Randomized Dimension Reduction for Monte Carlo Simulations

Nabil Kahalé *

December 14, 2017

Abstract

We present a new unbiased algorithm that estimates the expected value of \( f(U) \) via Monte Carlo simulation, where \( U \) is a vector of \( d \) independent random variables, and \( f \) is a function of \( d \) variables. We assume that \( f \) does not depend equally on all its arguments. Under certain conditions we prove that, for the same computational cost, the variance of our estimator is lower than the variance of the standard Monte Carlo estimator by a factor of order \( d \). Our method can be used to obtain a low-variance unbiased estimator for the expectation of a function of the state of a Markov chain at a given time-step. We study applications to volatility forecasting and time-varying queues. Numerical experiments show that our algorithm dramatically improves upon the standard Monte Carlo method for large values of \( d \), and is highly resilient to discontinuities.

Keywords: dimension reduction; variance reduction; effective dimension; Markov chains; Monte Carlo methods

1 Introduction

Markov chains arise in a variety of fields such as finance, queuing theory, and social networks. While much research has been devoted to the study of steady-states of Markov chains, several practical applications rely on the transient behavior of Markov chains. For example, the volatility of an index can be modelled as a Markov chain using the GARCH model (Hull 2014, Ch. 23). Financial institutions conducting stress tests may need to estimate the probability that the volatility exceeds a given level in a few years from now. Also, due to the nature of human activity, queuing systems in areas such as health-care, manufacturing, telecommunication and transportation networks, have often time-varying features and do not have a steady-state. For instance, empirical data show significant daily variation in traffic in wide-area networks (Paxson 1994, Thompson, Miller and Wilder 1997) and vehicular flow on roads (Nagel, Wagner and Woesler 2003). Estimating the expected delay of packets in a wide-area network at a specific time of the day (12pm, say) could be used to dimension such networks. Similarly, estimating the velocity of cars in a region at 6pm could be used to design transportation networks. In the same vein, consider the problem of estimating the queue-length at the end of a business day in a call center that operates with fixed hours. In such call centers, knowing how many calls would still need to be answered at 5pm could be an important metric that would be needed in estimating their staffing requirements. Methods to determine appropriate staffing levels in call centers and other many-server queueing systems with time-varying arrival rates have been designed in (Feldman, Mandelbaum, Massey and Whitt 2008). Also, approximation tools have been developed to study time-varying queues (see (Whitt 2017) and references therein). However, in many situations, there are no analytical tools, except Monte Carlo simulation, to study accurately systems modeled by a Markov chain. A drawback of Monte Carlo simulation is its

*ESCP Europe, Big data research center, and Labex ReFi, 75011 Paris, France; e-mail: nkahale@escpeurope.eu.
high computation cost. This motivates the need to design efficient simulation tools to study the transient behavior of Markov chains, with or without time-varying features.

This paper gives a new unbiased algorithm to estimate $E(f(U))$, where $U = (U_1, \ldots, U_d)$ is a vector of $d$ independent random variables $U_1, \ldots, U_d$ taking values in a measurable space $F$, and $f$ is a real-valued Borel-measurable function on $F^d$ such that $f(U)$ is square-integrable. For instance, $F$ can be equal to $\mathbb{R}$ or to any vector space over $\mathbb{R}$. Under certain conditions, we show that our algorithm yields substantial lower variance than the standard Monte Carlo method for the same computational effort. Our techniques can be used to efficiently estimate the expected value of a function of the state of a Markov chain at a given time-step $d$, for a class of Markov chains driven by independent random variables. An alternative algorithm for Markov chains estimation, based on Quasi-Monte Carlo sequences, that substantially improves upon standard Monte Carlo in certain numerical examples, is given in (L’Ecuyer, Lécot and Tuffin 2008), with bounds on the variance proven for special situations where the state space of the chain is a subset of the real numbers.

In a standard Monte Carlo scheme, $E(f(U))$ is estimated by simulating $n$ independent vectors in $F^d$ having the same distribution as $U$, and taking the average of $f$ over the $n$ vectors. In the related Quasi-Monte Carlo method (see (Glasserman 2004, Ch. 5)), $f$ is evaluated at a predetermined deterministic sequence of points. In several applications, the efficiency of Quasi-Monte Carlo algorithms can be improved by reordering the $U_i$’s and/or making a change of variables, so that the value of $f(U)$ depends mainly on the first few $U_i$’s. For instance, the Brownian bridge construction and principal components analysis have been used (Caflisch, Morokoff and Owen 1997, Acworth, Broadie and Glasserman 1998, Åkesson and Lehoczky 2000) to reduce the error in the valuation of financial derivatives via Quasi-Monte Carlo methods (see (Caflisch 1998) for related results). The relative importance of the first variables can formally be measured by calculating the effective dimension in the truncation sense, a concept defined in (Caflisch, Morokoff and Owen 1997): when the first variables are important, the effective dimension in the truncation sense is low in comparison to the nominal dimension. It is proven in (Sloan and Woniakowski 1998) that Quasi-Monte Carlo methods are effective for a class of functions where the importance of $U_i$ decreases with $i$. The truncation dimension and a related notion, the effective dimension in the superposition sense, are studied in (Sobol 2001, Owen 2003, Liu and Owen 2006). It is shown in (Wang and Fung 2003, Wang and Sloan 2005, Wang 2006) that the Brownian bridge and/or principal components analysis algorithms substantially reduce the truncation dimension of certain financial instruments. Alternative linear transformations have been proposed in (Imai and Tan 2006, Wang and Sloan 2011, Wang and Tan 2013) to reduce the effective dimension of financial derivatives and improve the performance of Quasi-Monte Carlo methods.

It is well known that the previously mentioned change of variables techniques do not modify the variance of the standard Monte Carlo estimator, even though they decrease the error in Quasi-Monte Carlo schemes. On the other hand, the multilevel Monte Carlo (MLMC) method introduced in (Giles 2008), which relies on strong, low dimensional approximations of the function to be estimated, dramatically reduces the computational complexity of estimating an expected value arising from a stochastic differential equation. Related randomized multilevel methods that produce unbiased estimators for equilibrium expectations of functionals defined on homogeneous Markov chains have been provided in (Glynn and Rhee 2014). These methods apply to the class of positive Harris recurrent Markov chains, and to chains that are contracting on average. It is shown in (Rhee and Glynn 2015) that similar randomized multilevel methods can be used to efficiently compute unbiased estimators for expectations of functionals of solutions to stochastic differential equations. The MLMC method has had numerous other applications (e.g., (Rosenbaum and Staum 2017)).

The basic idea behind our algorithm is that, if $f$ does not depend equally on all its arguments, the standard Monte Carlo method can be inefficient because it simulates all $d$ arguments of $f$
at each iteration. In contrast, our algorithm simulates at each iteration a random subset of arguments of \( f \), and reuses the remaining arguments from the previous iteration. Under certain conditions, we show that \( E(f(U)) \) can be estimated with variance \( \epsilon^2 \) in \( O(d + \epsilon^{-2}) \) expected time. In comparison, assuming that the expected time needed to simulate \( f(U) \) is of order \( d \) and that the variance of \( f(U) \) is upper and lower-bounded by constants, the time needed to achieve variance \( \epsilon^2 \) by standard Monte Carlo simulation is \( \Theta(de^{-2}) \).

In order to optimize the tradeoff between the statistical error and the running time, we use a new geometric algorithm that solves in \( O(d) \) time a \( d \)-dimensional optimization problem. Our geometric algorithm is of independent interest and can be used to solve an optimization problem of the same type that was solved in (Rhee and Glynn 2015, Section 3) in \( O(d^3) \) time. We are not aware of other previous algorithms that solve this problem. This work extends the research in (Kahalé 2016).

The rest of the paper is organized as follows. §2 presents our generic randomized dimension reduction algorithm and analyse its performance. §3 describes the aforementioned geometric algorithm and gives a numerical implementation of the randomized dimension reduction algorithm. §4 provides applications to Markov chains. §5 presents numerical simulations. §6 compares our algorithm to a class of MLMC algorithms. Concluding remarks are given in §7. Most proofs are contained in the appendix.

2 The generic randomized dimension reduction algorithm

2.1 The algorithm description

We assume that all random variables in this paper are defined on the same probability space. Our algorithm estimates \( E(f(U)) \) by performing \( n \) iterations, where \( n \) is an arbitrary positive integer. The algorithm samples more often the first arguments of \( f \) than the last ones. It implicitly assumes that, roughly speaking, the importance of the \( i \)-th argument of \( f \) decreases with \( i \). In many Markov chain examples, the last random variables are more important than the first ones, but our algorithm can still be used efficiently after re-ordering the random variables, as described in detail in §4.

Let

\[
A = \{(q_0, \ldots, q_{d-1}) \in \mathbb{R}^d : 1 = q_0 \geq q_1 \geq \cdots \geq q_{d-1} > 0\}.
\]

Throughout the paper, \( q = (q_0, \ldots, q_{d-1}) \) denotes an element of \( A \). Our generic algorithm takes such a vector \( q \) as parameter. Let \( (N_k) \), \( k \geq 1 \), be a sequence of independent random integers in \([1, d]\) such that \( \Pr(N_k > i) = q_i \) for \( 0 \leq i \leq d - 1 \) and \( k \geq 1 \). The algorithm simulates \( n \) copies \( V^{(1)}, \ldots, V^{(n)} \) of \( U \) and consists of the following steps:

1. First iteration. Simulate a vector \( V^{(1)} \) that has the same distribution as \( U \) and calculate \( f(V^{(1)}) \).

2. Loop. In iteration \( k + 1 \), where \( 1 \leq k \leq n - 1 \), let \( V^{(k+1)} \) be the vector obtained from \( V^{(k)} \) by redrawing the first \( N_k \) components of \( V^{(k)} \), and keeping the remaining components unchanged. Calculate \( f(V^{(k+1)}) \).

3. Output the average of \( f(V^{(1)}), \ldots, f(V^{(n)}) \).

More formally, consider a sequence \( (U^{(k)}) \), \( k \geq 1 \), of independent copies of \( U \) such that the two sequences \( (N_k) \), \( k \geq 1 \), and \( (U^{(k)}) \), \( k \geq 1 \), are independent. Define the sequence \( (V^{(k)}) \), \( k \geq 1 \), in \( F^d \) as follows: \( V^{(1)} = U^{(1)} \) and, for \( k \geq 1 \), the first \( N_k \) components of \( V^{(k+1)} \) are the same as the corresponding components of \( U^{(k+1)} \), and the remaining components of \( V^{(k+1)} \) are the same as the corresponding components of \( V^{(k)} \). The algorithm then outputs

\[
f_n \triangleq \frac{f(V^{(1)}) + \cdots + f(V^{(n)})}{n}.
\]
Note that \( f_n \) is an unbiased estimator of \( E(f(U)) \) since \( V^{(k)} \overset{d}{=} U \) for \( 1 \leq k \leq n \).

### 2.2 Performance analysis

For ease of presentation, we ignore the time needed to generate \( N_k \) and the running time of the third step of the algorithm. For \( 1 \leq i \leq d \), let \( t_i \) be an upper bound on the expected time needed to generate \( V^{(k+1)} \) and calculate \( f(V^{(k+1)}) \) when \( N_k = i \). Equivalently, \( t_i \) is an upper-bound on the expected time needed to perform Step 2 of the algorithm when \( N_k = i \). Thus, \( t_i \) is an upper bound on the expected time needed to re-draw the first \( i \) components of \( U \) and recalculate \( f(U) \), and \( t_d \) is an upper bound on the expected time needed to simulate \( f(U) \).

By convention, \( t_0 = 0 \). We will assume for simplicity that \( t_d \) is a strictly increasing function of \( i \). In many examples (see \( \S 2.4 \) and \( \S 4 \)), it can be shown that \( t_i = O(i) \). As \( \Pr(N_k = i) = q_{i-1} - q_i \) for \( 1 \leq i \leq d \) and \( k \geq 1 \), where \( q_d = 0 \), the expected running time of a single iteration of our algorithm, excluding the first one, is upper bounded by \( T \), where

\[
T \triangleq \sum_{i=1}^{d} (q_{i-1} - q_i) t_i = \sum_{i=0}^{d-1} q_i (t_{i+1} - t_i). \tag{2.1}
\]

For \( 0 \leq i \leq d \), define

\[
C(i) \triangleq \text{Var}(E(f(U)|U_{i+1}, \ldots, U_d)).
\]

Thus, \( C(0) = \text{Var}(f(U)) \), while \( C(d) = 0 \), and we can interpret \( C(i) \) as the variance captured by the last \( d - i \) components of \( U \). Roughly speaking, if the last \( d - i \) arguments of \( f \) are not important, the variance \( C(i) \) of the conditional expectation \( E(f(U)|U_{i+1}, \ldots, U_d) \) should be small. Proposition \( 2.1 \) below shows that \( (C(i)) \), \( 0 \leq i \leq d \), is always a decreasing sequence, and gives an alternative expression for \( C(i) \), which can be viewed as a variant of Theorem 2 of (Sobol 2001).

