

Sudoku Latin Square Sampling for Markov Chain Simulation

Rami El Haddad, Joseph El Maalouf, Christian Lécot, and Pierre L'Ecuyer

Abstract We are interested in Monte Carlo simulations of discrete-time Markov chains on discrete and totally ordered state spaces. To improve simulation efficiency, we use a technique previously introduced in the context of quasi-Monte Carlo simulation of an array of N Markov chains. This method simulates the N copies of the chain simultaneously, reorders the chains at each step by increasing order of their states, and samples the next state by using N two-dimensional points in the unit square. The first coordinate of each point is used to match a chain, and the second coordinate is used to sample the next state by inversion from its cumulative distribution function conditional on the current state. We study the case where the N points are obtained at each step from Sudoku Latin square sampling, which means that (1) if the unit square is uniformly divided into N identical subsquares, exactly one point lies in each subsquare, (2) for each axis, the N projections of the points are distributed with exactly one projection in each of the N subintervals of length $1/N$ that partition the unit interval, and (3) in both cases, each individual point has the uniform distribution in the subsquare and interval to which it belongs. We prove that the variance of the Sudoku Latin square sampling estimator is of order $\mathcal{O}(N^{-3/2})$. The same convergence rate is obtained when property (2) is removed, which gives simple stratified sampling. However, in our numerical experiments, we observe empirically a much smaller variance and better efficiency with the Sudoku Latin square sampling than with simple stratified sampling alone.

Rami El Haddad · Joseph El Maalouf
Laboratoire de Mathématiques et Applications, U.R. Mathématiques et modélisation, Faculté des sciences, Université Saint-Joseph, B.P. 7-5208, Mar Mikhaël Beyrouth 1104 2020, Liban, e-mail: rami.haddad@usj.edu.lb, e-mail: joseph.maalouf@usj.edu.lb

Christian Lécot
Université Grenoble Alpes, Université Savoie Mont Blanc, CNRS, LAMA, 73000 Chambéry, France, e-mail: christian.lecot@univ-smb.fr

Pierre L'Ecuyer
DIRO, Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal, H3C 3J7, Canada, e-mail: lecuyer@iro.umontreal.ca

1 Introduction

We consider a discrete-time Markov chain $\{X_n, n \geq 0\}$ over a countable state space \mathcal{X} , where $X_n \in \mathcal{X}$ is the state at step n , and we are interested in estimating by simulation the expected cost at step n , $\mathbb{E}[c(X_n)]$, for one or several cost functions $c : \mathcal{X} \rightarrow [0, \infty)$ (we assume non-negative cost functions for simplicity). If the state space is finite with small cardinality and sometimes when the chain has a very special structure, it is possible to compute the exact distribution of X_n and the exact expected cost at step n , for any n . Otherwise, one can use standard Monte Carlo (MC): simulate the chain until step n , repeat N times independently, and average the N realizations of $c(X_n)$. The main drawback of this general approach is its slow convergence: The variance of the Monte Carlo estimator of $\mathbb{E}[c(X_n)]$ typically converges as $\mathcal{O}(N^{-1})$ for any n .

A (deterministic) *quasi-Monte Carlo* (QMC) method for Markov chains has been proposed in [9] for the case where the chain has a totally ordered state space. The method simulates an array of N copies of the chain in parallel. At each step n , it reorders the chains by increasing order of their states, and it uses two-dimensional quasi-random points to move them ahead by one step. Convergence (in the deterministic sense) of the average cost to the expectation when $N \rightarrow \infty$ was established, and the QMC approach outperformed plain MC in numerical experiments. However, QMC error bounds are typically too loose and inconvenient for practical error assessment. A *randomized quasi-Monte Carlo* (RQMC) approach named *Array-RQMC*, which resembles the previous QMC scheme, was proposed and analyzed in [11, 12], in the setting of a Markov chain model with general state space. The method was shown to provide an unbiased estimator of $\mathbb{E}[c(X_n)]$ for any n , and variance bounds for this estimator were proved under certain conditions. In particular, it was proved that for a Markov chain with a one-dimensional state space, if stratified sampling as in [2, 8] is used at each step to advance the array of chains by one step, and under some technical conditions, the variance converges as $\mathcal{O}(N^{-3/2})$, which beats Monte Carlo. In numerical experiments with Markov chains having one-dimensional and higher-dimensional states, the empirical variance was typically much smaller than the Monte Carlo variance, and was observed to decrease often at better rates than $\mathcal{O}(N^{-3/2})$, sometimes even faster than $\mathcal{O}(N^{-2})$: see [12, 13, 14], for example. However, no proof of these faster rates is available so far, and the $\mathcal{O}(N^{-3/2})$ rate has been proved only for ordinary stratification of the unit square in identical subsquares, for a one-dimensional state. A related convergence-rate result worth mentioning was obtained in [7], in the context of particle filters. The authors proved that if the RQMC point set used at each step is a (t, m, s) -net with a nested uniform scramble [18] and if the states are sorted using a Hilbert-curve when their dimension is larger than 1, then the variance of the Array-RQMC estimator converges as $o(N^{-1})$, which is faster than Monte Carlo.

The aim of this paper is to increase our theoretical understanding of the method by expanding the class of sampling methods for which an $\mathcal{O}(N^{-3/2})$ convergence rate is proved. We revisit the simple stratified sampling (SSS) setting and we consider a *Sudoku Latin square sampling* (SLSS) setting, which combines two-dimensional

stratified sampling with Latin hypercube sampling [15, 20]. Our theoretical results are consolidated by three numerical experiments in which we observe a significantly lower variance with SLSS than with simple stratification.

SLSS turns out to be a special case of the *U-sampling* method of [21] for sampling N points in the unit hypercube. The U-sampling first generates a random *orthogonal array-based Latin hypercube design*, which is a selection of N small cubic boxes of side size $1/N$ that form an orthogonal array of strength t [16, 17] and a Latin hypercube at the same time. Then it samples one point uniformly inside each selected small box, independently across the boxes. For the special case where $t = 2$, this type of design (the selection of the boxes) gives a *Sudoku Latin square* [19] for each two-dimensional projection of the points. Thus, each two-dimensional projection satisfies the properties (1) to (3) mentioned in the abstract. An example of a Sudoku Latin square is given in Fig. 1. Since our SLSS is in two dimensions, there is a single two-dimensional projection and it must form a Sudoku Latin square. Sudoku Latin squares are also studied in [22], although these authors are only considering discrete designs and space filling constructions, and not in sampling random points uniformly in the unit hypercube.

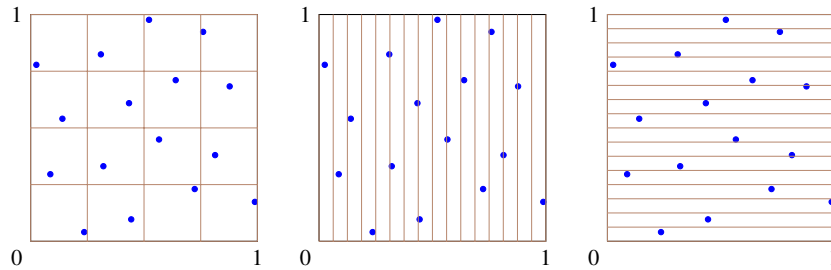


Fig. 1 Example of a Sudoku Latin square with 16 points.

A different sampling method that generalizes the SLSS to more than two dimensions was studied in [5]. In d dimensions, that method generates $N = p^d$ points in a way that (1) there is always one point per subcube when we partition the d -dimensional unit cube into N identical subcubes, (2) there is one value in each subinterval when we project all the points over a single coordinate to obtain N values in the unit interval, and we partition the unit interval into N subintervals of length $1/N$, and (3) each point taken individually has the uniform distribution in the subcube to which it belongs. Variance bounds have been obtained when the integral of a function over the unit cube is estimated by the average of the function values at the N points, under certain assumptions on the integrand. Different types of variance results, in terms of the ANOVA decomposition of the integrand, were proved in [21] for the same type of integration problem with U-sampling. SLSS is the two-dimensional special case of each of these two methods.

Our paper is the first to study the use of SLSS in the context of simulating an array of Markov chains. SLSS gives stronger constructions than simple stratified points over the unit square. Our aim is to investigate if, and how much, this strengthening has an impact on the variance of expected cost estimators.

