A COMBINATION OF RANDOMIZED QUASI-MONTE CARLO WITH SPLITTING FOR RARE-EVENT SIMULATION

Valérie Demers and Pierre L'Ecuyer
Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville
Montréal, H3C 3J7, CANADA
E-mail:{demersv,lecuyer}@iro.umontreal.ca

Bruno Tuffin
IRISA-INRIA
Campus Universitaire de Beaulieu
35042 Rennes Cedex, FRANCE
E-mail: Bruno.Tuffin@irisa.fr

KEYWORDS

Monte Carlo, Markov chain, Variance reduction

ABSTRACT

The splitting method is one of the primary approaches to make important rare events happen more frequently in a simulation and yet recover an unbiased estimator of the target performance measure, in the context where this performance measure is highly influenced by the rare event. In many rare-event situations, simulation is impractical (because the estimators are much too noisy) unless such a method is used. Randomized quasi-Monte Carlo (RQMC) is another class of methods for reducing the noise of simulation estimators, by sampling more evenly than with standard Monte Carlo (MC). It typically works well for simulations that depend mostly on very few random numbers. In splitting, on the other hand, we simulate Markov chains whose sample paths are usually a function of a long sequence of independent random numbers generated during the simulation. In this paper, we show how a new RQMC technique called *array-RQMC* can be used together with splitting to obtain estimators with smaller variance than what can be obtained by either of the two methods alone, and discuss the difficulties that have to be tackled to further increase the efficiency. We do that in a setting where the goal is to estimate the probability of reaching a given set B before returning to the set A when starting from state $x_0 \in A$, where A and B are two disjoint subsets of the state space and B is very rarely reached. This problem has several practical applications.

1 INTRODUCTION

We consider a discrete-time Markov chain $\{X_j, j \geq 0\}$ with arbitrary state space $\mathscr{X}$. Let A and B be two disjoint subsets of $\mathscr{X}$ and let $x_0 \in A$, the initial state. The chain starts in state $X_0 = x_0$, eventually leaves the set A, and then may eventually reach B or return to A. Suppose the first exit time from A is at time 0 (this is when we start counting time). Let $\tau_A = \inf\{j > 0 : X_j \in A\}$, the first time when the chain returns to A after leaving it, and $\tau_B = \inf\{j > 0 : X_j \in B\}$, the first time when the chain reaches the set B. The goal is to estimate μ, the probability that the chain reaches B before it returns to A, i.e. $\mu = \mathbb{P}[\tau_B < \tau_A]$. This probability is assumed to be very small, e.g., 10^{-10} or even less.

This problem occurs in many practical situations; see, e.g., Nicola, Nakayama, Heidelberger, and Goyal (1991), Goyal, Shahabuddin, Heidelberger, Nicola, and Glynn (1992), Heidelberger (1995). For example, suppose we want to estimate the expected time until failure for a complex multicomponent system whose initial state is "new". Components fail once in a while and are replaced by new ones after some random delay. When the set of working components satisfies certain conditions, the system is operational, otherwise it is in the *failed* state. Let $A = \{x_0\}$, the set that contains only the "new" state, and let B be the set of failed states. Suppose we are interested in estimating $\mathbb{E}[\tau_B]$, the expected time until failure for a new system. By a standard argument (Goyal, Shahabuddin, Heidelberger, Nicola, and Glynn 1992), we have

$$\mathbb{E}[\tau_B] = \mathbb{E}[\min(\tau_A, \tau_B)]/\mu.$$

In this expression, $\mathbb{E}[\min(\tau_A, \tau_B)]$ is easy to estimate by standard simulation, but μ is often very difficult to estimate because it is very small. For example, if $\mu = 10^{-10}$ and we do straightforward simulations to estimate it, by running n copies of the chain up to the stopping time $\tau = \min[\tau_A, \tau_B]$, we must take $n = 10^{12}$ (a huge number) to be able to expect that the event $\{\tau_B < \tau_A\}$ occurs about 100 times. For $n < 10^{10}$, we are likely to observe *no single occurrence* of this event, in which case the estimator of μ takes the value 0 and is rather useless.

A similar problem occurs in a queueing system when we want to estimate the expected time until the number of customers in the queue exceeds a given number (Parekh and Walrand 1989, Sadowsky 1991). For example, the customers

could be packets in a telecommunication network, the number to exceed could be the size of the buffer used to store the packets waiting to be transfered at a communication switch, the set B could be the set of states for which the buffer overflows, and A would be the states where the buffer is empty. Then, $\mathbb{E}[\tau_B]$ represents (roughly) the average time between buffer overflows and μ is the probability that the buffer overflows before returning to empty.

