A Randomized Quasi-Monte Carlo Simulation Method for Markov Chains

Pierre L'Ecuyer

GERAD and Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal (Québec), H3C 3J7, Canada
lecuyer@iro.umontreal.ca

Christian Lécot

Laboratoire de Mathématiques, Université de Savoie, 73376 Le Bourget-du-Lac Cedex, France,
Christian.Lecot@univ-savoie.fr

Bruno Tuffin

IRISA-INRIA, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France
Bruno.Tuffin@irisa.fr

Abstract. We introduce and study a randomized quasi-Monte Carlo method for estimating the state distribution at each step of a Markov chain, under the assumption that the chain has a totally ordered (discrete or continuous) state space. The number of steps in the chain can be random and unbounded. The method simulates n copies of the chain in parallel, using a $(d+1)$-dimensional low-discrepancy point set of cardinality n, randomized independently at each step, where d is the number of uniform random numbers required at each transition of the Markov chain. This technique is effective in particular to obtain a low-variance unbiased estimator of the expected total cost up to some random stopping time, when state-dependent costs are paid at each step.

We provide numerical illustrations where the variance reduction with respect to standard Monte Carlo is substantial. The variance is reduced by factors of several thousands in some cases. We prove bounds on the convergence rate of the worst-case error and variance for special situations. In line with what is typically observed in RQMC contexts, our empirical results indicate much better convergence than what these bounds guarantee.

Subject classifications: Simulation: efficiency. Probability: Markov processes.
Area of review: simulation.
Keywords: Variance reduction, randomized quasi-Monte Carlo, Markov chain, random walk, regenerative process, digital nets, lattice rules.

1 Introduction

A wide variety of real-life systems can be modeled as Markov chains with a large state space. In most interesting situations, analytic formulas are not available for these Markov chains and matrix-based numerical methods require too much time, so Monte Carlo simulation becomes the standard way of estimating performance measures for these systems.

In this paper, we propose a novel randomized quasi-Monte Carlo (RQMC) method to improve simulation efficiency for discrete-time Markov chains with a totally ordered state space. The method simulates n copies of the chain in parallel and induces negative dependence between the corresponding sample paths by using some form of generalized antithetic variates (Wilson, 1983; Ben-Ameur et al., 2004). Our aim is that the empirical distribution of the states of these n chains, at any given step j, is a better approximation of the corresponding theoretical distribution than if the n chains were simulated independently. As a result, performance measure estimators obtained by taking an average across the n copies of the chain, will typically have much smaller variance.

More specifically, we consider a Markov chain $\{X_j, \ j \geq 0\}$ with state space $\mathcal{X}$, which evolves according to the stochastic recurrence:

$$X_0 = x_0, \qquad X_j = \varphi_j(X_{j-1}, \mathbf{U}_j), \ j \geq 1, \tag{1}$$

where the $\mathbf{U}_j$ are i.i.d. random vectors uniformly distributed over the d-dimensional unit hypercube $[0,1)^d$. Every discrete-time Markov chain that can be simulated on a computer fits this framework with $d = 1$ (it suffices to define the chain so that each newly generated uniform random number in the simulation corresponds to one step of the chain). Here, we simply assume that d is a finite constant. The random vector $\mathbf{U}_j$ represents the uniform random numbers required to simulate one step of the chain.

We want to estimate the expected total cost $\mu = E[Y]$ where

$$Y = \sum_{j=1}^{\tau} c_j(X_j), \tag{2}$$

each $c_j : \mathcal{X} \to \mathbb{R}$ is a *cost function*, τ is a stopping time with respect to the filtration $\{\mathcal{F}_j, \ j \geq 0\}$ generated by $\{(j, X_j), \ j \geq 0\}$, $E[\tau] < \infty$, and we assume that the functions φ_j and c_j are easy to

evaluate at any given point, for each j. We also assume (implicitly) that $\mathcal{X}$, the φ_j's, and the c_j's satisfy the appropriate measure-theoretic requirements so that all objects of interest in this paper are well-defined.

The *standard Monte Carlo* (MC) method simulates the random variable Y as follows. For $j = 1, \ldots, \tau$, generate $\mathbf{U}_j \sim U[0,1)^d$, compute $X_j = \varphi_j(X_{j-1}, \mathbf{U}_j)$, and add the value of $c_j(X_j)$ to an accumulator, which at the end will contain the value of Y. This can be replicated n times independently, and the sample mean and variance of the n values of Y can be taken as unbiased estimators of the exact mean and variance of Y. From this, one can compute a confidence interval on μ, e.g., via the central-limit theorem.

Let $s = \sup_\omega d\tau$, where the supremum is taken over all possible sample paths ω, so $s = \infty$ if τ is unbounded. In this setting, the random variable Y can be written as a function of a sequence of s i.i.d. $U(0,1)$ random variables, say $Y = f(U_1, \ldots, U_s)$, for some complicated function f, where $\mathbf{U}_j = (U_{d(j-1)+1}, \ldots, U_{dj})$ for each j. The MC method generates n independent random points $\mathbf{V}_i = (U_{i,1}, \ldots, U_{i,s})$, $i = 1, \ldots, n$, in the s-dimensional unit hypercube $[0,1)^s$, evaluates the function f at each of these points, and takes the average $\bar{Y}_n$ of these n evaluations as an estimator of μ.

The idea of RQMC is to induce negative dependence between the points $\mathbf{V}_i$ by generating them in a way that

(i) each point $\mathbf{V}_i$ is still uniformly distributed in $[0,1)^s$ and

(ii) the point set $\{\mathbf{V}_1, \ldots, \mathbf{V}_n\}$ covers the hypercube $[0,1)^s$ more evenly (in some sense) than a set of independent random points.

A point set that satisfies these two conditions is called a *RQMC point set*. The estimator $\bar{Y}_n$ is computed by averaging the values of f over the n points in the same way as for MC. These n values are not independent, but we can estimate the variance of $\bar{Y}_n$ by replicating this scheme m times, with independent randomizations of the same point set. Under simple conditions on the randomization, the sample mean and sample variance of these m averages are unbiased estimators

of the exact mean and variance of $\bar{Y}_n$. Further details on this *classical RQMC* approach (and variants of it for high-dimensional contexts) can be found in Owen (1998); L'Ecuyer and Lemieux (2000, 2002) and other references given there. This approach is typically more efficient than MC when s is small (e.g., less than 20 or so) or if the function f has *low effective dimension* in some sense, as explained in Owen (1998) and L'Ecuyer and Lemieux (2002).

The RQMC method proposed in this paper operates differently. We simulate n copies of the chain in parallel. To simulate step j for all copies, we use a randomized $(d+1)$-dimensional highly-uniform (or "low-discrepancy") point set $P'_{n,j}$ of cardinality n, as explained in the next section, where $d \ll s$ typically. These point sets are randomized independently at the different steps, in a way that the sample path of any given copy of the chain obeys the correct probability law (the same as with the MC method). As a result, we have an unbiased estimator for the average cost. The aim of the proposed method is to induce dependence across the n copies so that the empirical distribution of the n values of X_j (at step j) gives a much better approximation of the distribution F_j of the random variable X_j than if the chains were simulated independently.

Our approach is an adaptation of a *deterministic method* proposed and studied by Lécot and Tuffin (2004), based on ideas of Lécot and Ogawa (2002), for approximating transient measures over a *fixed* number of steps, for discrete-time and discrete-state Markov chains with a totally ordered state space. That method uses a $(0,2)$-sequence in base 2. At step j of the chain, it reorders the n copies according to their current states and "simulates" the transitions (next states) for the n copies by employing the elements nj to $nj+n-1$ of the $(0,2)$-sequence in place of uniform random numbers to drive the simulation. It assumes that simulating each transition of the chain requires a single uniform random variate. Convergence to the correct value was proved by Lécot and Tuffin (2004) under a condition on the structure of the transition probability matrix of the Markov chain. In contrast, our new method is a randomized algorithm that provides an unbiased estimator. It also applies to Markov chains with continuous state space, with a random and unbounded number of steps, and the number d of uniform random variates that are required to generate the next state in one step of the Markov chain can be larger than 1. It thus covers a much broader range of applications.

4

We have theoretical results on the convergence rate of the variance of the mean estimator (as $n \to \infty$) for special cases only, but our empirical results with a variety of examples indicate that this variance goes down much faster (as a function of n) with the proposed method than for standard MC and also faster than classical RQMC in many situations. Note that this gap between proven error (or variance) bounds and empirical performance also exists for traditional QMC and RQMC methods.

In the next section, we motivate and define our RQMC sampling algorithm, called *array-RQMC*. Convergence results for special settings are given in Section 3. Section 4 illustrates the method via several numerical examples, showing that it can improve the simulation efficiency by large factors compared with standard MC. Our experiments also provide strong empirical evidence of the effectiveness of applying a baker transformation to a randomly-shifted lattice rule for both classical RQMC and array-RQMC, and of applying a linear matrix scrambling to a Sobol' net.

2 The Array-RQMC Algorithm

We now assume (for the remainder of the paper) that $\mathcal{X} \subseteq \mathbb{R}^{\ell} \cup \{\infty\}$, and that a *total order* has been defined on $\mathcal{X}$, for which ∞ is the largest state. The state ∞ is an *absorbing state* used to indicate that we have reached the stopping time τ. That is, $X_j = \infty$ for $j > \tau$, and $c_j(\infty) = 0$.

