\, M \geq x) = \frac{\gamma}{\bar{F}(x)} \,, \end{aligned} \tag{3}$$

where $\bar{F}$ is the survival function of M: $\bar{F}(x) = \mathbb{P}(M \geq x)$. It is evident that this overflow probability is increasing monotonically in the starting state x.

Suppose the variance of $\hat{\gamma}$ in (2) is minimal for the success probabilities $p_1, \ldots, p_m$ and splitting numbers $r_1, \ldots, r_m$. Then, the levels $L_1, \ldots, L_m$ should be chosen such that

$$\gamma_{L_k} = p_{k+1} \cdots p_m, \quad k = 1, \ldots, m-1 \,.$$

In view of (3) we therefore have

$$L_k = \inf\{x \,|\, \bar{F}(x) \leq \frac{\gamma}{p_{k+1} \cdots p_m}\} \,,$$

as kth optimal splitting level.

As explained at the end of the previous section, the optimal $p_1, \ldots, p_m$ are approximately equal ($p_k = p$). If γ is exponentially decaying in L, e.g., $\lim_{x\to\infty} \mathbb{P}(M > L)\, e^{\theta L} = c$, for some $\theta, c \in \mathbb{R}^+$, then

$$p \approx \frac{\mathbb{P}(M > L_{k+1})}{\mathbb{P}(M > L_k)} \approx \frac{c\, e^{-\theta L_{k+1}}}{c\, e^{-\theta L_k}} = e^{-\theta(L_{k+1} - L_k)}$$

and the threshold L_{k+1} is given by $L_{k+1} = L_k - \log(p)/\theta$. In other words, the thresholds are spaced equidistantly for stages large enough. This formula also shows that for a strong decay (θ large) the thresholds will have to be close together, whereas for a weak decay (θ small) we will see distant asymptotic thresholds.

3.2 Multi-dimensional State Space

When X is a multi-dimensional process, the choice of the IF is not straightforward. The intuitive approach is to simply take f equal to h. That is, to let the splitting surfaces of X be $\mathcal{L}_k = \{x \in E \,|\, h(x) = L_k\}$, $k = 0, 1, \ldots, m$ for some choice of m and intermediate levels $L_1, \ldots, L_{m-1}$. However, choosing the IF in this naive fashion will generally not be optimal (in the sense of minimal variance of $\hat{\gamma}$). The reason is that the naive approach uses no information regarding the system's behaviour. We illustrate this with an example.

Example 1 Consider figure 2. We wish to estimate the probability that the process starting from 0 hits set $A = \{x \,|\, h(x) = L\}$ before hitting $B = \{x \,|\, x_2 = 0\}$ by using a splitting procedure with intermediate splitting surface/level $\{x \,|\, h(x) = L/2\}$, effectively using h as our importance function.

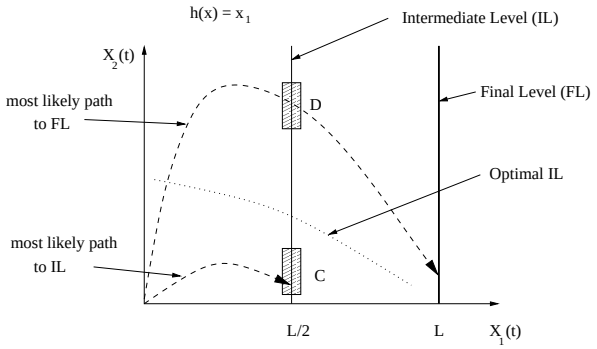

Figure 2: *An illustration of the importance function.*

Using this "natural" IF will create lots of runs that hit the intermediate level in Region C. However, we want many runs to cross the intermediate level in Region D, because those will have a high probability of actually reaching set A, whereas reaching A from Region C might even be impossible. Defining $P(C)$ as the probability of reaching set C before hitting B, and $P(C \,|\, A)$ the probability

of reaching set C given that we have reached set A, we analyse an example of how wrong the simple choice of IF can be. Suppose for example that $P(C) = 0.1, P(D) = 0.01, P(A) = 0.2, P(A \mid C) = 10^{-4}, P(A \mid D) = 0.1$. We see that $P(C \mid A) \approx 5 \cdot 10^{-5}$ and $P(D \mid A) \approx 5 \cdot 10^{-3}$. Clearly region D must be much more important than region C with respect to the rare event. We need to choose a different intermediate splitting surface (and hence a different IF f) such that it is more likely to reach A via the intermediate level. This could well be established by an optimal threshold represented by the dotted line. This new IF is more likely to favour paths that pass through region D and spend little effort on paths crossing region C, as we would want the optimal simulation to occur.