The two primary techniques for dealing with rare-event simulation are *importance sampling* and *splitting*. Importance sampling changes the probability laws that drive the evolution of the system, to increase the probability of the rare event, and multiplies the estimator by an appropriate likelihood ratio so that it has the correct expectation (e.g., remains unbiased for μ in the above setting). A major difficulty in general is to find a good way of changing the probability laws. We refer the reader to Glynn and Iglehart (1989), Glynn (1994), Heidelberger (1995), Bucklew (2004) for the details.

In the splitting method, the probability laws of the system remain unchanged, but an artificial drift toward the rare event is created by terminating the trajectories that seem to get away from it and cloning (i.e., *splitting*) those that are going in the right direction. Again, an unbiased estimator is recovered by multiplying the original estimator by an appropriate factor. We give more details in the next section. The method can be traced back to Kahn and Harris (1951) and has been studied by several authors, including Bayes (1972), Villén-Altamirano and Villén-Altamirano (1991), Villén-Altamirano and Villén-Altamirano (1994), Garvels and Kroese (1998), Glasserman, Heidelberger, Shahabuddin, and Zajic (1998), Glasserman, Heidelberger, Shahabuddin, and Zajic (1999), Garvels (2000).

In this paper, we concentrate on the splitting method and examine how it can be combined with *randomized quasi-Monte Carlo* (RQMC) to further reduce the variance. L'Ecuyer, Lécot, and Tuffin (2005) recently proposed a new RQMC approach called *array-RQMC*, based on earlier work by Lécot and Tuffin (2004), and designed primarily for Markov chains having a totally ordered state space and which evolve for a large number of steps. At first sight, this method seems to be highly compatible with splitting. The goal of this paper is to examine the degree of improvement obtained by their combination, as well as the difficulties encountered and which have to be tackled to obtain an additional gain.

The remainder of the paper is organized as follows. In the next section, we recall the main principles of splitting in the setting where we want to estimate $\mu = \mathbb{P}[\tau_B < \tau_A]$. In Section 3, we describe the array-RQMC method and how it can be implemented in our setting. We additionally discuss the potential difficulties of the method. Numerical illustrations are given in Section 4 with two examples: firstly an Ornstein-Uhlenbeck (mean-reverting) process for which B is the set of states that exceed a given threshold, and secondly a tandem queue where B is the set of states where the number of customers waiting at the second queue exceeds a given value. We finally conclude and provide hints for future research in Section 5.

2 MULTILEVEL SPLITTING

To define the splitting algorithm, it is convenient to introduce a function $h : \mathscr{X} \to \mathbb{R}$ that assigns a real number to each state of the chain. This h is called the *importance function* (Garvels 2000, Garvels, Kroese, and Van Ommeren 2002). Define the real-valued process $\{Z_j = h(X_j), j \geq 0\}$. We assume that $A = \{x \in \mathscr{X} : h(x) \leq 0\}$ and $B = \{x \in \mathscr{X} : h(x) \geq L\}$ for some constant $L > 0$. In the multilevel splitting method, we partition the interval $[0, L)$ into m subintervals with boundaries $0 = L_0 < L_1 < \cdots < L_m = L$. For $k = 1, \ldots, m$, define $T_k = \inf\{j > 0 : Z_j \geq L_k\}$, let $D_k = \{T_k < \tau_A\}$ denote the event that the process Z reaches level L_k before returning to level 0, and define the conditional probabilities $p_k = \mathbb{P}[D_k \mid D_{k-1}]$ for $k > 1$, and $p_1 = \mathbb{P}[D_1]$. Since $D_m \subset D_{m-1} \subset \cdots \subset D_1$, we see immediately that

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

The basic idea of splitting is to estimate each probability p_k "separately", by starting a large number of chains in states that are generated from the distribution of $X_{T_{k-1}}$ conditional on the event D_{k-1}. This conditional distribution is called the *entrance distribution at threshold* L_{k-1} and we shall denote it by G_{k-1}.