The *array-RQMC* algorithm works as follows. At *step 1*, we take a RQMC point set $P_{n,1} = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$ in $[0,1)^d$, define

$$X_{i,1} = \varphi_1(x_0, \mathbf{u}_i) \qquad \text{for } i = 0, \ldots, n-1,$$

and estimate the distribution F_1 of X_1 by the empirical distribution $\hat{F}_1$ of $X_{0,1}, \ldots, X_{n-1,1}$. This

gives the following approximation, where I denotes the indicator function:

$$
\begin{aligned}
F_1(x) &= P[X_1 \leq x] \\
&= \int_{[0,1)^d} I(\varphi_1(x_0, \mathbf{u}) \leq x) \, d\mathbf{u} \qquad (3) \\
&\approx \frac{1}{n} \sum_{i=0}^{n-1} I(\varphi_1(x_0, \mathbf{u}_i) \leq x) \qquad (4) \\
&= \frac{1}{n} \sum_{i=0}^{n-1} I(X_{i,1} \leq x) \stackrel{\text{def}}{=} \hat{F}_1(x),
\end{aligned}
$$

which amounts to estimating the integral (3) by RQMC in (4).

At *step j*, we use the empirical distribution $\hat{F}_{j-1}$ of $X_{0,j-1}, \ldots, X_{n-1,j-1}$ as an approximation of the distribution F_{j-1} of X_{j-1}. Let $P_{n,j} = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$ be a RQMC point set in $[0,1)^d$ such that the $(d+1)$-dimensional point set $P'_{n,j} = \{\mathbf{u}'_i = ((i+0.5)/n, \mathbf{u}_i), 0 \leq i < n\}$ is "highly uniform" (or has "low discrepancy") in $[0,1)^{d+1}$, in a sense that we leave open for the moment (specific definitions in the asymptotic sense, as $n \to \infty$, are adopted in the propositions of Section 3). We estimate F_j by the empirical distribution $\hat{F}_j$ of the values $X_{i,j} = \varphi_j(X_{(i),j-1}, \mathbf{u}_i)$, $i = 0, \ldots, n-1$, where $X_{(0),j-1}, \ldots, X_{(n-1),j-1}$ are the *sorted states* at step $j-1$. This can be interpreted as follows, where $F^{-1}(v)$ is defined as $\inf\{x \in \mathbb{R} : F(x) \geq v\}$:

$$
\begin{aligned}
F_j(x) &= P[X_j \leq x] = E[I(\varphi_j(X_{j-1}, \mathbf{U}_j) \leq x)] \\
&= \int_{\mathcal{X}} \int_{[0,1)^d} I(\varphi_j(y, \mathbf{u}) \leq x) \, d\mathbf{u} \, dF_{j-1}(y) \qquad (5) \\
&\approx \int_{\mathcal{X}} \int_{[0,1)^d} I(\varphi_j(y, \mathbf{u}) \leq x) \, d\mathbf{u} \, d\hat{F}_{j-1}(y) \qquad (6) \\
&= \int_{[0,1)^{d+1}} I(\varphi_j(\hat{F}_{j-1}^{-1}(v), \mathbf{u}) \leq x) \, d\mathbf{u} \, dv \qquad (7) \\
&\approx \frac{1}{n} \sum_{i=0}^{n-1} I(\varphi_j(\hat{F}_{j-1}^{-1}((i+0.5)/n), \mathbf{u}_i) \leq x) \qquad (8) \\
&= \frac{1}{n} \sum_{i=0}^{n-1} I(\varphi_j(X_{(i),j-1}, \mathbf{u}_i) \leq x) \\
&= \frac{1}{n} \sum_{i=0}^{n-1} I(X_{i,j} \leq x) \stackrel{\text{def}}{=} \hat{F}_j(x).
\end{aligned}
$$

In (6), we have replaced F_{j-1} in (5), by its approximation $\hat{F}_{j-1}$. In (8), we approximate the integral in (7) by RQMC over $[0,1)^{d+1}$ with the point set $P'_{n,j}$. Observe that this point set gives a perfect

6

stratification of the distribution $\hat{F}_{j-1}$, with exactly one observation per stratum (the strata are the jumps of $\hat{F}_{j-1}$). On the other hand, these observations are not independent across the strata.

Putting these pieces together, we get the following algorithm (the "for" loops are written using the C/C++/Java syntax and indentation alone indicates the scope of the loops):

Array-RQMC algorithm:

Initialization. Select a d-dimensional QMC point set $\tilde{P}_n = (\tilde{\mathbf{u}}_0, \ldots, \tilde{\mathbf{u}}_{n-1})$ and a randomization of $\tilde{P}_n$ such that (a) each randomized point is uniform over $[0, 1)^d$ and (b) if $P_n = (\mathbf{u}_0, \ldots, \mathbf{u}_{n-1})$ denotes the randomized version, then $P'_n = \{((i + 0.5)/n, \mathbf{u}_i), 0 \le i < n\}$ has "low discrepancy".

Simulate chains. Simulate in parallel n copies of the chain, numbered $0, \ldots, n-1$, as follows:

For $(j = 1;\ X_{0,j-1} < \infty;\ j{+}{+})$
   Randomize $\tilde{P}_n$ afresh into $P_n = \{\mathbf{u}_0, \ldots, \mathbf{u}_{n-1}\}$;
   For $(i = 0;\ i < n$ and $X_{i,j-1} < \infty;\ i{+}{+})\ \ X_{i,j} = \varphi_j(X_{i,j-1}, \mathbf{u}_i)$;
   Sort (and renumber) the chains for which $X_{i,j} < \infty$ by increasing order
      of their states. (The sorted states $X_{0,j}, \ldots, X_{n-1,j}$ provide $\hat{F}_j$).

Output. Return the average $\bar{Y}_n$ of the n values of Y as an estimator of μ.

Let $\bar{Y}_{n,j}$ denote the average cost at step j, across the n copies. We have $\bar{Y}_n = \sum_{j=1}^{\infty} \bar{Y}_{n,j}$.

Proposition 1 *The averages $\bar{Y}_{n,j}$ and $\bar{Y}_n$ are* unbiased *estimators of $E[c_j(X_j)]$ and μ, respectively.*

Proof. The successive steps of the chain use independent randomizations. Therefore, for each chain, the vectors that take place of the $\mathbf{U}_j$'s in the recurrence (1) to generate the successive steps j of the chain are independent random variables uniformly distributed over $[0, 1)^d$. Thus, any given copy of the chain obeys the correct probabilistic model defined by (1) and (2), so $c_j(X_j)$ and Y have the correct expectations, $E[c_j(X_j)]$ and μ, and their averages over the n copies as well. $\quad\square$

To estimate the variance of $\bar{Y}_n$ and compute a confidence interval on μ, we can replicate this entire procedure independently m times. That is, across the m replications, all randomizations are independent. With this, we have:

Proposition 2 *The empirical variance of the m copies of $\bar{Y}_n$ is an unbiased estimator of $Var[\bar{Y}_n]$.*

Proof. This follows from the fact that the m copies of $\bar{Y}_n$ are i.i.d. unbiased estimators of μ. These m copies are independent because randomized points from different copies at any given step are independent. $\qquad\square$

This proposition implies that the variance of the overall average converges as $O(1/m)$ when $m \to \infty$. In the next section, we examine the convergence rate as a function of n when $n \to \infty$.

3 Convergence

Classical error bounds for numerical integration by quasi-Monte Carlo (QMC) methods are based on the Koksma-Hlawka inequality, which bounds the worst-case integration error by the product of the Hardy-Krause total variation $V(f)$ of the integrand f and the rectangular star discrepancy $D_n^*(P_n)$ of the point set P_n (see Niederreiter, 1992, for the definitions and further details). That is,

$$\|\bar{Y}_n - \mu\| \leq V(f)D_n^*(P_n)$$

if $\bar{Y}_n$ represents the average value of f over the n points and μ is its theoretical average over the unit hypercube. The star discrepancy of a point set $P_n \in [0,1)^s$ is defined as

$$D_n^*(P_n) = \sup_{\mathbf{x}=(x_1,\ldots,x_s)\in[0,1)^s} |\text{card}(P_n \cap [\mathbf{0},\mathbf{x}))/n - \text{vol}([\mathbf{0},\mathbf{x}))|$$

where $[\mathbf{0},\mathbf{x})$ is the rectangular box with opposite corners $\mathbf{0}$ and $\mathbf{x}$, and vol denotes its volume. The definition of $V(f)$ is more complicated. We just point out that it is infinite whenever f has a discontinuity not aligned with one of the axes (Owen, 2004). The integrand $I(\varphi_j(\hat{F}_{j-1}^{-1}(v), \mathbf{u}) \leq x)$ in (7) is 1 in part of the unit cube, and 0 elsewhere. The shape and complexity of the boundary

between these two regions depends on $\varphi_1, \ldots, \varphi_j$. We assume that these regions are at least Jordan-measurable sets. For continuous state spaces $\mathcal{X}$, the Hardy-Krause total variation of this indicator function is typically infinite, because the boundary is typically not aligned with the axes, so the classical Koksma-Hlawka inequality is not helpful to bound the integration error in (8).