**Proposition 2.1.** The sequence \( (C(i)) \), \( 0 \leq i \leq d \), is decreasing. If \( U'_1, \ldots, U'_d \) are random variables such that \( U'_j \overset{d}{=} U_j \) for \( 1 \leq j \leq i \), and \( U'_1, \ldots, U'_d, U \) are independent, then

\[
C(i) = \text{Cov}(f(U), f(U'_1, \ldots, U'_i, U_{i+1}, \ldots, U_d)). \tag{2.2}
\]

Theorem \( 2.1 \) below establishes a formal relationship between the variance of \( f_n \) and the \( C(i) \)’s. Let \( \nu^* \) be the vector of \( \mathbb{R}^{d+1} \) with \( \nu^*_0 = C(0) \) and \( \nu^*_i = 2C(i) \) for \( 1 \leq i \leq d \).

**Theorem 2.1.** For \( n \geq 1 \),

\[
n \text{Var}(f_n) \leq \sum_{i=0}^{d-1} \frac{\nu^*_i - \nu^*_{i+1}}{q_i}. \tag{2.3}
\]

Furthermore, the LHS of \( (2.3) \) converges to its RHS as \( n \) goes to infinity.

As, for \( \nu = (\nu_0, \ldots, \nu_d) \in \mathbb{R}^d \times \{0\} \),

\[
\sum_{i=0}^{d-1} \frac{\nu_i - \nu_{i+1}}{q_i} = \nu_0 + \sum_{i=1}^{d-1} \nu_i \left( \frac{1}{q_i} - \frac{1}{q_{i-1}} \right), \tag{2.4}
\]

the RHS of \( (2.3) \) is a weighted combination of the \( C(i) \)'s, with positive weights. Thus, the smaller the \( C(i) \)'s, the smaller the RHS of \( (2.3) \). Furthermore, as \( C(i) \) is the variance of the conditional expectation \( E(f(U)|U_{i+1}, \ldots, U_d) \), which can be considered as a smoothed version of \( f(U) \), we expect our algorithm to be resilient to discontinuities of \( f \).

We use \( \mathbb{R}_+ \) to denote the set of nonnegative real numbers. For \( \nu = (\nu_0, \ldots, \nu_d) \in \mathbb{R}_+^d \times \{0\} \), and \( q \in A \), set

\[
R(q; \nu) = \left( \sum_{i=0}^{d} q_i (t_{i+1} - t_i) \right) \left( \sum_{i=0}^{d-1} \frac{\nu_i - \nu_{i+1}}{q_i} \right). \tag{2.5}
\]
The expected time needed to perform \( n \) iterations of the algorithm, including the first one, is at most \( T_n \), where \( T_n \equiv (n - 1)T + t_d \). Theorem 2.1 and 2.1 imply that \( T_n \text{Var}(f_n) \) converges to \( R(q; \nu^*) \) as \( n \) goes to infinity. By (2.4), \( R(q; \nu) \) is an increasing function with respect to \( \nu \), i.e. \( R(q; \nu) \leq R(q; \nu') \) for \( \nu \leq \nu' \), where the symbol \( \leq \) between vectors represents componentwise inequality. Let \( T^\text{tot}(q, \epsilon) \) be the total expected time it takes for our algorithm to guarantee that \( \text{Std}(f_n) \leq \epsilon \). Corollary 2.1 below gives an upper bound on \( T^\text{tot}(q, \epsilon) \) in terms of \( R(q; \nu^*) \).

Furthermore, if \( \epsilon > 0 \),

\[
T^\text{tot}(q, \epsilon) \leq t_d + R(q; \nu^*)\epsilon^{-2}. 
\]  

(2.6)

In light of above, we will use \( R(q; \nu^*) \) to measure the performance of our algorithm. The smaller the \( C(i)'s \) and \( t_i's \), the smaller \( R(q; \nu^*) \), and the higher the performance of our algorithm. Proposition 2.3 below gives upper bounds on \( R(q; \nu^*) \) for \( 1 \leq i \leq d \), if \( f_i \) is a measurable function from \( F^i \) to \( \mathbb{R} \) such that \( f_i(U_1, \ldots, U_i) \) is square-integrable, then

\[
C(i) \leq \text{Var}(f(U) - f_i(U_1, \ldots, U_i)). 
\]

2.3 Explicit and semi-explicit distributions

An optimal choice for \( q \) is a one that minimizes \( R(q; \nu^*) \). A numerical algorithm that performs such minimization is presented in §3. This subsection gives explicit or semi-explicit choices for \( q \), with corresponding upper-bounds on \( R(q; \nu^*) \).

Proposition 2.3 below gives upper bounds on \( R(q; \nu^*) \) if \( t_i = O(i) \) and \( (C(i)) \) decreases at a sufficiently high rate. It implies in particular that, if \( t_i = O(i) \) and \( (C(i)) = O((i + 1)^\gamma) \) with \( \gamma < -1 \), then \( T^\text{tot}(q, \epsilon) = O(d + \epsilon^{-2}) \).

Proposition 2.3. Assume that \( d \geq 2 \) and there are constants \( c \) and \( c' \) and \( \gamma < 0 \) independent of \( d \) such that \( t_i = ci \) and \( C(i) \leq c'(i + 1)^\gamma \) for \( 0 \leq i \leq d \). Then, for \( q_i = (i + 1)^{\gamma - 1}/2 \), \( 0 \leq i \leq d - 1 \), there is a constant \( c_i \) independent of \( d \) such that

\[
R(q; \nu^*) \leq \begin{cases} 
  c_1, & \gamma < -1, \\
  c_1 \ln^2(d), & \gamma = -1, \\
  c_1 d^{\gamma + 1}, & -1 < \gamma < 0.
\end{cases} 
\]  

(2.8)

When upper-bounds on the \( C(i)'s \) satisfying a convexity condition are known, Proposition 2.3 below gives a vector \( q \) which can be shown to be optimal (see Theorem 3.1).
Proposition 2.4. Assume that \( \nu_0, \ldots, \nu_{d-1} \) are positive real numbers such that \( C(i) \leq \nu_i \) for \( 0 \leq i \leq d - 1 \), and that the sequence

\[
\theta_i = \frac{\nu_{i+1} - \nu_i}{t_{i+1} - t_i},
\]

\( 0 \leq i \leq d - 1 \), is increasing (by convention, \( \nu_d = 0 \)). Then, for \( q_i = \sqrt{\theta_i/\theta_0}, 0 \leq i \leq d - 1 \),

\[
R(q; \nu^*) \leq 2 \left( \sum_{i=0}^{d-1} \sqrt{(\nu_i - \nu_{i+1})(t_{i+1} - t_i)} \right)^2.
\]

Proof. We first observe that \( \theta_i \leq \theta_{d-1} < 0 \) for \( 0 \leq i \leq d - 1 \). Thus \( q \) is well-defined and belongs to \( A \). As \( \nu^* \leq 2\nu \) and \( R(q;.) \) is increasing with respect to its second argument, it follows that \( R(q; \nu^*) \leq R(q; 2\nu) \). This concludes the proof. \( \square \)

Proposition 2.5 below yields an upper bound on \( \sqrt{R(q; \nu^*)} \) in terms of a weighted sum of the square roots of the \( C(i) \)'s, for a semi-explicit vector \( q \).

Proposition 2.5. Assume that \( C(d - 1) > 0 \). If, for \( 0 \leq i \leq d - 1 \),

\[
q_i = \sqrt{\frac{t_i C(i)}{t_{i+1} C(0)}},
\]

then

\[
R(q; \nu^*) \leq 8 \left( \sum_{i=0}^{d-1} \left( \sqrt{t_{i+1}} - \sqrt{t_i} \right) \sqrt{C(i)} \right)^2. \tag{2.9}
\]

Proposition 2.6 below gives an explicit distribution which is optimal up to a logarithmic factor, without requiring any prior knowledge on the \( C(i) \)'s.

Proposition 2.6. For any \( q \in A \),

\[
R(q; \nu^*) \geq \sum_{i=0}^{d-1} C(i)(t_{i+1} - t_i). \tag{2.10}
\]

Furthermore, if \( q_i = t_1/t_{i+1} \) for \( 0 \leq i \leq d - 1 \), then

\[
R(q; \nu^*) \leq 2(1 + \ln(t_d/t_1)) \sum_{i=0}^{d-1} C(i)(t_{i+1} - t_i). \tag{2.11}
\]

2.4 A Lipschitz function example

Assume that \( F = \mathbb{R} \) and that \( U_1, \ldots, U_d \) are square-integrable real-valued random variables, with \( \sigma_1 \geq \cdots \geq \sigma_d > 0 \), where \( \sigma_i \) is the standard deviation of \( U_i \). Assume also that \( f(x_1, \ldots, x_d) = g(\sum_{j=1}^{d} x_j) \) for \( (x_1, \ldots, x_d) \in \mathbb{R}^d \), where \( g \) is a real-valued 1-Lipschitz function on \( \mathbb{R} \) that can be calculated in constant time. For instance, \( f(x_1, \ldots, x_d) = \max(\sum_{j=1}^{d} x_j - K, 0) \), where \( K \) is a constant, satisfies this condition. Assume further that each \( U_i \) can be simulated in constant time. For \( 1 \leq k \leq n \), let \( S_k \) be the sum of all components of \( V^{(k)} \). Thus \( S_{k+1} \) can be calculated recursively in \( O(N_k) \) time by adding to \( S_k \) the first \( N_k \) components of \( V^{(k+1)} \) and subtracting the first \( N_k \) components of \( V^{(k)} \). Hence, we can set \( t_i = ci \), for some constant \( c \).
In order to bound the $C(i)$'s, we show that $f(U)$ can be approximated by $f_i(U_1, \ldots, U_i)$, where $f_i(x_1, \ldots, x_i) = g(\sum_{j=1}^i x_j + \sum_{j=i+1}^d E(U_j))$ for $(x_1, \ldots, x_i) \in \mathbb{R}^i$. Let $||Z|| = \sqrt{E(Z^2)}$ for a real-valued random variable $Z$. By Proposition 2.4, with $\nu = \{\nu_t \}$ below the segment $[(\nu_1, \nu_0), (\nu_d, \nu_0)]$, such that (3.1) holds, calculate recursively the ordered subset $B$ for a real-valued random variable $Z$. Then $C(i) \leq ||f(U) - f_i(U_1, \ldots, U_i)||^2 = \| \sum_{j=i+1}^d (U_j - E(U_j)) \|^2 = \text{Var}(\sum_{j=i+1}^d U_j) = \sum_{j=i+1}^d \sigma_j^2$.

The second equation follows from the assumption that $g$ is 1-Lipschitz. By applying Proposition 2.4 with $\nu_i = \sum_{j=i+1}^d \sigma_j^2$, and setting $q_i = \sigma_i/\sigma_1, 0 \leq i \leq d - 1$, we infer that

$$R(q; \nu^*) \leq 2c(\sum_{i=1}^d \sigma_i^2).$$

Thus, if $\sigma_i = O(i^\gamma)$, with $\gamma < -1$, then $R(q; \nu^*) = O(1)$ and $T^{\text{tot}}(q, \epsilon) = O(d + \epsilon^{-2})$.

3 The optimal distribution

We now seek to calculate a vector $q$ that minimizes $R(q; \nu^*)$, in the same spirit as stratified sampling (see (Glasserman 2004, Section 4.3)), MLMC (Giles 2008), and related methods (Rhee and Glynn 2015). Given a vector $\nu$ in $\mathbb{R}^d \times \{0\}$ whose first $d$ components are positive, Theorem 3.1 below gives a geometric algorithm that finds in $O(d)$ time a vector $q^*$ that minimizes $R(q; \nu)$ under the constraint that $q \in A$. In (Rhee and Glynn 2015, Section 3), a dynamic programming algorithm that calculates such a vector $q^*$ in $O(d^2)$ time has been described.