The remainder is organized as follows. In Sect. 2, we define our setting for Monte Carlo simulation of discrete-state Markov chains and explain how we proceed with plain (standard) Monte Carlo, with SSS, and with SLSS. With SSS, the way we map the points to chain states at each step follows [6, 9] and differs from what was done in [12]. In Sect. 3, we analyze the variance of these schemes. We prove that the variance of the simulation estimator of an expected state-dependent cost at any given step n is $\mathcal{O}(N^{-3/2})$ for both SSS and SLSS. This beats the known rate of $\mathcal{O}(N^{-1})$ for standard Monte Carlo. Results of computational experiments and comparison between standard Monte Carlo, SSS, and SLSS are given in Sect. 4. The empirical convergence rates of the variance for SSS and SLSS are close to those established in Sect. 3, but the variance with SLSS is significantly smaller than with SSS. In Sect. 5, we give the technical proofs of some results and we conclude in Sect. 6.

2 Monte Carlo Simulations of Markov Chains

Let $\{X_n, n \geq 0\}$ be a stationary discrete-time Markov chain over a countable and ordered state space \mathcal{X} . Without loss of generality one can assume that $\mathcal{X} = \mathbb{N}$ or \mathbb{Z} . Let $P(i, j) = \mathbb{P}(X_{n+1} = j | X_n = i)$ denote the transition probabilities and $\mathbf{P} = (P(i, j) : (i, j) \in \mathcal{X}^2)$ the transition probability matrix. We denote by $\mu_n(i) = \mathbb{P}(X_n = i)$ the state probabilities at step n and $\boldsymbol{\mu}_n = (\mu_n(i) : i \in \mathcal{X})$ the probability vector for step n . We assume that the initial probability vector $\boldsymbol{\mu}_0$ is given (often, it is degenerate over a single state).

For $i, j \in \mathcal{X}$ we set

$$q_j(i) := \sum_{h \leq j} P(i, h). \quad (1)$$

We define the conditional cumulative distribution function $F_i(j) := \mathbb{P}(X_{n+1} \leq j | X_n = i) = q_j(i)$. If I denotes the unit interval $(0, 1]$ we have a disjoint union $I = \bigcup_{j \in \mathcal{X}} I_{i,j}$, where $I_{i,j} := (q_{j-1}(i), q_j(i)]$. So that for any $i \in \mathcal{X}$ and $u \in I$, there exists a unique $j \in \mathcal{X}$ such that $u \in I_{i,j}$: we denote it by $F_i^{-1}(u)$. If $X_n = i$, then $F_i^{-1}(u)$ is the next state.

Let δ_i be the Dirac measure at i , defined by

$$\delta_i(j) = \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{otherwise.} \end{cases}$$

For any integer n , the distribution $\boldsymbol{\mu}_n$ is approximated by

$$\widehat{\mu}_n := \frac{1}{N} \sum_{k=1}^N \delta_{i_k^n},$$

where N is a fixed integer and i_1^n, \dots, i_N^n are calculated iteratively.

First, a set $(i_k^0 : 1 \leq k \leq N)$ of N states is sampled from μ_0 : several techniques are proposed in [4]. In many applications, the initial state of the chain is fixed, and then $\widehat{\mu}_0 = \mu_0$.

We describe the transition from step n to $n+1$ for three Monte Carlo methods. We introduce $\widetilde{\mu}_{n+1} := \widehat{\mu}_n \mathbf{P}$ as an intermediate distribution (which is not used in effective calculations). This $\widetilde{\mu}_{n+1}$ is an approximation of μ_{n+1} , but it is generally not an equally-weighted sum of Dirac measures, like $\widehat{\mu}_n$, so that an additional step is needed. We formulate this step as a quadrature: the MC methods correspond to quadrature algorithms, possibly combined with variance reduction techniques. To that end, let us consider an arbitrary sequence $s = (s(i) : i \in \mathcal{X})$ (a column vector); we assume that s is non-negative, just to avoid worrying about convergence of series. Then

$$\widetilde{\mu}_{n+1} s = \widehat{\mu}_n \mathbf{P} s = \frac{1}{N} \sum_{k=1}^N \sum_{j \in \mathcal{X}} P(i_k^n, j) s(j).$$

Let 1_k be the indicator function of the interval $I_k := ((k-1)/N, k/N]$ and $1_{i,j}$ denote the indicator function of the interval $I_{i,j}$. If we associate to s the function C_s^n defined by

$$C_s^n(u) := \sum_{k=1}^N \sum_{j \in \mathcal{X}} 1_k(u_1) 1_{i_k^n, j}(u_2) s(j), \quad u = (u_1, u_2) \in I^2, \quad (2)$$

then we have

$$\widetilde{\mu}_{n+1} s = \int_{I^2} C_s^n(u) du.$$

We obtain $\widehat{\mu}_{n+1}$ by approximating the integral with Monte Carlo estimation. In the following, if m is an integer, we denote $[1, m] := \{1, 2, \dots, m\}$. The notation $U \sim \mathcal{U}(\mathcal{E})$ means that U is a random variable uniformly distributed over the set \mathcal{E} .

2.1 Standard Monte Carlo

The transition from step n to step $n+1$ acts as follows: if the state of the chain is i , i.e. $X_n = i$, then a random number U with $U \sim \mathcal{U}(I)$ is generated and the new state of the chain is $F_i^{-1}(U)$, i.e. $X_{n+1} = F_{X_n}^{-1}(U)$. The operation is repeated N times independently, in order to advance N copies of the chain. With our notations, this may be written as follows. Let $\{U_k : 1 \leq k \leq N\}$ be independent random variables with $U_k \sim \mathcal{U}(I)$, then

$$i_k^{n+1} = F_{i_k^n}^{-1}(U_k), \quad 1 \leq k \leq N.$$

That is, if, for any non-negative sequence s ,

$$\widehat{X}_s^{n+1} := \frac{1}{N} \sum_{k=1}^N C_s^n \left(\frac{k-1}{N}, U_k \right), \quad (3)$$

then

$$\widehat{\mu}_{n+1}s = \widehat{X}_s^{n+1}. \quad (4)$$

2.2 Simple Stratified Sampling

We suppose that $N = p^2$, for some integer $p > 0$. The transition from n to $n+1$ has two steps: renumbering of the states and numerical integration.

- (S1) The states are relabeled so that $i_1^n \leq \dots \leq i_N^n$. The technique was used in the QMC context and ensures theoretical and numerical convergence of the scheme (see [9]).
- (S2) Consider a partition of I^2 into N squares: $I_\ell = H_{\ell_1} \times H'_{\ell_2}$, where, for $\ell \in [1, p]^2$: $H_{\ell_1} := ((\ell_1 - 1)/p, \ell_1/p]$ and $H'_{\ell_2} := ((\ell_2 - 1)/p, \ell_2/p]$. Let $\{V_\ell : \ell \in [1, p]^2\}$ be independent random variables, where $V_\ell = (V_{\ell,1}, V_{\ell,2}) \sim \mathcal{U}(I_\ell)$.

For an arbitrary non-negative sequence s , let

$$\widehat{Y}_s^{n+1} := \frac{1}{N} \sum_{\ell \in [1, p]^2} C_s^n(V_\ell), \quad (5)$$

then

$$\widehat{\mu}_{n+1}s = \widehat{Y}_s^{n+1}. \quad (6)$$

If $u \in I$, let

$$\kappa(u) := \lceil Nu \rceil, \quad (7)$$

where $\lceil x \rceil$ is the least integer greater than or equal to x . Hence equation (6) means that the next states are calculated as follows:

$$i_{(\ell_1-1)p+\ell_2}^{n+1} = F_{i_{\kappa(V_{\ell,1})}^n}^{-1}(V_{\ell,2}), \quad \ell \in [1, p]^2$$

(the numbering of the states i_k^{n+1} is arbitrary). The first projection $V_{\ell,1}$ of V_ℓ is used for selecting the state at step n and the second projection $V_{\ell,2}$ is used for performing the transition to step $n+1$. Note that with this scheme, the mapping between the N points and the N states is not necessarily one-to-one: it is possible to pick the same state more than once and leave out some of the states. This differs from the SSS scheme used in [12, 14].

2.3 Sudoku Latin Square Sampling

As before, we assume $N = p^2$, and the transition from n to $n + 1$ has two steps: renumbering of the states and numerical integration.