In what follows, we prove bounds on the convergence rate directly from first principles, for special cases for which $d = 1$ (so $\mathcal{X} \subseteq \mathbb{R}$ and the distribution function F_j is defined over $\mathbb{R}$). We will end up bounding $|\bar{Y}_{n,j} - E[c_j(X_j)]|$ by the product of the Kolmogorov distance between $\hat{F}_j$ and F_j defined as

$$\Delta_j = \sup_{x \in \mathcal{X}} |\hat{F}_j(x) - F_j(x)| = \sup_{x \in \mathbb{R}} |\hat{F}_j(x) - F_j(x)|$$

and the total variation of the cost function c_j. This is Proposition 8. To bound this product, we will obtain bounds on Δ_j by using notions of histogram and integral discrepancies whose properties are examined in the next subsection.

Related results have been obtained by Lécot (1996) in a more general setting (general real-valued functions defined over the unit hypercube and having bounded variation) but under the stronger assumption that P_n is a (t, m, s)-net and with different methods of proof.

3.1 Histogram and Integral Discrepancies

A *histogram with L intervals* over the unit square is defined as follows. Partition the unit interval $[0, 1)$ at the bottom of the square into L subintervals, say of lengths $q_1, \ldots, q_L$ where $q_1 + \cdots + q_L = 1$. Over the ith interval, put a rectangle of height h_i, where $0 \le h_i \le 1$, and with the same width as the interval. The histogram H is the union of these rectangles. We say that the histogram is *monotone* (increasing or decreasing) if $h_1 \le \cdots \le h_L$ or $h_1 \ge \cdots \ge h_L$.

Let $\mathcal{H}(L)$ be the family of all histograms with L intervals over the unit square, and $\mathcal{H}^+(L)$ the subfamily of all monotone histograms. The *L-histogram discrepancy* of a point set P_n in the unit square is defined as

$$D_{\mathrm{h}}(L, P_n) = \sup_{H \in \mathcal{H}(L)} |\mathrm{card}(P_n \cap H)/n - \mathrm{area}(H)|$$

9

where area(H) denotes the area of H. If $\mathcal{H}(L)$ is replaced by $\mathcal{H}^+(L)$, we get the *L-staircase discrepancy* of P_n, denoted $D_{\mathrm{h}}^+(L, P_n)$. The following lemma is rather straightforward:

Lemma 3 *Let P_n denotes the first n points of a two-dimensional low-discrepancy sequence whose rectangular star discrepancy satisfies $D_n^*(P_n) = O(n^{-1} \log n)$. Then there is a constant K such that for all L,*

$$D_{\mathrm{h}}^+(L, P_n) \leq D_{\mathrm{h}}(L, P_n) \leq K L n^{-1} \log n.$$

Proof. In every histogram H, each rectangle can be written as a difference of two rectangular boxes anchored at the origin. Thus, H can be written as a sum and difference of $2L$ such boxes. But we know that the rectangular star discrepancy of P_n is in $O(n^{-1} \log n)$ and the last inequality follows. The other inequality is obvious. $\qquad\square$

Several two-dimensional sequences that satisfy this requirement are readily available (Niederreiter, 1992); for instance, one can take the two-dimensional Sobol' sequence. However, the bound in Lemma 3 is linear in L so it is practically not very useful for Markov chains with large state spaces (assuming that each state of the chain is associated with one of the L subintervals and vice-versa). The next lemma provides a bound that does not depend on L. It is based on Assumption 1 and a notion of integral discrepancy obtained by sending L to infinity.

For a function $f : [0, 1] \to \mathbb{R}$, the *total variation* of f is defined as

$$V(f) = \sup_{L \geq 1,\, 0 = x_0 < \cdots < x_L = 1} \sum_{j=1}^{L} |f(x_j) - f(x_{j-1})| = \int_0^1 df(x).$$

It is well known that a function of bounded variation over a given interval is Riemann-integrable over that interval. Let $\mathcal{F}(v)$ denote the set of functions $f : [0, 1] \to [0, 1]$ such that $V(f) \leq v$.

For $f : [0, 1] \to [0, 1]$, let $H(f) = \{(x, y) \in [0, 1]^2 : 0 \leq y \leq f(x)\}$, the surface under f. We define the *integral discrepancy at variation v*, for a point set P_n, by

$$D_{\mathrm{i}}(P_n, v) = \sup_{f \in \mathcal{F}(v)} |\mathrm{card}(P_n \cap H(f))/n - \mathrm{area}(H(f))| . \tag{9}$$

If f has bounded variation, $H(f)$ can be approximated arbitrarily closely by an histogram H having L rectangles of heights $h_1 = f(x_1), \ldots, h_L = f(x_L)$ where $0 < x_1 < \cdots < x_L = 1$ for some large L. This histogram has total variation

$$V(H) = \sum_{i=2}^{L} |h_i - h_{i-1}| \leq V(f).$$

Reciprocally, any such histogram defines a piecewise-constant function f of total variation $V(f) = V(H)$. Hence, we have that

$$D_{\mathrm{i}}(P_n, v) = \sup_{L \geq 1, \, H \in \mathcal{H}(L), \, V(H) \leq v} |\mathrm{card}(P_n \cap H)/n - \mathrm{area}(H)|.$$

Assumption 1 *Suppose that n is a square, so $\sqrt{n}$ is an integer, and that if we partition of the unit square into n subsquares of size $n^{-1/2} \times n^{-1/2}$, each of those subsquares contains exactly one point from P_n.*

Lemma 4 *Under Assumption 1, for any $v \geq 0$, we have $D_{\mathrm{i}}(P_n, v) \leq (v+1)n^{-1/2}$.*

    Proof. Consider a function $f : [0,1] \to [0,1]$ with $V(f) \leq v$. We define the *extended graph* of f, denoted $G(f)$, as the boundary between $H(f)$ and $[0,1]^2 \setminus H(f)$. This is the graph of f, $\{(x, f(x)) : 0 \leq x \leq 1\}$, to which we add vertical lines that link the graph pieces where f is discontinuous. The idea of the proof is to bound the number of subsquares that intersect $G(f)$ and then bound the error in terms of this number. The n squares are partitioned into $\sqrt{n}$ columns that correspond to $\sqrt{n}$ intervals on the horizontal axis. Suppose that $G(f)$ goes into ℓ_j different subsquares in column j. Clearly, these ℓ_j subsquares must form a connected rectangle. For any of these subsquares S, we have an *overestimation* of $\mathrm{area}(H(f) \cap S)$ if the point of P_n lying in subsquare S is in $H(f)$ and *underestimation* otherwise. The total error in any given column is the total amount of overestimation minus the total amount of underestimation. Let $S_{\mathrm{t},j}$ and $S_{\mathrm{b},j}$ denote the top and bottom subsquares from this rectangle, respectively. They may be the same subsquare, if $\ell_j = 1$.

    Suppose that we are overestimating in $S_{\mathrm{t},j}$. If we are underestimating in $S_{\mathrm{b},j}$, or if the top and bottom subsquares are the same, the combined error in the top and bottom subsquares cannot

11

exceed $1/n$, so the error in column j is at most $(\ell_j - 1)/n$. Otherwise, i.e., if we are also overestimating in $S_{b,j}$ and $\ell_j \geq 2$, then the error in the bottom subsquare is the surface of this subsquare which is above $G(f)$. This surface cannot exceed $V_{b,j} n^{-1/2}$ where $V_{b,j}$ is the variation of f in this subsquare. Then in this second case, the error in column j is at most $(\ell_j - 1)/n + V_{b,j} n^{-1/2}$. But in both cases, the total variation V_j in column j satisfies $V_j \geq (\ell_j - 2) n^{-1/2} + V_{b,j}$, so the error in column j cannot exceed $V_j n^{-1/2} + n^{-1}$. This same error bound can be obtained by a symmetrical argument in the case where we are underestimating in $S_{t,j}$.

By adding these inequalities over all columns, we obtain that the total error cannot exceed $V(f) n^{-1/2} + n^{1/2} n^{-1} = (V(f) + 1) n^{-1/2} \leq (v + 1) n^{-1/2}$. By taking the sup over f as in (9), the result follows. $\qquad\square$

Let $\mathcal{F}^+$ be the set of *monotone* functions $f : [0,1] \to [0,1]$ and define

$$D_i^+(P_n) = \sup_{f \in \mathcal{F}^+} |\text{card}(P_n \cap H(f))/n - \text{area}(H(f))|.$$

For $f \in \mathcal{F}^+$, $V(f)$ cannot exceed 1. This gives:

Corollary 5 *Under Assumption 1, we have $D_i^+(P_n) \leq 2n^{-1/2}$.*

To see that this discrepancy bound is tight, consider the constant function $f(x) = n^{-1/2}(1 + \epsilon)$ for a small $\epsilon > 0$ and suppose that each column has two points in $H(f)$. Then $V(f) = 0$ and the error is $(1 - \epsilon) n^{-1/2}$, which can be arbitrarily close to the bound $n^{-1/2}$.