Let $\nu' = (\nu'_0, \ldots, \nu'_d) \in \mathbb{R}^{d+1}$ be such that the set $\{(t_i, \nu'_i) : 0 \leq i \leq d\}$ forms the lower hull of the set $\{(t_i, \nu_i) : 0 \leq i \leq d\}$. In other words, $\nu'$ is the supremum of all sequences in $\mathbb{R}^{d+1}$ such that $\nu' \leq \nu$ and the sequence $(\theta_i)$ is increasing, where

$$\theta_i = \frac{\nu'_{i+1} - \nu'_i}{t_{i+1} - t_i},$$

(3.1)

$0 \leq i \leq d - 1$. For instance, if $d = 6$, with $t_i = i$ and $\nu = (20, 21, 13, 8, 7, 2, 0)$, then $\nu' = (20, 16, 12, 8, 5, 2, 0)$, as illustrated in Fig. 1. Fig. 1 shows how to calculate $\nu'$ in $O(d)$ time.

**Theorem 3.1.** Let $\nu$ be a vector in $\mathbb{R}^d \times \{0\}$ whose first $d$ components are positive. For $0 \leq i \leq d - 1$, set $q^*_i = \sqrt{\theta_i/\theta_0}$, where $\theta_i$ is given by (3.1), and let $q^* = (q^*_1, \ldots, q^*_{d-1})$. Then $q^* = \arg \min_{q \in A} R(q; \nu)$, and

$$R(q^*; \nu) = \left(\sum_{i=0}^{d-1} \sqrt{(\nu'_{i+1} - \nu'_i)(t_{i+1} - t_i)})^2.\right.$$  \hfill (3.2)

3.1 Lower hull calculation

Given $\nu$, the following algorithm, due to (Andrew 1979), calculates $\nu'$ in $O(d)$ time. First, calculate recursively the ordered subset $B(j)$ of $\{1, \ldots, d\}$, $2 \leq j \leq d$, as follows. Let $B(2) = \{1, 2\}$. Assume $B(j - 1) = \{i_1, \ldots, i_m\}$. Let $k$ be the largest index such that $(\nu_k, \nu_{k-1})$ lies below the segment $[(t_{i_{k-1}}, \nu_{i_{k-1}}), (t_j, \nu_j)]$, if such an index exists, otherwise let $k = 1$. Set $B(j) = \{i_1, \ldots, i_k, j\}$. For $1 \leq i \leq d$, let $i'$ and $i''$ be two elements of $B(d)$ with $i' \leq i \leq i''$. Set $\nu'_i$ so that $(t_i, \nu'_i)$ lies on the segment $[(t_{i'}, \nu_{i'}), (t_{i''}, \nu_{i''})]$. 


3.2 Estimating the $C(i)$’s

The calculation of $q^*$ requires the knowledge of the $C(i)$’s. Proposition 3.1 below can be used to estimate $C(i)$ via Monte Carlo simulation. Assuming that $f(U)$ is strongly approximated by a function of its first $i$ arguments, we expect that both components of the product in the RHS of (3.3) to be small, on average. Thus, (3.3) can be considered as a “control variate” version of (2.2), and should yield a more accurate estimate of $C(i)$ via Monte Carlo simulation for large values of $i$.

**Proposition 3.1.** Assume that $U'_1, \ldots, U'_d$, and $U''_{i+1}, \ldots, U''_d$, are random variables such that $U'_j = U_j$ for $1 \leq j \leq d$, and $U'_j = U_{j+1}$ for $i+1 \leq j \leq d$, and $U'_1, \ldots, U'_d$, $U$, $U''_{i+1}, \ldots, U''_d$ are independent. Then

$$C(i) = E((f(U) - f(U_1, \ldots, U_i, U'_{i+1}, \ldots, U'_d))$$

$$- (f(U'_1, \ldots, U_{i+1}, \ldots, U'_d) - f(U'_1, \ldots, U'_i, U''_{i+1}, \ldots, U''_d)).$$

(3.3)

### Numerical algorithm

Building upon the previously discussed elements, the algorithm that we have used for our numerical experiments is as follows. It constructs a vector $(\nu_0, \ldots, \nu_d)$ and uses it as a proxy for $\nu^*$.

1. For $i = 0$ to $d - 1$, if $i + 1$ is a power of 2, estimate $C(i)$ by Monte Carlo simulation with 1000 samples via Proposition 3.1.
2. Set $\nu_0 = C(0)$, and $\nu_d = 0$. For $1 \leq i \leq d - 1$, let $\nu_i = 2C(j)$, where $j$ is the largest index in $[0, i]$ such that $j + 1$ is a power of 2.
3. For $i = d - 1$ down to 1, set $\nu_i \leftarrow \max(\nu_i, \nu_{i+1})$. Set $\nu_0 \leftarrow \max(\nu_0, \nu_1/2)$.
4. Let $(\nu'_0, \ldots, \nu'_d) \in \mathbb{R}^{d+1}$ be such that the set $\{(t_i, \nu'_i) : 0 \leq i \leq d\}$ forms the lower hull of the set $\{(t_i, \nu_i) : 0 \leq i \leq d\}$. For $0 \leq i \leq d - 1$, set $q_i = \sqrt{\theta_i/\theta_0}$, where $\theta_i$ is given by (3.1).
5. Calculate $T$ via (2.1). For $0 \leq i \leq d - 1$, set

$$q_i \leftarrow \min(1, \max(q_i, \frac{T}{t_{i+1} \ln(t_d/t_1)})).$$
6. Run steps 1 through 3 of the generic randomized dimension reduction algorithm of \(^{(2.1)}\) using \(q\).

The purpose of Steps 3 and 5 is to reduce the impact on \(q\) of statistical errors that arise in Step 1. Using \((2.1)\) and the proof of Proposition \(2.6\) and assuming that \(t_d \geq 2t_1\), it can be shown that Step 5 increases \(T\) by at most a constant multiplicative factor. An alternative way to implement our algorithm is to skip Steps 1 through 5 and run the generic algorithm with \(q_i = t_i/t_{i+1}\) for \(0 \leq i \leq d - 1\). By Proposition \(2.6\) the resulting vector \(q\) is optimal up to a logarithmic factor.

4 Applications to Markov chains

In queueing systems, the performance metrics at a specific time instant are heavily dependent on the last busy cycle, i.e., the events that occurred after the queue was empty for the last time. Thus, the performance metrics depend a lot more on the last random variables driving the system than on the initial ones. Nevertheless, we can apply our algorithm to queueing systems by using a time-reversal transformation inspired from (Glynn and Rhee 2014). More generally, using such a time-reversal transformation, this section shows that our algorithm can efficiently estimate the expected value of a function of the state of a Markov chain at time-step \(d\), for a class of Markov chains driven by independent random variables.

Let \((X_m), 0 \leq m \leq d\), be a Markov chain with state-space \(F'\) and deterministic initial value \(X_0\). Assume that there are independent random variables \(Y_i, 0 \leq i \leq d - 1\), that take values in \(F\), and measurable functions \(g_i\) from \(F' \times F\) to \(F'\) such that \(X_{i+1} = g_i(X_i, Y_i)\) for \(0 \leq i \leq d - 1\). We want to estimate \(E(g(X_d))\) for a given positive integer \(d\), where \(g\) is a deterministic real-valued measurable function on \(F'\) such that \(g(X_d)\) is square-integrable. For \(1 \leq i \leq d\), set \(U_i = Y_{d-i}\). It can be shown by induction that \(X_d = G_i(U_1, \ldots, U_i, X_{d-i})\), where \(G_i\), \(0 \leq i \leq d\), is a measurable function from \(F_i \times F'\) to \(F'\), and so there is a real-valued measurable function \(f\) on \(F_d\) with \(g(X_d) = f(U_1, \ldots, U_d)\). We can thus use our randomized dimension reduction algorithm to estimate \(E(g(X_d))\). Recall that, in iteration \(k+1\) in Step 2 of the generic algorithm of \((2.1)\) conditioning on \(N_k = i\), the first \(i\) arguments of \(f\) are re-drawn, and the remaining arguments are unchanged. This is equivalent to re-drawing the last \(i\) random variables driving the Markov chain, and keeping the first \(d - i\) variables unchanged. In light of above, the generic randomized dimension reduction algorithm for Markov chains estimation takes as parameter a vector \(q \in A\) and consists of the following steps:

1. First iteration. Generate recursively \(X_0, \ldots, X_d\). Calculate \(g(X_d)\).

2. Loop. In iteration \(k+1\), where \(1 \leq k \leq n-1\), keep \(X_0, \ldots, X_{d-N_k}\) unchanged, and calculate recursively \(X_{d-N_{k+1}}, \ldots, X_d\) by re-drawing \(Y_{d-N_{k+1}}, \ldots, Y_{d-1}\), where \(N_k\) is a random integer in \([1, d]\) such that \(Pr(N_k > i) = q_i\). Calculate \(g(X_d)\).

3. Output the average of \(g\) over the \(n\) copies of \(X_d\) generated in the first two steps.

We assume that \(g\) and the \(g_i\)'s can be calculated in constant time, and that the expected time needed to simulate each \(Y_i\) is upper-bounded by a constant independent of \(d\). Thus, given \(N_k\), the expected time needed to perform iteration \(k + 1\) is \(O(N_k)\). Hence, we can set \(t_i = ci\), for some constant \(c\) independent of \(d\). Proposition \(4.1\) below shows that, roughly speaking, \(C(i)\) is small if \(X_{d-i}\) and \(X_d\) are weakly dependent.

**Proposition 4.1.** For \(0 \leq i \leq d\), we have \(C(i) = \text{Var}(E(g(X_d)|X_{d-i}))\).

By Proposition \(2.3\) if there are constants \(c' > 0\) and \(\gamma < -1\) independent of \(d\) such that \(C(i) \leq c'(i + 1)^\gamma\) for \(0 \leq i \leq d - 1\), then \(R(q, \nu)\) is upper-bounded by a constant independent
of \( d \), where \( q_i = (i + 1)^{(\gamma - 1) / 2} \) for \( 0 \leq i \leq d - 1 \). The analysis in (Asmussen and Glynn 2007, Section IV.1a), combined with Proposition 4.1 suggests that \( C(i) \) decreases exponentially with \( i \) for a variety of Markov chains.

For \( x \in \mathbb{F}^t \), and \( 0 \leq i \leq d \), let
\[
X_{i,x} = G_i(U_1, \ldots, U_i, x).
\]
In other words, \( X_{i,x} \) is the state of the chain at time-step \( d \) if the chain is at state \( x \) at time-step \( d - i \). Intuitively, we expect \( X_{i,x} \) to be close to \( X_d \) for large \( i \) if \( X_d \) depends mainly on the last \( Y_j \)'s. By Proposition 2.2, if \( g(X_{i,x}) \) is square-integrable,
\[
C(i) \leq \| g(X_d) - g(X_{i,x}) \|^2. \tag{4.1}
\]
In the following examples, we prove that under certain conditions, \( R(q, \nu^*) \) is upper-bounded by a constant independent of \( d \) for an explicit vector \( q \in A \), and so \( T^{\text{tot}}(q, \epsilon) = O(d + \epsilon^{-2}) \).

### 4.1 GARCH volatility model

In the GARCH(1,1) volatility model (see (Hull 2014, Ch. 23)), the variance \( X_i \) of an index return between day \( i \) and day \( i + 1 \), as estimated at the end of day \( i \), satisfies the following recursion:
\[
X_{i+1} = \omega + \alpha X_i Y_i^2 + \beta X_i,
\]
\( i \geq 0 \), where \( \omega, \alpha \) and \( \beta \) are positive constants with \( \alpha + \beta < 1 \), and \( Y_i, i \geq 0 \), are independent standard Gaussian random variables. The variable \( Y_i \) is known at the end of day \( i + 1 \). At the end of day \( 0 \), given \( X_0 \geq 0 \), a positive integer \( d \) and a real number \( z \), we want to estimate \( \Pr(X_d > z) \). In this example, \( F = F' = \mathbb{R} \), and \( g_i(x, y) = \omega + \alpha xy^2 + \beta x \), with \( g(u) = 1 \{ u > z \} \) for \( u \in \mathbb{R} \). Proposition 4.2 shows that \( C(i) \) decreases exponentially with \( i \).

**Proposition 4.2.** There is a constant \( \kappa \) independent of \( d \) such that \( C(i) \leq \kappa(\alpha + \beta)^{i/2} \) for \( 0 \leq i \leq d - 1 \).

By applying Proposition 2.2 with \( \nu_i = \kappa(\alpha + \beta)^{i/2} \) and setting \( q_i = (\alpha + \beta)^{i/4} \) for \( 0 \leq i \leq d - 1 \), we infer that \( R(q; \nu^*) \) is upper-bounded by a constant independent of \( d \).