- (S1) The states are relabeled so that $i_1^n \leq \dots \leq i_N^n$.
(S2) We consider the same partition of I^2 as before: I_ℓ for $\ell \in [1, p]^2$. Let $\{W_\ell : \ell \in [1, p]^2\}$ be random variables, where $W_\ell = (W_{\ell,1}, W_{\ell,2})$, with

$$W_{\ell,1} = \frac{\ell_1 - 1}{p} + \frac{\sigma_1(\ell_2) - 1 + \xi_\ell^1}{p^2} \quad W_{\ell,2} = \frac{\ell_2 - 1}{p} + \frac{\sigma_2(\ell_1) - 1 + \xi_\ell^2}{p^2}.$$

Here σ_1 and σ_2 are random permutations of $[1, p]$ and $\xi_\ell^1 \sim \mathcal{U}(I)$ and $\xi_\ell^2 \sim \mathcal{U}(I)$. All these random variables being independent. The set of values of the random variable W_ℓ is included in I_ℓ and has the properties:

- (P1) for any $\ell \in [1, p]^2$, there is a unique point of this set in each square I_ℓ ,
(P2) for any $k \in [1, N]$, there is a unique point of this set in each rectangle $I \times I_k$ or $I_k \times I$.

In addition $W_\ell \sim \mathcal{U}(I_\ell)$. For an arbitrary non-negative sequence s , let

$$\widehat{Z}_s^{n+1} := \frac{1}{N} \sum_{\ell \in [1, p]^2} C_s^n(W_\ell). \quad (8)$$

Then

$$\widehat{\mu}_{n+1} s = \widehat{Z}_s^{n+1}. \quad (9)$$

Due to property (P2), the mapping (see (7))

$$\ell := (\ell_1, \ell_2) \in [1, p]^2 \rightarrow \kappa(W_{\ell,1}) \in [1, N]$$

is one-to-one: each state of step n is considered exactly once for a transition (this is not the case with SSS). Equation (9) means that the next states are calculated as follows:

$$i_{(\ell_1-1)p+\ell_2}^{n+1} = F_{i_{\kappa(W_{\ell,1})}^n}^{-1}(W_{\ell,2}), \quad \ell \in [1, p]^2$$

(as before, the numbering of the states i_k^{n+1} is arbitrary). The first projection $W_{\ell,1}$ of W_ℓ is used for selecting the state at step n and the second projection $W_{\ell,2}$ is used for performing the transition to step $n + 1$.

3 Convergence Analysis

In this section we prove, for each method, that the estimator of the expected cost at each step is unbiased and we establish that the variance of the estimator used is

$\mathcal{O}(N^{-1})$ for standard MC and $\mathcal{O}(N^{-3/2})$ for SSS and SLSS, where N is the number of simulation paths. In the following, λ is the Lebesgue measure and λ_2 the two-dimensional Lebesgue measure; we put $|\mathcal{E}|$ for the number of elements of a set \mathcal{E} . We use the sequence s_h , for $h \in \mathcal{X}$:

$$s_h(i) := \begin{cases} 1 & \text{if } i \leq h, \\ 0 & \text{otherwise.} \end{cases}$$

The total variation of a sequence $s = (s(i) : i \in \mathcal{X})$ is defined by

$$TV(s) := \sum_{i \in \mathcal{X}} |s(i+1) - s(i)|. \quad (10)$$

We use below the total variation of q_h , for $h \in \mathcal{X}$. We recall that we have from (1):

$$q_h(i) = \mathbb{P}(X_{n+1} \leq h | X_n = i),$$

and from (10):

$$TV(q_h) = \sum_{i \in \mathcal{X}} |\mathbb{P}(X_{n+1} \leq h | X_n = i+1) - \mathbb{P}(X_{n+1} \leq h | X_n = i)|.$$

In the following, we assume that

$$M := \sup_{h \in \mathcal{X}} TV(q_h) < +\infty.$$

There are situations for which q_h is monotone and situations for which $M < 1$ (or both), but this is not always true. See [13, 14] for examples and further discussion. Our M corresponds to Λ_j in [12].

3.1 Standard Monte Carlo

Lemma 1. *Let s be a non-negative sequence. The standard Monte Carlo estimator of $\tilde{\mu}_{n+1}^s$:*

$$\widehat{X}_s^{n+1} := \frac{1}{N} \sum_{k=1}^N C_s^n \left(\frac{k-1}{N}, U_k \right)$$

has the following properties.

1. \widehat{X}_s^{n+1} is unbiased.
2. If $s = s_h$, for $h \in \mathcal{X}$, then

$$\text{Var}(\widehat{X}_{s_h}^{n+1}) \leq \frac{1}{4N}.$$

Proof.

1. We have

$$\mathbb{E} \left[C_s^n \left(\frac{k-1}{N}, U_k \right) \right] = \sum_{j \in \mathcal{X}} P(i_k^n, j) s(j),$$

so that $\mathbb{E}[\widehat{X}_s^{n+1}] = \widetilde{\mu}_{n+1}s$.

2. The variable

$$C_{s_h}^n \left(\frac{k-1}{N}, U_k \right) = \sum_{j \in \mathcal{X}, j \leq h} 1_{i_k^n, j}(U_k)$$

is a Bernoulli random variable, with variance $\leq 1/4$. Hence the result. \square

We then obtain an error bound by using the same techniques as in [12]. We assume that, for any non-negative sequence s , the standard Monte Carlo estimator $\widehat{\mu}_0s$ of μ_0s is unbiased and that, for any $h \in \mathcal{X}$,

$$\text{Var}(\widehat{\mu}_0s_h) \leq \frac{x_0}{N},$$

for some $x_0 \geq 0$ (as noticed before, in many applications, $\widehat{\mu}_0 = \mu_0$).

Proposition 1. *For the standard Monte Carlo method, it holds:*

1. *for any non-negative sequence s*

$$\mathbb{E}[\widehat{\mu}_n s] = \mu_n s,$$

2. *for any $h \in \mathcal{X}$,*

$$\text{Var}(\widehat{\mu}_n s_h) \leq \frac{x_n}{N},$$

where $x_{n+1} = M^2 x_n + 1/4$ ($n \geq 0$).

Proof.

1. We have

$$\mu_{n+1}s - \widehat{\mu}_{n+1}s = \mu_{n+1}s - \widetilde{\mu}_{n+1}s + \widetilde{\mu}_{n+1}s - \widehat{\mu}_{n+1}s = \mu_n \mathbf{P}s - \widehat{\mu}_n \mathbf{P}s + \widetilde{\mu}_{n+1}s - \widehat{X}_s^{n+1},$$

so, by using Lemma 1, the result follows by induction.

2. The variables $\mu_{n+1}s_h - \widetilde{\mu}_{n+1}s_h$ and $\widetilde{\mu}_{n+1}s_h - \widehat{\mu}_{n+1}s_h$ are uncorrelated and $\widetilde{\mu}_{n+1}s_h - \widehat{\mu}_{n+1}s_h = \widetilde{\mu}_{n+1}s_h - \widehat{X}_s^{n+1}$ is a centered variable, consequently

$$\text{Var}(\widehat{\mu}_{n+1}s_h) = \mathbb{E}[(\mu_{n+1}s_h - \widetilde{\mu}_{n+1}s_h)^2] + \mathbb{E}[(\widetilde{\mu}_{n+1}s_h - \widehat{\mu}_{n+1}s_h)^2]. \quad (11)$$

For any $i \in \mathcal{X}$, we have $\mathbf{P}s_h(i) = F_i(h)$, hence

$$\begin{aligned} \mu_{n+1}s_h - \widetilde{\mu}_{n+1}s_h &= \mu_n \mathbf{P}s_h - \widehat{\mu}_n \mathbf{P}s_h = \sum_{i \in \mathcal{X}} \mu_n(i) F_i(h) - \sum_{i \in \mathcal{X}} \widehat{\mu}_n(i) F_i(h) \\ &= - \sum_{i \in \mathcal{X}} \mu_n s_i (F_{i+1}(h) - F_i(h)) + \sum_{i \in \mathcal{X}} \widehat{\mu}_n s_i (F_{i+1}(h) - F_i(h)) \\ &= \sum_{i \in \mathcal{X}} (\widehat{\mu}_n s_i - \mu_n s_i) (F_{i+1}(h) - F_i(h)). \end{aligned}$$

