Array-RQMC to Speed Up the Simulation for Estimating the Hitting-Time Distribution to a Rare Set of a Regenerative System

Marvin K Nakayama, Bruno Tuffin

Array-RQMC to Speed Up the Simulation for Estimating the Hitting-Time Distribution to a Rare Set of a Regenerative System

Marvin K. Nakayama and Bruno Tuffin

Abstract Estimating the distribution of the hitting time to a rarely visited set of states presents substantial challenges. We recently designed simulation-based estimators to exploit existing theory for regenerative systems that a scaled geometric sum of independent and identically distributed random variables weakly converges to an exponential random variable as the geometric's parameter vanishes. The resulting approximation then reduces the estimation of the distribution to estimating just the mean of the limiting exponential variable. The present work examines how randomized quasi-Monte Carlo (RQMC) techniques can help to reduce the variance of the estimators. Estimating hitting-time properties entails simulating a stochastic (here Markov) process, for which the so-called array-RQMC method is suited. After describing its application, we illustrate numerically the gain on a standard rare-event problem. This chapter combines ideas from several areas in which Pierre L'Ecuyer has made fundamental theoretical and methodological contributions: randomized quasi-Monte Carlo methods, rare-event simulation, and distribution estimation.

1 Introduction

Monte Carlo (MC) simulation provides a primary tool to estimate the probability of rare events or related indicators [27]. The extensive related literature focuses mainly on estimating the *mean* of a relevant random variable, but its *distribution* provides valuable additional information. For example, suppose a manufacturer wants to specify an appropriate length of a warranty. While the product's mean time to

Marvin K. Nakayama
Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA, e-mail: marvin@njit.edu

Bruno Tuffin
Inria, Univ Rennes, CNRS, IRISA, Campus de Beaulieu, 35042 Rennes, France, e-mail: bruno.tuffin@inria.fr

failure (MTTF) yields some relevant details, more useful are the random failure time's quantiles (i.e., inverse distribution). Setting the warranty length to, say, the 0.9-quantile leads to 10% of products resulting in warranty claims.

Distribution determination of a hitting time, especially related to a rare event (e.g., system failure), poses numerous challenges. But when the simulated stochastic process is *regenerative* [10], existing theory [11] shows that the hitting time to a rare set converges weakly to an exponential random variable if the probability to hit the rare set before regenerating converges to zero. This suggests approximating the hitting-time distribution by an exponential, reducing distribution estimation to estimating just its mean, which is broadly covered in the rare-event simulation literature. Our papers [4, 5] present two MC estimators exploiting such approximations. The *exponential estimator* directly applies this idea, further employing *measure-specific importance sampling* (MSIS) [6] to efficiently estimate the mean. The other is the *convolution estimator*, which first applies an exponential approximation to the distribution of the geometric sum of cycle lengths (i.e., the times elapsing between successive regenerations) completed before the first visit to the rare set, and then convolves this with the distribution of the hitting time given that it occurs in a cycle.

This chapter investigates how randomized quasi-Monte Carlo (RQMC) can be used to improve the accuracy of the above estimators and the potential associated gains. By distributing the sample points more evenly than independent sampling on the considered domain, RQMC methods can reduce the variance of estimators and even increase the convergence speed to the true value [13]. A naive implementation of RQMC to simulate a stochastic process entails generating sequences whose dimension is at least the number of transitions in a simulated path, which is typically large or even unbounded. But RQMC often performs poorly in large or infinite dimensions. Array-RQMC [15, 17, 19] has been designed precisely to simulate Markov chains while retaining the power of RQMC, the dimension of the generated sequences being "just" the required number of random values to simulate a single step of the chain. Basically, array-RQMC simulates in parallel a set of realizations of a Markov chain and makes use of a "sorting function" to reorder the chains according to their states after each simulation step. We describe the array-RQMC implementations of the exponential and convolution estimators and illustrate numerically the gains that can be derived from it.

Interestingly, this work combines several research interests of Pierre L'Ecuyer: rare-event simulation [16, 18]; RQMC techniques [13], among which array-RQMC [15, 17, 19] is specifically designed by Pierre and his coauthors to simulate Markov chains; and distribution determination [1, 21].

The remainder of this chapter unfolds as follows. Section 2 reviews the exponential and convolution estimators devised in [4]. Section 3 recalls array-RQMC simulation methods and how it can be implemented for our problem. As an illustration, we apply the approach in Section 4 to a standard rare-event problem in the literature: the hitting time to a large buffer threshold in an M/M/1 queue. Finally, Section 5 concludes the paper and provides further research directions to pursue on these ideas.

2 Regenerative-Simulation-Based Estimators of the Distribution of the Hitting Time to a Rarely Visited Set

2.1 Assumptions and Notations

For $\mathbb{R}_+$ denoting the set of nonnegative real numbers, we consider a positive recurrent Markov chain $(X(t) : t \in \mathbb{R}_+)$ defined on a discrete state space $\mathcal{S}$ for ease of exposition. Our goal is to estimate the cumulative distribution function (cdf) F of the hitting time $T = \inf\{t \geq 0 : X(t) \in \mathcal{A}\}$ of a subset $\mathcal{A}$ of $\mathcal{S}$, as well as the q-quantile $\xi = \xi_q = F^{-1}(q) = \inf\{t : F(t) \geq q\}$ of F (or of T) for some $q \in (0, 1)$.

Define regeneration times $0 = \Gamma_0 < \Gamma_1 < \cdots$ (always existing with our assumptions of discrete $\mathcal{S}$ and recurrence: it suffices to consider return times to a fixed state as regeneration times) and $\tau_k = \Gamma_k - \Gamma_{k-1}$, the length between regeneration $k-1$ and regeneration k for $k \geq 1$. The process "probabilistically restarts" at each regeneration time Γ_k. The process between successive regenerations is called a *cycle*, the k-th cycle being $(X(\Gamma_{k-1} + s) : 0 \leq s < \tau_k)$. The couples $(\tau_k, (X(\Gamma_{k-1} + s) : 0 \leq s < \tau_k) : k \geq 1)$ are independent and identically distributed (i.i.d.), and let τ denote a generic copy of τ_k. Let $T_k = \inf\{t \geq 0 : X(\Gamma_{k-1} + t) \in \mathcal{A}\}$ be the first hitting time to $\mathcal{A}$ after regeneration time Γ_{k-1}. We further define $M = \inf\{i \geq 1 : T_i < \tau_i\} - 1$ as the number of cycles completed before first hitting $\mathcal{A}$. As the cycles are i.i.d., M obeys a geometric distribution with parameter $p = \mathbb{P}(T < \tau)$ and support starting from 0; i.e., $\mathbb{P}(M = k) = (1-p)^k p$ for each $k \in \{0, 1, 2, \ldots\}$.

We can express

$$T = S + V \equiv \sum_{i=1}^{M} \tau_i + T_{M+1}, \tag{1}$$

where the regenerative property ensures the geometric sum $S = \sum_{i=1}^{M} \tau_i$ is independent of $V = T_{M+1}$. Define G as the cdf of S, and H the cdf of V. Note that H is the conditional cdf of T_1, given $T_1 < \tau_1$. Figure 1 illustrates the notation, with the state space $\mathcal{S}$ on the vertical axis and $\mathcal{A}$ the subset above the dashed line.