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

In the second stage, we start N_1 chains from these R_1 entrance states, by cloning (splitting) some chains if we want $N_1 > R_1$, and continue the simulation of these chains independently up to time $\min(\tau_A, T_2)$. Then $\hat{p}_2 = R_2/N_1$ is an unbiased estimator of p_2, where R_2 is the number of those chains for which D_2 occurs. This procedure is repeated at each stage. In stage k, we pick N_{k-1} states out of the R_{k-1} that are available (by cloning if necessary), simulate independently from these states up to time $\min(\tau_A, T_k)$, and estimate p_k by $\hat{p}_k = R_k/N_{k-1}$ where R_k is the number of chains for which D_k occurs.

Even though the $\hat{p}_k$'s are not independent, it turns out that

the product $\hat{p}_1 \cdots \hat{p}_m = (R_1/N_0)(R_2/N_1)\cdots(R_m/N_{m-1})$ is an unbiased estimator of μ (Garvels 2000).

There are many ways of doing the splitting (Garvels 2000). For example, one may clone each of the R_k chains that reached level k in c_k copies for a fixed integer c_k, in which case $N_k = c_k R_k$ is random. This is called *fixed splitting*. In contrast, in the *fixed effort* method, we fix a priori each value of N_k and make just the right amount of splitting to reach this target value. One way of doing this is by sampling the N_k starting states at random, with replacement, from the R_k available states. This is called *random assignment* and is equivalent to sampling from the empirical distribution of the states. In a *fixed assignment*, on the other hand, we would split each of the R_k states the same number of times (or approximately the same number of times, in the case where N_k is not a multiple of R_k). In practice, the fixed effort method tends to perform better, because it reduces the variance of the number of chains that are simulated at any given stage, and we prefer a fixed assignment strategy to a random assignment because it amounts to using stratified sampling over the empirical distribution, and thus typically reduces the variance.

Under a number of simplifying assumptions (e.g., that $\mathbb{P}[D_k \mid D_{k-1}, X_{T_{k-1}} = x]$ does not depend on x) and for the fixed splitting setting, it has been shown (Villén-Altamirano, Martinez-Marrón, Gamo, and Fernández-Cuesta 1994, Garvels and Kroese 1998) that the efficiency of the splitting method is maximized by selecting the thresholds so that $p_k \approx e^{-2} \approx 0.135$ and $\mathbb{E}[N_k] = N_0$ for each k. This gives $m \approx -(\ln \mu)/2$ stages. However, these simplifying assumption typically do not hold, so these results only give guidelines, and more importantly the p_k's are unknown in practice and selecting the appropriate threshold may be difficult. Moreover, the choice of the importance function h may have a large impact on the performance of the method and is not trivial (Garvels, Kroese, and Van Ommeren 2002).

3 ARRAY-RQMC

3.1 Array-RQMC for simulating Markov chains

Array-RQMC is a simulation method recently designed by L'Ecuyer, Lécot, and Tuffin (2005) to simulate a Markov chain $\{X_j,\ j \geq 0\}$ defined by some distribution ν_0 for the initial state X_0 and a stochastic recurrence

$$X_{j+1} = \varphi(X_j, \mathbf{U}_j) \tag{1}$$

where the $\mathbf{U}_j$ are independent random vectors uniformly distributed over $[0,1)^d$. The method assumes that the state space $\mathscr{X}$ is totally ordered. It simulates N copies of the Markov chain in parallel, using at each step of the chain a so-called *highly-uniform point set*, which contains N points that are more evenly distributed in the unit hypercube than typical random points. This induces a negative correlation among the copies of the chains, resulting in a better approximation of the probability distribution of X_j than with standard Monte Carlo (MC), for each j, and consequently a variance reduction of the performance estimator of interest.

The basic idea is to simulate the N chains in parallel as follows. The N initial states $X_{i,0}$, $i = 0, \ldots, N$, are generated from the initial distribution ν_0, using an RQMC point set $P_{N,0} = \{\mathbf{u}_{0,0}, \ldots, \mathbf{u}_{N-1,0}\}$ in $[0,1)^{d_0}$, where $\mathbf{u}_{i,0}$ is used to generate $X_{i,0}$, assuming that at most d_0 uniform random numbers are required to generate the initial state. The N chains are then sorted in increasing order of their state, to get the empirical distribution function of X_1. The states at the next time step are selected from the previously sorted ones. We assume that d uniforms are necessary to generate a transition according to the recurrence (1). An RQMC point set $P_{N,1} = \{\mathbf{u}_{0,1}, \ldots, \mathbf{u}_{N-1,1}\}$ in $[0,1)^d$, randomized independently from the previous one, is used, where $\mathbf{u}_{i,1}$ serves to generate $X_{i,1}$ from $X_{i,0}$. The chains are sorted again according to their state, and this process is repeated at successive steps with independent RQMC point sets until all chains have reached their stopping times. At each step, only the chains that have not yet reached their stopping times are considered and sorted; the other chains are ignored and for convenience their state is assumed to be ∞ in the algorithm.