It is therefore essential that we construct our simulations using information about the system's behaviour, much as it is commonplace in Importance Sampling, where a lot of attention has been focused on creating paths that are compliant with the large deviations behaviour of the system.

Glasserman et al. [2] also look at this problem and prove that in certain cases the simple approach will not yield asymptotically efficient estimates, and will even show apparent bias with high probability. This effect is seen in our Example 1 by the large waste of effort on uninteresting samples, resulting in high variance and apparent bias in the estimator.

Next, we investigate how to choose a "good" IF, i.e., an IF which leads to a relatively small variance for $\hat{\gamma}$. We are led by the following two-stage example from [3].

Consider a two-stage splitting simulation. Suppose at the second stage we have R_1 saved states $S_1, \ldots, S_{R_1}$. Assume for simplicity that R_1 is *fixed*. We start a total of $r_2 := n_2 R_1$ new runs; n_2 independent paths from each starting state. The success probability of reaching the next level, starting from state s will be denoted by $p_2(s)$. Let Y_i be the number of successful runs (that reach the next level) starting from state S_i. Since we have only one intermediate stage, the S_i's are i.i.d. and $\mathbb{E}p_2(S_i) = p_2$. Consequently, $\hat{p}_2 := \sum_{i=1}^{R_1} Y_i / r_2$ is an unbiased estimator of p_2. It is not difficult to see ([3]) that

$$\mathbb{V}\text{ar}\,(\hat{p}_2) = \frac{1}{r_2}\left\{p_2(1-p_2) + (n_2 - 1)\mathbb{V}\text{ar}\,(p_2(S_1))\right\}. \tag{4}$$

The only factor in (4) that is dependent on the choice of the IF is $\mathbb{V}\text{ar}\,(p_2(S_1))$. We can minimise (4) by choosing the intermediate splitting surface in such a way that the probability of reaching the next level, starting from any state on the splitting surface, is *constant* (p_2). This suggest the following rule:

Rule 1 *For a two-stage splitting simulation choose the IF such that the probability of reaching the next level does not depend on the starting state.*

It is not difficult to extend this rule to $m \geq 2$ stages, using an induction argument. Suppose for m stages the "optimal" IF is such that the probability of reaching the next levels is constant within the splitting surfaces. We will show that this should be the case for $m+1$ stages as well. First, let $p_1, \ldots, p_{m+1}$ be the probabilities of reaching the next surfaces, as in Section 2. By the induction assumption the optimal IF for estimating $p_1 \ldots p_m$ is such that the probability of reaching the next levels is constant over the splitting surfaces. The only problem then becomes the choice of the last intermediate level. But now we are in exactly the same situation as the 2-level splitting case considered earlier. In particular, the final splitting surface should be chosen such that the probability of reaching the final level (or set A) is constant, p_{m+1}, within that level.

Rule 2 *For an m-stage splitting procedure choose the IF such that the probability of reaching any intermediate and final level does not depend on the starting state.*

Note that an IF of the form above makes the stages *independent*, as the success or failure of a stage $k+1$ path does not depend on the entrance state produced by its parent path in stage k.

Another observation to make is that the "optimal" IF f can be directly related to the probabilities $\gamma_x = \mathbb{P}(T_A < T_B \mid X_0 = x)$. Namely, Rule 2 above implies that γ_x is constant for all x within each splitting surface. This is established by taking $f(x) = g(\gamma_x)$ for some monotone increasing function g. The actual *form* of this function g will be given in section 4.