2.2 Exponential Limit

We consider a rare-event setting where the probability p to reach $\mathcal{A}$ before regeneration is small. To examine the asymptotic properties of estimators as the probability shrinks, we index the model and all notation by a rarity parameter $\epsilon > 0$, such that $p \equiv p_\epsilon \to 0$ as $\epsilon \to 0$. But when unambiguous, we will omit the index ϵ to simplify notation. Two well-known rare-event contexts having $p_\epsilon \to 0$ as $\epsilon \to 0$ are the following [7]:

- Stable queueing system: For a single-server queue with first-in-first-out discipline, we want to estimate the distribution of the hitting time T to a large buffer size buffer

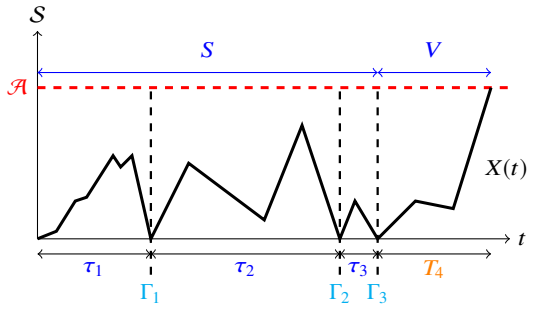

Fig. 1 Illustration of the notation used to represent and analyze regenerative processes, where $M = 3$.

size $b \equiv b_\epsilon = \lceil 1/\epsilon \rceil$. Specifically, $X(t)$ denotes the total number of customers in the system at time $t \geq 0$, and the state space is $\mathcal{S} = \{0, 1, 2, \ldots\}$. Thus, the hitting time is $T = \inf\{t \geq 0 : X(t) \in \mathcal{A}\}$ with $\mathcal{A} = \mathcal{A}_\epsilon = \{b_\epsilon, b_\epsilon + 1, \ldots\}$. For a G/G/1 queue, regenerations occur when a customer arrives to an empty system, which we assume occurs at time $t = 0$. In our numerical illustrations with an M/M/1 queue, returns to any fixed state constitutes a regeneration sequence, where we take the fixed state to be 0 and $X(0) = 0$. The transition kernel does not depend on ϵ, and rarity arises from b_ϵ being large for small ϵ, and [28] shows that $p_\epsilon \to 0$ as $\epsilon \to 0$. .

- Highly reliable Markovian system (HRMS) considered in dependability analysis: The system consists of components of different types, each having a specified redundancy. Each component is subject to failures and repairs, all being exponentially distributed with rates depending on the component type. Failure propagations can occur, i.e., a component failure can cause others to simultaneously fail. A state $x \in \mathcal{S}$ specifies the number of components failed of each type, as well as any other necessary information (e.g., about queueing of failed components waiting for repair) so that the resulting stochastic process on state space $\mathcal{S}$ is a Markov chain. The entire system is considered down (i.e., in $\mathcal{A}$) when specified combinations of components are currently failed. We may want to estimate the distribution of the hitting time to $\mathcal{A}$ when all components are operational at time $t = 0$. Rarity comes from failure rates being small (depending on ϵ) with respect to repair rates, leading to probabilistically long hitting times, and [29] provides conditions ensuring that $p_\epsilon \to 0$ as $\epsilon \to 0$.

Let $\mu_\epsilon = \mathbb{E}_\epsilon[T_\epsilon]$ be the mean hitting time, with $\mu_\epsilon \to \infty$ as $\epsilon \to 0$. Then existing limit results (see [10, 11]) show that if $p_\epsilon \to 0$ as $\epsilon \to 0$, then the normalized random variable T_ϵ/μ_ϵ converges weakly to an exponential, i.e., for $t \in \mathbb{R}$ and $t^+ = \max(t, 0)$,

$$\lim_{\epsilon \to 0} \mathbb{P}_\epsilon(T_\epsilon/\mu_\epsilon \leq t) = 1 - e^{-t^+}. \tag{2}$$

2.3 Exponential Estimators with Monte Carlo (MC)

From the limiting behavior (2), we can write for fixed small $\epsilon > 0$

$$F_\epsilon(t) = \mathbb{P}_\epsilon(T_\epsilon < t) = \mathbb{P}_\epsilon(T_\epsilon/\mu_\epsilon < t/\mu_\epsilon) \approx 1 - e^{-t^+/\mu_\epsilon} \equiv \widetilde{F}_\epsilon(t). \tag{3}$$

Thus to compute the cdf of hitting time T (dropping now the subscript ϵ to ease notation, as we will often but not always do in the following), we asymptotically need to know "just" its mean. (This is analogous to the central limit theorem (CLT), where the asymptotic normal cdf is fully specified through simply its mean and variance). Estimating the mean μ has been extensively studied in the literature. Doing this by averaging i.i.d. copies of T can be time-consuming because generating each observation of T typically entails lengthy simulations (e.g., many transitions) for small ϵ. Instead, we exploit the regenerative structure to rewrite μ as (see [6])

$$\mu = \frac{\mathbb{E}[T \wedge \tau]}{\mathbb{P}(T < \tau)} \equiv \frac{\zeta}{p}, \tag{4}$$

where $x \wedge y = \min(x, y)$. The key point is that (4) expresses μ in terms of cycle-based quantities, ζ and p, each of which can be estimated by simulating only cycles. The numerator $\zeta = \mathbb{E}[T \wedge \tau]$ in (4) can usually be estimated well by crude Monte Carlo, while the denominator $p = \mathbb{P}(T < \tau)$ is a small probability for which rare-event simulation techniques have to be applied for efficient estimation. Measure-specific importance sampling [6] employs independent simulations to estimate the numerator and denominator using crude simulation (CS) and importance sampling (IS), respectively. Given a computation budget of simulating n cycles in total to estimate μ, MSIS allocates a proportion $\gamma \in (0, 1)$ (resp., $1 - \gamma$) of the budget for CS (resp., IS). More specifically,

- We use $n_{\mathrm{CS}} \equiv \gamma n$ cycles to estimate the numerator ζ in (4) by CS via

$$\widehat{\zeta}_n = \frac{1}{n_{\mathrm{CS}}} \sum_{i=1}^{n_{\mathrm{CS}}} T_i \wedge \tau_i \tag{5}$$

  from n_{CS} independent observations $T_i \wedge \tau_i$ $(1 \le i \le n_{\mathrm{CS}})$ generated using the original system dynamics, denoted by $\mathbb{P}$.
- Because CS is unlikely to observe the event $T < \tau$ when p is small, MSIS instead estimates the denominator p in (4) using $n_{\mathrm{IS}} \equiv (1-\gamma)n$ cycles generated using IS. IS entails simulating under another probability measure $\mathbb{P}'$ rather than the original measure $\mathbb{P}$, where $\mathbb{P}'$ is chosen so that $T < \tau$ is more likely and can depend on ϵ. Letting $\mathcal{I}(\cdot)$ be the indicator function, we apply a "change of measure" to write

$$p = \mathbb{E}[\mathcal{I}(T < \tau)] = \int \mathcal{I}(T < \tau)\, \mathrm{d}\mathbb{P} = \int \mathcal{I}(T < \tau) L\, \mathrm{d}\mathbb{P}' = \mathbb{E}'[\mathcal{I}(T < \tau)L],$$

  with $L = \mathrm{d}\mathbb{P}/\mathrm{d}\mathbb{P}'$ the likelihood ratio, and $\mathbb{E}'$ denotes expectation under measure $\mathbb{P}'$. An unbiased estimator of p is then

$$\widehat{p}_n = \frac{1}{n_{\rm IS}} \sum_{i=1}^{n_{\rm IS}} \mathcal{I}(T_i' < \tau_i') L_i', \tag{6}$$

for i.i.d. copies $(\mathcal{I}(T_i' < \tau_i'), L_i')$, $i = 1, 2, \ldots, n_{\rm IS}$, of $(\mathcal{I}(T < \tau), L)$ under $\mathbb{P}'$.