For intuitive justifications, additional details, and illustrations of the degrees of improvement that are obtained in practice, the reader is referred to L'Ecuyer, Lécot, and Tuffin (2005). A confidence interval is easily obtained by considering independent replications (i.e., randomizations) of groups of N chains. For a one-dimensional state space and under certain additional conditions, these authors have shown that the array-RQMC technique converges in $O(N^{-1/2})$ in the *worst case*. In contrast, the MC method converges in $O(N^{-1/2})$ in the (probabilistic) sense that the width of a confidence interval converges at this rate. L'Ecuyer, Lécot, and Tuffin (2005) have also shown that for a special variant of array-RQMC, under certain assumptions, the *variance* converges to zero as $O(N^{-3/2})$.

3.2 Array-RQMC combined with splitting

In order to apply array-RQMC to the splitting approach, an adaptation is required. The probabilities p_k, $k = 1, \cdots, m$, are estimated one after the other. Since μ is the product of the p_k's, the principle is to successively estimate each p_k by the array-RQMC method previously described, and to use the product of estimators as the overall estimator, as realized in the standard splitting methodology. Let $X^{(k)}$ denote the Markov chain $\{X_j,\ j \geq 0\}$ between times T_{k-1} and $\min(\tau_A, T_k)$.

We start with N_0 chains. We first estimate p_1 by using the

array-RQMC algorithm for the Markov chain $X^{(1)}$: the N_0 chains are simulated in parallel according to (1) and sorted after each time step. Each chain evolves until the stopping time $\min(T_1, \tau_A)$. If R_1 is the number of chains for which D_1 occurs, R_1/N_0 is an unbiased estimator of p_1. The states of these R_1 chains are stored and their empirical distribution is used as an unbiased estimator of the distribution of X_{T_1}.

At the second level, N_1 chains are started from those R_1 states according to one of the aforementioned splitting policies, and are simulated in parallel using the same array-RQMC procedure on $X^{(2)}$, each chain being simulated until its stopping time $\min(T_2, \tau_A)$. The probability p_2 is estimated by R_2/N_1 where R_2 is the number of chains for which D_2 occurs. These R_2 chains are then split again, and so on, until all the probabilities $p_3, \cdots, p_m$ have been estimated. As in L'Ecuyer, Lécot, and Tuffin (2005), it can be readily verified that the estimator of each p_k is unbiased.

The algorithm is described in Figure 1. It basically consists in adding a loop to the algorithm of L'Ecuyer, Lécot, and Tuffin (2005) for the estimation of the probability p_k of reaching each successive level. For simplicity, we assume that $d_0 = d$ and consider a single replication. This entire procedure must be repeated using independent randomizations to get a confidence interval.

This algorithm is not very complicated in principle. However, several practical issues must be addressed for the combination of splitting and array-RQMC to be really effective. These difficulties include the following.

1. Recall that the number of time steps before reaching the next level or coming back to A is random and may have large variability, especially when k is large, because the chain then starts farther from A and requires a larger number of steps to come back to A. For the array-RQMC method, this causes the number of points used at each step of the chain to decrease with the step number, within a given splitting level, thus reducing the RQMC efficiency because only a few points from the RQMC point set are used in the later steps.

2. The empirical distribution of the entrance states in D_k tends to deteriorate (as an approximation of the exact distribution) as k increases, due the fact that it is derived from the empirical distribution at the previous level, so the approximation error accumulates from level to level. As a result, the variance reduction from array-RQMC is expected to decrease with k.

3. In the case of multidimensional state spaces, both the choice of the importance function for selecting the splitting levels and the ordering of the states are (related) non-trivial issues (Garvels, Kroese, and Van Ommeren 2002, L'Ecuyer, Lécot, and Tuffin 2005). In general, the states could be ordered by the value of the importance function, in which case its choice has a double impact.

4. The optimal number of levels to minimize the variance of the estimator, derived by Villén-Altamirano, Martinez-Marrón, Gamo, and Fernández-Cuesta (1994), Garvels and Kroese (1998), is valid only for the *fixed splitting* algorithm and under additional conditions. This optimal number may differ significantly for the *fixed effort* method, used in our experiments in the next section. Our empirical investigations indicate that it tends to be larger. The optimal number of chains to simulate at each level may also differ from level to level. Finding these (jointly) optimal numbers is not necessarily easy. Fortunately, rough approximations may suffice because the variance is often not very sensible to small changes in these numbers.