On the one hand, we write

$$\begin{aligned} \mathbb{E} [(\mu_{n+1s_h} - \tilde{\mu}_{n+1s_h})^2] &= \mathbb{E} \left[\left(\sum_{i \in \mathcal{X}} (\hat{\mu}_n s_i - \mu_n s_i) (q_h(i+1) - q_h(i)) \right)^2 \right] \\ &= \sum_{(i,j) \in \mathcal{X}^2} \mathbb{E} [(\hat{\mu}_n s_i - \mu_n s_i) (q_h(i+1) - q_h(i)) (\hat{\mu}_n s_j - \mu_n s_j) (q_h(j+1) - q_h(j))] \\ &\leq \sum_{(i,j) \in \mathcal{X}^2} \sqrt{\text{Var}(\hat{\mu}_n s_i) \text{Var}(\hat{\mu}_n s_j)} |q_h(i+1) - q_h(i)| \times |q_h(j+1) - q_h(j)|. \end{aligned}$$

On the other hand, Lemma 1 gives

$$\mathbb{E} [(\tilde{\mu}_{n+1s_h} - \hat{\mu}_{n+1s_h})^2] = \mathbb{E} [(\tilde{\mu}_{n+1s_h} - \hat{X}_{s_h}^{n+1})^2] \leq \frac{1}{4N}.$$

So, by using (11), the result follows by induction. \square

The bounds for $\text{Var}(\hat{Y}_{s_h}^{n+1})$ (SSS) and $\text{Var}(\hat{Z}_{s_h}^{n+1})$ (SLSS) are not so easily obtained, and the proofs of Lemma 2 and 3 are given in Sect. 5.

3.2 Simple Stratified Sampling

Lemma 2. *Let s be a non-negative sequence. The SSS estimator of $\tilde{\mu}_{n+1s}$:*

$$\hat{Y}_s^{n+1} := \frac{1}{N} \sum_{\ell \in [1,p]^2} C_s^n(V_\ell)$$

has the following properties.

1. \hat{Y}_s^{n+1} is unbiased.
2. If $s = s_h$, for $h \in \mathcal{X}$, then

$$\text{Var}(\hat{Y}_{s_h}^{n+1}) \leq \frac{1}{4N^{3/2}} (TV(q_h) + 2).$$

A similar result (with the same $N^{-3/2}$ order) was established in [12], but the SSS method studied there differs from the one used here, so we provide a different proof (see Sect. 5). Intermediate results from this proof (eqs. (12) and (13)) will be re-used afterwards for the analysis of SLSS.

The proof of the next result is similar to the proof of Proposition 1. We assume that, for any non-negative sequence s , the SSS estimator $\hat{\mu}_{0s}$ of μ_{0s} is unbiased and that, for any $h \in \mathcal{X}$,

$$\text{Var}(\hat{\mu}_{0s_h}) \leq \frac{y_0}{N^{3/2}}$$

for some $y_0 \geq 0$.

Proposition 2. *For the SSS method, it holds:*

1. *for any non-negative sequence s*

$$\mathbb{E}[\widehat{\mu}_n s] = \mu_n s,$$

2. *for any $h \in \mathcal{X}$,*

$$\text{Var}(\widehat{\mu}_n s_h) \leq \frac{y_n}{N^{3/2}},$$

where $y_{n+1} = M^2 y_n + (M+2)/4$ ($n \geq 0$).

3.3 Sudoku Latin Square Sampling

Lemma 3. *Let s be a non-negative sequence. The SLSS estimator of $\widetilde{\mu}_{n+1} s$:*

$$\widehat{Z}_s^{n+1} := \frac{1}{N} \sum_{\ell \in [1,p]^2} C_s^n(W_\ell)$$

has the following properties.

1. *\widehat{Z}_s^{n+1} is unbiased.*

2. *If $s = s_h$, for $h \in \mathcal{X}$, and if q_h is a piecewise monotonic sequence, with r pieces, then*

$$\text{Var}(\widehat{Z}_{s_h}^{n+1}) \leq \frac{1}{N^{3/2}} \left(\left(\frac{13}{4} + r \right) (TV(q_h) + 2) + 2(TV(q_h) + 2)^2 \right).$$

The constant involved in the $\mathcal{O}(N^{-3/2})$ bound of $\text{Var}(\widehat{Z}_{s_h}^{n+1})$ (SLSS) is larger than the corresponding constant for $\text{Var}(\widehat{Y}_{s_h}^{n+1})$ (SSS), since $r \geq 1$ in Lemma 3; this would suggest degraded performance. But in the examples of Sect. 4 we see that it is not necessarily the case.

The proof of the next result is similar to the proof of Proposition 1. We assume that, for any non-negative sequence s , the SLSS estimator $\widehat{\mu}_0 s$ of $\mu_0 s$ is unbiased and that, for any $h \in \mathcal{X}$,

$$\text{Var}(\widehat{\mu}_0 s_h) \leq \frac{z_0}{N^{3/2}},$$

for some $z_0 \geq 0$.

Proposition 3. *For the SLSS method, it holds:*

1. *for any non-negative sequence s*

$$\mathbb{E}[\widehat{\mu}_n s] = \mu_n s,$$

2. *for any $h \in \mathcal{X}$,*

$$\text{Var}(\widehat{\mu}_n s_h) \leq \frac{z_n}{N^{3/2}},$$

where $z_{n+1} = M^2 z_n + (13/4 + r)(M + 2) + 2(M + 2)^2 \quad n \geq 0$.

Remark 1. The bounds in Propositions 1, 2, and 3 increase exponentially with n when $M > 1$, and remains bounded if $M < 1$, which is not uncommon; see [14].

Remark 2. The variance of each estimator is bounded for a test sequence of the form s_h , for $h \in \mathcal{X}$. We obtain a bound for a nonnegative cost function c by the same reasoning as in Proposition 1 (see [12]):

$$\mu_n c - \hat{\mu}_n c = \sum_{i \in \mathcal{X}} (\hat{\mu}_n s_i - \mu_n s_i)(c(i+1) - c(i)),$$

hence

$$\mathbb{E} [(\mu_n c - \hat{\mu}_n c)^2] \leq \sum_{(i,j) \in \mathcal{X}^2} \sqrt{\text{Var}(\hat{\mu}_n s_i) \text{Var}(\hat{\mu}_n s_j)} |c(i+1) - c(i)| \times |c(j+1) - c(j)|,$$

and then

$$\text{Var}(\hat{\mu}_n c) \leq TV(c)^2 \times \sup_{h \in \mathcal{X}} \text{Var}(\hat{\mu}_n s_h).$$

4 Numerical Examples

In this section, we compare standard Monte Carlo to the variance reduction strategies analyzed previously: SSS and SLSS, for three examples. For each example, each strategy, and each N considered, we compute the unbiased estimator $\hat{\mu}_n s$ of $\mu_n s$ for the selected n , replicate this 100 times, and compute the empirical variance Var of the 100 realizations of $\hat{\mu}_n s$. We then plot $\log_{10} \text{Var}$ as a function of $\log_{10} N$. Assuming that $\text{Var} \approx KN^{-\alpha}$ for some positive constants K and α , we estimate the variance rate α by linear regression. We also compute the (empirical) *efficiency* of each simulation estimator, defined as the inverse of the product of Var by the CPU time [10], and we plot \log_{10} efficiency as a function of $\log_{10} N$. Note that for standard Monte Carlo, the efficiency does not depend on N . For SSS and SLSS, it takes into account the additional work to compute the estimators.

4.1 A Geo/Geo/1 Queue

We consider a discrete-time Geo/Geo/1 queue (see [1]): the queue is empty at the initial time. During each unit of time, the customer in service (if there is one) completes it with probability 0.5, and one new customer arrives with probability 0.6. We estimate the mean number of customers in the queue at time $n = 12$. Figure 2 (top) shows $\log_{10} \text{Var}$ as a function of $\log_{10} N$ on the left and \log_{10} efficiency as a function of $\log_{10} N$ on the right, for $N = 10^2, 50^2, 100^2, \dots, 1000^2$. We find from the

plots that SSS and SLSS give not only smaller variances than standard MC (for the same N), but also better efficiencies, and that SLSS outperforms SSS. The regression estimates of α are given in the first row of Table 1. They match the upper bounds of $\mathcal{O}(N^{-3/2})$ established in Sect. 3.