Continuous and Discrete State Spaces

When X has a continuous state space splitting surfaces of the form $\{x \in E \mid \gamma_x = c\}$ are generally well-defined, and are hit by any sample path that reaches the rare event set A. For discrete state spaces things become a bit more difficult; the (nearly) optimal splitting surfaces should be chosen such that γ_x is approximately constant on such surfaces. When all the probabilities γ_x are known, this could be solved by a classical Set Partitioning problem, which is proven to be hard to solve. Thus, in actual implementations we have to resort to approximate methods for choosing the splitting surfaces.

Remark 2 We note that the optimal way of choosing the IF is the same in splitting and RESTART simulation. The idea that the optimal IF should minimise the variance of the probability of reaching the next level was already suggested (for RESTART) in [5] and [6], and was further developped in [10].

4 IMPLEMENTATION

In this section we pay attention to the various implementation issues that need to be resolved in order to make

effective use of the implicit results found in the previous section 3. In particular, we will focus on how the "optimal" IF could be implemented.

4.1 DEFINING THE OPTIMAL IF

So far we have only obtained an implicit characterisation of our optimal IF as a function of $x \mapsto \gamma_x$. We now want to give a more explicit expression that could be used in actual implementations.

First, let

$$c_i = \prod_{j=i}^{m} c_j, \quad i = 1, \ldots, m$$

be the probability of reaching the final level starting from threshold $i-1$. Second, we assume that splitting surfaces are of the form

$$\mathcal{L}_i = \{x \,|\, f(x) = i\}, \quad i = 0, 1, \ldots, m \; .$$

It follows that the "optimal" IF f, as described in the previous section, is constructed such that for a given x

$$c_i \leq \gamma_x \Leftrightarrow i \leq f(x) \quad . \tag{5}$$

When we also assume that the process does not cross more than one threshold at a time, we arrive at the following tighter relation

$$c_i \leq \gamma_x < c_{i+1} \Leftrightarrow i \leq f(x) < i+1 \; . \tag{6}$$

Since we want an explicit expression for f we restrict it further by enforcing the following simple rules which we found as the optimal implementation techniques as presented in section 2. Make

1. the success probability in each stage equal,
2. the number of stages equal to $m = -\log(\gamma)/2$,
3. the number of samples per stage equal.

Since we have

$$c_i = p, \quad i = 1, 2, \ldots, m$$

for the optimal implementation, we obtain $c_i = p^{m-i}$, and since $c_0 = p^m = \gamma$, we obtain $\log(p) = \log(\gamma)/m$ resulting in

$$\gamma^{(m-i)/m} \leq \gamma_x < \gamma^{(m-i+1)/m} \Leftrightarrow i \leq f(x) < i+1 \; .$$

The definition of

$$f(x) = m\big(1 - \frac{\log \gamma_x}{\log \gamma}\big)$$

is easily seen to fulfill the requirement above, as well as making the function f continuous in the case that the success probability is continuous over the state space E. Filling in the optimal m we obtain

$$f(x) = \frac{1}{2} \log\big(\frac{\gamma_x}{\gamma}\big) \; . \tag{7}$$

The remaining problem is that the quantities γ_x are not known, in general, because it is as hard to solve as the original problem (substitute $x = X_0$). Therefore, we look at methods to find estimators for the unknown parameters.

4.2 REVERSE TIME ANALYSIS

We consider a reverse time simulation scheme to generate an estimate for the optimal IF. We focus on a specific type of system for which we can derive good estimates for the IF, which will feature in our simulation results section 5. Specifically we assume that X is a discrete-time Markov chain on a (countable) space E – for example, the discrete skeleton of a continuous-time Markov chain. Let the one-step transition probability from i to j be given by $p(i,j)$, for all i and j in E. We assume X is irreducible and has a stationary distribution π.

When we view the Markov chain X "backwards in time", we obtain the *time reversed* Markov chain $Y = (Y_n)$, which is a Markov chain on E with one-step probabilities

$$p'(i,j) = \pi(j)\, p(j,i) \,/\, \pi(i),$$

see, e.g., [13]. Moreover, Y has the same stationary probability distribution π as X. When we have $P = \tilde{P}$ we call X *reversible*.