Instead of asking only for the points of P_n to be evenly distributed among the n subsquares, we could have the stronger requirement that they form a $(0, k, 2)$-net in base 2, assuming that $n = 2^k$ for some integer k. This means that for every partition of the unit square into n rectangles of width 2^{-q} and height 2^{-k+q} for some $q = 0, 1, \ldots, k$, every rectangle contains exactly one point. Consider the function $f(x) = x$. Proposition 5 of Lécot (1996) shows that for this example, there is a $(0, k, 2)$-net (namely the Hammersley point set in base 2) for which the error is $n^{-1/2}/2$. In other words, the rate of the bound of Lemma 4 is tight even under this stronger assumption.

3.2 Error Bounds on the State Distribution

Here we derive bounds on the distance Δ_j between $\hat{F}_j$ and F_j, under the following assumption, which implies that each φ_j is nondecreasing with respect to its second argument.

Assumption 2 *The Markov chain has a one-dimensional state space $\mathcal{X} \subseteq \mathbb{R}$, so $d = 1$, and at each step j, we use inversion from a single uniform to generate the next state X_j from its conditional distribution given X_{j-1}.*

At step j of the Markov chain, for $x \in \mathcal{X}$ and $\ell \in \mathbb{R}$, let

$$
\begin{aligned}
F_j(\ell) &= P[X_j \leq \ell], \\
F_j(\ell \mid x) &= P[X_j \leq \ell \mid X_{j-1} = x], \\
\Lambda_j(\ell) &= \int_{-\infty}^{\infty} |dF_j(\ell \mid x)|, \quad \text{and} \\
\Lambda_j &= \sup_{\ell \in \mathbb{R}} \Lambda_j(\ell),
\end{aligned}
$$

where the Stieltjes differential $dF_j(\ell \mid x)$ in the definition of $\Lambda_j(\ell)$ is with respect to x. Thus, $\Lambda_j(\ell)$ is the total variation of the function $F_j(\ell \mid \cdot)$. If the Markov chain is stochastically monotone, then each $F_j(\ell \mid \cdot)$ is monotone and Λ_j cannot exceed 1.

Let $\hat{F}_j$ be the empirical distribution of the states of the n copies of the chain at step j and

$$
\tilde{F}_j(\ell) = \int_{-\infty}^{\infty} F_j(\ell \mid x) d\hat{F}_{j-1}(x) = \frac{1}{n} \sum_{i=0}^{n-1} F_j(\ell \mid X_{(i),j-1}),
$$

so $\tilde{F}_j$ is the conditional distribution function of X_j when X_{j-1} is generated from $\hat{F}_{j-1}$. The value of $\tilde{F}_j(\ell)$ is equal to the area of the histogram $H_{j,\ell}$ whose height over the interval $[i/n, (i+1)/n)$ is $F_j(\ell \mid X_{(i),j-1})$, for $i = 0, \dots, n-1$. This histogram $H_{j,\ell}$ is inside the unit square $[0,1]^2$. We also have that

$$
V[H_{j,\ell}] = \sum_{i=1}^{n-1} \left| F_j(\ell \mid X_{(i-1),j-1}) - F_j(\ell \mid X_{(i),j-1}) \right| \leq \Lambda_j(\ell) \leq \Lambda_j. \tag{10}
$$

Define $\Delta_j(\ell) = \hat{F}_j(\ell) - F_j(\ell)$. We have $\Delta_0 = 0$ and

$$
\Delta_j = \sup_{\ell \in \mathbb{R}} |\Delta_j(\ell)|.
$$

The following proposition provides bounds on Δ_j.

Proposition 6 *Let Assumption 2 holds.*

(a) Suppose that the Markov chain has a finite state space $\mathcal{X} = \{1, \ldots, L\}$ and that the star discrepancy of each $P'_{n,j}$ satisfies $D^*_n(P'_{n,j}) = O(n^{-1} \log n)$ w.p.1 (this can easily be achieved by taking a $(0,2)$-sequence in some base b). Then

$$|\hat{F}_j(\ell) - \tilde{F}_j(\ell)| \leq KLn^{-1} \log n$$

for some constant K. If this holds for all $j \geq 1$ and $\ell \in \mathbb{R}$, then

$$\Delta_j \leq KLn^{-1} \log n \sum_{k=1}^{j} \prod_{i=k+1}^{j} \Lambda_i. \tag{11}$$

for all j, for some constant K.

(b) If $P'_{n,j}$ satisfies Assumption 1, then for all $j \geq 1$ and $\ell \in \mathbb{R}$,

$$|\hat{F}_j(\ell) - \tilde{F}_j(\ell)| \leq (\Lambda_j(\ell) + 1)n^{-1/2}$$

and

$$\Delta_j \leq n^{-1/2} \sum_{k=1}^{j} (\Lambda_k + 1) \prod_{i=k+1}^{j} \Lambda_i. \tag{12}$$

Proof. At step j, we have

$$|\hat{F}_j(\ell) - F_j(\ell)| \leq |\hat{F}_j(\ell) - \tilde{F}_j(\ell)| + |\tilde{F}_j(\ell) - F_j(\ell)|. \tag{13}$$

To bound Δ_j we will bound the two quantities on the right of (13). We have

$$\begin{aligned} \tilde{F}_j(\ell) - F_j(\ell) &= \int_{-\infty}^{\infty} F_j(\ell \mid x) d\hat{F}_{j-1}(x) - \int_{-\infty}^{\infty} F_j(\ell \mid x) dF_{j-1}(x) \\ &= \int_{-\infty}^{\infty} (F_{j-1}(x) - \hat{F}_{j-1}(x)) dF_j(\ell \mid x) \end{aligned} \tag{14}$$

where the second equality is obtained via integration by parts. Therefore,

$$|\tilde{F}_j(\ell) - F_j(\ell)| \leq \int_{-\infty}^{\infty} |\hat{F}_{j-1}(x) - F_{j-1}(x)||dF_j(\ell \mid x)| \leq \Lambda_j(\ell)\Delta_{j-1} \leq \Lambda_j \Delta_{j-1}. \tag{15}$$

From (15) and (13), we have

$$\Delta_j \leq \Lambda_j \Delta_{j-1} + \sup_{\ell \in \mathbb{R}} |\hat{F}_j(\ell) - \tilde{F}_j(\ell)|.$$

14

Together with the fact that $\Delta_0 = 0$, this gives

$$\Delta_j \leq \sum_{k=1}^{j} \sup_{\ell \in \mathbb{R}} |\hat{F}_k(\ell) - \tilde{F}_k(\ell)| \prod_{i=k+1}^{j} \Lambda_i, \tag{16}$$

where an empty product is assumed to be 1.

We recall that $\tilde{F}_j(\ell)$ is the area of the histogram $H_{j,\ell}$ and observe that $\hat{F}_j(\ell)$ is the fraction of the points of $P'_{n,j}$ that fall in $H_{j,\ell}$. Therefore,

$$\hat{F}_j(\ell) - \tilde{F}_j(\ell) = \text{card}(P'_{n,j} \cap H_{j,\ell})/n - \text{area}(H_{j,\ell}), \tag{17}$$

which implies that

$$|\hat{F}_j(\ell) - \tilde{F}_j(\ell)| \leq D_\mathrm{i}(P'_{n,j}, V(H_{j,\ell})) \leq D_\mathrm{i}(P'_{n,j}, \Lambda_j(\ell)). \tag{18}$$

This, together with Lemmas 3 and 4, proves the proposition. $\qquad\square$

Corollary 7 *If the Markov chain is also stochastically increasing, i.e., $P[X_j \geq x \mid X_{j-1} = y]$ is non-decreasing in y for each j, the bound (12) becomes*

$$\Delta_j \leq 2jn^{-1/2}.$$

Proof. Recall that in that case, $F_j(\ell \mid x)$ is non-decreasing in x, so $\Lambda_j \leq 1$ for each j. $\qquad\square$

3.3   Worst-Case Error Bounds on the Expected Average Cost

The next step is to bound the error on the expected cost at step j. Let

$$V(c_j) = \int_{-\infty}^{\infty} |dc_j(x)|,$$

the total variation of the cost function c_j. Our bounds on Δ_j proved in the preceding section readily provide bounds on the error $|\bar{Y}_{n,j} - E[c_j(X_j)]|$ as follows, in the case where c_j has bounded variation.

Proposition 8 *We have*

$$\left| \bar{Y}_{n,j} - E[c_j(X_j)] \right| \leq \Delta_j V(c_j).$$

15

Proof. Using integration by parts for the third equality, we get

$$
\begin{aligned}
\left| \bar{Y}_{n,j} - E[c_j(X_j)] \right| &= \left| \frac{1}{n} \sum_{i=0}^{n-1} c_j(X_{i,j}) - E[c_j(X_j)] \right| \\
&= \left| \int_{-\infty}^{\infty} c_j(\ell) d\hat{F}_j(\ell) - \int_{-\infty}^{\infty} c_j(\ell) dF_j(\ell) \right| \\
&= \left| \int_{-\infty}^{\infty} (\hat{F}_j(\ell) - F_j(\ell)) dc_j(\ell) \right| \\
&\leq \int_{-\infty}^{\infty} |\Delta_j(\ell)| |dc_j(\ell)| \\
&\leq \Delta_j V(c_j).
\end{aligned}
$$

$\square$

This proposition tells us that the square error of $\bar{Y}_{n,j}$ converges at worst at the same rate as Δ_j^2 when $n \to \infty$. If τ is bounded, this implies that $|\bar{Y}_n - \mu|^2$ converges as $O(\sum_{j=1}^{\tau} \Delta_j^2) = O(1/n)$ in the worst case.