4.2 A Gambler in a Casino

A gambler is going to a casino for four hours. He plans to play the same game every ten seconds (so he will play 1 440 times). At this game, for each Euro that he bids, he gets 0 with probability 0.9 and $m \in \{1, 2, \dots, 10\}$ with probability 0.01 each. His policy is the following: if he has more than 100 Euros, he plays 2 Euros, but if he has 100 Euros or less, he plays only 1. To make sure that he can play during the four hours, he brings 2 780 Euros with him. The model is a Markov chain on state space $E = [0, 28\,700]$. We estimate the probability that the gambler has more than 1 500 Euros at the end of the game. Here we use $N = 10^2, 20^2, 30^2, \dots, 200^2$. The results are reported in the middle rows of Fig. 2 and Table 1. We find that SSS and SLSS produce both smaller variances and better efficiencies than standard MC for large enough N , and that SLSS outperforms SSS. The regression estimate of α for SLSS corresponds to the bound established in Sect. 3, but for SSS it is better. However, the variance itself is smaller for SLSS than for SSS, and the better rate of SSS might not hold beyond the observed range (it is likely caused by a few poor values of Var for the smallest values of N).

4.3 Diffusion

The 1-D diffusion equation

$$\frac{\partial c}{\partial t}(x, t) = D \frac{\partial^2 c}{\partial x^2}(x, t), \quad x \in \mathbb{R}, t > 0 \quad \text{and} \quad c(x, 0) = c_0(x), \quad x \in \mathbb{R}$$

(where $c_0 \geq 0$ and $\int_{\mathbb{R}} c_0(x) dx = 1$) may be discretized with a time step Δt and a spatial step Δx and the solution is approximated using a random walk: $P(i, i-1) = P(i, i+1) = D\Delta t/\Delta x^2$, $P(i, i) = 1 - 2D\Delta t/\Delta x^2$ (we refer to [3] in a QMC context). Here we specify $D = 1$ and c_0 is the indicator function of the interval $[-1/2, 1/2]$; we want to approximate $\int_{-1/2}^{1/2} c(x, T) dx$. We choose $T = 1$, with $\Delta t = 6.25 \cdot 10^{-4}$ and $\Delta x = 5.0 \cdot 10^{-2}$. Here we take $N = 11^2, 19^2, 31^2, \dots, 199^2$. In a previous version of the paper, we had $N = 10^2, 20^2, 30^2, \dots, 200^2$, but this gave oscillations in SLSS outcomes, with better results when $p = \sqrt{N}$ was a multiple of 20, because of interactions with other discretization parameters. To avoid this, we changed our choices of p . So we use prime numbers near the previous ones for p . The bottom part of Figure 2 and the

last row of Table 1 give the results, which are very similar to those of the previous example. Again, SLSS outperforms the other methods.

Table 1 Calculation of order α of the sample variance: comparison of standard Monte Carlo (MC), SSS, and SLSS for three examples and estimators

	Calculation	MC	SSS	SLSS
Geo/Geo/1 queue	$E[X_{12}]$	0.99	1.50	1.51
Gambler in a casino	$\mathbb{P}(X_{1440} > 1500)$	1.02	1.76	1.53
Diffusion	$\int_{-1/2}^{1/2} c(x, T) dx$	0.98	1.61	1.49

5 The Proofs

5.1 Proof of Lemma 2

Proof.

1. We have

$$\mathbb{E}[C_s^n(V_\ell)] = N \sum_{k=1}^N \sum_{j \in \mathcal{X}} \left(\int_{I_\ell} 1_k(u_1) 1_{i_k^n, j}^n(u_2) du \right) s(j).$$

Hence

$$\mathbb{E}[\widehat{Y}_s^{n+1}] = \frac{1}{N} \sum_{k=1}^N \sum_{j \in \mathcal{X}} P(i_k^n, j) s(j) = \widetilde{\mu}_{n+1s}.$$

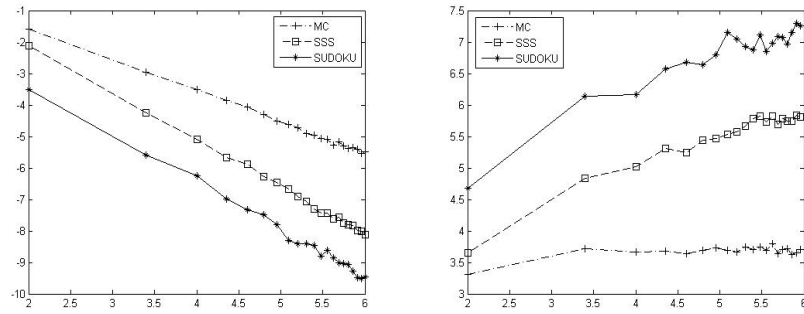
2. The function $C_{s_h}^n$ is the indicator function of the set

$$J_h^n := \bigcup_{k=1}^N \left(I_k \times \bigcup_{j \in \mathcal{X}, j \leq h} I_{i_k^n, j}^n \right) = \bigcup_{k=1}^N I_k \times (0, q_h(i_k^n)]. \quad (12)$$

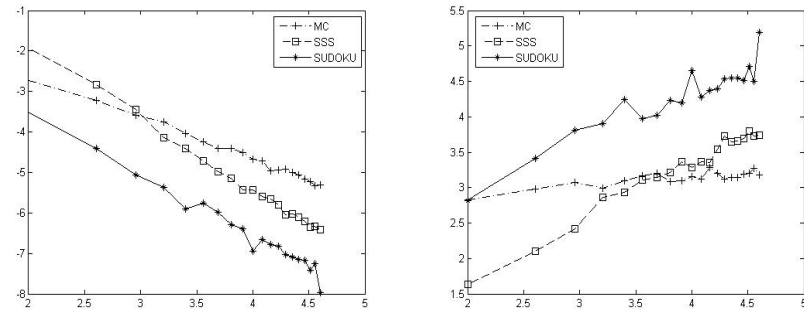
The variable $C_{s_h}^n(V_\ell)$ is a Bernoulli random variable, with expectation $f_{h, \ell}^n = N \lambda_2(J_h^n \cap I_\ell)$. Here $f_{h, \ell}^n = 1$ if $I_\ell \subset J_h^n$ and $f_{h, \ell}^n = 0$ if $I_\ell \cap J_h^n = \emptyset$. Consequently, $\text{Var}(C_{s_h}^n(V_\ell)) = f_{h, \ell}^n(1 - f_{h, \ell}^n) \leq 1/4$ and $\text{Var}(C_{s_h}^n(V_\ell)) = 0$ if $I_\ell \subset J_h^n$ or if $I_\ell \cap J_h^n = \emptyset$, so that

$$\text{Var}(\widehat{Y}_{s_h}^{n+1}) \leq \frac{1}{4N^2} |\{\ell \in [1, p]^2 : I_\ell \not\subset J_h^n \text{ and } I_\ell \cap J_h^n \neq \emptyset\}|.$$

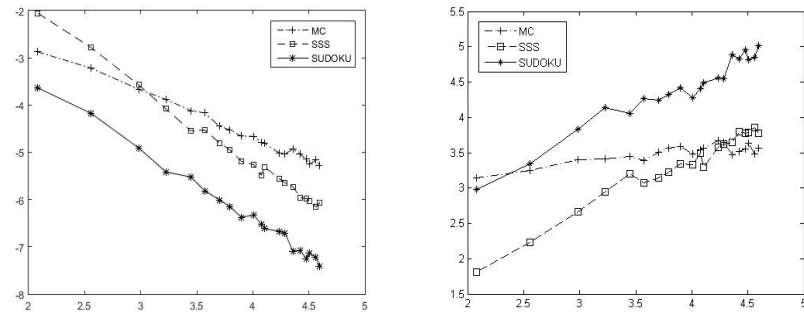
a. If $I_\ell \not\subset J_h^n$, then there exists (u_1, u_2) which belongs to I_ℓ and not to J_h^n ; since this u_1 is in some I_k , we have:



1. Geo/Geo/1 queue: sample variance (left) and efficiency (right) of 100 copies of the calculation of $E[X_{12}]$ as a function of N ($N = 10^2, 50^2, 100^2, \dots, 1000^2$) (log-log scale)



2. Gambler in a casino: sample variance (left) and efficiency (right) of 100 copies of the calculation of $\mathbb{P}(X_{1440} > 1500)$ as a function of N ($N = 10^2, 20^2, 30^2, \dots, 200^2$) (log-log scale)



3. Diffusion: sample variance (left) and efficiency (right) of 100 copies of the calculation of $\int_{-1/2}^{1/2} c(x, T) dx$ as a function of N ($N = 11^2, 19^2, 31^2, \dots, 199^2$) (log-log scale)