Let D_x be the event that process X reaches A before B, starting from $x \in E$. We wish to estimate the probability $\gamma_x = \mathbb{P}(D_x)$. Denote by Ω_x the set of "paths" of the form $(i_0, i_1, \ldots, i_n)$ starting at $i_0 = x$ and ending at a state in A, such that none of the intermediate states are in B. Also, let Ω'_x be the set of paths of Y, of the form $(j_0, \ldots, j_n)$ for some n, which start somewhere at A and reach x before B, without returning to A. Let D'_x denote the corresponding event that Y reaches x before B and A, starting from A. Note that each path $(i_0, i_1, \ldots, i_n)$ in Ω_x corresponds to a path $(i_n, i_{n-1}, \ldots, i_0)$ in Ω'_x and vise versa. Moreover, for each such path we have

$$\begin{aligned}
&\mathbb{P}(X_0 = x_0, \ldots, X_n = x_n) \\
&\quad = \; p(x_0, x_1) \cdots p(x_{n-1}, x_n) \\
&\quad = \; p'(x_1, x_0) \frac{\pi(x_1)}{\pi(x_0)} \cdots p'(x_n, x_{n-1}) \frac{\pi(x_n)}{\pi(x_{n-1})} \\
&\quad = \; \mathbb{P}(Y_0 = x_n, \ldots, Y_n = x_0) \, \frac{\pi(x_n)}{\pi(x)} \; .
\end{aligned}$$

Consequently, we have

$$\gamma_x = \mathbb{E} I_{D_x} = \mathbb{E} I_{D'_x} L,$$

where $L = \pi(Y_0)/\pi(x)$. L can be view. Namely, it is the quotient of the likelihood of a forward path of the X process and the likelihood of the corresponding backward path for the Y process. The expression above gives rise to a simulation scheme which will estimate γ_x for all x in E as follows:

1. Generate N samples of Y; run each sample until B is hit.
2. If the ith sample ($i = 1, \ldots, N$) hits x before B and A, let W_i be equal to the likelihood ratio of the ith sample (as defined by L above) or else let $W_i = 0$.
3. Estimate γ_x by the mean

$$\widehat{\gamma_x} = N^{-1} \sum_{i=1}^{N} W_i \ .$$

4. Use this estimator to obtain the IF value for x.

The whole procedure gives fast estimates for the IF in the Jackson networks we simulate in section 5.

Distribution of Y_0

The last problem we have in the algorithm presented is that we do not always know where to start, i.e., what to choose for the starting state in a sample path of Y. If we define T as the first time at which X hits set A, and D_x as the event that X hits A before B, starting from x, then the distribution of Y_0 should by definition be equal to the conditional distribution of X_T *given* the event D_x. Any failure to comply to the actual entrance distribution will lead to a bias in the generated estimates. We usually do not know the real distribution of this variable since this is hard to obtain in general.

As an approximation we propose to choose Y_0 from the steady state probabilities of the border of A, ∂A say, as follows:

$$\mathbb{P}(Y_0 = k) = \frac{\pi(k)}{\pi(\partial A)}, k \in \partial A \ .$$

Another possibility is to generate actual samples of X_{N_A} and use these in a bootstrapping manner. Such samples may be obtained using any splitting method, even for inefficient splitting methods we can usually obtain a few hits of the rare event set.

The fact that this entrance distribution is unknown in general makes the algorithm unfit for estimation of γ in the pilot run because an unknown bias will be introduced by sampling from another distribution than the true entrance distribution.

We will try to find out how close this heuristic is to the optimal function for a number of systems in the following section.

5 NUMERICAL EXPERIMENTS

In this section we present a number of numerical experiments, illustrating the use of the Importance Function. Our focus will be on the 2- and 3-node tandem queue. The 2-node tandem queue, in particular, has served as a convenient reference model for rare event simulation, see e.g., [14, 15, 16, 17, 18, 2, 19, 20, 11, 10]. For this system, rare events probabilities such as the overflow of the second queue or the overflow of the total populaton have proved to be difficult to estimate, both using importance sampling and the splitting method (see e.g. [2, 19, 14]).