3.4   Variance Bounds for array-RQMC

Proposition 6(b) gives a *worst-case deterministic* bound of $O(1/n)$ for the square error of $\bar{Y}_{n,j}$, in contrast with an *expected* square error of $O(1/n)$ for ordinary Monte Carlo. In what follows, we obtain a better bound on the *convergence rate of the variance* by exploiting randomization. The proof is given in a setting where the following holds:

Assumption 3 *If the unit square $[0,1)^2$ is partitioned into n subsquares as in Assumption 1, each subsquare contains exactly one point of $\tilde{P}'_{n,j}$, and the positions of these points within their respective subsquares are pairwise independent and uniformly distributed over the subsquare.*

This is an instance of *stratified sampling* over the unit square, with independent samples across the strata. The array-RQMC algorithm does not satisfy this assumption, because the positions of the points in the subsquares that are on top of each other are not independent. However, proving the result without the independence assumption is more difficult and we leave this for future work. Typically, the overall dependence created by the array-RQMC algorithm will be negative and will

16

tend to *reduce* the variance of the expressions in Proposition 9, so we should expect the variance in the setting of the RQMC algorithm to be even smaller. We made numerical experiments that did confirmed that, for a few examples.

Define

$$\Gamma_j = \frac{1}{4n^{3/2}} \sum_{k=1}^{j} (\Lambda_k + 1) \prod_{i=k+1}^{j} \Lambda_i^2.$$

Proposition 9 *Under Assumptions 1, 2, and 3, for each j and ℓ, we have*

$$\mathrm{Var}[\hat{F}_j(\ell) - \tilde{F}_j(\ell)] \leq (\Lambda_j + 1)n^{-3/2}/4, \tag{19}$$

$$\mathrm{Var}[\hat{F}_j(\ell) - F_j(\ell)] \leq \Gamma_j, \tag{20}$$

and

$$E\left[\left(\bar{Y}_{n,j} - E[c_j(X_j)] \right)^2 \right] \leq \Gamma_j V(c_j)^2.$$

Proof. To prove (19), we first recall that $\tilde{F}_j(\ell)$ is the area of the histogram $H_{j,\ell}$ whereas $\hat{F}_j(\ell)$ is the fraction of $P'_{n,j}$ that falls in this histogram. We enumerate the n subsquares with $i = 0, \ldots, n-1$. Let S_i be the ith subsquare and $\delta_j(\ell, i) = \mathrm{card}(\tilde{P}'_{n,j} \cap H_{j,\ell} \cap S_i) - n \, \mathrm{area}(H_{j,\ell} \cap S_i)$. We have

$$\hat{F}_j(\ell) - \tilde{F}_j(\ell) = \mathrm{card}(P'_{n,j} \cap H_{j,\ell})/n - \mathrm{area}(H_{j,\ell}) = \sum_{i=0}^{n-1} \delta_j(\ell, i)/n.$$

For any given j and each i, $\delta_j(\ell, i)$ is a Bernoulli random variable minus its mean, so $E[\delta_j(\ell, i)] = 0$ and $\mathrm{Var}[\delta_j(\ell, i)] \leq 1/4$. These Bernoulli random variables have nonzero variance only for the subsquares that intersect the histogram boundary (that separates $H_{j,\ell}$ from $[0,1]^2 \setminus H_{j,\ell}$), because for the other subsquares they are constant.

Here we consider a fixed j and drop the subscript j for a moment. Let $S_{\mathrm{b},c}$ and $S_{\mathrm{t},c}$ denote the lowest and highest subsquares that intersect the histogram boundary in column c, and let $V_{\mathrm{b},c}$ and $V_{\mathrm{t},c}$ be $n^{1/2}$ times the variation of the histogram in these two subsquares. Let p_c be n times the area of $S_{\mathrm{t},c}$ contained in the histogram and q_c be n times the area of $S_{\mathrm{b},c}$ *not* contained in the histogram. We suppose, until indicated otherwise, that $S_{\mathrm{b},c}$ and $S_{\mathrm{t},c}$ are not the same subsquare.

17

Then the Bernoulli variables $\delta_j(\ell, i)$ that correspond to these two subsquares are independent and have variances bounded by $p_c(1 - p_c)$ and $q_c(1 - q_c)$ (which cannot exceed $1/4$), respectively. We will now prove the following bound on the sum of their variances:

$$p_c(1 - p_c) + q_c(1 - q_c) \leq \frac{V_{b,c} + V_{t,c} + 1}{4}. \tag{21}$$

Denote $v = V_{b,c} + V_{t,c}$. If $v \geq 1$, (21) holds trivially, so let us assume that $v \leq 1$. At any given point on the horizontal axis, the histogram boundary cannot be in $S_{b,c}$ and $S_{t,c}$ at the same time. Let ρ be the fraction of the horizontal interval in column c where the histogram is in $S_{t,c}$. Then we have $p_c \leq \rho V_{t,c}$ and $q_c \leq (1 - \rho)V_{b,c}$. We now observe that either $\rho V_{t,c} \leq v/4$ or $(1 - \rho)V_{b,c} \leq v/4$. To see this, consider a rectangle of width 1 and height v, with bottom left corner at point $(0, 0)$, divided into four subrectangles by a vertical line at ρ and an horizontal line at $V_{t,c}$. The quantities $\rho V_{t,c}$ and $(1 - \rho)V_{b,c}$ are the surfaces of two opposite subrectangles of this rectangle, so their surfaces cannot be *both* larger than a quarter of the rectangle's surface v. Indeed, suppose that $\rho V_{t,c} > v/4$. Then, $V_{t,c} > v/(4\rho)$ and therefore $(1 - \rho)(v - V_{t,c}) > v/4$ would imply that $(1 - \rho)(v - v/(4\rho)) > v/4$, i.e., $0 < (1 - \rho)(4\rho - 1) - \rho = -(2\rho - 1)^2$, which is a contradiction. Therefore, we have either $p_c(1 - p_c) \leq v/4$ or $q_c(1 - q_c) \leq v/4$, and since these two quantities never exceed $1/4$, the bound (21) follows. If $S_{b,c}$ and $S_{t,c}$ are the same subsquare, the variance of $\delta_j(\ell, i)$ in this subsquare cannot exceed $1/4$.

If other subsquares intersect the histogram in column c, between $S_{b,c}$ and $S_{t,c}$, then in each of these subsquares the histogram variation is at least $n^{-1/2}$ and the variance of the corresponding $\delta_j(\ell, i)$ is at most $1/4$. By adding the above inequalities over all the columns, we obtain that the sum (over i) of variances of all Bernoulli variables $\delta_j(\ell, i)$ is bounded by $(\Lambda_j + 1)n^{1/2}/4$.

Since these $\delta_j(\ell, i)$'s are pairwise independent across the different values of i, we obtain

$$\text{Var}\left[\hat{F}_j(\ell) - \tilde{F}_j(\ell)\right] = \sum_{i=0}^{n-1} \text{Var}[\delta_j(\ell, i)/n] \leq n^{1/2}(\Lambda_j + 1)/(4n^2) = (\Lambda_j + 1)n^{-3/2}/4,$$

which proves (19).

We now prove (20) by induction on j. It obviously holds for $j = 0$, because $\hat{F}_0 = F_0$. Suppose it holds for $j - 1$, for all ℓ. Observe that from the proof of Proposition 1, $E[\hat{F}_j(x)] = P[X_j \leq x] =$

$F_j(x)$, so $E[\Delta_j(x)] = 0$. Then,

$$E[\Delta_{j-1}(x)\Delta_{j-1}(y)] \;=\; \mathrm{Cov}[\Delta_{j-1}(x), \Delta_{j-1}(y)] \;\leq\; \sup_{\ell \in \mathbb{R}} \mathrm{Var}[\Delta_{j-1}(\ell)] \;\leq\; \Gamma_j$$

for all states x, y. Therefore, using (14) for the first equality and assuming that we can interchange the expectation and integral in the third equality,

$$
\begin{aligned}
&E[(\tilde{F}_j(\ell) - F_j(\ell))^2] \\
=\; & E\left[\left(\int_{-\infty}^{\infty}(\hat{F}_{j-1}(x) - F_{j-1}(x))dF_j(\ell \mid x)\right)^2\right] \\
=\; & E\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(\Delta_{j-1}(x)\Delta_{j-1}(y))dF_j(\ell \mid x)dF_j(\ell \mid y)\right] \\
=\; & \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E[\Delta_{j-1}(x)\Delta_{j-1}(y)]dF_j(\ell \mid x)dF_j(\ell \mid y) \\
=\; & \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Gamma_j dF_j(\ell \mid x)dF_j(\ell \mid y) \\
=\; & \Lambda_j^2(\ell)\Gamma_j.
\end{aligned}
$$