5. RQMC has been proved to be asymptotically more effective than MC only when the integrand is a *smooth* function (Owen 1998, L'Ecuyer and Lemieux 2002). But here, we estimate the probability p_k of reaching the next level by an average of indicator functions, for each k. Indicator functions are definitely not smooth, so it is unclear a priori if array-RQMC can bring any improvement, even asymptotically.

In the next section, we show that despite these difficulties, the combination can still bring significant variance reductions. We also discuss specific ways of tackling them for our examples. Future research on how to better address these issues should lead to further efficiency improvements.

4 EXAMPLES

The following examples are toy problems, for which the exact answer is known beforehand. They are used to illustrate and evaluate the performance of the proposed method.

4.1 The Ornstein-Uhlenbeck Process

The Ornstein-Uhlenbeck process is a continuous-time stochastic process $\{R(t), t \geq 0\}$ that obeys the stochastic differential equation

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

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

Initialization.
Select m d-dimensional QMC point sets $\tilde{P}_{k,N_{k-1}} = (\tilde{\mathbf{u}}_0, \ldots, \tilde{\mathbf{u}}_{N_{k-1}-1})$, $1 \le k \le m$, and a randomization of each $\tilde{P}_{k,N_{k-1}}$ such that (a) each randomized point is a uniform random variable over $[0,1)^d$ and (b) if $P_{k,N_{k-1}} = (\mathbf{u}_0, \ldots, \mathbf{u}_{N_{k-1}-1})$ denotes the randomized version, then $P'_{k,N_{k-1}} = \{((i+0.5)/N_{k-1}, \mathbf{u}_i), 0 \le i < N_{k-1}\}$ is "highly uniform" in $[0,1)^{d+1}$.
Select the m thresholds $0 = L_0 < L_1 < \cdots < L_m = L$.

Estimate each p_k.
For ($k = 1$; $k <= m$; k++)
Simulate in parallel N_{k-1} copies of the chain, numbered $0, \ldots, N_{k-1}-1$ as follows:
Initialize the chains, according to the initial distribution if $k = 1$, or according to the splitting policy and the R_{k-1} states if $k > 1$.
Sort the chains according to their state.
For ($j = 1$; $X^{(k)}_{0,j-1} < \infty$; j++)
Randomize $\tilde{P}_{k,N_{k-1}}$ afresh into $P_{k,N_{k-1}} = \{\mathbf{u}_0, \ldots, \mathbf{u}_{N_{k-1}-1}\}$;
For ($i = 0$; $i < N_{k-1}$ and $X^{(k)}_{i,j-1} < \infty$; i++)
$X^{(k)}_{i,j} = \varphi(X^{(k)}_{i,j-1}, \mathbf{u}_i)$;
Sort (and renumber) the chains for which $X^{(k)}_{i,j} < \infty$ by increasing order of their states. The empirical distribution of the sorted states $X^{(k)}_{0,j}, \ldots, X^{(k)}_{N_{k-1}-1,j}$ provides an estimator of the distribution of $X^{(k)}_j$.

Output.
Return $\prod_{k=1}^{m} R_k/N_{k-1}$ as an estimator of μ.

Figure 1: Combined array-RQMC/splitting algorithm

is high and attracted upward when it is low. The constant σ indicates the strength of the noise.

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

$$\varphi(x_j, U_j) = x_j e^{-a\delta} + \frac{\sigma\sqrt{1-e^{-2a\delta}}}{\sqrt{2a}}\Phi^{-1}(U_j)$$

where Φ is the standard normal distribution function and U_j is uniformly distributed over $[0,1)$.

The levels L_k are defined simply as equidistant thresholds on the value of the state X_j. If we were considering the continuous-time process, the entrance distribution G_k at each level L_k would be degenerate at L_k. But because of the time discretization, the entrance distribution has positive support over the entire interval $[L_k, \infty)$. In this setting, the simulation starts from a fixed state only at the first level. The simulation at level L_k determines $\hat{p}_k$ and $\hat{G}_{k+1}$, which becomes the initial distribution for the simulation at level L_{k+1}. Previous analyzes of splitting algorithms (e.g., Garvels 2000) assume that only one level can be crossed at a time and this condition does not hold for the discrete-time Ornstein-Uhlenbeck process (an arbitrary number of thresholds can be crossed in a single jump). We nevertheless recover an unbiased estimator by considering explicitly the possibility that the chain crosses the next threshold in zero steps.