The simulations were carried out on a Sun system running the Solaris operating system version 2.6 and equipped with 6 UltraSparcII 336 MHz processors online and 3 GB of RAM.

In all tables γ denotes the rare event probability of interest. The estimate of γ is given by $\hat{\gamma}$. For each $\hat{\gamma}$ the corresponding estimate of the *Relative Error* (RE) is included. As a measure of the efficiency of the estimator $\hat{\gamma}$ we use the *Relative Time Variance product* (RTV), which we define as the simulation time (in seconds) multiplied by squared (estimate of the) relative error of $\hat{\gamma}$. Notice that the RTV is equivalent to the 'work-balanced variance" used in [21]. Once a stable estimate of the variance is reached, the RTV becomes constant. This constant is smaller for more efficient simulation schemes. Practically, if scheme 1 gives a RTV which is half that of scheme 2, it would take twice as long to estimate γ within a certain accuracy via scheme 2 than via scheme 1. We introduce the gain of using scheme 1 rather than using scheme 2 as

$$\text{Gain}(\widehat{\gamma}_1, \widehat{\gamma}_2) = \frac{\text{RTV}_2}{\text{RTV}_1} = \frac{\text{RE}_2^2 t_2}{\text{RE}_1^2 t_1} \approx \frac{\mathbb{V}\text{ar}(\widehat{\gamma}_2) t_2}{\mathbb{V}\text{ar}(\widehat{\gamma}_1) t_1}$$

and it is clear that the gain is now equal to the speed-up ratio in simulation time t_2/t_1 for achieving a fixed variance when using estimator $\widehat{\gamma}_1$ instead of $\widehat{\gamma}_2$.

The number of simulation runs in each example is typically 10^7, all other splitting parameters are chosen in an optimal fashion by a pilot run.

5.1 TWO-NODE TANDEM QUEUE

Consider the following 2-node tandem queue, see figure 3. Customers arrive at the first buffer according to a Poisson process with parameter λ. The first buffer has a fixed capacity C, the second buffer has infinite capacity. The servers operate at Poisson rates μ_1 and μ_2 respectively.

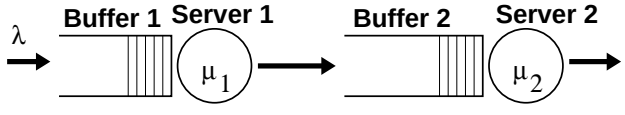

Figure 3: *The 2-node tandem queue*

The underlying Markov process is given by $X = (X_t, t > 0)$ where $X_t = (X_{1,t}, X_{2,t})$, in which the stochastic variable $X_{i,t}$ denotes the number of clients in buffer number i at time t. We wish to estimate the probability γ that the second buffer fills up to at least level L before it empties.

In our simulations we use $X_0 = (1, 0)$, $C = 40, \lambda = 1$, and $\mu_2 = 3$, leaving μ_1 as a variable.

Anantharam et al. [15] look specifically at this system and derive the limiting "flow" that the successful paths (paths that lead to a buffer overflow) take through reverse–time analysis. The results show that a typical path with the first buffer being the bottleneck, leading to overflow of the second buffer, starts with queue 1 building up on its own, and then emptying until buffer 2 hits the overflow level and buffer 1 is empty.

Generating a sample path that has these characteristics is very hard to do in splitting, and it is, at first, not clear what the optimal IF will look like. Fortunately, it is not too difficult to actually *calculate* the probability γ_x in (7), for all x, and hence find the optimal IF. For a detailed description of how to the evaluate γ_x, we refer to [22, 4]. Of course, in this case we can also find the overflow probability γ itself by substituting the starting state.

In figure 4 we depict the optimal IF graphically; the contour levels for multiples of five of the IF only are drawn for clarity. These represent the optimal thresholds for the splitting method. Note that the "naive" IF would have given vertical contour lines.