The resulting MSIS estimator of the mean μ in (4) is the ratio estimator

$$\widehat{\mu}_n = \frac{\widehat{\zeta}_n}{\widehat{p}_n}. \tag{7}$$

The proportion γ can be selected during a presimulation run to minimize the variance per unit of computational budget of $\widehat{\mu}_n$ (see [6] for details). To summarize [4]:

Definition 1 The *exponential estimator* of the cdf $F(t)$ of T is

$$\widehat{F}_{{\rm exp},n}(t) = 1 - e^{-t^+/\widehat{\mu}_n}. \tag{8}$$

For fixed $q \in (0,1)$, the exponential estimator of the q-quantile $\xi = F^{-1}(q)$ is $\widehat{\xi}_{{\rm exp},n} = \widehat{F}^{-1}_{{\rm exp},n}(q) = -\widehat{\mu}_n \ln(1-q)$.

As a notational convention, for an unknown parameter (e.g., F), we use a tilde to signify a non-simulation approximation (e.g., $\widetilde{F}_\epsilon$ in (3)) based on a weak-convergence result, as in (2). A hatted variable (e.g., $\widehat{F}_{{\rm exp},n}$) denotes a simulation estimator.

The exponential estimators in Definition 1 result from approximating the true cdf F by $\widetilde{F}_\epsilon$ in (3), with (2) showing that the approximation becomes exact as the rarity parameter $\epsilon \to 0$. But any actual system has a small but *fixed* $\epsilon > 0$, which typically leads to $\widetilde{F}_\epsilon \neq F$. Because the exponential estimators are estimating quantities related to $\widetilde{F}_\epsilon$ and not the actual F, the estimators have bias that does not vanish as the computing budget $n \to \infty$. For example, for fixed $\epsilon > 0$ and $t > 0$, we have that as $n \to \infty$, $\widehat{F}_{{\rm exp},n}(t) \equiv \widehat{F}_{{\rm exp},n,\epsilon}(t)$ converges almost surely to $\widetilde{F}_\epsilon(t) = 1 - e^{-t/\mu_\epsilon}$, not to $F(t)$.

For fixed $\epsilon > 0$, the exponential estimators obey CLTs as $n \to \infty$, but the CLTs will employ centering constants computed from $\widetilde{F}_\epsilon$ rather than F. For example, for fixed $\epsilon > 0$ and $t > 0$, the exponential cdf estimator satisfies $\sqrt{n}[\widehat{F}_{{\rm exp},n}(t) - \widetilde{F}_\epsilon(t)] \Rightarrow \mathcal{N}(0, \psi_t^2)$ as $n \to \infty$ for an asymptotic variance $\psi_t^2 \equiv \psi_{t,\epsilon}^2$ that can be derived using the delta method, where $\Rightarrow$ denotes weak convergence and $\mathcal{N}(a, b^2)$ is a normal random variable with mean a and variance b^2. Similarly, for fixed $\epsilon > 0$ and $q \in (0,1)$, the exponential q-quantile estimator also obeys a CLT (as $n \to \infty$) with centering constant $\widetilde{\xi}_\epsilon \equiv -\mu_\epsilon \ln(1-q)$ rather than the true q-quantile $\xi = F^{-1}(q)$. Based on these two CLTs, we can then provide confidence intervals (CIs) for the true values $F(t)$ and ξ, but the CIs are biased from fixing $\epsilon > 0$, so the coverage probabilities will converge to 0 as $n \to \infty$. A CI may still have reasonable coverage when the estimator's bias makes a negligible contribution to its mean square error. This may be difficult to determine in practice, as quantifying the bias is nontrivial, but may occur for large (but not too large) n and fixed small $\epsilon > 0$.

2.4 Convolution Estimators with Monte Carlo

Rather than directly approximating the cdf F by an exponential, as done for the exponential estimator, we instead can use the decomposition $T = S + V$ in Equation (1), leading to expressing the cdf F as the convolution

$$F = G \star H, \quad \text{recalling that } S \sim G \text{ and } V \sim H \text{ are independent,} \tag{9}$$

where $(G \star H)(t) = \int H(t-y)\,\mathrm{d}G(y)$. Typically, the exponential limit (2) for the scaled hitting time arises from S (scaled by its mean $\eta = \mathbb{E}[S] = \mathbb{E}[M]\cdot\mathbb{E}[\tau \mid \tau < T]$) converging weakly to an exponential as $\epsilon \to 0$; e.g., see [11, Theorem 3.2.5]. Thus, for small $\epsilon > 0$, we approximate $G(y)$ by $\widetilde{G}_{\exp}(y) \equiv 1 - e^{-y^+/\eta}$. As before with the exponential estimator in (8), the approximation reduces estimation of the cdf G to estimating just its mean η. Writing $\mathbb{E}[M] = (1-p)/p$ and $\mathbb{E}[\tau \mid \tau < T] = \mathbb{E}[\tau \mathcal{I}(\tau < T)]/(1-p)$ suggests estimating $\eta = (1/p)\mathbb{E}[\tau \mathcal{I}(\tau < T)]$ by

$$\widehat{\eta}_n = \frac{1}{\widehat{p}_n\, n_{\mathrm{CS}}} \sum_{i=1}^{n_{\mathrm{CS}}} \tau_i \mathcal{I}(\tau_i < T_i),$$

where we can employ the same CS and IS cycle data from (5) and (6) used for the exponential estimator. This then yields the MSIS estimator of G in (9) as

$$\widehat{G}_{\exp,n}(t) = 1 - e^{-t/\widehat{\eta}_n}. \tag{10}$$

Estimating the cdf H of V in (9) also requires rare-event simulation techniques. As $H(x) = \mathbb{P}(T \le x \mid T < \tau) = \mathbb{P}(T \le x, T < \tau)/p$, a change of measure gives

$$H(x) = \frac{1}{p}\mathbb{E}[\mathcal{I}(T \wedge \tau \le x,\, T < \tau)] = \frac{1}{p}\mathbb{E}'[\mathcal{I}(T \wedge \tau \le x,\, T < \tau)\, L].$$

Applying IS produces a sample $(T_i' \wedge \tau_i', \mathcal{I}(T_i' < \tau_i'), L_i')$, $i = 1, 2, \ldots, n_{\mathrm{IS}}$, of $(T \wedge \tau, \mathcal{I}(T < \tau), L)$ under $\mathbb{P}'$ from n_{IS} cycles (as for the exponential estimator), leading to

$$\widehat{H}_n(x) = \frac{1}{\widehat{p}_n\, n_{\mathrm{IS}}} \sum_{i=1}^{n_{\mathrm{IS}}} \mathcal{I}(T_i' \wedge \tau_i' \le x, T_i' < \tau_i') L_i' \tag{11}$$

as an estimator of H. Convolving the two distributions $\widehat{G}_{\exp,n}$ from (10) and $\widehat{H}_n$ from (11), [4] obtains the following estimator of cdf F in (9).

Definition 2 The *convolution estimator* of the cdf $F(t)$ is

$$\widehat{F}_{\mathrm{conv},n}(t) = (\widehat{G}_{\exp,n} \star \widehat{H}_n)(t) = 1 - \frac{1}{\widehat{p}_n \cdot n_{\mathrm{IS}}} \sum_{i=1}^{n_{\mathrm{IS}}} \mathcal{I}(T_i' < \tau_i')\, L_i'\, e^{-(t-(T_i' \wedge \tau_i'))^+/\widehat{\eta}_n}.$$

The convolution estimator of the q-quantile $\xi = F^{-1}(q)$ is $\widehat{\xi}_{\mathrm{conv},n} = \widehat{F}^{-1}_{\mathrm{conv},n}(q)$, which typically requires numerical methods to compute.

The basis of the convolution estimators is the weak convergence of the geometric sum S_ϵ in (1) (scaled by its mean) to an exponential as $\epsilon \to 0$, which can hold even when the exponential limit for $T_\epsilon = S_\epsilon + V_\epsilon$ in (2) does not. This can happen when V_ϵ is not "negligible" compared to S_ϵ, as occurs, e.g., for the model described in [23, Section 5]. By explicitly taking into consideration the contribution of V_ϵ to T_ϵ, the convolution estimator can then have smaller bias than its exponential counterpart, as seen in Figure 5 of [5].