### 4.2 \( G_t/D/1 \) queue

Consider a queue where customers arrive at time-step \( i \), \( 1 \leq i \leq d \), and are served by a single server in order of arrival. Service times are all equal to \( 1 \). Assume the system starts empty at time-step \( 0 \), and that \( A_i \) customers arrive at time-step \( i \), \( 0 \leq i \leq d \), where \( A_0 = 0 \) and the \( A_i \)'s are independent square-integrable random variables. Let \( X_i \) be the number of customers waiting in the queue at time-step \( i \). Then \( X_0 = 0 \) and \( (X_i) \) satisfies the Lindley equation
\[
X_{i+1} = (X_i + Y_i)^+,
\]
for \( 0 \leq i \leq d - 1 \), with \( Y_i = A_{i+1} - 1 \). We want to estimate \( E(X_d) \). In this example, \( g \) is the identity function, \( F = F' = \mathbb{R} \), and \( g_i(x, y) = (x + y)^+ \). Proposition 4.3 below shows that \( C(i) \) decreases exponentially with \( i \) under certain conditions on the service times.

**Proposition 4.3.** If there are constants \( \gamma > 0 \) and \( \kappa < 1 \) independent of \( d \) such that
\[
E(e^{\gamma Y_i}) \leq \kappa \tag{4.2}
\]
for \( 0 \leq i \leq d - 1 \), then \( C(i) \leq \gamma' \kappa^i \) for \( 0 \leq i \leq d - 1 \), where \( \gamma' \) is a constant independent of \( d \).

By applying Proposition 2.3 with \( \nu_i = \gamma' \kappa^i \) and \( q_i = \kappa^{i/2} \) for \( 0 \leq i \leq d - 1 \), we conclude that, under the assumption of Proposition 4.3, \( R(q; \nu^*) \) is upper-bounded by a constant independent of \( d \). The assumption in Proposition 4.3 can be justified as follows. Given \( i \in [0, d - 1] \), if \( E(A_i) < 1 \) and the function \( h(\gamma) = E(e^{\gamma Y_i}) \) is bounded on a neighborhood of 0, then \( h'(0) = E(Y_i) < 0 \). As \( h(1) = 1 \), there is \( \gamma > 0 \) such that \( h(\gamma) < 1 \), and (4.2) holds for \( \kappa = h(\gamma) \). The assumption in Proposition 4.3 says that \( \gamma \) and \( \kappa \) can be chosen independently of \( i \) and of \( d \).
4.3 $M_t/GI/1$ queue

Consider a $M_t/GI/1$ queue where customers are served by a single server in order of arrival. We assume that customers arrive according to a Poisson process with positive and continuous time-varying rate $\lambda_t \leq \lambda^*$, where $\lambda^*$ is a fixed positive real number. The service times are assumed to be i.i.d. and independent of the arrival times. Assume that the system starts empty at time $0$. For simplicity, we assume that the number of customers that arrive in any bounded time interval is finite (rather than finite with probability 1). Consider a customer present in the system at a given time $s$. If the customer has been served for a period of length $\tau$, its remaining service time is equal to its service time minus $\tau$, and if the customer is in the queue, its remaining service time is equal to its service time. The residual work $W_s$ at time $s$ is defined as the sum of remaining service times of customers present in the system at $s$. We want to estimate the expectation of $W_0$, where $\theta$ is a fixed time. Let $d = \lfloor \lambda^* \theta \rfloor$, and assume that $d \geq 2$.

For $0 \leq i \leq d$, let $X_i = W_{i\theta/d}$ be the residual work at time $i\theta/d$. For $0 \leq i \leq d - 1$, let $Y_i$ be the vector that consists of arrival and service times of customers that arrive during the interval $(i\theta/d, (i + 1)\theta/d)$. In this example, $g$ is the identity function, $F$ is equal to the set of real-valued sequences with finite support, and $F' = \mathbb{R}$. Let $0 \leq t < t'$. If no customers arrive in $(t, t']$ then $W_{t'} = (W_t - t' + t)^+$. On the other hand, if no customers arrive in $(t, t']$ and a customer with service time $S$ arrives at $t'$, then $W_{t'} = S + (W_t - t' + t)^+$. Thus, given the set of arrival and service times of customers that arrive in $(t, t']$, we can calculate iteratively $W_{t'}$ from $W_t$.

This implies that $X_{i+1}$ is a deterministic measurable function of $X_i$ and $Y_i$, for $0 \leq i \leq d - 1$. Proposition 4.4 below shows that $C(i)$ decreases exponentially with $i$ under certain conditions on the arrival and service times.

**Proposition 4.4.** For $0 \leq t \leq \theta$, let $Z_\theta(t)$ be the cumulative service time of customers that arrive in $[t, \theta]$. Assume there are constants $\gamma > 0$ and $\kappa < 1$ independent of $d$ such that, for $0 \leq t \leq t' \leq \theta$ and $t' - t \leq 1/\lambda^*$,

$$E(e^{\gamma(Z_\theta(t') - Z_\theta(t) - 1/\lambda^*)}) \leq \kappa. \quad (4.3)$$

Then $C(i) \leq \gamma'\kappa^{i/2}$ for $0 \leq i \leq d - 1$, where $\gamma'$ is a constant independent of $d$.

By applying Proposition 2.3 with $\nu_i = \gamma'\kappa^{i/2}$ and $q_i = \kappa^{i/4}$ for $0 \leq i \leq d - 1$, we conclude that, under the assumption of Proposition 4.4, $R(q; \nu^*)$ is upper-bounded by a constant independent of $d$. The assumption in Proposition 4.4 can be justified as follows. For $0 \leq t \leq t' \leq \theta$ and $t' - t \leq 1/\lambda^*$, the cumulative service times of customers that arrive in $[t, t']$ is $Z_{t'} - Z_0(t)$. If $E(Z_{t'} - Z_0(t)) < t' - t$ and $h(\gamma) = E(e^{\gamma(Z_{t'} - Z_0(t) - 1/\lambda^*)})$ is bounded on a neighborhood of 0, then $h'(0) < 0$. Thus $h(\gamma) < 1$ for some $\gamma > 0$ and (4.3) holds for $\kappa = h(\gamma)$. The assumption in Proposition 4.4 says that $\gamma$ and $\kappa$ can be chosen independently of $d$, $t$ and $t'$.

5 Numerical experiments

Our simulation experiments, using the examples in §4, were implemented in the C++ programming language. In the RDR algorithm, described in §4.3, $n$ was chosen so that the expected total number of simulations of the $U_i$’s in iterations 2 through $n$ is approximately $10d$. The actual total number of simulations of the $U_i$’s, denoted by “Cost” in our computer experiments, is about $11d$ because it includes the $d$ simulations of the first iteration.

We have also implemented the multilevel algorithm (MLMC) described in §6.1 with $L = \lceil \log_2(d) \rceil + 1$, and $m_l = \lceil 2^{l-L}d \rceil$ for $1 \leq l \leq L$, and $\phi_l = f(U_1, \ldots, U_{m_l}, X_0, \ldots, X_l)$. The $U_i$’s were estimated by Monte Carlo simulation with 1000 samples, and the $n_l$’s were scaled up so that the actual total number of simulations of the $U_i$’s is about $11d$. 

11
In Tables 1 through 5, the variable Std refers to the standard deviation of $f_n$ for the RDR algorithm, and to the standard deviation of $\phi$ for the MLMC algorithm. The variable Std and a 90% confidence interval for $E(f(U))$ were estimated using 1000 independent runs of these two algorithms. The variance reduction factor VRF is defined as

$$VRF = \frac{d\text{Var}(f(U))}{\text{Cost} \times \text{Std}^2}.$$  

We estimated $\text{Var}(f(U))$ by using 10000 independent samples of $U$. The standard deviation of the running time of the RDR algorithm is not reported because it is negligible in comparison to the running time.

### 5.1 GARCH volatility model

Table 1 shows results of our simulations of the GARCH volatility model for estimating $\Pr(X_d > z)$, with $z = 4.4 \times 10^{-5}$, $X_0 = 10^{-4}$, $\alpha = 0.06$, $\beta = 0.9$, and $\omega = 1.76 \times 10^{-6}$. The 90% confidence intervals for the RDR and MLMC algorithms are consistent with each other. As expected, the variable Cost is about 11$d$ for the RDR and MLMC algorithms. For both algorithms, the variable Cost $\times$ Std$^2$ is roughly independent of $d$, and the variance reduction factors are roughly proportional to $d$. The RDR algorithm outperforms the MLMC algorithm by about a factor of 10.

### 5.2 $G_t/D/1$ queue

Assume that $A_t$ has a Poisson distribution with time-varying rate $\lambda_i = 0.75 + 0.5 \cos(\pi t/50)$, for $1 \leq i \leq d$, (recall that $A_0 = 0$). These parameters are taken from (Whitt and You 2016). Table 2 estimates $E(X_d)$, and Table 3 gives VRFs in the estimation of $\Pr(X_d > z)$, for selected values of $z$. Once again, for both the RDR and MLMC algorithms, the variable Cost $\times$ Std$^2$ is roughly independent of $d$, and the variance reduction factors are roughly proportional to $d$. The VRFs of the RDR algorithm in Table 3 are greater than or equal to the corresponding VRFs in Table 2, which confirms the resiliency of the RDR algorithm to discontinuities of $g$. In contrast, the VRFs of the MLMC algorithm in Table 3 are lower than the corresponding VRFs in Table 2. The RDR algorithm outperforms the MLMC algorithm by a factor ranging from 1 to 2 in Table 2 and a factor ranging from 2 to 17 in Table 3.

### 5.3 $M_t/GI/1$ queue

Assume that $\lambda_t = 0.75 + 0.5 \cos(\pi t/50)$ for $t \geq 0$. These parameters are taken from (Whitt and You 2016). Assume further that, for $j \geq 1$, the service time $S_j$ for the $j$-th customer has a Pareto distribution with $\Pr(S_j \geq z) = (1 + z/\alpha)^{-3}$ for $z \geq 0$, for some constant $\alpha > 0$. A simple calculation shows that $E(S_j) = \alpha/2$. In our simulations, we have set $d = [\theta]$. Table 4 gives our simulation results for estimating $\Pr(W_0 > 1)$ when $\alpha = 2$, and Table 5 lists VRFs for estimating $\Pr(W_0 > 1)$ for selected values of $\alpha$. Here again, for both the RDR and MLMC
with $m$ to a deterministic measurable function of $l$ that, as the algorithm improves upon this class of MLMC algorithms by a factor of order $d$. The MLMC algorithms description and analysis

Table 2: $E(X_d)$ estimation in $M_l/D/1$ queue, 1000 samples, where $X_d$ is the number of customers in the queue at time-step $d$.

| $d$  | RDR       | Std         | Cost       | Cost × Std² | VRF |
|------|------------|-------------|------------|-------------|-----|
| 10²  | 2295.5     | 8.8 × 10⁻²  | 1.1 × 10⁰  | 8.5 × 10²   | 1.8 × 10² |
| 10⁵  | 710.5      | 1.1 × 10⁻¹  | 1.1 × 10⁵  | 1.4 × 10³   | 1.1 × 10² |
| 10⁶  | 23391.5    | 2.8 × 10⁻²  | 1.1 × 10⁵  | 8.6 × 10²   | 1.8 × 10³ |
|      | 62815.5    | 3.8 × 10⁻²  | 1.1 × 10⁶  | 1.6 × 10³   | 9.5 × 10² |
|      | 661151.5   | 9.5 × 10⁻³  | 1.1 × 10⁷  | 1.0 × 10³   | 1.5 × 10⁴ |

Table 3: VRFs for $\Pr(X_d > z)$ estimation in $M_l/D/1$ queue.

| $z$ | 0    | 2    | 4    | 6    | 8    | 10   |
|-----|------|------|------|------|------|------|
| 10² | 2.8 × 10¹ | 2.3 × 10² | 2.1 × 10² | 2.1 × 10² | 1.9 × 10² | 1.8 × 10² |
| 10⁵ | 3.1 × 10¹ | 3.7 × 10¹ | 5.1 × 10¹ | 6.5 × 10¹ | 7.7 × 10¹ | 7.2 × 10¹ |
| 10⁶ | 3.3 × 10¹ | 2.2 × 10² | 2.1 × 10³ | 1.9 × 10⁴ | 1.8 × 10⁴ | 1.7 × 10⁴ |

For instance, the RDR and MLMC algorithms become less efficient as $\alpha$ increases. This can be explained by noting that, as $\alpha$ increases, the length of the last busy cycle increases as well, which renders $W_\theta$ more dependent on the first $Y_t$’s.