Fig. 2 Comparison of standard Monte Carlo (MC) to SSS and SLSS (Sudoku) for three examples

$$\exists k \in [1, N], \exists (u_1, u_2) \in I_\ell : u_1 \in I_k \subset \left(\frac{\ell_1 - 1}{p}, \frac{\ell_1}{p} \right] \text{ and } u_2 \notin (0, q_h(i_k^n)],$$

so that

$$\exists k \in \{p(\ell_1 - 1) + 1, p(\ell_1 - 1) + 2, \dots, p\ell_1\}, \exists u_2 \in \left(\frac{\ell_2 - 1}{p}, \frac{\ell_2}{p} \right] : u_2 > q_h(i_k^n),$$

consequently

$$\ell_2 > p \min_{p(\ell_1 - 1) < k \leq p\ell_1} q_h(i_k^n).$$

- b. Analogously, if $I_\ell \cap J_h^n \neq \emptyset$, then there exists (u_1, u_2) which belongs to I_ℓ and also to J_h^n ; and we eventually obtain:

$$\ell_2 < p \max_{p(\ell_1 - 1) < k \leq p\ell_1} q_h(i_k^n) + 1.$$

We then have the following bounds

$$\begin{aligned} & |\{\ell \in [1, p]^2 : I_\ell \not\subset J_h^n \text{ and } I_\ell \cap J_h^n \neq \emptyset\}| \\ & \leq p \left(\sum_{\ell_1=1}^p \left(\max_{p(\ell_1 - 1) < k \leq p\ell_1} q_h(i_k^n) - \min_{p(\ell_1 - 1) < k \leq p\ell_1} q_h(i_k^n) \right) + 2 \right) \\ & \leq N^{1/2} \left(\sum_{i \in \mathcal{X}} |q_h(i+1) - q_h(i)| + 2 \right), \end{aligned}$$

because the states are relabeled so that $i_1^n \leq \dots \leq i_N^n$. Consequently,

$$|\{\ell \in [1, p]^2 : I_\ell \not\subset J_h^n \text{ and } I_\ell \cap J_h^n \neq \emptyset\}| \leq N^{1/2} (TV(q_h) + 2), \quad (13)$$

and the result follows. \square

5.2 Proof of Lemma 3

Proof.

1. Since $W_\ell \sim \mathcal{U}(I_\ell)$, the demonstration is the same as in Lemma 2.
2. In the following, we have many summations with $\ell, \ell', m, m' \in [1, p]^2$. In order to lighten the notations, we omit this set. We have

$$\text{Var}(\widehat{Z}_{s_h}^{n+1}) = V_0(\widehat{Z}_{s_h}^{n+1}) + \frac{1}{N^2} \sum_{(\ell, \ell') : \ell \neq \ell'} \text{Cov}(C_{s_h}^n(W_\ell), C_{s_h}^n(W_{\ell'})),$$

where

$$V_0(\widehat{Z}_{s_h}^{n+1}) := \frac{1}{N^2} \sum_{\ell} \text{Var}(C_{s_h}^n(W_{\ell})).$$

- a. The function $C_{s_h}^n$ is the indicator function of the set J_h^n defined by (12). Since $W_{\ell} \sim \mathcal{U}(I_{\ell})$, we have, as in Lemma 2:

$$V_0(\widehat{Z}_{s_h}^{n+1}) \leq \frac{1}{4N^2} |\{\ell \in [1, p]^2; I_{\ell} \not\subseteq J_h^n \text{ and } I_{\ell} \cap J_h^n \neq \emptyset\}|.$$

From the bound (13), we deduce

$$V_0(\widehat{Z}_{s_h}^{n+1}) \leq \frac{1}{4N^{3/2}} (TV(q_h) + 2).$$

- b. We split $\text{Var}(\widehat{Z}_{s_h}^{n+1}) = V_0(\widehat{Z}_{s_h}^{n+1}) + V_1(\widehat{Z}_{s_h}^{n+1}) + V_2(\widehat{Z}_{s_h}^{n+1}) + V_3(\widehat{Z}_{s_h}^{n+1})$, with

$$V_1(\widehat{Z}_{s_h}^{n+1}) := \frac{1}{N^2} \sum_{(\ell, \ell'): \ell_1 \neq \ell'_1, \ell_2 = \ell'_2} \text{Cov}(C_{s_h}^n(W_{\ell}), C_{s_h}^n(W_{\ell'})),$$

$$V_2(\widehat{Z}_{s_h}^{n+1}) := \frac{1}{N^2} \sum_{(\ell, \ell'): \ell_1 = \ell'_1, \ell_2 \neq \ell'_2} \text{Cov}(C_{s_h}^n(W_{\ell}), C_{s_h}^n(W_{\ell'})),$$

$$V_3(\widehat{Z}_{s_h}^{n+1}) := \frac{1}{N^2} \sum_{(\ell, \ell'): \ell_1 \neq \ell'_1, \ell_2 \neq \ell'_2} \text{Cov}(C_{s_h}^n(W_{\ell}), C_{s_h}^n(W_{\ell'})).$$

We introduce the N^2 squares $I_{\ell, m} = H_{\ell_1, m_1} \times H_{\ell_2, m_2}$, where, for $(\ell, m) \in [1, p]^4$:

$$H_{\ell_1, m_1} := ((\ell_1 - 1)/p + (m_1 - 1)/N, (\ell_1 - 1)/p + m_1/N],$$

$$H_{\ell_2, m_2} := ((\ell_2 - 1)/p + (m_2 - 1)/N, (\ell_2 - 1)/p + m_2/N].$$

We have

$$\begin{aligned}
V_1(\widehat{Z}_{S_h}^{n+1}) &= \sum_{(\ell, \ell'): \ell_1 \neq \ell'_1, \ell_2 = \ell'_2} \left(\frac{N}{p-1} \sum_{(m, m'): m_1 = m'_1, m_2 \neq m'_2} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n) \right. \\
&\quad \left. - \lambda_2(I_\ell \cap J_h^n) \lambda_2(I_{\ell'} \cap J_h^n) \right), \\
V_2(\widehat{Z}_{S_h}^{n+1}) &= \sum_{(\ell, \ell'): \ell_1 = \ell'_1, \ell_2 \neq \ell'_2} \left(\frac{N}{p-1} \sum_{(m, m'): m_1 \neq m'_1, m_2 = m'_2} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n) \right. \\
&\quad \left. - \lambda_2(I_\ell \cap J_h^n) \lambda_2(I_{\ell'} \cap J_h^n) \right), \\
V_3(\widehat{Z}_{S_h}^{n+1}) &= \sum_{(\ell, \ell'): \ell_1 \neq \ell'_1, \ell_2 \neq \ell'_2} \left(\frac{N}{(p-1)^2} \sum_{(m, m'): m_1 \neq m'_1, m_2 \neq m'_2} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n) \right. \\
&\quad \left. - \lambda_2(I_\ell \cap J_h^n) \lambda_2(I_{\ell'} \cap J_h^n) \right).
\end{aligned}$$

i. We have

$$V_1(\widehat{Z}_{S_h}^{n+1}) = \sum_{\ell: I_\ell \not\subset J_h^n, I_\ell \cap J_h^n \neq \emptyset} \sum_{\ell': \ell_1 \neq \ell'_1, \ell_2 = \ell'_2} V_1(\ell, \ell'),$$

where

$$\begin{aligned}
V_1(\ell, \ell') &:= \frac{N}{p-1} \sum_{(m, m'): m_1 = m'_1, m_2 \neq m'_2} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n) \\
&\quad - \lambda_2(I_\ell \cap J_h^n) \lambda_2(I_{\ell'} \cap J_h^n).
\end{aligned}$$

We split $V_1(\ell, \ell') = \widehat{V}_1(\ell, \ell') + \check{V}_1(\ell, \ell')$, with

$$\begin{aligned}
\widehat{V}_1(\ell, \ell') &:= \frac{N}{p-1} \sum_{(m, m'): m_1 = m'_1, m_2 \neq m'_2} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n) \\
&\quad - \frac{N}{p} \sum_{(m, m'): m_1 = m'_1} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n), \\
\check{V}_1(\ell, \ell') &:= p \\
&\quad \times \sum_{(m, m'): m_1 = m'_1} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n) - \lambda_2(I_\ell \cap J_h^n) \lambda_2(I_{\ell'} \cap J_h^n).
\end{aligned}$$