Constructing (biased, as discussed in the last paragraph of Section 2.3) CIs based on the convolution estimators requires that the estimators obey corresponding CLTs (with centering constants derived from $\widetilde{G}_{\exp} \star H$ rather than $G \star H$), which we have not yet established (but are working on). This statement even applies for batching CIs, as they also rely on an underlying CLT for each batch. A complication in establishing such CLTs is that in contrast to, e.g., the exponential cdf estimator in (8), the convolution cdf estimator is not simply a function of sample means, so the delta method does not directly apply.

3 Array-RQMC Implementation of Regenerative-Simulation-Based Estimators of Quantiles

3.1 RQMC and Array-RQMC

Quasi-Monte Carlo (QMC) is a deterministic numerical integration method (usually considered over the s-dimensional unit cube $[0,1]^s$ without much loss of generality) to approximate an integral $I \equiv \int_{[0,1]^s} f(x)\,\mathrm{d}x$ of a given function f. QMC approximates I by an average of evaluations of f over m values from a deterministic sequence $\mathcal{P} = (\theta_i)_{1 \le i \le m}$ of points from $[0,1]^s$; i.e., the QMC estimator of the integral is $\frac{1}{m}\sum_{i=1}^m f(\theta_i)$. The sequence $\mathcal{P}$ of points is designed to "evenly" cover the space $[0,1]^s$ and is known as a *low-discrepancy sequence*. The most common constructions are lattice points and digital nets, including Sobol' sequences [2, 25]. Error bounds exist [25] under restrictive assumptions, showing that the QMC error shrinks at a rate in $O(m^{-1}(\log m)^s)$ as $m \to \infty$ (and sometimes even faster), better than the $O(m^{-1/2})$ convergence rate of MC's root-mean-square error. (For non-negative functions g_1 and g_2, "$g_1(m) = O(g_2(m))$ as $m \to \infty$" means there are positive constants c and m_0 such that $g_1(m) \le c g_2(m)$ for all $m \ge m_0$.) But applying such bounds is impractical: they are very difficult to compute and can be extremely loose for a given integrand f or a given value of m. RQMC, which has several advantages over QMC, randomizes the sequence $\mathcal{P}$ such that each point of the sequence is uniformly distributed over $[0,1]^s$ but the points are correlated and keep the low discrepancy to gain the improved convergence rate with respect to MC [13]. We can apply a central limit theorem on $r \to \infty$ i.i.d. randomizations to obtain a confidence interval for I (see [24] for conditions).

QMC and RQMC efficiency is sensitive to the problem's dimension s (or actually to the *effective dimension* representing the number of coordinates encompassing "most of the variability" of the problem; see [26] for more details). But in a naive implementation of (R)QMC to simulate paths of a Markov chain, the dimension s corresponds to the maximum length of a simulated path, which can be large, even infinite in many cases (as when generating paths up to an unbounded hitting time). In most such situations, RQMC is typically considered useless, yielding no improvement with respect to MC except if the effective dimension is small, which happens only in restricted cases.

To cope with this dimensionality issue, the array-RQMC method has been designed in [15] and further developed in [17] to adapt RQMC to the simulation of Markov chains. As a randomization of the deterministic QMC version presented in [12], array-RQMC simulates a Markov chain $(X_j,\ j \geq 0)$ defined on a state space $\mathcal{S}$ as follows. It assumes a total ordering function h of states in $\mathcal{S}$. Let the initial state X_0 be distributed according to some distribution ν_0. (In our regenerative setting of Section 2, we will assume that ν_0 is degenerate, so there is a single fixed starting state, but we recall here the nondegenerate-ν_0 version introduced in [15] for sake of generality.) Transitions of the chain are defined by the stochastic recurrence

$$X_j = \varphi(X_{j-1}, U_j), \quad (j \geq 1), \tag{12}$$

for a given transition kernel φ, where U_j (independent for different j) is a random vector uniformly distributed over $[0,1)^d$, meaning that d uniforms are used to simulate a single transition step.

While MC typically simulates n chains sequentially and independently, array-RQMC instead generates m chains *in parallel*, simulating the jth step of all the m paths in a negatively correlated way (to reduce the variance in the estimation) before moving to the next step for each path. For $i = 1, 2, \ldots, m$, let $(X_{i,j} : j = 0, 1, 2, \ldots)$ be the ith path generated, with $X_{i,j}$ as the state visited after the jth step. To begin, m initial states $X_{i,0}$ (for $i = 0, \ldots, m$) are generated from the initial distribution ν_0 using an RQMC point set $\mathcal{P}_{m,0} = \{U_{0,0}, \ldots, U_{m-1,0}\}$ in $[0,1)^{d_0}$ (that is, at most d_0 uniforms are used to generate an initial state); from the property of RQMC points being well distributed over the space, this results in m "well spread" (according to ν_0) initial points for the m chains. The m chains are then sorted (say in increasing order of their state) according to h. Then for the transitions from step $j-1$ to step j (for $j \geq 1$), the next state for each of the m chains is sampled from the previously sorted ones. An RQMC point set $\mathcal{P}_{m,j} = \{U_{0,j}, \ldots, U_{m-1,j}\}$ in $[0,1)^d$ independent from previous RQMC point sets is used such that for all $i \in \{1, \ldots, m\}$, the transition of the i-th (ordered) chain is generated using the i-th point of $\mathcal{P}_{m,j}$:

$$X_{i,j} = \varphi(X_{i,j-1}, U_{i,j}).$$

And again the states are re-ordered according to h. The process is iterated up to the end of the paths. If the chains have different stopping times, the below algorithm ignores those terminated paths (and not simulated anymore) and their states are

specified as ∞ (an absorbing state used to indicate that those simulated paths have already reached the stopping time.)

The algorithm can therefore be described as follows for a discrete-time Markov chain, which we will later modify (at the end of Section 3.2) to handle a continuous-time Markov chain, as needed for the array-RQMC convolution estimator:

Array-RQMC algorithm [15]:
1 (Initialization).
Generate a RQMC point set, $\mathcal{P}_0 = \{U_{0,0}, \ldots, U_{m-1,0}\} \subset [0,1)^{d_0}$;
$\forall\, i \in \{0, 1, \ldots, m-1\}$, generate $X_{i,0}$ from $U_{i,0}$;
2 (Simulate chains).
Simulate in parallel m copies of the chain, numbered $0, \ldots, m-1$, as follows:
For ($j = 1$; $X_{0,j-1} < \infty$; j++)
Generate an RQMC point set $\mathcal{P}_{m,j} = \{U_{0,j}, \ldots, U_{m-1,j}\} \subset [0,1)^d$
(independent of previous ones);
For all non-terminated chains i, let $X_{i,j} = \varphi_j(X_{i,j-1}, U_{i,j})$;
For terminated chains (i.e., stopping time reached), set $X_{i,j} = \infty$;
Sort (and renumber) the chains for which $X_{i,j} < \infty$ by increasing
order of their states (based on the ordering function h);
(The sorted states $X_{0,j}, \ldots, X_{n-1,j}$ result in an estimator $\hat{F}_j$
of the cdf F_j of the chain at the jth step X_j.)
3 (Output).
Return the estimator obtained from the m generated paths.

The algorithm simulates each transition step across the m chains according to an RQMC point set with good coverage properties over the sampling space. The re-ordering helps to obtain an empirical cdf of the random variable X_j at $(j-1)$-th step of the chain, so that the RQMC point set at step j is actually generating step-j values from this empirical cdf, from a $(d+1)$-dimensional point set where the first coordinate of the i-th point is i/m and the d other coordinates $U_{i,j}$ (see [15, 17]).