6 Comparison with a class of multilevel algorithms

We compare our method to a class of MLMC algorithms, adapted from (Giles 2008), that efficiently estimate $E(f(U))$ under the assumption that $f$ is strongly approximated, in the $L^2$ sense, by functions of its first arguments. Under conditions described in [6.1] we prove that, up to a constant, the randomized dimension reduction algorithm is at least as efficient as this class of MLMC algorithms. [6.2] gives an example where the randomized dimension reduction algorithm improves upon this class of MLMC algorithms by a factor of order $d$.

6.1 The MLMC algorithms description and analysis

Let $L$ be a positive integer and let $(m_l)$, $0 \leq l \leq L$, be a strictly increasing integral sequence, with $m_0 = 0$ and $m_L = d$. For $1 \leq l \leq L$, let $\phi_l$ be a square-integrable random variable equal to a deterministic measurable function of $U_1, \ldots, U_{m_l}$, with $\phi_L = f(U)$. The $\phi_l$’s are chosen so that, as $l$ increases, $\phi_l$ gets closer to $f(U)$, in the $L^2$ sense. For instance, $L$ can be proportional

Table 4: $\Pr(W_\theta > 1)$ estimation in $M_l/GI/1$ queue, $\alpha = 2$, with 1000 samples, where $W_\theta$ is the residual work at time $\theta$.

| $\theta$ | n     | 90% confidence interval | Std     | Cost   | Cost × Std² |
|----------|-------|-------------------------|---------|--------|-------------|
| 10³      | RDR   | 3.9 × 10³                | 0.85389 ± 4.9 × 10⁻⁴ | 9.4 × 10⁻³ | 1.1 × 10³   | 9           |
|          | MLMC  | 1.0 × 10⁴                | 0.8541 ± 1.6 × 10⁻³ | 3.1 × 10⁻² | 1.1 × 10⁵   | 106         |
| 10⁵      | RDR   | 3.0 × 10⁴                | 0.85385 ± 1.6 × 10⁻⁴ | 3.0 × 10⁻³ | 1.1 × 10⁶   | 10          |
|          | MLMC  | 8.3 × 10⁴                | 0.85322 ± 4.9 × 10⁻⁴ | 9.5 × 10⁻³ | 1.1 × 10⁶   | 98          |
| 10⁶      | RDR   | 2.5 × 10⁵                | 0.853762 ± 5.1 × 10⁻⁵ | 9.8 × 10⁻⁴ | 1.1 × 10⁷   | 10          |
|          | MLMC  | 8.5 × 10⁵                | 0.8539 ± 1.6 × 10⁻⁴ | 3.2 × 10⁻³ | 1.1 × 10⁷   | 111         |
Table 5: VRF for Pr($W_d > 1$) estimation in $M_t/GI/1$ queue.

| $d$      | $RDR$ | $MLC$ |
|----------|-------|-------|
| $10^4$   | 4.3 x $10^2$ | 3.2 x $10^2$ |
| $10^5$   | 2.6 x $10^2$ | 1.4 x $10^2$ |
| $10^6$   | 2.3 x $10^3$ | 2.1 x $10^3$ |

for some $x \in F$. For $1 \leq l \leq L$, let $\hat{\phi}_l$ be the average of $n_l$ independent copies of $\hat{\phi}_l - \phi_{l-1}$ (with $\phi_0 \triangleq 0$), where $n_l$ is an integer to be specified later. Assume that the estimators $\hat{\phi}_1, \ldots, \hat{\phi}_L$ are independent. As

$$E(f(U)) = \sum_{l=1}^{L} E(\phi_l - \phi_{l-1}),$$

$$\hat{\phi} = \sum_{l=1}^{L} \hat{\phi}_l$$

is an unbiased estimator of $E(f(U))$. Following the analysis in (Giles 2008),

$$\text{Var}(\hat{\phi}) = \sum_{l=1}^{L} \frac{V_l}{n_l},$$

where $V_l \triangleq \text{Var}(\phi_l - \phi_{l-1})$ for $1 \leq l \leq L$. The expected time needed to simulate $\hat{\phi}$ is $T_{MLC} \triangleq \sum_{l=1}^{L} n_l \hat{t}_l$, where $\hat{t}_l$ is the expected time needed to simulate $\phi_l - \phi_{l-1}$. As shown in (Giles 2008), the time-variance product $T_{MLC} \text{Var}(\hat{\phi})$ is minimized when the $n_l$'s are proportional to $\sqrt{V_l/\hat{t}_l}$ (ignoring the integrality constraints on the $n_l$'s), in which case

$$T_{MLC} \text{Var}(\hat{\phi}) = \left( \sum_{l=1}^{L} \sqrt{V_l/\hat{t}_l} \right)^2.$$  (6.1)

As the variance of the average of $n$ i.i.d. square-integrable random variables is proportional to $1/n$, for $\epsilon > 0$, the number of independent samples of $\hat{\phi}$ needed to achieve an estimator variance $\epsilon^2$ is $\lceil \text{Var}(\hat{\phi}) \epsilon^{-2} \rceil$. Thus the total expected time $T_{MLC}(\epsilon)$ needed for the MLMC algorithm to estimate $E(f(U))$ with variance $\epsilon^2$ satisfies the relation

$$T_{MLC}(\epsilon) = \Theta(T_{MLC} + T_{MLC} \text{Var}(\hat{\phi}) \epsilon^{-2}).$$  (6.2)

In line with (Giles 2008, Theorem 3.1), if $\hat{t}_l = O(2^l)$ and $\|\phi_l - \phi_L\|^2 = O(2^{2l})$, with $\beta < -1$, where the constants behind the $O$-notation do not depend on $d$, then $T_{MLC} \text{Var}(\hat{\phi})$ is upper-bounded by a constant independent of $d$. This can be shown by observing that

$$V_l \leq \|\phi_l - \phi_{l-1}\|^2 \leq (\|\phi_l - \phi_L\| + \|\phi_{l-1} - \phi_L\|)^2.$$  

Theorem 5.1 below shows that, under certain conditions, the randomized dimension reduction method is, up to a multiplicative constant, at least as efficient as the class of MLMC methods described above. Indeed, under the assumptions of Theorem 5.1 by (2.6), $T^{\text{tot}}(q, \nu) = O(d + T_{MLC} \text{Var}(\hat{\phi}) \epsilon^{-2}).$ On the other hand, since $T_{MLC} = \Omega(d)$, it follows from (6.2) that $T_{MLC}(\epsilon) = \Omega(d + T_{MLC} \text{Var}(\hat{\phi}) \epsilon^{-2}).$

**Theorem 5.1.** Assume that there are constants $c$ and $\check{c}$ independent of $d$ such that $\hat{t}_i = ci$ for $1 \leq i \leq d$, and $\hat{t}_l \geq \check{c}m_l$ for $1 \leq l \leq L$, and that $C(d - 1) > 0$. Then $R(q; \nu^*) \leq (32c/\check{c})T_{MLC} \text{Var}(\hat{\phi})$ if, for $0 \leq i \leq d - 1,$

$$q_i = \sqrt{\frac{C(i)}{(i+1)C(0)}}.$$
6.2 Example

We describe an academic example where the randomized dimension reduction algorithm outperforms the class of MLMC methods described in [6,1] by a factor of order \( d \). Let \( U_1, \ldots, U_d \) be independent centered random variables with variance 1 that can be simulated in constant time. Set

\[
\hat{f}(x_1, \ldots, x_d) = x_1 \sum_{j=2}^d x_j.
\]

Let \( L \) be a positive integer and let \( (m_t), 0 \leq t \leq L, \) be a strictly increasing integral sequence, with \( m_0 = 0 \) and \( m_L = d \). For \( 1 \leq t \leq L \), let \( \phi_t \) be a square-integrable random variable equal to a deterministic measurable function of \( U_1, \ldots, U_{m_t} \), with \( \phi_t = f(U) \). Under the assumptions of Proposition 6.1 for \( 0 < \epsilon < 1 \), we infer that \( T_{\text{tot}}(q, \epsilon) = O(d^{c-2}) \) via (2.6), and that \( T_{\text{MLMC}}(\epsilon) = \Omega(d^2 \epsilon^{-2}) \) via (5.2).

**Proposition 6.1.** If we set \( q_0 = 1 \) and \( q_i = 1/d \) for \( 1 \leq i \leq d-1 \), then \( R(q; \nu^*) = O(d) \). If there is a constant \( \tilde{c} \) independent of \( d \) such that \( \tilde{c} \geq \tilde{c} m_l \) for \( 1 \leq l \leq L \), then \( T_{\text{MLC}} \text{Var}(\hat{\phi}) = \Omega(d^2) \).

7 Conclusion

We have described a randomized dimension reduction algorithm that estimates \( E(f(U)) \) via Monte Carlo simulation, assuming that \( f \) does not depend equally on all its arguments. We formally prove that under some conditions, in order to achieve an estimator variance \( \epsilon^2 \), our algorithm requires \( O(d + \epsilon^{-2}) \) computations as opposed to \( O(d \epsilon^{-2}) \) under the standard Monte Carlo methods. Our algorithm can be used to efficiently estimate the expected value of a function of the state of a Markov chain at time-step \( d \), for a class of Markov chains driven by random variables. The numerical implementation of our algorithm uses a new geometric procedure of independent interest that solves in \( O(d^3) \) time a \( d \)-dimensional optimisation problem that was previously solved in \( O(d^5) \) time. We have argued intuitively that our method is resilient to discontinuities of \( f \). Our numerical experiments confirm that our method highly outperforms the standard Monte Carlo method for large values of \( d \), and show its high resilience to discontinuities.

A Proof of Proposition 2.1

For \( 0 \leq i \leq d \), let

\[
f^{(i)} = E(f(U)|U_{i+1}, \ldots, U_d).
\]

Thus, \( C(i) = \text{Var}(f^{(i)}) \). By the tower law, for \( 0 \leq i \leq d-1 \),

\[
E(f^{(i)}|U_{i+2}, \ldots, U_d) = f^{(i+1)},
\]

and so

\[
C(i+1) = \text{Var}(E(f^{(i)}|U_{i+2}, \ldots, U_d)).
\]

As the variance decreases by taking the conditional expectation, it follows that \( C(i+1) \leq C(i) \), as desired.

We now prove (2.2). Let \( W = (U'_1, \ldots, U'_l, U_{i+1}, \ldots, U_d) \). Since \( U \) and \( W \) are conditionally independent given \( U_{i+1}, \ldots, U_d \), and \( E(f(W)|U_{i+1}, \ldots, U_d) = f^{(i)} \),

\[
E(f(U)f(W)|U_{i+1}, \ldots, U_d) = (f^{(i)})^2.
\]

Hence, by the tower law,

\[
E(f(U)f(W)) = E((f^{(i)})^2).
\]
On the other hand, using the tower law once again,

\[ E(f(U)) = E(f(W)) = E(f(i)), \]

and so the RHS of (2.2) is equal to \( \text{Var}(f(i)) \), as required. \( \square \)

**B Proof of Theorem 2.1**

We first prove Lemma B.1 below, which follows by classical calculations (see e.g. (Asmussen and Glynn 2007, Section IV.6a)).

**Lemma B.1.** Let \((Z_k), k \geq 1,\) be an homogeneous stationary Markov chain in \(\mathbb{R}^d,\) and let \(g\) be a real-valued Borel-measurable function on \(\mathbb{R}^d\) such that \(g(Z_1)\) is square-integrable, and \(a_j = \text{Cov}(g(Z_1), g(Z_{1+j}))\) is non-negative for \(j \geq 0.\) Assume that \(\sum_{j=1}^{\infty} a_j\) is finite. Then

\[
\frac{n^{-1}}{\text{Var}}\left(\sum_{m=1}^{n} g(Z_m)\right) \leq a_0 + 2 \sum_{j=1}^{\infty} a_j. \tag{B.1}
\]

Furthermore, the LHS of (B.1) converges to its RHS as \(n\) goes to infinity.