On one side

$$\begin{aligned} \hat{V}_1(\ell, \ell') &= N \sum_m \lambda_2(I_{\ell, m} \cap J_h^n) \\ &\quad \times \left(\frac{1}{p(p-1)} \sum_{m': m'_1 = m_1, m'_2 \neq m_2} \lambda_2(I_{\ell', m'} \cap J_h^n) - \frac{1}{p} \lambda_2(I_{\ell', m} \cap J_h^n) \right). \end{aligned}$$

Since both terms inside the parentheses are bounded by $1/(pN^2)$, we have $|\hat{V}_1(\ell, \ell')| \leq 1/(pN^2)$ and so

$$\left| \sum_{\ell': \ell'_1 \neq \ell_1, \ell'_2 = \ell_2} \hat{V}_1(\ell, \ell') \right| \leq \frac{p-1}{pN^2}.$$

On the other side

$$\check{V}_1(\ell, \ell') = \sum_{m_1 \in [1, p]} \lambda_2((H_{\ell_1, m_1} \times H_{\ell_2}) \cap J_h^n) \sum_{m'_1 \in [1, p]} \check{V}_1(\ell, \ell', m_1, m'_1),$$

where $\check{V}_1(\ell, \ell', m_1, m'_1) := \lambda_2((H_{\ell_1, m_1} \times H_{\ell_2}) \cap J_h^n) - \lambda_2((H_{\ell_1, m'_1} \times H_{\ell_2}) \cap J_h^n)$. We have

$$\begin{aligned} \check{V}_1(\ell, \ell', m_1, m'_1) &= \frac{1}{N} \left(\lambda \left(H_{\ell_2} \cap (0, q_h(i_{p(\ell_1-1)+m_1}^n)) \right) \right. \\ &\quad \left. - \lambda \left(H_{\ell_2} \cap (0, q_h(i_{p(\ell_1-1)+m'_1}^n)) \right) \right). \end{aligned}$$

As we have

$$\begin{aligned} |\check{V}_1(\ell, \ell', m_1, m'_1)| &\leq \frac{1}{N} \\ &\quad \times \lambda \left(H_{\ell_2} \cap \left[\min_{p(\ell_1-1) < k \leq p\ell'_1} q_h(i_k^n), \max_{p(\ell_1-1) < k \leq p\ell'_1} q_h(i_k^n) \right] \right), \end{aligned}$$

we deduce

$$|\check{V}_1(\ell, \ell')| \leq \frac{1}{Np} \lambda \left(H_{\ell_2} \cap \left[\min_{p(\ell_1-1) < k \leq p\ell'_1} q_h(i_k^n), \max_{p(\ell_1-1) < k \leq p\ell'_1} q_h(i_k^n) \right] \right).$$

Consequently

$$\begin{aligned} \left| \sum_{\ell': \ell'_1 \neq \ell_1, \ell'_2 = \ell_2} \check{V}_1(\ell, \ell') \right| &\leq \frac{1}{Np} \\ &\quad \times \sum_{\ell'_1 \in [1, p]: \ell'_1 \neq \ell_1} \lambda \left(H_{\ell_2} \cap \left[\min_{p(\ell_1-1) < k \leq p\ell'_1} q_h(i_k^n), \max_{p(\ell_1-1) < k \leq p\ell'_1} q_h(i_k^n) \right] \right). \end{aligned}$$

Since q_h is a piecewise monotonic sequence, and because the states are relabeled so that $i_1^n \leq \dots \leq i_N^n$ the intervals

$$\left(\min_{p(\ell'_1-1) < k \leq p\ell'_1} q_h(i_k^n), \max_{p(\ell'_1-1) < k \leq p\ell'_1} q_h(i_k^n) \right), \quad \ell'_1 \in [1, p]$$

are pairwise disjoint on each of the r pieces where q_h is monotonic, and we obtain

$$\left| \sum_{\ell': \ell'_1 \neq \ell_1, \ell'_2 = \ell_2} \check{V}_1(\ell, \ell') \right| \leq \frac{r}{Np} \lambda(H_{\ell_2}) \leq \frac{r}{N^2}.$$

And so, using the bound (13):

$$|V_1(\widehat{Z}_{s_h}^{n+1})| \leq \frac{(r+1)p-1}{pN^{3/2}} (TV(q_h) + 2).$$

ii. We have

$$V_2(\widehat{Z}_{s_h}^{n+1}) = \sum_{\ell: I_\ell \not\subset J_h^n, I_\ell \cap J_h^n \neq \emptyset} \sum_{\ell': \ell'_1 = \ell_1, \ell'_2 \neq \ell_2} V_2(\ell, \ell'),$$

where

$$\begin{aligned} V_2(\ell, \ell') &:= \frac{N}{p-1} \\ &\times \sum_{(m, m'): m_1 \neq m'_1, m_2 = m'_2} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n) - \lambda_2(I_\ell \cap J_h^n) \lambda_2(I_{\ell'} \cap J_h^n). \end{aligned}$$

We split $V_2(\ell, \ell') = \hat{V}_2(\ell, \ell') + \check{V}_2(\ell, \ell')$, with

$$\begin{aligned} \hat{V}_2(\ell, \ell') &:= \frac{N}{p-1} \sum_{(m, m'): m_1 \neq m'_1, m_2 = m'_2} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n) \\ &\quad - \frac{N}{p} \sum_{(m, m'): m_2 = m'_2} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n), \\ \check{V}_2(\ell, \ell') &:= p \sum_{(m, m'): m_2 = m'_2} \lambda_2(I_{\ell, m} \cap J_h^n) \lambda_2(I_{\ell', m'} \cap J_h^n) - \lambda_2(I_\ell \cap J_h^n) \lambda_2(I_{\ell'} \cap J_h^n). \end{aligned}$$

On one side

$$\begin{aligned} \hat{V}_2(\ell, \ell') &= N \sum_m \lambda_2(I_{\ell, m} \cap J_h^n) \\ &\quad \times \left(\frac{1}{p(p-1)} \sum_{m': m'_1 \neq m_1, m'_2 = m_2} \lambda_2(I_{\ell', m'} \cap J_h^n) - \frac{1}{p} \lambda_2(I_{\ell', m} \cap J_h^n) \right). \end{aligned}$$

Since both terms inside the parentheses are bounded by $1/(pN^2)$, we have $|\hat{V}_2(\ell, \ell')| \leq 1/(pN^2)$ and so

$$\left| \sum_{\ell': \ell'_1 = \ell_1, \ell'_2 \neq \ell_2} \hat{V}_2(\ell, \ell') \right| \leq \frac{p-1}{pN^2}.$$

On the other side

$$\check{V}_2(\ell, \ell') = \sum_{m_2 \in [1, p]} \lambda_2((H_{\ell_1} \times H_{\ell_2, m_2}) \cap J_h^n) \sum_{m'_2 \in [1, p]} \check{V}_2(\ell, \ell', m_2, m'_2),$$

where $\check{V}_2(\ell, \ell', m_2, m'_2) := \lambda_2((H_{\ell_1} \times H_{\ell_2, m_2}) \cap J_h^n) - \lambda_2((H_{\ell_1} \times H_{\ell_2, m'_2}) \cap J_h^n)$; we have

$$\begin{aligned} \check{V}_2(\ell, \ell', m_2, m'_2) &= \frac{1}{N} \sum_{m'_1 \in [1, p]} \left(\lambda \left(H_{\ell'_2, m_2} \cap (0, q_h(i_{p(\ell_1-1)+m'_1}^n)) \right) \right. \\ &\quad \left. - \lambda \left(H_{\ell'_2, m'_2} \cap (0, q_h(i_{p(\ell_1-1)+m'_1}^n)) \right) \right). \end{aligned}$$

Note that the difference in the parentheses is equal to 0 if $\ell'_2 \neq \ell'_2(\ell_1, m'_1) := \lceil nq_h(i_{p(\ell_1-1)+m'_1}^n) \rceil + 1$. Consequently

$$\begin{aligned} \left| \sum_{\ell': \ell'_1 = \ell_1, \ell'_2 \neq \ell_2} \check{V}_2(\ell, \ell') \right| &\leq \frac{1}{N} \sum_{m_2 \in [1, p]} \lambda_2((H_{\ell_1} \times H_{\ell_2, m_2}) \cap J_h^n) \\ &\quad \times \sum_{m'_1} \left| \lambda \left(H_{\ell'_2(\ell_1, m'_1), m_2} \cap (0, q_h(i_{p(\ell_1-1)+m'_1}^n)) \right) \right. \\ &\quad \left. - \lambda \left(H_{\ell'_2(\ell_1, m'_1), m'_2} \cap (0, q_h(i_{p(\ell_1-1)+m'_1}^n)) \right) \right| \\ &\leq \frac{1}{N} \sum_{m_2 \in [1, p]} \lambda_2((H_{\ell_1} \times H_{\ell_2, m_2}) \cap J_h^n) \leq \frac{1}{N} \lambda_2(I_\ell) = \frac{1}{N^2}. \end{aligned}$$