But the main advantage of using array-RQMC with respect to traditional RQMC techniques is that the dimension of the RQMC point sets is $\max(d, d_0)$ for array-RQMC, as compared to $d_0 + d \times \tau'$ for traditional RQMC, where τ' is an upper bound (possibly infinite) for the stopping time τ. Hence, array-RQMC drastically reduces the dimension, from which efficiency improvements can be expected. Actually it is shown in [15, 17] that if stratified sampling is used, the variance of a mean estimator can be $O(m^{-3/2})$ as $m \to \infty$, much faster than the $O(m^{-1})$ for MC. In fact, [17, 19] present numerical results that suggest variances can even shrink as $O(m^{-2})$.

We can easily obtain a confidence interval by considering $r \geq 2$ independent replications (i.e., randomizations) of groups of m chains.

The algorithm is sensitive to the choice of ordering function h. When the state space $\mathcal{S}$ is a (one-dimensional) subset of $\mathbb{R}$, the states have a natural order. But difficulties arise for state spaces of higher dimension, for which it may not be obvious how to design an effective ordering of the states. This issue is related to that of defining an importance function for the levels in the splitting technique in

rare-event simulation [16]: an effective ordering is problem-specific, depending on both the stochastic model and on what is being estimated.

3.2 Array-RQMC Exponential and Convolution Estimators

We explain now how we propose to apply array-RQMC to the exponential and convolution estimators of Section 2. Recall that we start in a fixed regenerative state so the initial distribution ν_0 in the Array-RQMC is degenerate; no sampling is required to specify the initial state. We start with the exponential estimator in Definition 1 of Section 2.3. Recall that this estimator exponentiates the ratio of the estimators $\widehat{\zeta}_n$ and $\widehat{p}_n$ in Equation (7), with $\widehat{\zeta}_n$ an average over n_{CS} cycles and $\widehat{p}_n$ averaging over n_{IS} cycles, where n_{CS} and n_{IS} may differ. For array-RQMC, we propose to consider a set of (a fixed number) m chains generated in parallel and to apply r_{CS} and r_{IS} independent randomizations of groups of m chains for estimating ζ and p, respectively, from (4). Because (R)QMC methods often work best for point sequences $\mathcal{P}$ of certain specific sizes (e.g., powers of 2), the array-RQMC exponential estimator specifies the same number m of chains for CS and IS, but the randomizations for CS and IS allow for unequal allocations (i.e., different r_{CS} and r_{IS}). By applying independent sets of replications to estimate ζ and p, we are able to estimate the variance of the exponential estimator and construct a (biased; see the discussion at the end of Section 2.3 confidence interval based on a CLT (with $1 - e^{-t/\mu}$ as the centering constant due to the bias from fixing $\epsilon > 0$ in (3)), provided $r_{\mathrm{CS}} \to \infty$ and $r_{\mathrm{IS}} \to \infty$.

Formally, denote by $\widehat{\zeta}_m^{(k)}$ ($k \in \{1, \ldots, r_{\mathrm{CS}}\}$) and $\widehat{p}_m^{(k)}$ ($k \in \{1, \ldots, r_{\mathrm{IS}}\}$) as the estimators of ζ and p respectively for the k-th independent group of cycles sampled from array-RQMC. Specifically, we have

$$\widehat{\zeta}_m^{(k)} = \frac{1}{m} \sum_{i=1}^{m} T_i^{(k)} \wedge \tau_i^{(k)}$$

with $T_i^{(k)} \wedge \tau_i^{(k)}$ the minimum of the hitting time and cycle length for the i-th generated array-RQMC chain of the k-th independent replication of groups under crude simulation. Also, we get

$$\widehat{p}_m^{(k)} = \frac{1}{m} \sum_{i=1}^{m} \mathcal{I}(T_i^{'(k)} < \tau_i^{'(k)}) L_i^{'(k)},$$

with $\mathcal{I}(T_i^{'(k)} < \tau_i^{'(k)})$ and $L_i^{'(k)}$ as the indicator of hitting $\mathcal{A}$ before regenerating and the likelihood ratio, respectively, for the i-th generated array-RQMC chain of the k-th independent replication of groups under IS.

The estimators of ζ, p and mean hitting time μ are then

$$\widehat{\zeta}_{m,r}^{\text{aRQMC}} = \frac{1}{r_{\text{CS}}} \sum_{k=1}^{r_{\text{CS}}} \widehat{\zeta}_m^{(k)}, \quad \widehat{p}_{m,r}^{\text{aRQMC}} = \frac{1}{r_{\text{IS}}} \sum_{k=1}^{r_{\text{IS}}} \widehat{p}_m^{(k)}, \quad \widehat{\mu}_{m,r}^{\text{aRQMC}} = \frac{\widehat{\zeta}_{m,r}^{\text{aRQMC}}}{\widehat{p}_{m,r}^{\text{aRQMC}}},$$

from which the *array-RQMC exponential estimator* of the cdf $F(t)$ of T is

$$\widehat{F}_{\text{exp},m,r}^{\text{aRQMC}}(t) = 1 - e^{-t/\widehat{\mu}_{m,r}^{\text{aRQMC}}}. \tag{13}$$

From the independent replications of groups of m parallel chains, we can obtain variance estimators of $\widehat{\zeta}_{m,r}^{\text{aRQMC}}$, $\widehat{p}_{m,r}^{\text{aRQMC}}$, and $\widehat{\mu}_{m,r}^{\text{aRQMC}}$, leading to a (biased) CI for $F(t)$ derived similarly to what is done for MC in [4] from the CLT described in the last paragraph of Section 2.3, where an estimator of the asymptotic variance ψ_t^2 can be computed from the sample variances of $\widehat{\zeta}_m^{(k)}$, $k = 1, 2, \ldots, r_{\text{CS}}$, and $\widehat{p}_m^{(k)}$, $k = 1, 2, \ldots, r_{\text{IS}}$.

We specify an allocation of the $r = r_{\text{CS}} + r_{\text{IS}}$ independent groups of chains between the crude and IS simulations with $r_{\text{CS}} = \gamma' r$ and $r_{\text{IS}} = (1 - \gamma')r$ for a user-specified constant $\gamma' \in (0, 1)$. From a pre-simulation, we can choose γ' with the goal to minimize the work-normalized variance [18] of the mean-hitting-time estimator $\widehat{\mu}_{m,r}^{\text{aRQMC}}$, similarly to what is done for MC [6, 4]. The optimal allocation parameter γ' for array-RQMC can differ from γ for MC in Section 2.3.

As explained in the last paragraph of Section 2.4, providing a CI (even with batching) using the convolution estimator requires a CLT, which we have not yet established for MC (although we are currently working on it). If we then decide to forgo a CI based on the convolution estimator, then we could just consider a single group (i.e., $r = 1$) to decompose the full budget $n = r \times m = m$ into m_{CS} and m_{IS} with $m_{\text{CS}} + m_{\text{IS}} = m$. But then the convolution estimator has an unfair advantage over the exponential estimator with the same total budget because the former is based on a larger QMC point sequence (and QMC has faster convergence than MC, which corresponds to the randomizations). As such, our numerical experiments in Section 4 construct the convolution estimator with the same allocation (with r_{CS} and r_{IS}) that is used for the exponential estimator.

As studied in [17, 19], the efficacy of array-RQMC depends critically on the choice of the ordering function h, but we do not pursue that issue here. When the state space $\mathcal{S}$ is a one-dimensional subset of $\mathbb{R}$, as in the M/M/1 example that we will study numerically in Section 4, there is a natural ordering of states, which can be effective.