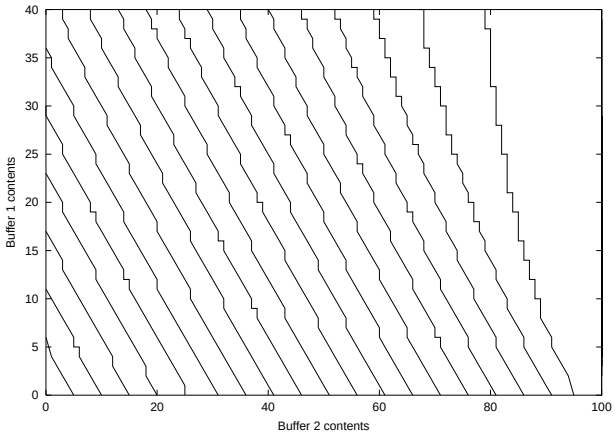

Figure 4: *The optimal IF for the two-node tandem queue.*

In Table 1, we compate the efficiency of the splitting method using the naive IF and the numerically obtained optimal IF.

To see what the effect is of *estimating* the IF via the reverse-time method, we repeated the experiment by using the estimated optimal IF instead of the true optimal IF. The results are given in Table 2.

We interpret these results by looking at the efficiency gain which is the quotient of the RTV's. The efficiency gain is then equal to the ratio of the time needed by method b to reach a certain fixed accuracy and the time needed by method a to reach that same accuracy. It is thus the simulation time speedup factor of method a over method b. We see that the efficiency gain of the numerically obtained optimal IF compared to the standard splitting typically lies between 2 to 200 for the simulated tandem queue. Comparing the proposed time-reversal method for the IF and the standard splitting we observe gains ranging from 2 to about 100, and the gain using the numerically obtained IF over the IF obtained by the time-reversal method is also usually between 2 and 10. This causes us to believe that the proposed estimation method works well but could be improved upon to obtain the optimal gain. One such method could be the diversion of more simulation effort towards the estimation of the optimal IF; in our test beds this has typically been 10% of the total effort. Comparing the optimal IF to the IS results, we see that the IS gain over the optimal splitting typically ranges from 1 to 10, indicating that this IS approach does a better job. Note however that the RTV seems to grow linearly in the buffer capacity for all cases, indicating that the complexity properties of both methods are equal. A last conclusion is that the RTV decreases dramatically for the IS method for an increasing μ_1, whereas the splitting RTV is very constant for all the optimal splitting implementations. We expected this behaviour for the splitting method as in all cases the optimal efficiency is obtained; the remaining variability is caused by the rounding effects of the thresholds and the different costs of simulation for different μ_1. This insensitivity of the efficiency of the optimal splitting method with respect to the model parameters can be seen as a sign of robustness of the optimal splitting method, as defined in [23] page 5.

Table 1: *Simulation results for the two-node tandem queue using the optimal IF obtained numerically.*

		Optimal IF splitting			Naive splitting		
μ_1	N	$\hat{\gamma}$	RE	RTV	$\hat{\gamma}$	RE	RTV
2	50	1.009e-26	3.0e-2	6.0e+0	1.166e-26	2.0e-1	2.8e+3
2	75	1.399e-39	3.2e-2	1.2e+1	1.417e-39	1.5e-1	2.7e+3
2	100	2.123e-52	5.6e-2	5.1e+1	2.383e-52	3.9e-1	3.5e+4
3	50	7.145e-25	1.6e-2	1.3e+0	7.035-25	4.0e-2	1.5e+2
3	75	6.951e-37	2.0e-2	2.5e+0	1.023e-36	3.5e-1	2.3e+4
3	100	7.217e-49	2.9e-2	6.0e+0	5.541e-49	1.0e-1	1.9e+3
4	50	1.772e-24	1.1e-2	3.5e-1	1.785e-24	2.7e-2	2.1e+1
4	75	2.070e-36	1.4e-2	1.0e+0	2.005e-36	8.4e-2	4.5e+2
4	100	2.494e-48	2.0e-2	2.4e+0	2.165e-48	4.0e-2	7.8e+1

Table 2: *Simulation results for the two-node tandem queue using the estimated IF obtained by time-reversal.*