We ran three types of simulations for this Ornstein-Uhlenbeck model, each one based on a fixed-effort splitting technique. The first one is the standard MC splitting algorithm, used as a reference. The other two are RQMC techniques, namely classical-RQMC and array-RQMC, applied on top of the splitting algorithm. For the *classical-RQMC* method, the N chains are simulated with the N points of an infinite-dimensional RQMC point set. The steps of any given chain use successive coordinates of a fixed point. The points must be infinite-dimensional because the number of steps is random and unbounded. The results presented here were obtained with randomly-shifted Korobov lattice rules with parameters taken from L'Ecuyer and Lemieux (2000), complemented with a baker's transformation (Hickernell 2002). For the array-RQMC algorithm, we used Sobol' digital nets with a random digital shift (L'Ecuyer and Lemieux 2002). The variance for the RQMC methods was estimated by making 30 independent replications of the entire procedure.

The results given here correspond to a time-discretized

Ornstein-Uhlenbeck process with parameters $a = 0.1$, $\sigma = 0.3$, $x_0 = 0.1$, and $\delta = 0.1$. We want to estimate the probability that the discrete-time process $\{X_j = R(t_j),\, j > 0\}$ exceeds $L = 4$ before getting below 0. With these parameters, the formula $-\ln \mu/2$ gives $m = 9$ as the optimal number of splitting levels, but this formula is valid only *for fixed splitting and under additional conditions which are not met here*.

Table 1: Results for the Ornstein-Uhlenbeck model

Method	Mean	Variance	VRF
$N = 2^{10}$, with 8 equidistant levels			
Standard MC	1.8E-8	7.2E-17	
Classical-RQMC	1.6E-8	1.9E-17	3.8
Array-RQMC	1.6E-8	6.9E-18	10.5
$N = 2^{10}$, with 16 equidistant levels			
Standard MC	1.6E-8	1.0E-17	
Classical-RQMC	1.6E-8	2.5E-18	4.0
Array-RQMC	1.6E-8	1.3E-18	7.8
$N = 2^{10}$, with 32 equidistant levels			
Standard MC	1.7E-8	7.1E-18	
Classical-RQMC	1.6E-8	3.0E-18	2.4
Array-RQMC	1.6E-8	1.1E-18	6.7
$N = 2^{12}$, with 8 equidistant levels			
Standard MC	1.6E-8	1.0E-17	
Classical-RQMC	1.6E-8	3.3E-18	3.0
Array-RQMC	1.6E-8	1.7E-18	6.0
$N = 2^{12}$, with 16 equidistant levels			
Standard MC	1.6E-8	2.6E-18	
Classical-RQMC	1.6E-8	8.5E-19	3.0
Array-RQMC	1.6E-8	1.6E-19	16.0
$N = 2^{16}$, with 16 equidistant levels			
Standard MC	1.6E-8	2.0E-19	
Classical-RQMC	1.6E-8	4.1E-20	4.8
Array-RQMC	1.6E-8	5.6E-21	35.0

Table 1 provides results obtained by using $m = 8$, 16, and 32 levels, with $N = 2^{10}$, 2^{12}, and 2^{16} chains simulated at each threshold. In the table, "mean" is the empirical mean (the estimate of μ), "variance" is the empirical variance of the average of all $30N$ chains, and VRF stands for the variance reduction factor with respect to MC with splitting, i.e., the variance with MC divided by the variance with the given RQMC method.

We see that both RQMC methods reduce the variance compared with MC. The classical-RQMC method only brings a modest improvement, by a factor of 3 to 5, which does not seem to increase much when we increase N. With array-RQMC, on the other hand, the VRF clearly increases with N and is quite significant for $N = 2^{16}$.

4.2 Buffer Overflow in a Tandem Queue

As in Parekh and Walrand (1989), Glasserman, Heidelberger, Shahabuddin, and Zajic (1999), Garvels (2000) (among others), we consider an open tandem Jackson queueing network with two queues. The arrival rate at the first queue is $\lambda = 1$ while the mean service time is ρ_i at queue i, for $i = 1, 2$. The events are the arrivals and service completions (at any queue) and $X_j = (X_{1,j}, X_{2,j})$ is the number of customers in each of the two queues immediately after the jth event. The set A contains only the empty state $(0,0)$ and $B = \{(x_1, x_2) : x_2 \geq \ell\}$ for some fixed threshold ℓ, i.e., the set of states for which the length of the second queue is at least ℓ.