Recall also that d is the number of uniforms employed to simulate a single transition step in (12). The exponential estimator in Definition 1 requires estimating only the mean μ in (4), so discrete-time conversion [3, 9] can be applied. Specifically, to estimate μ, we need to generate only the embedded discrete-time Markov chain (DTMC), replacing the exponential holding times in each successive state visited by its conditional mean (given the DTMC). In addition to reducing the number of uniforms needed to generate each transition, discrete-time conversion (as a form of conditional Monte Carlo) also reduces (asymptotic) variance. Thus, generating a path of the DTMC typically has $d = 1$ in (12), as a single DTMC step requires only a single uniform, even if for some applications using $d > 1$ may lead to more efficient

implementations. But for the convolution estimator, discrete-time conversion cannot be applied when estimating the cdf H in (11) since it further requires the actual exponential holding times in each state visited. Thus, the number of uniforms needed to generate each transition is in this case $d' = d + d_g$, where d_g stands for the number of uniforms to generate the random holding time once the new state is selected. In our examples, we will typically have $d_g = 1$, those times being exponentially distributed, generated from the inversion procedure of a single uniform.

4 Numerical Illustration of the Gain on the Simulation of an M/M/1 Queue

To study the effectiveness of array-RQMC, consider the simulation of an M/M/1 queue, also studied in [15], with arrival rate $\lambda = 1.0$ and service rate $\mu' = 4.0$. For the process $(X(t) : t \in \mathbb{R}_+)$ with $X(t)$ denoting the total number of customers in the system at time t and $X(0) = 0$, our goal is to estimate quantiles and the cdf F of the hitting time T to a given buffer size N. As in [4], we apply MSIS, where the IS swaps the arrival and service rates, an approach known to be efficient when using the ratio estimator (7) of the mean hitting time as the buffer size increases, and therefore hitting times typically increase too. As explained in Section 3.2, the exponential and convolution estimators will use $r = r_{\text{CS}} + r_{\text{IS}}$ independent sets of randomizations of m parallel chains to estimate the variances of $\widehat{\zeta}_{m,r}^{\text{aRQMC}}$ and $\widehat{p}_{m,r}^{\text{aRQMC}}$ and obtain a (biased; see the last paragraph of Section 2.3) confidence interval. All the results are each time compared with the MC exponential and convolution estimators with a total of $n = m \times r$ MSIS cycles.

Our experiments test different sets of RQMC point sets, among classical ones:

- Sobol' with a left matrix scrambling (named Matoušek scrambling [22]);
- Randomly-shifted lattice rule [14, 31] with lattice points selected using [20];
- (The same) Randomly-shifted lattice rule plus baker's transformation [8];
- Randomly-shifted Sobol' sequence (often yielding good numerical results, see for example [30]).

Table 1 displays the outputs for three different quantiles $F^{-1}(q)$ (when $q = 0.1$, 0.5 and 0.9). For constructing the exponential estimator, array-RQMC uses $r = 100$ independent randomizations of $m = 2^{14}$ parallel chains, compared with $r \times m$ cycles for MC. To simplify the following discussion about array-RQMC, we will focus on the exponential estimator (but similar comments also apply to the convolution estimator). The array-RQMC exponential q-quantile estimator applies array-RQMC to independently estimate ζ by CS and p by IS to handle the ratio μ from (4). The performance of an array-RQMC estimator depends critically on the choice of the ordering function h (Section 3), which should be tailored for the particular estimand. We could try to select different h for ζ and p (see [16, 17] for discussions on this), taking into account, e.g., the accumulated "reward" (time already spent for CS or accumulated likelihoood ratio for IS) to which an approximation of the remaining

reward is appended. But our experiments instead simply had that CS and IS used the same ordering function h (the number of customers in the system), which gave similar results.

Table 1 Results for the M/M/1 queue with $\lambda = 1.0$, $\mu' = 4.0$ when estimating q-quantiles $F^{-1}(q)$ of hitting times to $N = 10$. We consider $r = 100$ and $m = 2^{14}$ for RQMC and a total of $r \times m$ cycles for MC. Exact values are 4.91036e+04 for $q = 0.1$, 3.230287e+05 for $q = 0.5$ and 1.073074e+06 for $q = 0.9$.

q	Method	Exp. Est.	Conf. Interval	Variance	Conv. Est.
0.1	MC	4.9056e+04	(4.8990e+04, 4.9121e+04)	1.12e+03	4.9146e+04
0.1	Matoušěk scrambling	4.9102e+04	(4.9099e+04, 4.9104e+04)	2.08e+00	4.9102e+04
0.1	Lattice-shift	4.9104e+04	(4.9099e+04, 4.9110e+04)	7.81e+00	4.9105e+04
0.1	Lattice-shift + baker	4.9104e+04	(4.9099e+04, 4.9109e+04)	5.66e+00	4.9100e+04
0.1	Sobol-shift	4.9102e+04	(4.9099e+04, 4.9105e+04)	2.14e+00	4.9101e+04
0.5	MC	3.2273e+05	(3.2230e+05, 3.2316e+05)	4.85e+04	3.2330e+05
0.5	Matoušěk scrambling	3.2303e+05	(3.2301e+05, 3.2305e+05)	8.98e+01	3.2301e+05
0.5	Lattice-shift	3.2305e+05	(3.2301e+05, 3.2309e+05)	3.38e+02	3.2303e+05
0.5	Lattice-shift + baker	3.2305e+05	(3.2302e+05, 3.2308e+05)	2.45e+02	3.2300e+05
0.5	Sobol-shift	3.2303e+05	(3.2301e+05, 3.2305e+05)	9.25e+01	3.2300e+05
0.9	MC	1.0721e+06	(1.0706e+06, 1.0735e+06)	5.36e+05	1.0740e+06
0.9	Matoušěk scrambling	1.0731e+06	(1.0730e+06, 1.0731e+06)	9.91e+02	1.0730e+06
0.9	Lattice-shift	1.0731e+06	(1.0730e+06, 1.0733e+06)	3.73e+03	1.0731e+06
0.9	Lattice-shift + baker	1.0731e+06	(1.0730e+06, 1.0732e+06)	2.70e+03	1.0730e+06
0.9	Sobol-shift	1.0731e+06	(1.0730e+06, 1.0732e+06)	1.02e+03	1.0730e+06

Column 5 of Table 1 shows that compared to MC for the same total number of cycles generated, array-RQMC drastically reduces the variance of the estimators for each quantile level q. The variance-reduction factor (i.e., ratio of variances for MC and array-RQMC) is always well over 100, with the specific amount depending on the randomization technique and choice of the low-discrepancy sequence. For this example, the array-RQMC variances differ by up a factor of 4, with Matoušěk scrambling and randomly shifted Sobol' sequence the most effective. From the numerically computed exact quantile values 4.91036e+04 for $q = 0.1$, 3.230287e+05 for $q = 0.5$ and 1.073074e+06 for $q = 0.9$, we see that array-RQMC estimators are more accurate than MC ones, and all competitive. Convolution estimators are accurate as well, expected to reduce the existing bias with respect to exponential ones [4, 5].

Figure 2 displays in a log-log scale the standard deviation of the exponential estimators in terms of $m = 2^k$ with fixed $r = 128$ for the various array-RQMC methods as well as for MC with $n = r \times m$ total cycles. We display only the results for the $q = 0.1$ quantile since all other quantiles have the same curve up to a multiplicative constant.

All array-RQMC estimators are of the same order of magnitude and outperform the MC one. Larger m yields greater variance reduction with respect to MC, as expected due to the benefit of the generated sequences' low discrepancy.