		Estimated Optimal IF splitting		
μ_1	L	$\hat{\gamma}$	RE	RTV
2	50	1.050e-26	2.7e-2	1.5e+1
2	75	1.510e-39	1.1e-1	4.0e+2
2	100	2.669e-52	1.0e-1	5.0e+2
3	50	7.185e-25	1.1e-2	2.7e+0
3	75	7.208e-37	2.4e-2	2.0e+1
3	100	7.744e-49	6.2e-2	1.4e+2
4	50	1.786e-24	5.5e-3	6.4e-1
4	75	2.134e-36	1.3e-2	7.8e+0
4	100	2.635e-48	2.1e-2	1.1e+1

5.2 THREE-NODE TANDEM QUEUE

Consider the following 3-node tandem queue, as depicted in figure 5. Customers arrive at the first buffer according to a Poisson process with parameter λ. The buffer capacities for the first, second and third buffer are C_1, C_2 and ∞, respectively. The first and second server operate at Poisson rates μ_1 and μ_2, respectively.

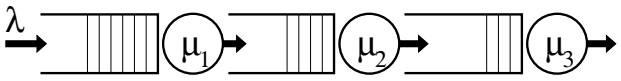

Figure 5: *The 3-node tandem queue.*

The underlying Markov process in this case is $X = (X_t, t > 0)$ with a three-dimensional state space: $X_t = (X_{1,t}, X_{2,t}, X_{3,t})$ in which again $X_{i,t}$ denotes the number of clients present in queue number i at time t. We wish to estimate the probability γ that the third buffer fills up to at least L customers before it empties out.

In our simulation we use the following values for the parameters: starting state $X_0 = (1, 1, 0)$, buffer capacities $C_1 = 40$ and $C_2 = 20$, arrival rate $\lambda = 1$ and service rates $\mu_1 = 2, \mu_2 = 2$ and $\mu_3 = 4$.

Unlike the 2-node case, the optimal IF is difficult, if not impossible to calculate. But, we can still estimate the optimal IF via the time-reverse method. However, the state space for this model is much bigger than the previous model, which results in a greater variance of the estimates of the IF. This increased variance in turn can cause the level process $(Z_t := f(X_t), t \geq 0)$ to jump erratically, which can create a bias in the estimate. In order to overcome this effect we used a smoothing technique that reduces the variance in the IF estimates, for more details see [4].

The results of the simulations carried out with this system are given in Table 3. From these simulations we see

Table 3: *Simulation results for the 3-node tandem queue using the proposed time-reversal method for the IF.*

	Estimated Optimal IS splitting			Naive IF splitting		
L	$\hat{\gamma}$	RE	RTV	$\hat{\gamma}$	RE	RTV
10	1.188e-7	4.2e-3	5.4e-2	1.182e-7	5.2e-3	4.3e-1
15	2.338e-11	2.9e-2	3.4e-1	2.338e-11	2.9e-2	2.8e+1
20	5.310e-15	2.0e-2	3.8e+0	5.481e-15	1.6e-1	1.4e+3

that the new method generates more efficient results, the efficiency gain ranges between 10 and 100, just as in the previous simulations with the two-node tandem queue. Typically the gain increases with increased rarity of the rare event set. This behaviour is exactly what we aimed for, since we devised our method especially for those extremely small probabilities. Note that the last result ($L = 20$, standard splitting) does not produce reliable results because the RTV is so large.

6 CONCLUSIONS

In this paper we have looked at the optimisation of the splitting method with respect to the decision of when to split. This is done by choosing an Importance Function that gives a certain weight to every system state. The exact optimum is hard to find, since it involves exact knowledge of the quantities to be estimated. A time-reversal method is proposed and evaluated that works well for Jackson networks; typical efficiency gains ranging from 2 to 200 are achieved with this method. The methodology is simple and adaptable to a broader range of systems; other heuristics for estimating a good IF may even perform better. Future research will need to determine such heuristics.

Knowledge about the behaviour of the system leading to the rare event is necessary for finding the optimal IF. Interestingly, the same information is necessary when implementing an optimal Importance Sampling strategy. The optimal IF will change the complexity properties from a worst case scenario of Monte Carlo complexity of $O(\gamma^{-1})$ (in the case of extremely high dependencies), to a complexity of $O((-\log(\gamma))^2)$ (in the case of independence, see [3]), making the efficiency of the splitting method a comparable to Importance Sampling . The robustness of the splitting simulation method combined with its complexity properties make it a very attractive rare event simulation method.