Combining this with (19) and (13), and observing that $\hat{F}_j(\ell) - \tilde{F}_j(\ell)$ has mean 0 and is uncorrelated with $\tilde{F}_j(\ell) - F_j(\ell)$, we obtain that

$$
\begin{aligned}
E[(\hat{F}_j(\ell) - F_j(\ell))^2] \;=\; & \mathrm{Var}[(\hat{F}_j(\ell) - F_j(\ell))^2] \\
=\; & \mathrm{Var}[(\hat{F}_j(\ell) - F_j(\ell))^2] + \mathrm{Var}[(\tilde{F}_j(\ell) - F_j(\ell))^2] \\
=\; & E[(\hat{F}_j(\ell) - F_j(\ell))^2] + E[(\tilde{F}_j(\ell) - F_j(\ell))^2] \\
\leq\; & \Lambda_j^2(\ell)\Gamma_{j-1} + (\Lambda_j + 1)n^{-3/2}/4 \;\leq\; \Gamma_j
\end{aligned}
$$

and this completes the induction. To prove the last part, we have

$$
\begin{aligned}
& E\left[\left|\frac{1}{n}\sum_{i=0}^{n-1}c_j(X_{i,j}) - E[c_j(X_j)]\right|^2\right] \\
=\; & E\left[\left|\int_{-\infty}^{\infty} c_j(\ell)d\hat{F}_j(\ell) - \int_{-\infty}^{\infty} c_j(\ell)dF_j(\ell)\right|^2\right] \\
=\; & E\left[\left|\int_{-\infty}^{\infty}(\hat{F}_j(\ell) - F_j(\ell))dc_j(\ell)\right|^2\right] \\
\leq\; & \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E(\Delta_j(x)\Delta_j(y))|dc_j(x)||dc_j(y)| \\
\leq\; & \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \Gamma_j|dc_j(x)||dc_j(y)| \\
\leq\; & (V(c_j))^2\Gamma_j.
\end{aligned}
$$

19

□

We thus have a $O(n^{-3/2})$ convergence rate for the variance of the cost at step j if the chain is one-dimensional, $d = 1$, the Λ_j's are bounded, and the cost function has bounded variation. When the chain is stochastically increasing, we have $\Lambda_j \leq 1$ for all j and the variance bound becomes $(n^{-3/2}/2)j(V(c_j))^2$.

This could be generalized to higher-dimensional settings. For this, we need a counterpart of Lemma 4. Here we just sketch how this can be done for $d = 2$ for a stochastically increasing Markov chain, so that we have a two-dimensional increasing histogram in the unit cube. Partition the cube into n subcubes by partitioning each axis into $n^{1/3}$ equal parts (assuming that $n^{1/3}$ is an integer). The histogram boundary is now a surface. If we fix one of the two horizontal coordinates to a multiple of $n^{-1/3}$, this determines a vertical plane and the intersection of this plane with the histogram boundary can be covered by at most $2n^{1/3}$ subcubes in a similar manner as in the proof of Lemma 4. We can do this for each multiple of $n^{-1/3}$, and repeat in each direction. We find that the histogram boundary can be covered by at most $Kn^{2/3}$ subcubes for some constant K. In general, for a $d + 1$-dimensional cube, we conjecture that the histogram boundary can be covered by $Kn^{-d/(d+1)}$ subcubes for some constant K that may depend on d but not on n. This can be turned into a variance bound of $O(n^{2-d/(d+1)}) = O(n^{1+1/(d+1)})$.

The result could also be generalized to the case where c_j has infinite variation (e.g., unbounded state spaces and cost functions) if we assume that large-variation areas have low probability.

In our numerical experiments of Section 4, instead of generating the points independently in the different squares as in the assumptions of Proposition 9, we will generate them according to RQMC schemes that provide more uniformity, with the aim of inducing a larger amount of negative dependence (i.e., more variance reduction) than with straightforward stratification. Some of these RQMC schemes (e.g., the two-dimensional Sobol' nets) have one point per subsquare as in Proposition 9, but none of them really satisfies the assumptions of Proposition 9 because the locations of the points in two different squares are not independent. These RQMC schemes turn out to work very well in practice, but so far we have no counterpart of Proposition 9 for them.

4 A Numerical Illustration

4.1 An $M/M/1$ Queue with $d = 1$

Consider a single-server queue with i.i.d. exponential interarrival times with mean 1 and i.i.d. exponential service times with mean $\rho < 1$. This ρ is also the utilization factor of the server. We want to estimate the expected average waiting time of the first t customers, denoted μ. This μ could be computed numerically without simulation; we just use this simple academic example to illustrate our method.

Let W_j denote the waiting time of customer j in this system, where the first customer (who arrives to the empty system) has number 0. These W_j's satisfy the *Lindley recurrence*: $W_0 = 0$ and $W_j = \max(0, W_{j-1} + S_{j-1} - A_j)$ for $j \geq 1$. We estimate μ by the sample average $Y = (W_0 + \cdots + W_{t-1})/t$. To compute Y, we need to generate the $2(t-1)$ random variates $S_0, A_1, \ldots, S_{t-1}, A_t$. This estimator Y is unbounded (so the Koksma-Hlawka inequality gives an infinite bound for it), but it has bounded variance.

We consider a Markov chain that moves by one step each time one of these random variates is generated. That is, $X_0 = W_0$, $X_1 = W_0 + S_0$, $X_2 = W_1$, $X_3 = W_1 + S_1$, and so on. In this case, $d = 1$ and $s = 2(t-1)$. Later, we will also consider the case where the chain moves by one step every $d/2$ customers (where d is even), so $X_j = W_{jd/2}$ and $s = (t-1)/d$. In all cases, this Markov chain is stochastically increasing.

We tried both *classical RQMC* and *array-RQMC* for this example, with various possibilities for the point set P_n. Here we report some results with $t = 100$. Table 1 gives the estimated variance reduction factors compared with standard MC, i.e., the empirical variance per observation for the MC estimator divided by that of the method considered, for some RQMC methods. All values are rounded to the nearest integer. For the RQMC methods, the sample variance of $m = 100$ independent copies of the average $\bar{Y}_n$ is multiplied by n to estimate the variance *per observation*, i.e., per simulated copy of the Markov chain. The MC variance was estimated by making 100×2^{18} independent simulation runs. The simulations were performed in Java, using the SSJ simulation

library (L'Ecuyer, 2004b).

The values of μ and of the MC variance per run, σ^2, are $\mu = 0.04922$ and $\sigma^2 = 0.0005393$ for $\rho = 0.2$, $\mu = 0.48000$ and $\sigma^2 = 0.06307$ for $\rho = 0.5$, and $\mu = 2.48004$ and $\sigma^2 = 3.1544$ for $\rho = 0.8$. These numbers are accurate (roughly) up to the given digits.

Table 1: Empirical variance reduction factors of RQMC with respect to MC, for the average waiting time of 100 customers, with $n \approx 2^k$ points.

ρ		$k = 10$	$k = 12$	$k = 14$	$k = 16$	$k = 18$	$k = 20$
0.2	Classical-Korobov-Baker	5	8	15	16	59	117
	Classical-Sobol	1	1	3	1	13	28
	Array-Korobov	18	55	49	292	850	2169
	Array-Korobov-Baker	43	159	306	991	3168	10590
	Array-Sobol	87	282	836	3705	10640	47850
	Array-Sobol-NoGray	46	112	276	874	2914	7429
0.5	Classical-Korobov-Baker	10	7	13	6	14	14
	Classical-Sobol	2	1	4	5	9	10
	Array-Korobov	14	46	33	231	686	2034
	Array-Korobov-Baker	44	200	241	1155	3540	15650
	Array-Sobol	123	504	1083	5651	13830	55160
	Array-Sobol-NoGray	55	130	302	1188	3507	11260
0.8	Classical-Korobov-Baker	11	2	15	17	21	26
	Classical-Sobol	3	2	4	6	10	11
	Array-Korobov	15	85	33	337	727	5119
	Array-Korobov-Baker	70	463	287	2225	10080	75920
	Array-Sobol	370	1281	3240	19730	57290	233100
	Array-Sobol-NoGray	117	288	996	4580	13210	48660

If the required CPU time to simulate the mn copies of the chain with RQMC is approximately α

times that required to simulate mn independent copies, and if the variance reduction factor is γ, we can say that MC requires γ/α times more CPU time to achieve a given precision for the estimator of μ. For this example, we have $\alpha \approx 2$ for array-RQMC (due to the overhead of maintaining several copies of the chain in parallel and sorting them) and $\alpha \approx 1$ for classical RQMC.

The table contains results for the classical RQMC approach with a $2(t-1)$-dimensional Korobov lattice rule using parameters taken from Table 1 of L'Ecuyer and Lemieux (2000) (where n is the largest prime smaller than 2^k), to which we applied a random shift modulo 1 followed by the baker's transformation (Hickernell, 2002), denoted *Classical-Korobov-Baker* in the table, and $2(t-1)$-dimensional Sobol nets with $n = 2^k$ randomized by a left (upper triangular) matrix scrambling followed by a random digital shift (L'Ecuyer and Lemieux, 2002; Owen, 2003), denoted *Classical-Sobol*.