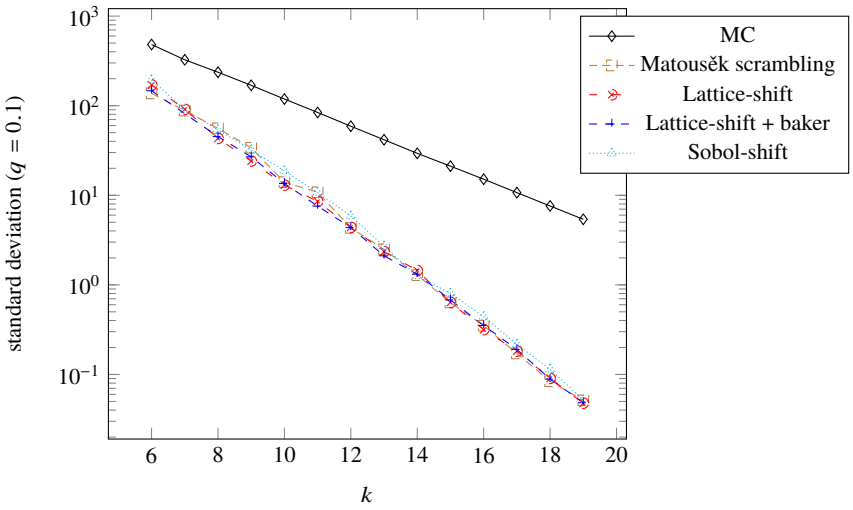

Fig. 2 Standard deviation of array-RQMC exponential estimators as a function of $m = 2^k$, for the M/M/1 queue with $\lambda = 1.0$ and $\mu' = 4.0$ when estimating q-quantiles of hitting times to $N = 10$ with $r = 128$. For MC, we consider $n = m \times r$.

The log-log curves in Figure 2 are close to linear. It is interesting to investigate the convergence rate of the standard deviation in terms of m. A standard procedure for convergence-rate estimation of QMC and RQMC methods applies log-log regression. Assume that the standard deviation σ_m as a function of m satisfies $\sigma_m \approx am^{-b}$ for some $a, b > 0$, which is equivalent to

$$\ln(\sigma_m) \approx \ln(a) - b \ln(m).$$

Applying a classical regression for the values of $m = 2^k$ with $k \in \{6, 7, \ldots, 19\}$ in Figure 2 leads to the regression coefficients in Table 2. The table verifies the $m^{-0.5}$

Table 2 Log-log regression corresponding to values of Figure 2 for the standard deviation of array-RQMC and MC exponential estimators as a function of m, on the M/M/1 queue model with $\lambda = 1.0, \mu' = 1.0$ when estimating q-quantiles for of hitting times to $N = 10$. For MC, we consider $n = r \times m$.

Method	$q = 0.1$
MC	$3702 \times m^{-0.4965}$
Matoušek scrambling	$7943 \times m^{-0.9055}$
Lattice-shift	$7068 \times m^{-0.8971}$
Lattice-shift + baker	$6615 \times m^{-0.8902}$
Sobol-shift	$8470 \times m^{-0.8969}$

convergence of MC. For all array-RQMC techniques, the standard deviation shrinks at about rate $m^{-0.9}$, much faster than MC.

From the known exact values for this M/M/1 model, Figure 3 displays in a log-log scale the error of the exponential *and convolution* estimators in terms of m with $r = 128$ fixed for the various array-RQMC methods as well as for MC with $n = r \times m$ ($r \times m$ is also used for all convolution estimators). Each plotted point is the average of the absolute errors obtained over $K = 10$ independent replications to smooth the curves with respect to drawing the error for a single replication.

Figure 3 shows that all array-RQMC techniques are of the same order of magnitude of accuracy, and order(s) of magnitude better than the corresponding MC accuracy. Also, for the "extreme" quantiles with $q = 0.1$ and $q = 0.9$, as $m = 2^k$ increases, the array-RQMC errors for the estimators seem to stabilize and converge to a positive (even if small) value, which results from the rarity parameter $\epsilon > 0$ being fixed in (3) because N is fixed. This suggests that the standard deviation is becoming negligible with respect to bias for the exponential estimation, meaning that the bias for the exponential estimator is larger than for the convolution estimator; this is more visible when q is small, e.g., for $q = 0.1$ (see also [5, Figure 5]). (Also see the related discussion for MC in the last two paragraphs of Section 2.3.) The stabilizing constant seems smaller for the convolution estimators than for the exponential ones, indicating a smaller bias for the convolution estimator. As noted before for MC in the penultimate paragraph of Section 2.4, the convolution estimator more explicitly accounts for the contribution of V to $R = S + V$ in (1) than the exponential estimator.

5 Conclusions

Estimating distributions of hitting times by simulation presents substantial challenges, especially when related to rarely visited sets. We previously designed [4, 5] MC methods to estimate distributions and quantiles of hitting times in a regenerative context, when hitting the rare set before regeneration is rare. Based on the limiting behavior of a geometric distribution converging to an exponential as when the success probability tends to zero, [4, 5] design two "simple" estimators using previous importance sampling designed to compute means. We proposed in this paper to combine the estimators with array-RQMC, a simulation method simulating paths of the Markov chain in parallel and distributing the sample points to cover more efficiently the space, hence reducing variance. We have illustrated on a standard example that the combination can reduce the variance by several orders of magnitude.

There are nevertheless several questions warranting further study. As noted in the last two paragraphs of Section 2.3, variance is not the only component of the simulation error. There is also bias coming from the exponential approximations, which become exact as the rarity parameter $\epsilon \to 0$ in, e.g., (2), but in practice we always have a fixed $\epsilon > 0$, resulting in bias in (3). Since array-RQMC can substantially reduce the variance, bias may significantly contribute to the estimator's mean-squared error, and increasing the sample size will not eliminate this source of