And so, using the bound (13):

$$|V_2(\hat{Z}_h^{n+1})| \leq \frac{2p-1}{pN^{3/2}} (TV(q_h) + 2).$$

iii. We have

$$V_3(\hat{Z}_h^{n+1}) = \sum_{\ell: I_\ell \not\subset J_h^n, I_\ell \cap J_h^n \neq \emptyset} \sum_{\substack{\ell': \ell'_1 \neq \ell_1, \ell'_2 \neq \ell_2 \\ I_{\ell'} \not\subset J_h^n, I_{\ell'} \cap J_h^n \neq \emptyset}} V_3(\ell, \ell'),$$

where

$$V_3(\ell, \ell') := \frac{N}{(p-1)^2} \times \sum_{(m,m'): m_1 \neq m'_1, m_2 \neq m'_2} \lambda_2(I_{\ell,m} \cap J_h^n) \lambda_2(I_{\ell',m'} \cap J_h^n) - \lambda_2(I_{\ell} \cap J_h^n) \lambda_2(I_{\ell'} \cap J_h^n).$$

We split $V_3(\ell, \ell') = V_3^a(\ell, \ell') - V_3^b(\ell, \ell') - V_3^c(\ell, \ell') - V_3^d(\ell, \ell')$, with

$$\begin{aligned} V_3^a(\ell, \ell') &:= \left(\frac{N}{(p-1)^2} - 1 \right) \sum_{(m,m'): m_1 \neq m'_1, m_2 \neq m'_2} \lambda_2(I_{\ell,m} \cap J_h^n) \lambda_2(I_{\ell',m'} \cap J_h^n), \\ V_3^b(\ell, \ell') &:= \sum_{(m,m'): m_1 \neq m'_1, m_2 = m'_2} \lambda_2(I_{\ell,m} \cap J_h^n) \lambda_2(I_{\ell',m'} \cap J_h^n), \\ V_3^c(\ell, \ell') &:= \sum_{(m,m'): m_1 = m'_1, m_2 \neq m'_2} \lambda_2(I_{\ell,m} \cap J_h^n) \lambda_2(I_{\ell',m'} \cap J_h^n), \\ V_3^d(\ell, \ell') &:= \sum_m \lambda_2(I_{\ell,m} \cap J_h^n) \lambda_2(I_{\ell',m} \cap J_h^n). \end{aligned}$$

Since

$$V_3^a(\ell, \ell') \leq \frac{2p-1}{N^3}, \quad V_3^b(\ell, \ell') \leq \frac{p-1}{N^3}, \quad V_3^c(\ell, \ell') \leq \frac{p-1}{N^3}, \quad V_3^d(\ell, \ell') \leq \frac{1}{N^3},$$

using the bound (13), we obtain:

$$|V_3(\widehat{Z}_{s_h}^{n+1})| \leq \frac{2p-1}{N^2} (TV(q_h) + 2)^2.$$

Hence the final result. □

6 Conclusion

In this article, we analyze the convergence of Monte Carlo methods, possibly combined with variance reduction techniques, for the simulation of Markov chains on one-dimensional discrete state spaces. We prove a bound of the variance of estimators used in one step of the simulation. We show that the convergence order (relative to the number of simulation paths) corresponds to the experimental order as calculated in various numerical experiments.

Albeit the theoretical convergence rates of SSS and Sudoku Latin square sampling are the same, for the numerical examples we see that the variance reduction of the Sudoku Latin square sampling approach is superior to that of the SSS. The difference is smaller for pure integration problems (see [5]). A drawback of the SSS approach for simulation is that it is not guaranteed that each state is considered exactly once for a transition.

The study aims to fill a gap between the theoretical results on variance reduction techniques used in Monte Carlo simulations and the actual improvements observed in computations. The Sudoku Latin square sampling is suited to situations where the state space is one-dimensional, with a natural order. Our interest has been in physical problems, where the states are related to particles. A numerical constraint is that N must be a square number. The method should be extended in many directions, such as continuous state spaces and multi-dimensional problems. This will be the subject of forthcoming research.

References

1. A.S. Alfa. *Applied Discrete-Time Queues, 2nd ed.* Springer, 2016.
2. R.C.H. Cheng and T. Davenport. The problem of dimensionality in stratified sampling. *Management Science*, 35:1278–1296, 1989.
3. I. Coulibaly and C. Lécot. Simulation of diffusion using quasi-random methods. *Mathematics and Computers in Simulation*, 47:153–163, 1998.
4. L. Devroye. *Non-Uniform Random Variate Generation.* Springer-Verlag, 1986.
5. R. El Haddad, R. Fakhereddine, C. Lécot and G. Venkiteswaran. Extended Latin hypercube sampling for integration and simulation. In J. Dick, F.Y. Kuo, G.W. Peters and I.H. Sloan, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2012*, pages 317–330. Springer-Verlag, 2013.
6. R. Fakhereddine, R. El Haddad, C. Lécot, and J. El Maalouf, Stratified Monte Carlo simulation of Markov chains. *Mathematics and Computers in Simulation*, 135:51–62, 2017.
7. M. Gerber and N. Chopin. Sequential quasi-Monte Carlo. *Journal of the Royal Statistical Society: Series B*, 77:509–579, 2015.
8. S. Haber. A modified Monte Carlo quadrature. *Mathematics of Computation*, 20:361–368, 1966.
9. C. Lécot and B. Tuffin. Quasi-Monte Carlo methods for estimating transient measures of discrete time Markov chains. In H. Niederreiter, editor, *Monte Carlo and Quasi-Monte Carlo Methods 2002*, pages 329–344. Springer-Verlag, 2004.
10. P. L’Ecuyer. Efficiency improvement and variance reduction. In J.D. Tew, S. Manivannan, D.A. Sadowski and A.F. Seila, editors, *Proceedings of the 1994 Winter Simulation Conference*, pages 122–132. IEEE Press, 1994.
11. P. L’Ecuyer, C. Lécot and B. Tuffin. Randomized quasi-Monte Carlo simulation of Markov chains with an ordered state space. In H. Niederreiter and D. Talay, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 331–342. Springer-Verlag, 2006.
12. P. L’Ecuyer, C. Lécot and B. Tuffin. A randomized quasi-Monte Carlo simulation method for Markov chains. *Operations Research*, 56(4):958–975, 2008.
13. P. L’Ecuyer, C. Lécot, A. L’Archevêque-Gaudet. On array-RQMC for Markov chains: mapping alternatives and convergence rates. In P. L’Ecuyer and A. Owen, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2008*, pages 485–500. Springer-Verlag, 2009.
14. P. L’Ecuyer, D. Munger, C. Lécot and B. Tuffin. Sorting methods and convergence rates for Array-RQMC: Some empirical comparisons. *Mathematics and Computers in Simulation*, 143:191–201, 2018.
15. M.D. McKay, R.J. Beckman and W.J. Conover. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21:239–245, 1979.
16. A.B. Owen. Orthogonal arrays for computer experiments, integration and visualization. *Statistica Sinica*, 2:439–452, 1992.

17. A.B. Owen. Lattice sampling revisited: Monte Carlo variance of means over randomized orthogonal arrays. *The Annals of Statistics*, 22:930–945, 1994.
18. A.B. Owen. Monte Carlo variance of scrambled net quadrature. *SIAM Journal on Numerical Analysis*, 34:1884–1910, 1997.
19. R.M. Pedersen and T.L. Vis. Sets of mutually orthogonal Sudoku Latin squares. *The College Mathematics Journal*, 40:174–180, 2009.
20. M. Stein. Large sample properties of simulations using Latin hypercube sampling. *Technometrics*, 29:143–151, 1987.
21. B. Tang. Orthogonal array-based Latin hypercubes. *Journal of the American Statistical Association*, 88:1392–1397, 1993.
22. X. Xu, B. Haaland and P.Z.G. Qian. Sudoku-based space-filling designs. *Biometrika*, 98:711–720, 2011.