Garvels (2000) and Garvels, Kroese, and Van Ommeren (2002) study the application of splitting to this model. A very simple way to define the importance function is by setting its value to the number of customers in the second queue. We call this definition the "second queue function." One drawback of this definition is that it ignores the state of the first queue, neglecting its impact on the "distance" to the set B. This impact can be important, especially when the bottleneck is at the first queue (Garvels 2000).

Our second choice of importance function, which we call "minimal distance function", is defined by

$$h(x_1, x_2) = x_2 + \min(0, x_2 + x_1 - \ell), \tag{2}$$

which equals 2ℓ minus the minimal number of steps required to reach B from the current state. (To reach B, we need at least $\ell - \min(0, x_2 + x_1 - \ell)$ arrivals at the first queue and $\ell - x_2$ transfers to the second queue.)

The two-dimensional state space also means that an ordering of the states must be selected for the array-RQMC algorithm. This order should reflect the "size" of the tandem queue, just like the importance function. (More generally, the problem of choosing the importance function for the splitting algorithm is very similar to the problem of choosing the ordering function for the array-RQMC method.) Here, we will sort the states based on the value of the chosen function h (second queue function, or minimal distance function), and two states with the same value of h are not sorted.

For our numerical experiments with this example, we use the parameter values $\rho_1 = 4$, $\rho_2 = 2$, and $L = \ell = 30$. Table 2 gives the empirical mean, variance, and variance reduction factors of RQMC compared with MC, for the two importance functions defined above. Results are given for $m = 10$ and 20 equidistant levels, with $N = 2^{12}$ and 2^{14} chains for the RQMC methods, for again 30 independent replications of the entire process.

No variance reduction is observed with the first choice of importance function, but some reduction is observed for the second choice. This illustrates the importance of a

Table 2: Results for the tandem queue

Method	Mean	Variance	VRF
$N = 2^{12}$, $m = 10$, second queue function			
Standard MC	1.2E-9	4.7E-20	
Classical-RQMC	1.2E-9	7.1E-20	0.7
Array-RQMC	1.2E-9	5.3E-20	0.9
$N = 2^{12}$, $m = 10$, minimal distance function			
Standard MC	1.2E-9	3.9E-20	
Classical-RQMC	1.3E-9	1.9E-20	2.0
Array-RQMC	1.3E-9	9.6E-21	4.1
$N = 2^{12}$, $m = 20$, minimal distance function			
Standard MC	1.2E-9	2.1E-20	
Classical-RQMC	1.2E-9	4.4E-21	4.8
Array-RQMC	1.2E-9	6.8E-21	3.1
$N = 2^{14}$, $m = 20$, minimal distance function			
Standard MC	1.2E-9	3.1E-21	
Classical-RQMC	1.2E-9	1.5E-21	2.1
Array-RQMC	1.2E-9	7.8E-22	4.0

good selection of this function not only for the effectiveness of the splitting technique, but also for the effectiveness of the RQMC methods. In this example, array-RQMC and classical-RQMC provide comparable variance reductions.

5 CONCLUSION

Splitting is one of the main approaches to efficiently simulate rare events. RQMC techniques, on the other hand, are well known to reduce the variance with respect to MC in certain settings. In this paper, we have examined the combination of the two methods to obtain an increased efficiency, with a special focus on the array-RQMC method recently designed for the simulation of Markov chains. The degree of improvement was illustrated on two examples: an Ornstein-Uhlenbeck model and two queues in tandem. For the first example, a significant variance reduction was obtained compared with splitting alone. A modest reduction was obtained in the second example.

The improvement provided by array-RQMC over MC was not as spectacular as for the single-queue example of L'Ecuyer, Lécot, and Tuffin (2005), where variance reduction factors of several thousands were observed in certain cases. There are several tentative explanations for this, suggesting directions for further improvement of the method. A first possible reason is that the number of time steps between two thresholds is highly variable. Also, the average and variance of the number of steps that need to be simulated at a given splitting level generally increase as we are getting closer to the rare event, because it then takes longer to come back to the absorbing set A. A possible improvement would be to adopt a variant of the splitting technique as in the RESTART algorithm (Villén-Altamirano and Villén-Altamirano 1991, Villén-Altamirano and Villén-Altamirano 1994), where the simulation of most trajectories is stopped whenever the state goes below the lower threshold of the current level, instead of waiting until it reaches A. This would reduce both the average and the variance of the number of steps at a given level. Another important issue that deserves attention is the choice of importance function and of the state ordering for array-RQMC (both are strongly related). Another point worth noticing is that we estimate here a probability by the average of an indicator function, which is generally not the kind of function that favors RQMC methods. A possible improvement would be to replace the indicator function by an estimator based on conditional expectations. We are currently pursuing our investigations in these directions.