**Proof.** Since \((Z_k), k \geq 1,\) is homogeneous and stationary, \(\text{Cov}(g(Z_m)g(Z_{m+j})) = a_j\) for \(m \geq 1\) and \(j \geq 0.\) Thus,

\[
\text{Var}\left(\sum_{m=1}^{n} g(Z_m)\right) = \sum_{m=1}^{n} \text{Var}(g(Z_m)) + 2 \sum_{1 \leq m < m+j \leq n} \text{Cov}(g(Z_m)g(Z_{m+j}))
\]

\[
= n a_0 + 2 \sum_{j=1}^{n} (n-j)a_j.
\]

Hence

\[
\frac{n^{-1}}{\text{Var}}\left(\sum_{m=1}^{n} g(Z_m)\right) = a_0 + 2 \sum_{j=1}^{\infty} \frac{n-j}{n} a_j,
\]

which implies (B.1). Using Lebesgue’s dominated convergence theorem concludes the proof. \( \square \)

We now prove Theorem 2.1. Since the \(C(i)\)’s and the variance of \(f_n\) remain unchanged if we add a constant to \(f,\) we can assume without loss of generality that \(E(f(U)) = 0.\) For \(0 \leq i \leq d - 1,\) set \(p_i = 1 - q_i.\) Let \(m < k\) be two integers in \([1, n].\) For \(0 \leq i \leq d - 1,\)

\[
\text{Pr}( \max_{m \leq j < k} N_j \leq i ) = p_i^{k-m}
\]

and so, for \(1 \leq i \leq d - 1,\)

\[
\text{Pr}( \max_{m \leq j < k} N_j = i ) = p_i^{k-m} - p_{i-1}^{k-m}.
\]

For \(i \in [1, d],\) conditional on the event \(\max_{m \leq j < k} N_j = i,\) the first \(i\) components of \(V^{(m)}\) and \(V^{(k)}\) are independent, and the last \(d - i\) components of \(V^{(m)}\) and \(V^{(k)}\) are the same. This is because, if \(N_j = i,\) with \(m \leq j < k,\) the vector \(V^{(m)}\) and the first \(i\) components of \(V^{(j+1)}\) are independent. Thus, by Proposition 2.1

\[
E(f(V^{(m)})f(V^{(k)})) \left| \max_{m \leq j < k} N_j = i \right) = C(i).
\]
As \( C(d) = 0 \), it follows from Bayes’ formula that

\[
E(f(V^{(m)})f(V^{(k)})) = \sum_{i=1}^{d-1} \Pr(\max_{m \leq j < k} N_j = i)C(i) = \sum_{i=1}^{d-1} (p_i k - m - p_{i-1} k - m)C(i).
\]

Let \( a_j = \text{Cov}(f(V^{(1)}), f(V^{(1+j)})) \). Thus, for \( j > 0 \),

\[
a_j = \sum_{i=1}^{d-1} (p_i^j - p_{i-1}^j)C(i)
\]
is non-negative, and

\[
\sum_{j=1}^{\infty} a_j = \sum_{i=1}^{d-1} \left( \frac{p_i}{1-p_i} - \frac{p_i}{1-p_{i-1}} \right)C(i) = \sum_{i=1}^{d-1} \left( \frac{1}{1-p_i} - \frac{1}{1-p_{i-1}} \right)C(i) = -C(1) + \sum_{i=1}^{d-1} \frac{C(i) - C(i+1)}{q_i}
\]
is finite. Since \( a_0 = C(0) \), it follows that

\[
a_0 + 2 \sum_{j=1}^{\infty} a_j = C(0) - 2C(1) + 2 \sum_{i=1}^{d-1} \frac{C(i) - C(i+1)}{q_i}.
\]

We conclude the proof using Lemma B.1. \( \Box \)

**C  Proof of Proposition 2.2**

Let \( U'_1, \ldots, U'_i \) be random variables satisfying the conditions of Proposition 2.1 and \( W = (U'_1, \ldots, U'_i, U_{i+1}, \ldots, U_d) \). Since \( (U_1, \ldots, U_i) \) and \( W \) are independent,

\[
\text{Cov}(f_i(U_1, \ldots, U_i), f(W)) = 0.
\]

Similarly,

\[
\text{Cov}(f_i(U_1, \ldots, U_i), f_i(U'_1, \ldots, U'_i)) = \text{Cov}(f(U), f_i(U'_1, \ldots, U'_i)) = 0.
\]

Thus, by Proposition 2.1 and bilinearity of the covariance,

\[
C(i) = \text{Cov}(f(U) - f_i(U_1, \ldots, U_i), f(W) - f_i(U'_1, \ldots, U'_i)) \leq \text{Std}(f(U) - f_i(U_1, \ldots, U_i)) \text{Std}(f(W) - f_i(U'_1, \ldots, U'_i)) = \text{Var}(f(U) - f_i(U_1, \ldots, U_i)).
\]

The last equation follows by observing that \( f(W) - f_i(U'_1, \ldots, U'_i) \overset{d}{=} f(U) - f_i(U_1, \ldots, U_i) \). \( \Box \)
D  Proofs of Propositions 2.3 and 2.5

We first prove the following.

Proposition D.1. Let \( \nu = (\nu_0, \ldots, \nu_d) \) be an element of \( \mathbb{R}^d \times \{0\} \). Assume that \( \nu_0, \ldots, \nu_{d-1} \) are positive, and that the sequence \( \{\nu_i/t_{i+1}\}, 0 \leq i \leq d-1 \), is decreasing. For \( 0 \leq i \leq d-1 \), set \( q_i = \sqrt{t_{i+1}/(\nu_0 t_{i+1})} \). Then

\[
R(q; \nu) \leq 4 \left( \sum_{i=0}^{d-1} \sqrt{\nu_i/(\nu_{i+1} - \nu_i)} \right)^2.
\]

Proof. Applying the inequality \( x - y \leq 2\sqrt{x(\sqrt{x} - \sqrt{y})} \), which holds for \( x \geq 0 \) and \( y \geq 0 \), to \( x = \nu_i \) and \( y = \nu_{i+1} \) yields

\[
\sum_{i=0}^{d-1} \frac{\nu_i - \nu_{i+1}}{q_i} \leq 2 \sqrt{\frac{\nu_0}{t_1} \sum_{i=0}^{d-1} (\sqrt{\nu_i} - \sqrt{\nu_{i+1}} \sqrt{t_{i+1}})} = 2 \sqrt{\frac{\nu_0}{t_1} \sum_{i=0}^{d-1} \sqrt{\nu_i(\sqrt{t_{i+1}} - \sqrt{t_i})}}.
\]

Similarly, since \( t_{i+1} - t_i \leq 2\sqrt{t_{i+1}(\sqrt{t_{i+1}} - \sqrt{t_i})} \),

\[
\sum_{i=0}^{d-1} q_i(t_{i+1} - t_i) \leq 2 \sqrt{\frac{t_1}{\nu_0} \sum_{i=0}^{d-1} \sqrt{\nu_i(\sqrt{t_{i+1}} - \sqrt{t_i})}}.
\]

Taking the product completes the proof. \( \square \)

We now prove Proposition 2.3. Let \( \nu = (\nu_0, \ldots, \nu_{d-1}, 0) \), with \( \nu_i = c'(i+1)^{\gamma}, 0 \leq i \leq d-1 \). Then \( R(q; \nu^*) \leq R(q; 2\nu) \) since \( \nu^* \leq 2\nu \). By Proposition D.1

\[
R(q; 2\nu) \leq 8cc' \left( \sum_{i=0}^{d-1} (i+1)^{\gamma/2}(\sqrt{i+1} - \sqrt{i}) \right)^2.
\]

As \( \sqrt{i+1} - \sqrt{i} \leq (i+1)^{-1/2} \) for \( i \geq 0 \), it follows that

\[
R(q; \nu^*) \leq 8cc' \left( \sum_{i=1}^{d} i^{(\gamma-1)/2} \right)^2.
\]

The inequality

\[
\sum_{i=1}^{d} i^{(\gamma-1)/2} \leq 1 + \int_1^d x^{(\gamma-1)/2} dx
\]

implies that

\[
\sum_{i=0}^{d-1} i^{(\gamma-1)/2} \leq \begin{cases} 1 + 2/(\gamma + 1), & \gamma < -1, \\ 1 + \ln(d), & \gamma = -1, \\ 2d^{(\gamma+1)/2}/(\gamma + 1), & -1 < \gamma < 0. \end{cases}
\]

We conclude that there is a constant \( c_1 \) such that (2.8) holds. This concludes the proof of Proposition 2.3. \( \square \)

We now prove Proposition 2.5. By applying Proposition D.1 to \( \nu = (2C(0), \ldots, 2C(d-1), 0) \), it follows that \( R(q; \nu) \) is upper-bounded by the RHS of (2.9). Since \( R(q; \nu^*) \leq R(q; \nu) \), this implies (2.9). \( \square \)
E  Proof of Proposition 2.6

Since \( R(q; \nu) \) is increasing with respect to \( \nu \),
\[
R(q; \nu^*) \geq R(q; C(0), \ldots, C(d)) = \left( \sum_{i=0}^{d-1} \frac{C(i) - C(i + 1)}{q_i} \right) \left( \sum_{i=0}^{d-1} q_i(t_{i+1} - t_i) \right).
\]

But, as \( \sum_{j=0}^{d-1} q_j(t_{j+1} - t_j) \geq q_i t_{i+1} \) for \( 0 \leq i \leq d - 1 \), and since \( (C(i)) \) is a decreasing sequence,\[
\left( \sum_{i=0}^{d-1} \frac{C(i) - C(i + 1)}{q_i} \right) \left( \sum_{j=0}^{d-1} q_j(t_{j+1} - t_j) \right) \geq \sum_{i=0}^{d-1} (C(i) - C(i + 1))t_{i+1}.
\]

This implies (2.10) since
\[
\sum_{i=0}^{d-1} (C(i) - C(i + 1))t_{i+1} = \sum_{i=0}^{d-1} C(i)(t_{i+1} - t_i).
\]

Assume now that \( q_i = t_1/t_{i+1} \) for \( 0 \leq i \leq d - 1 \). Since \( R(q; \nu) \) is increasing with respect to \( \nu \),
\[
R(q; \nu^*) \leq R(q; 2C(0), \ldots, 2C(d))
\]
\[
= 2 \left( \sum_{i=0}^{d-1} \frac{C(i) - C(i + 1)}{q_i} \right) \left( \sum_{i=0}^{d-1} q_i(t_{i+1} - t_i) \right)
\]
\[
= 2 \left( \sum_{i=0}^{d-1} (C(i) - C(i + 1))t_{i+1} \right) \left( \sum_{i=0}^{d-1} \frac{t_{i+1} - t_i}{t_{i+1}} \right).
\]

But, since \( (t_{i+1} - t_i)/t_{i+1} \leq \ln(t_{i+1}/t_i) \) for \( 1 \leq i \leq d - 1 \),
\[
\sum_{i=0}^{d-1} \frac{t_{i+1} - t_i}{t_{i+1}} \leq 1 + \ln \left( \frac{t_d}{t_1} \right).
\]

Using (E.1) once again yields (2.11). \( \square \)

F  Proof of Theorem 3.1

We use the following proposition, whose proof follows immediately from (2.3).

**Proposition F.1.** If \( \nu \in \mathbb{R}^d \times \{0\} \) and \( \nu' \in \mathbb{R}^d \times \{0\} \) are such that \( \nu' \leq \nu \), and \( q \in A \), then \( R(q; \nu') \leq R(q; \nu) \), with equality if \( \nu_i = \nu'_i \) and, for \( 1 \leq i \leq d - 1 \), \( (\nu_i - \nu'_i)(q_{i-1} - q_i) = 0 \).