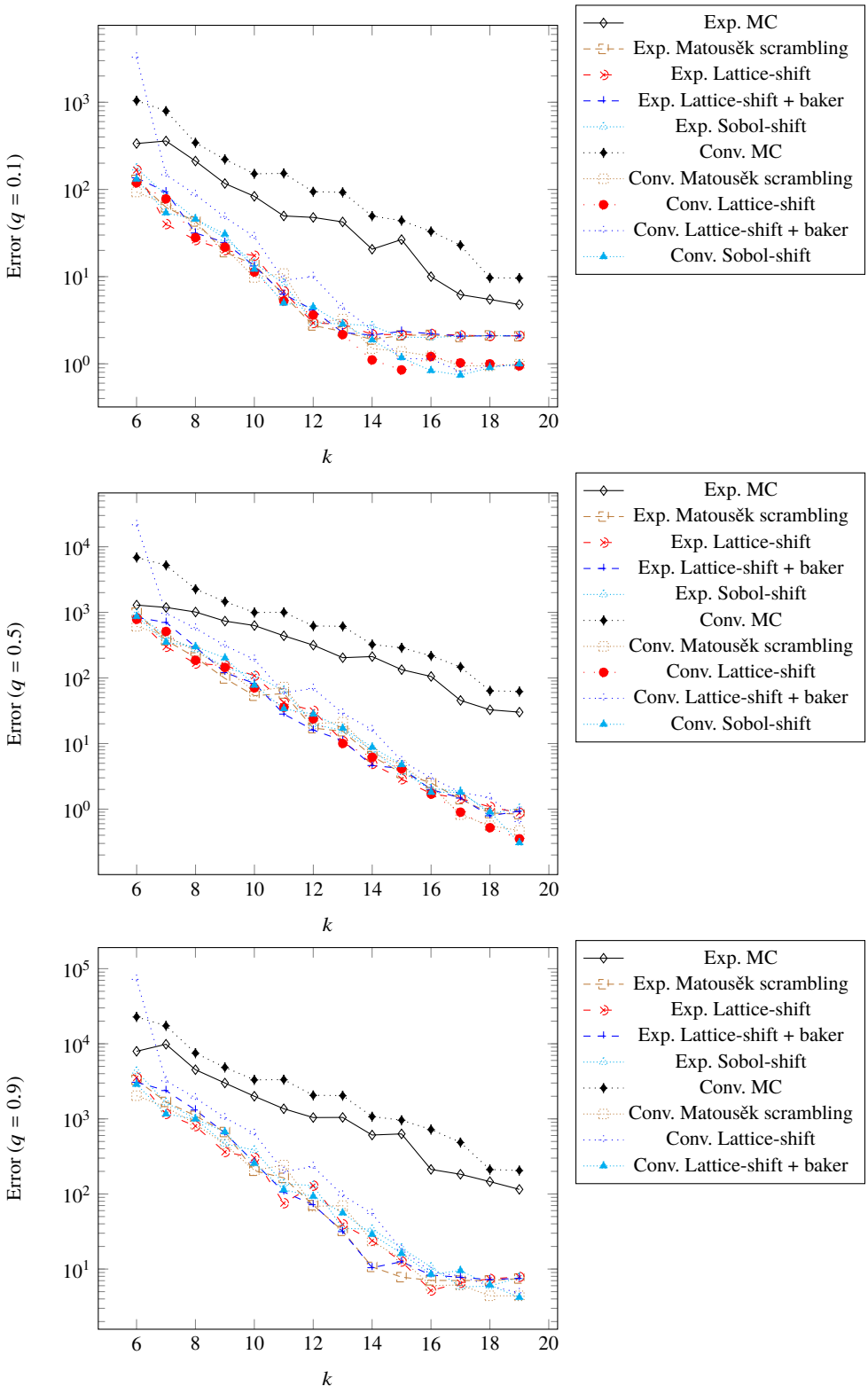

Fig. 3 Errors of the various exponential and convolution estimators as a function of m, for the M/M/1 queue with $\lambda = 1.0$, $\mu' = 4.0$ when estimating q-quantiles for of hitting times to $N = 10$ with $r = 128$. For MC, we consider $n = m \times r$. Each plotted point is the average of $K = 10$ independent replications.

bias. Thus, increasing the number m of parallel chains in array-RQMC can provide benefits up to a point, but eventually, bias from fixed $\epsilon > 0$ becomes the dominant issue. This issue deserves further study. Also, array-RQMC efficiency depends on the dimension of the state space $\mathcal{S}$ and of the RQMC point set. More investigations on this are required.

References

[1] Ben Abdellah A, L'Ecuyer P, Owen A, Puchhammer F (2021) Density estimation by randomized quasi-Monte Carlo. SIAM Journal on Uncertainty Quantification 9(1):280–301

[2] Dick J, Pillichshammer F (2010) Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration. Cambridge University Press, Cambridge, U.K.

[3] Fox BL, Glynn PW (1986) Discrete time conversion for simulating semi-Markov processes. Operations Research Letters 5:191–196

[4] Glynn PW, Nakayama MK, Tuffin B (2018) Using simulation to calibrate exponential approximations to tail-distribution measures of hitting times to rarely visited sets. In: Proceedings of the 2018 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, NJ

[5] Glynn PW, Nakayama MK, Tuffin B (2020) Comparing regenerative-simulation-based estimators of the distribution of the hitting time to a rarely visited set. In: Bae KH, Feng B, Kim S, Lazarova-Molnar S, Zheng Z, Roeder T, Thiesing R (eds) Proceedings of the 2020 Winter Simulation Conference, IEEE, Piscataway, New Jersey

[6] Goyal A, Shahabuddin P, Heidelberger P, Nicola V, Glynn PW (1992) A unified framework for simulating Markovian models of highly dependable systems. IEEE Transactions on Computers C-41(1):36–51

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

[8] Hickernell FJ (2002) Obtaining $O(N^{-2+\epsilon})$ convergence for lattice quadrature rules. In: Fang KT, Hickernell FJ, Niederreiter H (eds) Monte Carlo and Quasi-Monte Carlo Methods 2000, Springer-Verlag, Berlin, pp 274–289

[9] Hordijk A, Iglehart DL, Schassberger R (1976) Discrete-time methods for simulating continuous-time Markov chains. Advances in Applied Probability 8:772–788

[10] Kalashnikov V (1994) Topics on Regenerative Processes. CRC Press, Boca Raton

[11] Kalashnikov V (1997) Geometric Sums: Bounds for Rare Events with Applications. Kluwer Academic Publishers, Dordrecht, The Netherlands

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

[13] L'Ecuyer P (2018) Randomized quasi-Monte Carlo: An introduction for practitioners. In: Glynn PW, Owen AB (eds) Monte Carlo and Quasi-Monte Carlo Methods: MCQMC 2016, Springer, Berlin, pp 29–52

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

[15] L'Ecuyer P, Lécot C, Tuffin B (2006) Randomized quasi-Monte Carlo simulation of Markov chains with an ordered state space. In: Niederreiter H, Talay D (eds) Monte Carlo and Quasi-Monte Carlo Methods 2004, Springer-Verlag, Berlin, pp 331–342

[16] L'Ecuyer P, Demers V, Tuffin B (2007) Rare-events, splitting, and quasi-Monte Carlo. ACM Transactions on Modeling and Computer Simulation 17(2):Article 9, 45 pages

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

[18] L'Ecuyer P, Blanchet JH, Tuffin B, Glynn PW (2010) Asymptotic robustness of estimators in rare-event simulation. ACM Transactions on Modeling and Computer Simulation 20(1):Article 6

[19] L'Ecuyer P, Munger D, Lécot C, Tuffin B (2018) Sorting methods and convergence rates for Array-RQMC: Some empirical comparisons. Mathematics and Computers in Simulation 143:191–201

[20] L'Ecuyer P, Marion P, Godin M, Fuchhammer F (2020) A tool for custom construction of QMC and RQMC point sets. In: Monte Carlo and Quasi-Monte Carlo Methods: MCQMC 2020, https://arxiv.org/abs/2012.10263

[21] L'Ecuyer P, Puchhammer F, Ben Abdellah A (2021) Monte Carlo and quasi-Monte Carlo density estimation via conditioning. INFORMS Journal on Computing To appear. See also http://arxiv.org/abs/1906.04607

[22] Matoušek J (1998) On the L_2-discrepancy for anchored boxes. J of Complexity 14:527–556

[23] Nakayama MK, Tuffin B (2019) Efficient estimation of the mean hitting time to a set of a regenerative system. In: Mustafee N, Bae KH, Lazarova-Molnar S, Rabe M, Szabo C, Haas P, Son YJ (eds) Proceedings of the 2019 Winter Simulation Conference, Institute of Electrical and Electronics Engineers, Piscataway, New Jersey, pp 416–427

[24] Nakayama MK, Tuffin B (2021) Sufficient conditions for central limit theorems and confidence intervals for randomized quasi-Monte Carlo methods. techreport hal-03196085, INRIA, https://hal.inria.fr/hal-03196085

[25] Niederreiter H (1992) Random Number Generation and Quasi-Monte Carlo Methods, vol 63. SIAM, Philadelphia

[26] Owen AB (1998) Latin supercube sampling for very high-dimensional simulations. ACM Transactions on Modeling and Computer Simulation 8(1):71–102

[27] Rubino G, Tuffin B (eds) (2009) Rare Event Simulation using Monte Carlo Methods. John Wiley, Chichester, UK

[28] Sadowsky JS (1991) Large deviations theory and efficient simulation of excessive backlogs in a GI/GI/m queue. IEEE Transactions on Automatic Control 36:1383–1394

[29] Shahabuddin P (1994) Importance sampling for highly reliable Markovian systems. Management Science 40(3):333–352
[30] Tuffin B (1996) On the use of low discrepancy sequences in Monte Carlo methods. Monte Carlo Methods and Applications 2(4):295–320
[31] Tuffin B (1998) Variance reduction order using good lattice points in Monte Carlo methods. Computing 61(4):371–378