For array-RQMC, we give result for the d-dimensional RQMC point set P_n taken as (a) a $(d+1)$-dimensional Korobov lattice rule with its first coordinate skipped, randomized by a random shift modulo 1 (denoted *Array-Korobov*); (b) the same Korobov rule with the random shift followed by baker's transformation (*Array-Korobov-Baker*); (c) the first n points of a Sobol sequence randomized by a left (upper triangular) matrix scrambling followed by a random digital shift (L'Ecuyer and Lemieux, 2002; Owen, 2003), where the points are enumerated by order of their Gray code (*Array-Sobol*); and (d) the same P_n as in (c) but with the points enumerated in their natural order (*Array-Sobol-NoGray*).

For Array-Korobov, the multiplier a of the two-dimensional Korobov lattice rule was selected so that n is prime and a/n is close to the golden ratio. With this choice of a, the rule performs quite well in the two-dimensional spectral test. The points of the Korobov lattice are enumerated by order of the (skipped) first coordinate, so P'_n becomes the original Korobov lattice point set. For instance, if $d = 1$, the points of P_n before the shift are $0, a, 2a, \ldots, (n-1)a$ in that order. The points of the Sobol net are often enumerated by order of their Gray code because this is a bit faster than enumerating them in their natural order (see, e.g., L'Ecuyer, 2004b). Enumerating the points of P_n by their Gray code is equivalent to applying a permutation to the second coordinate of the points of P'_n, i.e., it performs an additional scrambling for this coordinate.

The array-RQMC methods clearly outperform both MC and classical RQMC in this example, even though classical RQMC is already significantly more efficient than MC (up to 100 times more efficient in one case). The improvement factor is larger when the queue has more traffic (i.e., for larger ρ, which is also when the variance is larger) and larger for the Sobol nets than for the Korobov rules. For the Korobov rules, the baker transform really makes a big difference. For the Sobol' nets, the results are better when the points are enumerated in Gray code order. The corresponding scrambling appears helpful, yet we do not have a clear explanation of why.

For classical RQMC, the improvement is much better for the Korobov rules with the baker transform than for the Sobol nets. Without the baker transform (not show here) the Korobov rules are just slightly better than the Sobol nets.

4.2 Increasing d

Table 2 shows the behavior of the estimated variance reduction factors when d increases, with $n \approx 2^{18}$. Here, at each step of the Markov chain, we generate d random variates to compute the waiting times of $d/2$ customers ($d = 1$ represents the case examined in the previous subsection). Note that for "Classical-Korobov," the exact variance reduction factor does not depend on d; the variation observed in the table is only statistical noise. It gives an idea of the precision of our variance-improvement estimators. For Array-Sobol, the variance reduction factors decrease with d, but not so fast. Moreover, the "NoGray" version becomes comparable to the regular one for $d > 2$. The gains are still substantial even for $d = 8$, where the RQMC method approximates 9-dimensional integrals at each step of the Markov chain.

4.3 Random dimension: a Regenerative System

So far in this example, s was fixed at $2(t - 1)$. We now modify the example so that $s = \infty$. Recall that the $M/M/1$ queue is a *regenerative* system that regenerates whenever a customer arrives to an empty system. Each regenerative cycle contains a random and unbounded number of customers. Suppose we want to estimate $\mu = E[Y]$, where we take the following two possibilities for Y: (i) the

Table 2: Estimated variance reduction factors of d-dimensional classical RQMC and array-RQMC with respect to MC, for selected values of d and $n \approx 2^{18}$.

		$d = 1$	$d = 2$	$d = 4$	$d = 8$
0.2	Classical-Korobov-Baker	59	70	73	78
	Classical-Sobol	13	13	12	12
	Array-Korobov-Baker	3168	571	283	137
	Array-Sobol	10640	4329	2247	352
	Array-Sobol-NoGray	2914	5294	2476	403
0.5	Classical-Korobov-Baker	14	22	16	18
	Classical-Sobol	9	6	9	7
	Array-Korobov-Baker	3540	918	152	150
	Array-Sobol	13830	8067	5566	667
	Array-Sobol-NoGray	3507	6206	5205	702
0.8	Classical-Korobov-Baker	21	22	20	28
	Classical-Sobol	10	12	14	10
	Array-Korobov-Baker	10080	2296	1074	597
	Array-Sobol	57290	33360	22550	2515
	Array-Sobol-NoGray	13210	23850	15570	2117

total waiting time of all customers in a regenerative cycle and (ii) the number of customers in a cycle whose waiting time exceeds c, for some constant $c > 0$. Note that changing the uniforms slightly may split or merge regenerative cycles, making Y highly discontinuous in both cases. Moreover, in the second case, Y is integer-valued, so it is not as smooth as in the first case. For our numerical illustration of case (ii), we take $c = 1$. The exact values of μ for case (i) are 0.0625, 1, and 16 for $\rho = 0.2$, 0.5 and 0.8. For case (ii), they are approximately 0.00458, 0.368, and 3.115 for $\rho = 0.2$, 0.5, and 0.8.

Table 3: Estimated variance reduction factors of classical RQMC and array-RQMC with respect to MC, for the regenerative example, case (i).

ρ		$k = 10$	$k = 12$	$k = 14$	$k = 16$	$k = 18$	$k = 20$
0.2	Classical-Korobov-Baker	3	5	6	5	14	24
	Array-Korobov-Baker	13	28	49	116	289	1093
	Array-Sobol	7	21	46	99	239	756
0.5	Classical-Korobov-Baker	2	3	3	1	6	5
	Array-Korobov-Baker	11	16	37	79	159	438
	Array-Sobol	6	11	24	72	228	469
0.8	Classical-Korobov-Baker	1	1	2	1	2	2
	Array-Korobov-Baker	6	12	22	36	151	237
	Array-Sobol	3	5	19	32	92	225

Tables 3 and 4 give the estimated variance reduction factors of classical RQMC and array-RQMC compared with standard MC, again with $m = 100$. The improvement factors are not as large as in the two previous tables, but they are still significant, increase with n, and are much larger for the array versions than for the classical ones. The gain decreases with ρ in case (i) and increases with ρ in case (ii). Note that in case (ii), "$Y > 0$" is a rare event when ρ is very small so in that case something else (such as importance sampling) would have to be done to reduce the variance. For the classical methods, we need infinite-dimensional RQMC point sets, because the number of steps of the chain is unbounded. The Sobol' nets are not infinite-dimensional, but the

Table 4: Estimated variance reduction factors of classical RQMC and array-RQMC with respect to MC, for the regenerative example, case (ii).

ρ		$k = 10$	$k = 12$	$k = 14$	$k = 16$	$k = 18$	$k = 20$
0.2	Classical-Korobov-Baker	1	1	2	2	3	2
	Array-Korobov-Baker	3	5	15	22	72	113
	Array-Sobol	2	5	9	23	46	108
0.5	Classical-Korobov-Baker	3	3	4	2	7	6
	Array-Korobov-Baker	22	35	146	253	540	1655
	Array-Sobol	13	33	85	245	645	1847
0.8	Classical-Korobov-Baker	2	1	3	2	3	3
	Array-Korobov-Baker	16	40	100	76	442	997
	Array-Sobol	10	27	81	198	629	1844

Korobov lattice rules have this property. The successive coordinates of each of their points can be generated by a linear congruential generator as explained in L'Ecuyer and Lemieux (2000).

4.4 Pricing a Bermudan-Asian option

We consider the pricing of a Bermudan-Asian option on a single asset whose value $S(\tau)$ at time τ obeys a geometric Brownian motion:

$$dS(\tau) = rS(\tau)d\tau + \sigma S(\tau)dB(\tau),$$

where r is the risk-free *interest rate*, σ is the (risk-neutral) *volatility* parameter, and $B(\cdot)$ is a standard Brownian motion. The option's value can be written as

$$\mu = E[e^{-rT}C_{\mathrm{a}}(T) \mid S(0)]$$

where

$$C_{\mathrm{a}}(T) = \max\left[0, \left(\frac{1}{s}\sum_{j=1}^{s} S(\tau_j)\right) - K\right]$$

27

and $0 < \tau_1 < \cdots < \tau_s = T$ are the discrete observation times. See, e.g., Hull (2000) and Glasserman (2004) for further details.

We have

$$S(\tau_j) = S(\tau_{j-1}) \exp[(r - \sigma^2/2)(\tau_j - \tau_{j-1}) + \sigma(\tau_j - \tau_{j-1})^{1/2} \Phi^{-1}(U_j)] \qquad (22)$$

for $j = 1, \ldots, s$, where the U_j's are independent $U(0,1)$ random variables and Φ is the standard normal distribution function. To get an unbiased estimator of μ it suffices to generate $S(\tau_1), \ldots, S(\tau_s)$ via (22), with s i.i.d. $U(0,1)$ random variates, and compute the estimator $X = e^{-rT} C_{\mathrm{a}}(T)$.

To apply the *array-RQMC* method, we define the *state* of the chain at step j as the two-dimensional vector $X_j = (S(\tau_j), \bar{S}_j)$, where $\bar{S}_j = (S(\tau_1) + \cdots + S(\tau_j))/j$. We order these states simply by increasing order of their value of $S(\tau_j)$ (there are other possibilities, perhaps better ones).