By definition of the lower hull, \( (\theta_i), \) \( 0 \leq i \leq d - 1 \), is an increasing sequence. Furthermore, \( \theta_{d-1} < 0 \) since it is equal to the slope of a segment joining \( (t_i, \nu_i) \) to \( (t_d, 0) \), for some \( i \in [0, d-1] \). Hence \( \theta_i < 0 \) for \( 0 \leq i \leq d - 1 \), and so \( q^* \) is well defined and belongs to \( A \). Furthermore, \( (\nu'_i) \), \( 0 \leq i \leq d \), is a decreasing sequence, and \( \nu'_d = \nu_d = 0 \). On the other hand, by (2.5),
\[
R(q^*; \nu') = \left( \sum_{i=0}^{d-1} \sqrt{(\nu'_i - \nu'_{i+1})(t_{i+1} - t_i)} \right)^2.
\]
Since, by the Cauchy-Schwartz inequality, for all non-negative sequences \((x_i)\) and \((y_i)\),

\[
\sum_{i=0}^{d-1} \sqrt{x_i y_i} \leq \left( \sum_{i=0}^{d-1} x_i \right) \left( \sum_{i=0}^{d-1} y_i \right),
\]

it follows that \(R(q^*; \nu') \leq R(q, \nu')\) for \(q \in A\). Furthermore, by Proposition \(\text{F.1}\) \(R(q; \nu') \leq R(q; \nu)\), and so \(R(q^*; \nu') \leq R(q, \nu)\). On the other hand, \((\nu_i - \nu'_i)(q^*_i - q'_i) = 0\) for \(1 \leq i \leq d - 1\). This is because, if \(\nu_i \neq \nu'_i\), then the point \((t_i, \nu_i)\) does not belong to the lower hull of the set \(\{(t_j, \nu_j): 0 \leq i \leq d\}\). Hence \((t_i, \nu'_i)\) belongs to the segment \((t_{i-1}, \nu'_{i-1}), (t_{i+1}, \nu'_{i+1})\), which implies that \(\theta_{i-1} = \theta_i\) and \(q^*_{i-1} = q'_i\). Thus, as \(\nu_0 = \nu'_0\), Proposition \(\text{F.1}\) shows that \(R(q^*; \nu) = R(q^*; \nu')\). This implies (3.2) and that \(R(q^*; \nu) \leq R(q; \nu)\) for \(q \in A\), as desired. \(\square\)

G Proof of Proposition 3.1

The random vectors \(W^{(1)} = (U_1, \ldots, U_i, U'_1, \ldots, U'_d)\), \(W^{(2)} = (U'_1, \ldots, U'_i, U_{i+1}, \ldots, U_d)\), and \(W^{(3)} = (U'_1, \ldots, U'_i, U_{i+1}, \ldots, U_d)\) have the same distribution as \(U\). Hence

\[
E((f(U) - f(W^{(1)}))(f(W^{(2)}) - f(W^{(3)}))) = \text{Cov}(f(U) - f(W^{(1)}), f(W^{(2)}) - f(W^{(3)})).
\]

As \(W^{(1)}\) and \(W^{(2)}\) are independent,

\[
\text{Cov}(f(W^{(1)}), f(W^{(2)})) = 0.
\]

Similarly,

\[
\text{Cov}(f(W^{(1)}), f(W^{(3)})) = \text{Cov}(f(U), f(W^{(3)})) = 0.
\]

By bilinearity of the covariance, it follows that

\[
E((f(U) - f(W^{(1)}))(f(W^{(2)}) - f(W^{(3)}))) = \text{Cov}(f(U), f(W^{(2)})).
\]

We conclude the proof using Proposition 2.1. \(\square\)

H Proof of Proposition 4.1

We first prove the following Markov property.

**Proposition H.1.** Let \(i \in [0, d]\). If \(H\) is a bounded random variable which is measurable with respect to the \(\sigma\)-algebra generated by \(X_i, Y_i, \ldots, Y_{d-1}\), then

\[
E(H|Y_0, \ldots, Y_{i-1}) = E(H|X_i).
\]

**Proof.** Let \(\mathcal{H}\) be the vector space of bounded real-valued random variables \(H\) satisfying (H.1). Clearly, the constant random variables belong to \(\mathcal{H}\). Let \((H_m), m \geq 0\), be an increasing sequence of positive elements of \(\mathcal{H}\) such that \(H = \sup_{m \geq 0} H_m\) is bounded. For \(m \geq 0\),

\[
E(H_m|Y_0, \ldots, Y_{i-1}) = E(H_m|X_i).
\]

By the conditional Lebesgue dominated convergence theorem (Shiryaev 1996, Theorem 2, p. 218), the LHS (resp. RHS) of (H.2) converges to \(E(H|Y_0, \ldots, Y_{i-1})\) (resp. \(E(H|X_i)\)) as \(m\) goes to infinity, and so \(H \in \mathcal{H}\).

Let \(\mathcal{G}\) (resp. \(\mathcal{G}'\)) be the set of bounded real-valued random variables which are measurable with respect to the \(\sigma\)-algebra generated by \(X_i\) (resp. \((Y_i, \ldots, Y_{d-1})\)), and let \(\mathcal{C}\) be the set of random variables of the form \(GG'\), with \(G \in \mathcal{G}\) and \(G' \in \mathcal{G}'\). For \(G \in \mathcal{G}\) and \(G' \in \mathcal{G}'\),

\[
E(GG'|Y_0, \ldots, Y_{i-1}) = GE(G'|Y_0, \ldots, Y_{i-1}) = GE(G').
\]

20
The first equation holds since $X_i$ is a measurable function of $Y_0, \ldots, Y_{i-1}$, which implies that $G$ is measurable with respect to the $\sigma$-algebra generated by $Y_0, \ldots, Y_{i-1}$. The second equation follows from the independence of $(Y_i, \ldots, Y_{d-1})$ and $(Y_0, \ldots, Y_{i-1})$. Similarly, since $(Y_i, \ldots, Y_{d-1})$ and $X_i$ are independent,
\[ E(G|X_i) = GE(G'|X_i) = GE(G'). \]
Thus, $GG' \in \mathcal{H}$, and so $C \subseteq \mathcal{H}$. As $C$ is closed under pointwise multiplication, by the monotone class theorem (Revuz and Yor 1999, Theorem 2.2, p. 3), $\mathcal{H}$ contains all bounded random variables which are measurable with respect to the $\sigma$-algebra generated by the elements of $C$. Since $Y_i, \ldots, Y_{d-1}$ and $X_i$ belong to $C$, it follows that $\mathcal{H}$ contains all bounded random variables which are measurable with respect to the $\sigma$-algebra generated by $X_i, Y_i, \ldots, Y_{d-1}$. This completes the proof. \hfill $\square$

As $g(X_d)$ is a measurable function of $(X_i, Y_i, \ldots, Y_{d-1})$, for any integer $m$, the random variable $\min(m, g^+(X_d))$ belongs to $\mathcal{H}$. By the conditional Lebesgue dominated convergence theorem, taking the limit as $m$ goes to infinity implies that
\[ E(g^+(X_d)|Y_0, \ldots, Y_{i-1}) = E(g^+(X_d)|X_i). \]
A similarly equation holds for $g^-(X_d)$. Thus,
\[ E(g(X_d)|Y_0, \ldots, Y_{i-1}) = E(g(X_d)|X_i). \]
Replacing $i$ with $d-i$ implies that
\[ E(g(X_d)|U_{i+1}, \ldots, U_d) = E(g(X_d)|X_{d-i}). \]
Taking the variance of both sides concludes the proof. \hfill $\square$

I Proof of Proposition 4.2

For $0 \leq i \leq d-1$, set $Z_i = \alpha Y_i^2 + \beta$, so that $X_{i+1} = \omega + Z_i X_i$. It can be shown by induction on $i$ that $X_d = \omega + \omega Z_{d-1} Z' + Z''$ for $2 \leq i \leq d$, where
\[ Z' = \sum_{j=1}^{i-1} \prod_{k=i-j}^j Z_{d-k} \]
and
\[ Z'' = Z_{d-i} \cdots Z_{d-1} X_{d-i}. \]

By convention, the product over an empty set is equal to 1. Since $X_{i,0}$ is equal to the state of the Markov chain at step $d$ if $X_{d-i} = 0$, it follows that $X_{i,0} = \omega + \omega Z_{d-1} Z'$. Note that $E(Z'') = (\alpha + \beta)^i E(X_{d-i})$ as $E(Y_i^2) = 1$. By (Hull 2014, Eq. 23.13), for $0 \leq m \leq d$,
\[ E(X_m) \leq \max(X_0, \frac{\omega}{1 - \alpha - \beta}), \]
and so
\[ E(Z'') \leq \max(X_0, \frac{\omega}{1 - \alpha - \beta})(\alpha + \beta)^i. \]
Since the density of $Y_{d-1}$ is upper-bounded by $1/2$, for $\gamma \leq \gamma'$,
\[ \Pr(\gamma \leq Z_{d-1} \leq \gamma') = 2 \Pr(\frac{(\gamma - \beta)^+}{\alpha} \leq Y_{d-1} \leq \frac{(\gamma' - \beta)^+}{\alpha}) \]
\[ \leq \sqrt{\frac{(\gamma' - \beta)^+}{\alpha}} - \sqrt{\frac{(\gamma - \beta)^+}{\alpha}} \]
\[ \leq \frac{\gamma' - \gamma}{\alpha}. \]
The last equation follows from the inequality \( \sqrt{y'} - \sqrt{y} \leq \sqrt{y' - y} \), which holds for \( 0 \leq y \leq y' \).

On the other hand, as \( X_{i,0} \leq X_d \),

\[
g(X_d) - g(X_{i,0}) = \begin{cases} 1 & \text{if } X_{i,0} \leq z < X_d, \\
0 & \text{otherwise}. \end{cases}
\]

Thus, by (4.1),

\[
C(i) \leq ||g(X_d) - g(X_{i,0})||^2 = \Pr(X_{i,0} \leq z < X_d).
\]

But

\[
\Pr(X_{i,0} \leq z < X_d) = \Pr\left(\frac{z - Z'' - \omega}{\omega Z'} \leq Z_{d-1} \leq \frac{z - \omega}{\omega Z'}\right)
= E\Pr\left(\frac{z - Z'' - \omega}{\omega Z'} \leq Z_{d-1} \leq \frac{z - \omega}{\omega Z'} \mid Z', Z''\right)
\leq E\left(\sqrt{\frac{Z''}{\alpha \omega Z'}}\right)
\leq \kappa (\alpha + \beta)^{1/2},
\]

where \( \kappa \) is a constant that depends only on \( \omega, X_0, \alpha \) and \( \beta \). The third equation follows from (1.1) and the independence of \( Z_{d-1} \) and \( Z', Z'' \). The last equation follows from Jensen’s inequality and the inequality \( Z' \geq 1 \). Hence \( C(i) \leq \kappa (\alpha + \beta)^{1/2} \) for \( 2 \leq i \leq d \). This inequality also holds for \( 0 \leq i \leq 1 \) by replacing \( \kappa \) with \( \max(\kappa, 1/(\alpha + \beta)) \).

J Proof of Proposition 4.3

By classical calculations (see, e.g., (Asmussen and Glynn 2007, §I, Eq. (1.4))), it can be shown by induction on \( d \) that \( X_d = \max_{0 \leq j \leq d} S_j \), where \( S_j = \sum_{k=d-j}^{d-1} Y_k \) for \( 1 \leq j \leq d \), with \( S_0 = 0 \). For \( 0 \leq i \leq d-1 \), since \( X_{i,0} \) is the number of customers in the queue at time-step \( d \) if there are no customers in the queue at time-step \( d - i \), it can be shown by induction on \( d \) that \( X_{i,0} = \max_{0 \leq j \leq i} S_j \). Hence

\[
X_d = \max(X_{i,0}, \max_{i+1 \leq j \leq d} S_j).
\]

Thus

\[
X_d - X_{i,0} \leq \max_{i+1 \leq j \leq d} S_j^+,
\]

and so

\[
C(i) \leq ||X_d - X_{i,0}||^2 \leq \sum_{j=i+1}^{d} ||S_j^+||^2.
\]

On the other hand, since the \( Y_i \)'s are independent, \( E(e^{\gamma S_j}) \leq \kappa^j \) for \( 0 \leq j \leq d \). Furthermore, as \( (x^+)^2 \leq 2e^x \) for \( x \in \mathbb{R} \), \( \gamma^2 (S_j^+)^2 \leq 2e^{\gamma S_j} \). Taking expectations implies that \( \gamma^2 ||S_j^+||^2 \leq 2\kappa^j \), and so \( C(i) \leq \gamma' \kappa^d \), where \( \gamma' = 2\gamma^{-2}/(1 - \kappa) \).
Proof of Proposition 4.4

It can be shown by induction on the number of arrivals in $(\theta', \theta]$ (see also (Whitt and Ma 2017)) that, if the system is empty at time $\theta' \in [0, \theta]$,

$$W_\theta = \sup_{t \in [\theta', \theta]} \{ Z_\theta(t) - (\theta - t) \}.$$

Hence

$$X_d = \sup_{t \in [0, \theta]} \{ Z_\theta(t) - (\theta - t) \}.$$

Also, for $0 \leq i \leq d$, since $X_{i,0}$ is the residual work at time $\theta$ if the residual work at time $(d-i)\theta/d$ is 0,

$$X_{i,0} = \sup_{t \in [(d-i)\theta/d, \theta]} \{ Z_\theta(t) - (\theta - t) \}.$$

Thus, by a calculation similar to the proof of Proposition 4.3, $X_d - X_{i,0} \leq \sup_{0 \leq t \leq (d-i)\theta/d} \{ Z_\theta(t) - (\theta - t) \}^+.$

For non-negative integer $j$, set

$$S_j = Z_\theta(\theta - j \lambda^*) - \frac{j - 1}{\lambda^*}.$$

Fix $t \in [0, (d-i)\theta/d]$, and let $j = \lceil (\theta - t)\lambda^* \rceil$. As $\theta - j\lambda^* \leq t$ and $Z_\theta$ is a decreasing function,

$$Z_\theta(t) \leq Z_\theta((\theta - j\lambda^*)^+).$$

Since $\theta - t \geq (j - 1)/\lambda^*$, this implies that $Z_\theta(t) - (\theta - t) \leq S_j$. But $2j \geq i$ as $j \geq i\lambda^*\theta/d$ and $\lambda^*\theta \geq d/2$. Hence

$$X_d - X_{i,0} \leq \sup_{j \geq i/2} S_j^+.$$

By calculations similar to the proof of Proposition 4.3 it follows that

$$C(i) \leq \sum_{j \geq i/2} ||S_j^+||^2.$$

On the other hand, (4.3) implies that $E(e^{\gamma(S_j+1-S_j)}) \leq \kappa$ for $j \geq 0$, and so $E(e^{\gamma S_j}) \leq e^{\gamma/\lambda^* \kappa^j}$. We conclude the proof in a way similar to the proof of Proposition 4.3.