ACKNOWLEDGMENTS

This work has been supported by an NSERC-Canada scholarship to the first author, NSERC-Canada grant No. ODGP0110050 and a Canada Research Chair to the second author, and EuroNGI Network of Excellence and SurePath ACI Security Project to the third author.

REFERENCES

Bayes, A. J. 1972. A minimum variance technique for simulation models. *Journal of the ACM* 19:734–741.

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

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

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

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

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

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

Glynn, P. W. 1994. Efficiency improvement techniques. *Annals of Operations Research* 53:175–197.

Glynn, P. W., and D. L. Iglehart. 1989. Importance sampling for stochastic simulations. *Management Sci-*

ence 35:1367–1392.

Goyal, A., P. Shahabuddin, P. Heidelberger, V. F. Nicola, and P. W. Glynn. 1992. A unified framework for simulating markovian models of highly reliable systems. *IEEE Transactions on Computers* C-41:36–51.

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

Hickernell, F. J. 2002. Obtaining $o(n^{-2+\varepsilon})$ convergence for lattice quadrature rules. In *Monte Carlo and Quasi-Monte Carlo Methods 2000*, ed. K.-T. Fang, F. J. Hickernell, and H. Niederreiter, 274–289. Berlin: Springer-Verlag.

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

Lécot, C., and B. Tuffin. 2004. Quasi-Monte Carlo methods for estimating transient measures of discrete time Markov chains. In *Monte Carlo and Quasi-Monte Carlo Methods 2002*, ed. H. Niederreiter, 329–343. Berlin: Springer-Verlag.

L'Ecuyer, P., C. Lécot, and B. Tuffin. 2005. Randomized quasi-Monte Carlo simulation of Markov chains with an ordered state space. In *Monte Carlo and Quasi-Monte Carlo Methods 2004*, ed. H. Niederreiter and D. Talay. To appear.

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

L'Ecuyer, P., and C. Lemieux. 2002. Recent advances in randomized quasi-Monte Carlo methods. In *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications*, ed. M. Dror, P. L'Ecuyer, and F. Szidarovszky, 419–474. Boston: Kluwer Academic Publishers.

Nicola, V. F., M. K. Nakayama, P. Heidelberger, and A. Goyal. 1991. Fast simulation of highly dependable systems with general failure and repair processes. *IEEE Transactions on Computers* 42 (8): 1440–1452.

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

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

Sadowsky, J. S. 1991. Large deviations and efficient simulation of excessive backlogs in a $GI/G/m$ queue. *IEEE Transactions on Automatic Control* AC-36:1383–1394.

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

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

Villén-Altamirano, M., A. Martinez-Marrón, J. Gamo, and F. Fernández-Cuesta. 1994. Enhancement of the accelerated simulation method restart by considering multiple thresholds. In *Proceedings of the 14th International Teletraffic Congress*, 797–810: Elsevier Science.

Villén-Altamirano, M., and J. Villén-Altamirano. 1991. Restart: A method for accelerating rare events simulations. In *Proceedings of the 13th International Teletraffic Congress*, 71–76: North-Holland.

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

AUTHOR'S BIOGRAPHIES

VALÉRIE DEMERS is a PhD student in the Département de Mathématiques et de Statistique at the Université de Montréal, Canada. Her research is on randomized quasi-Monte Carlo methods and other efficiency improvement techniques for simulation.

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

BRUNO TUFFIN received his PhD degree in applied mathematics from the University of Rennes 1 (France) in 1997. Since then, he has been with INRIA (Institut National de Recherche en Informatique et Automatique) at Rennes, France. He also spent 8 months at Duke University in 1999. His research interests include developing Monte Carlo and quasi-Monte Carlo simulation techniques for the performance evaluation of computer and telecommunication systems, and more recently developing new Internet pricing schemes. His e-mail address is <btuffin@irisa.fr>, and his web page is <www.irisa.fr/armor/lesmembres/Tuffin/Tuffin_en.htm>.