For a numerical illustration, let $S(0) = 100$, $K = 90$, $T = 120/365$ (in years), $\delta = 1/365$, $t_1 = T - s$, $r = \log 1.09$, $\sigma = 0.2$, and $s = 10$, 60, and 120. Table 5 gives the estimated variance reduction factors of RQMC compared with standard MC. The *classical* RQMC methods uses the straightforward simulation approach described above, with s-dimensional randomized point sets defined as in the previous examples. Efficiency can be further improved by combining RQMC with *bridge sampling* and other variance-reduction techniques such as control variates and importance sampling (Caflisch and Moskowitz, 1995; Glasserman, 2004; L'Ecuyer and Lemieux, 2000; L'Ecuyer, 2004a) but we do not go in that direction here.

Here, classical RQMC is already very effective when s is small, and array-RQMC is not really better. For larger s, however, array-RQMC eventually provides larger variance-reduction factors. On the other hand, it is disappointing to see that these factors eventually stabilize when we increase n. This suggests a $O(1/n)$ asymptotic rate of convergence of the variance for this particular implementation and choice of sorting criterion. Another important point to notice is that one again, the baker transformation applied on top of the Korobov rules really helps.

Table 5: Estimated variance reduction factors of classical RQMC and array-RQMC with respect to MC, for the Asian option.

s		$k=10$	$k=12$	$k=14$	$k=16$	$k=18$	$k=20$
10	Classical-Korobov	188	594	2601	5505	18050	11040
	Classical-Korobov-Baker	2629	10600	5104	83450	27560	93620
	Classical-Sobol	4844	11460	28740	46020	142900	222800
	Array-Korobov-Baker	4783	13280	23960	45990	36670	39950
	Array-Sobol	5080	13030	37460	38320	36360	32430
60	Classical-Korobov	81	17	352	406	552	497
	Classical-Korobov-Baker	481	567	919	610	1362	1745
	Classical-Sobol	282	488	907	787	1654	2413
	Array-Korobov-Baker	1187	1742	1218	2231	1680	1998
	Array-Sobol	1234	1742	2050	2203	2189	1866
120	Classical-Korobov	73	27	152	209	252	276
	Classical-Korobov-Baker	244	380	452	407	581	498
	Classical-Sobol	68	92	234	253	531	410
	Array-Korobov-Baker	816	1263	1355	1736	1456	1635
	Array-Sobol	1423	1485	1260	1390	1333	1477
240	Classical-Korobov	30	9	93	116	95	148
	Classical-Korobov-Baker	76	167	233	303	375	319
	Classical-Sobol	29	32	54	69	151	217
	Array-Korobov-Baker	445	703	375	758	773	601
	Array-Sobol	744	769	670	725	702	667

4.5 Estimating a small ruin probability with importance sampling and array-RQMC

A (simplified) insurance company receives *premiums* at constant rate $c > 0$ and *claims* according to a Poisson process $\{N(t), t \geq 0\}$ with rate $\lambda > 0$. The claim sizes C_j, $j \geq 1$, are i.i.d. random variables with density h. The *reserve* (amount of money in hand) at time t is

$$R(t) = R(0) + ct - \sum_{j=1}^{N(t)} C_j,$$

where $R(0)$ is the initial reserve. We want to estimate the *ruin* probability, i.e., the probability μ that $R(t)$ eventually becomes negative.

Note that ruin can occur only at the time of a claim. The reserve just after claim j is

$$X_j = X_{j-1} + A_j c - C_j, \qquad j \geq 1,$$

where $X_0 = R(0)$ and A_j is the time between claims $j-1$ and j. The process $\{X_j, j \geq 0\}$ is a random walk. This process cannot be simulated directly to estimate the ruin probability because: (1) we cannot be 100% sure that ruin does not occur if we simulate only for a finite time and (2) in practice ruin occurs very rarely. We can get around these difficulties by using *importance sampling* (IS) with *exponential twisting* as follows (Asmussen, 1985). Assuming that h has a finite moment generating function M_h around 0, we replace the density $h(x)$ by

$$h_\theta(x) = h(x) e^{\theta x} / M_h(\theta)$$

and increase the rate λ of the Poisson process to

$$\lambda_\theta = \lambda + \theta c,$$

where θ is the largest solution to the Lundberg equation $M_h(\theta) = (\lambda + \theta c)/\lambda$. Under this IS scheme, ruin occurs with probability 1 and the (unbiased) estimator of μ is

$$e^{\theta(X_\tau - X_0)}$$

where $\tau = \inf\{j : X_j < 0\}$.

Table 6: Estimated variance reduction factors for the ruin probability example.

c		$k = 10$	$k = 12$	$k = 14$	$k = 16$	$k = 18$	$k = 20$
10	Classical-Korobov-Baker	1	1	1	1	1	1
	Array-Korobov-Baker	3	3	7	3	15	27
	Array-Sobol	2	2	6	10	19	45
5	Classical-Korobov-Baker	1	1	1	1	2	1
	Array-Korobov-Baker	3	4	10	5	21	37
	Array-Sobol	2	4	8	13	33	73
3	Classical-Korobov-Baker	1	1	1	1	1	1
	Array-Korobov-Baker	2	4	8	7	24	38
	Array-Sobol	2	5	7	17	30	49

We are interested in seeing if a combination of IS with array-RQMC is more efficient than IS alone. Since the function

$$f(\mathbf{u}) = e^{\theta(X_\tau - X_0)}$$

is *sawtooth-like* (not smooth at all) with respect to each coordinate of $\mathbf{u}$, we do not expect RQMC to perform well a priori. For a numerical experiment, we take $\lambda = 1$, exponential claim sizes with mean $1/\beta = 2$, and $R(0) = 200$. We use $d = 1$, i.e., one step of the chain each time a uniform random number is generated. The number of steps before ruin occurs is random.

Table 6 gives the estimated variance reduction factors of classical RQMC and array-RQMC compared with MC, both with IS, for $c = 3$, 5, and 10. For classical RQMC we need an infinite-dimensional rule and we use Korobov as in Section 4.3. The exact ruin probability μ is approximately $\mu = 2.2 \times 10^{-15}$ for $c = 3$, $\mu = 3.5 \times 10^{-27}$ for $c = 5$, and $\mu = 3.6 \times 10^{-36}$ for $c = 10$. The gains are not as spectacular as for the previous examples, but they are nevertheless significant.

4.6 Conclusion

We have proposed a new RQMC method for Markov chains, proved some results on its convergence, and tested it numerically on a few examples. The new method provides large efficiency gains compared with standard MC in all the examples we tried and performs much better than classical RQMC in the examples where the Markov chain has a one-dimensional state space. Generally speaking, the performance of the array-RQMC method tends to degrade when the integrand has higher variability, or when the dimension of the state space becomes larger than 1 and there is no obvious "natural order" for the states. But even in these cases, there can be significant variance reductions compared with standard MC, and often compared with classical RQMC as well.

Our paper also provides novel empirical evidence of the effectiveness of applying the baker transformation over a randomly shifted lattice rule, an idea that was studied theoretically by Hickernell (2002).

Obtaining better convergence bounds for the variance is a goal that would certainly deserve further work. From the practical viewpoint, an interesting challenge would be to find good ways of ordering the states for specific classes of problems where the Markov chain has a multidimensional state space. Our on-going and future work also includes studying the application of array-RQMC to other settings that fit a general Markov chain framework, such as Markov Chain Monte Carlo methods and stochastic approximation algorithms, for example.

5 Acknowledgments

References

Asmussen, S. (1985). Conjugate processes and the simulation of ruin problems. *Stochastic Processes and their Applications*, 20:213–229.

Ben-Ameur, H., L'Ecuyer, P., and Lemieux, C. (2004). Combination of general antithetic transformations and control variables. *Mathematics of Operations Research*, 29(4):946–960.

Caflisch, R. E. and Moskowitz, B. (1995). Modified Monte Carlo methods using quasi-random sequences. In Niederreiter, H. and Shiue, P. J.-S., editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, number 106 in Lecture Notes in Statistics, pages 1–16, New York. Springer-Verlag.

Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*. Springer-Verlag, New York.

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

Hull, J. (2000). *Options, Futures, and Other Derivative Securities*. Prentice-Hall, Englewood-Cliff, N.J., fourth edition.

Lécot, C. (1996). Error Bound for Quasi-Monte Carlo Integration with Nets. *Mathematics of Computation*, 65(213):179–187.

Lécot, C. and Ogawa, S. (2002). Quasirandom walks methods. In Fang, K.-T., Hickernell, F. J., and Niederreiter, H., editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 63–85, Berlin. Springer-Verlag.

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

L'Ecuyer, P. (2004a). Quasi-Monte Carlo methods in finance. In Ingalls, R. G., Rossetti, M. D., Smith, J. S., and Peters, B. A., editors, *Proceedings of the 2004 Winter Simulation Conference*, Piscataway, New Jersey. IEEE Press.

L'Ecuyer, P. (2004b). *SSJ: A Java Library for Stochastic Simulation.* Software user's guide, Available at `http://www.iro.umontreal.ca/~lecuyer`.

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

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

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

Niederreiter, H. (1992). *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63 of *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia.

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

Owen, A. B. (2003). Variance with alternative scramblings of digital nets. *ACM Transactions on Modeling and Computer Simulation*, 13(4):363–378.

Owen, A. B. (2004). Multidimensional variation for quasi-Monte Carlo. Manuscript.

Wilson, J. R. (1983). Antithetic sampling with multivariate inputs. *American Journal of Mathematical and Management Sciences*, 3:121–144.