Proof of Theorem 6.1

We first prove the following lemma.

**Lemma L.1.** Let $(\nu_i)$, $0 \leq i \leq d$, be a decreasing sequence such that $\nu_{m_l} \leq \text{Var}(\phi_l - \phi_i)$ for $0 \leq l \leq L$, with $\nu_d = 0$. Then

$$\sum_{i=0}^{d-1} \sqrt{\frac{\nu_i}{i+1}} \leq 2 \sum_{l=1}^{L} \sqrt{m_l V_l}.$$

**Proof.** Since the sequence $(\nu_i)$ is decreasing and $(i + 1)^{-1/2} \leq 2(\sqrt{i+1} - \sqrt{i})$ for $i \geq 0$,

$$\sum_{i=j}^{k-1} \sqrt{\frac{\nu_i}{i+1}} \leq 2(\sqrt{k} - \sqrt{j})\sqrt{\nu_j},$$

On the other hand, (4.3) implies that $E(e^{\gamma(S_j+1-S_j)}) \leq \kappa$ for $j \geq 0$, and so $E(e^{\gamma S_j}) \leq e^{\gamma/\lambda^* \kappa^j}$. We conclude the proof in a way similar to the proof of Proposition 4.3.

Proof of Theorem 6.1

We first prove the following lemma.

**Lemma L.1.** Let $(\nu_i)$, $0 \leq i \leq d$, be a decreasing sequence such that $\nu_{m_l} \leq \text{Var}(\phi_l - \phi_i)$ for $0 \leq l \leq L$, with $\nu_d = 0$. Then

$$\sum_{i=0}^{d-1} \sqrt{\frac{\nu_i}{i+1}} \leq 2 \sum_{l=1}^{L} \sqrt{m_l V_l}.$$

**Proof.** Since the sequence $(\nu_i)$ is decreasing and $(i + 1)^{-1/2} \leq 2(\sqrt{i+1} - \sqrt{i})$ for $i \geq 0$,
for $0 \leq j \leq k \leq d$. Hence,

$$
\sum_{i=0}^{d-1} \sqrt{\frac{\nu_i}{i+1}} = \sum_{i=0}^{L-1} \sum_{i=m_i}^{m_{i+1}-1} \sqrt{\frac{\nu_i}{i+1}} \\
\leq 2 \sum_{l=0}^{L-1} (\sqrt{m_{l+1}} - \sqrt{m_l}) \sqrt{\text{Std}(\phi_L - \phi_l)} \\
\leq 2 \sum_{l=1}^{L} \sqrt{m_l} (\text{Std}(\phi_L - \phi_{l-1}) - \text{Std}(\phi_L - \phi_l)) \\
\leq 2 \sum_{l=1}^{L} \sqrt{m_l} V_{l},
$$

where the last equation follows by sub-linearity of the standard deviation.

We now prove Theorem 6.1. Since $\phi_l$ is square-integrable and is a measurable function of $U_1, \ldots, U_{m_l}$, by Proposition 2.2 $C(m_l) \leq \text{Var}(\phi_L - \phi_l)$ for $0 \leq l \leq L$. By Proposition 2.1, the sequence $(C(i))$, $0 \leq i \leq d$, is decreasing, and so it satisfies the conditions of Lemma L.1. Thus,

$$
\sum_{i=0}^{d-1} \sqrt{\frac{C(i)}{i+1}} \leq 2 \sum_{l=1}^{L} \sqrt{m_l} V_{l}.
$$

Furthermore, by Proposition 2.5

$$
R(q; \nu^*) \leq 8c \left( \sum_{i=0}^{d-1} (\sqrt{i+1} - \sqrt{i}) \sqrt{C(i)} \right)^2.
$$

Since $\sqrt{i+1} - \sqrt{i} \leq (i + 1)^{-1/2}$, it follows that

$$
R(q; \nu^*) \leq 32c \left( \sum_{l=1}^{L} \sqrt{m_l} V_{l} \right)^2.
$$

Using (6.1) concludes the proof.

**M Proof of Proposition 6.1**

We assume that $d \geq 2$. By standard properties of conditional expectations (Borovkov 2013, p. 98, Property 9), the minimum of $||\phi_L - Z||$ over all real-valued random variables $Z$ which are measurable with respect to the $\sigma$-algebra generated by $U_1, \ldots, U_{m_l}$ is attained at $Z = E(\phi_L|U_1, \ldots, U_{m_l})$. Thus, for any real-valued random variable $Z$ which is a deterministic measurable function of $U_1, \ldots, U_{m_l}$,

$$
||\phi_L - E(\phi_L|U_1, \ldots, U_{m_l})||^2 \leq ||\phi_L - Z||^2.
$$

In particular, when $Z = \phi_l + E(\phi_L - \phi_l)$, it follows that

$$
||\phi_L - E(\phi_L|U_1, \ldots, U_{m_l})||^2 \leq \text{Var}(\phi_L - \phi_l).
$$
But, for $1 \le i \le d$,
\[
f(U) - E(f(U)|U_1, \ldots, U_i) = U_1 \sum_{j=i+1}^{d} U_j,
\]
and so, by an easy calculation,
\[
||f(U) - E(f(U)|U_1, \ldots, U_i)||^2 = d - i.
\]
Thus $d - m_l \le \text{Var}(\phi_L - \phi_l)$ for $1 \leq l \leq L$. Since $\text{Var}(\phi_L) = d - 1$, we conclude that $2(d - m_l) \le \text{Var}(\phi_L - \phi_l)$ for $0 \leq l \leq L$. Thus the sequence $\nu_l = (d - l)/2$ verifies the conditions of Lemma \[1.1\] and so $\sum_{l=1}^{L} \sqrt{m_l} V_l = \Omega(d)$. Since $t_l \geq c m_l$ for $1 \leq l \leq L$, it follows that $T_{ML, \text{Var}}(\hat{\phi}) = \Omega(d^2)$. On the other hand, a simple calculation shows $C(i) = 0$ for $1 \leq i \leq d$, and so, if we set $q_0 = 1$ and $q_i = 1/d$ for $1 \leq i \leq d - 1$, then $R(q, \nu^*) = O(d)$.

References

Acworth, P. A., Broadie, M. and Glasserman, P. (1998). A comparison of some Monte Carlo and quasi Monte Carlo techniques for option pricing, in H. Niederreiter, P. Hellekalek, G. Larcher and P. Zinterhof (eds), Monte Carlo and Quasi-Monte Carlo Methods 1996, Vol. 127 of Lecture Notes in Statistics, Springer New York, pp. 1–18.

Äkesson, F. and Lehoczky, J. P. (2000). Path generation for quasi-Monte Carlo simulation of mortgage-backed securities, Management Science 46(9): 1171–1187.

Andrew, A. M. (1979). Another efficient algorithm for convex hulls in two dimensions, Inf. Process. Lett. 9(5): 216–219.

Asmussen, S. and Glynn, P. W. (2007). Stochastic simulation: algorithms and analysis, Vol. 57, Springer Science & Business Media.

Borovkov, A. A. (2013). Probability Theory, Universitext, Springer, London.

Caflisch, R. E. (1998). Monte Carlo and quasi-Monte Carlo methods, Acta Numerica 7: 1–49.

Caflisch, R. E., Morokoff, W. J. and Owen, A. B. (1997). Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension, Journal of Computational Finance 1: 27–46.

Feldman, Z., Mandelbaum, A., Massey, W. A. and Whitt, W. (2008). Staffing of time-varying queues to achieve time-stable performance, Management Science 54(2): 324–338.

Giles, M. B. (2008). Multilevel Monte Carlo path simulation, Operations Research 56(3): 607–617.

Glasserman, P. (2004). Monte Carlo methods in financial engineering, Vol. 53, Springer, New York.

Glynn, P. W. and Rhee, C.-h. (2014). Exact estimation for Markov chain equilibrium expectations, Journal of Applied Probability 51(A): 377–389.

Hull, J. (2014). Options, Futures and Other Derivatives, ninth edn, Prentice-Hall, Incorporated.

Imai, J. and Tan, K. S. (2006). A general dimension reduction technique for derivative pricing, Journal of Computational Finance 10(2): 129.
Kahalé, N. (2016). Optimized sampling for Monte Carlo simulations via dimension reduction, *9th NIPS Workshop on Optimization for Machine Learning*, Barcelona, Spain. http://opt-ml.org/index.html.

L’Ecuyer, P., Lécot, C. and Tuffin, B. (2008). A randomized quasi-Monte Carlo simulation method for Markov chains, *Operations Research* 56(4): 958–975.

Liu, R. and Owen, A. B. (2006). Estimating mean dimensionality of analysis of variance decompositions, *Journal of the American Statistical Association* 101(474): 712–721.

Nagel, K., Wagner, P. and Woesler, R. (2003). Still flowing: Approaches to traffic flow and traffic jam modeling, *Operations Research* 51(5): 681–710.

Owen, A. B. (2003). The dimension distribution and quadrature test functions, *Statistica Sinica* 13(1): 1–18.

Paxson, V. (1994). Empirically derived analytic models of wide-area TCP connections, *IEEE/ACM Transactions on Networking (TON)* 2(4): 316–336.

Revuz, D. and Yor, M. (1999). *Continuous martingales and Brownian motion*, Vol. 293 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*, third edn, Springer-Verlag, Berlin.

Rhee, C.-h. and Glynn, P. W. (2015). Unbiased estimation with square root convergence for SDE models, *Operations Research* 63(5): 1026–1043.

Rosenbaum, I. and Staum, J. (2017). Multilevel Monte Carlo Metamodeling, *Operations Research* 65(4): 1062–1077.

Shiryaev, A. N. (1996). *Probability*, Vol. 95 of *Graduate texts in mathematics*, Springer-Verlag, New York.

Sloan, I. H. and Woniakowski, H. (1998). When are quasi-Monte Carlo algorithms efficient for high dimensional integrals?, *Journal of Complexity* 14(1): 1 – 33.

Sobol, I. M. (2001). Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates, *Mathematics and computers in simulation* 55(1-3): 271–280.

Thompson, K., Miller, G. J. and Wilder, R. (1997). Wide-area internet traffic patterns and characteristics, *IEEE network* 11(6): 10–23.

Wang, X. (2006). On the effects of dimension reduction techniques on some high-dimensional problems in finance, *Operations Research* 54(6): 1063–1078.

Wang, X. and Fang, K.-T. (2003). The effective dimension and quasi-Monte Carlo integration, *Journal of Complexity* 19(2): 101 – 124.

Wang, X. and Sloan, I. H. (2005). Why are high-dimensional finance problems often of low effective dimension?, *SIAM Journal on Scientific Computing* 27(1): 159–183.

Wang, X. and Sloan, I. H. (2011). Quasi-Monte Carlo methods in financial engineering: An equivalence principle and dimension reduction, *Operations Research* 59(1): 80–95.

Wang, X. and Tan, K. S. (2013). Pricing and hedging with discontinuous functions: quasi-Monte Carlo methods and dimension reduction, *Management Science* 59(2): 376–389.

Whitt, W. (2017). Time-varying queues. Columbia University, New York, NY, http://www.columbia.edu/~ww2040/allpapers.html.
Whitt, W. and Ma, N. (2017). A rare-event simulation algorithm for periodic single-server queues. Columbia University, New York, NY, http://www.columbia.edu/~ww2040/allpapers.html.

Whitt, W. and You, W. (2016). Time-varying robust queueing. Columbia University, New York, NY, http://www.columbia.edu/~ww2040/allpapers.html.