ACKNOWLEDGEMENT

This research was partly supported by the Australian Research Council (ARC), grant number DP0209921.

REFERENCES

[1] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. A look at multilevel splitting. In H. Niederreiter, editor, *Monte Carlo and Quasi Monte Carlo Methods 1996, Lecture Notes in Statistics*, volume 127, pages 99–108. Springer Verlag, 1996.

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

[3] M.J.J. Garvels and D.P. Kroese. A comparison of RESTART implementations. In *Proceedings of the 1998 Winter Simulation Conference*, pages 601–609, Washington, DC, 1998.

[4] M. Garvels. *The splitting method in rare event simulation*. PhD thesis, University of Twente, The Netherlands, October 2000.

[5] M. Villén-Altamirano and J. Villén-Altamirano. RESTART: A method for accelerating rare event simulations. In J.W. Cohen and C.D. Pack, editors, *Proceedings of the 13th International Teletraffic Congress, Queueing, Performance and Control in ATM*, pages 71–76, 1991.

[6] M. Villén-Altamirano, A. Martínez-Marrón, J. Gamo, and F. Fernández-Cuesta. Enhancement of the accelerated simulation method RESTART by considering multiple thresholds. In J. Labetouille and J.W. Roberts, editors, *Proceedings of the 14th International Teletraffic Congress, The Fundamental Role Of Teletraffic in the Evolution of Telecommunications Networks*, pages 797–810, 1994.

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

[8] C. Görg and O. Fuß. Comparison and optimization of RESTART run time strategies. *AEÜ*, 52(3):197–204, 1998.

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

[10] M. Villén-Altamirano and J. Villén-Altamirano. About the efficiency of RESTART. In *Proceedings of the RESIM'99 Workshop*, pages 99–128. University of Twente, the Netherlands, 1999.

[11] A.A. Akyamac, Z. Haraszti, and J.K. Townsend. Efficient rare event simulation using DPR for multi-dimensional parameter spaces. In *Proceedings of the 16th ITC*, pages 767–776. Elsevier, 1999.

[12] Z. Haraszti and J. Townsend. Rare event simulation of delay in packet switching networks using DPR-based splitting. In *Proceedings of the RESIM Workshop, 11-12 March 1999*, pages 185–190. University of Twente, the Netherlands, 1999.

[13] F.P. Kelly. *Reversibility and Stochastic Networks*. Wiley Statistics Series, 1979.

[14] P. Glasserman and S-G. Kou. Analysis of an importance sampling estimator for tandem queues. *ACM Transactions of Modeling and Computer Simulation*, 5(1):22–42, 1995.

[15] V. Anantharam, P. Heidelberger, and P. Tsoucas. Analysis of rare events in continuous time Markov chains via time reversal and fluid approximation. *IBM Research Report*, 16280, 1990.

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

[17] M.R. Frater, T.M. Lenon, and B.D.O. Anderson. Optimally efficient estimation of the statistics of rare events in queueing networks. *IEEE Trans. Autom. Control*, 36:1395–1405, 1991.

[18] P. Tsoucas. Rare events in series of queues. *J. Appl. Probab.*, 29:168–175, 1992.

[19] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic. Splitting for rare event simulation: Analysis of simple cases. In *Proceedings of the 1996 Winter Simulation Conference*, pages 302–308, San Diego, California, 1996. IEEE Computer Society Press.

[20] D.P. Kroese and V.F. Nicola. Efficient simulation of a tandem Jackson network. *ACM Transactions of Modeling and Computer Simulation*, 2002. To appear.

[21] P.W. Glynn and W. Whitt. The asymptotic efficiency of simulation estimators. *Operations Research*, 40:505–520, 1992.

[22] M.J.J. Garvels and D.P. Kroese. On the entrance distribution in RESTART simulation. In *Proceedings of the RESIM'99 workshop*, pages 65–88. University of Twente, 11-12 March 1999, 1999.

[23] Peter J. Huber. *Robust Statistics*. John Wiley and Sons, Inc., 1981.