Available online at www.sciencedirect.com
ScienceDirect
Mathematics and Computers in Simulation 143 (2018) 191–201
MATHEMATICS AND COMPUTERS IN SIMULATION
www.elsevier.com/locate/matcom

Original articles

Sorting methods and convergence rates for Array-RQMC: Some empirical comparisons

Pierre L'Ecuyer[a,b,*], David Munger[a], Christian Lécot[c], Bruno Tuffin[b]

[a] *DIRO, Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal H3C 3J7, Canada*
[b] *Inria Rennes – Bretagne Atlantique, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France*
[c] *Université Savoie Mont Blanc, LAMA, UMR 5127 CNRS, 73376 Le Bourget-du-Lac Cedex, France*

Received 1 February 2016; received in revised form 19 July 2016; accepted 21 July 2016
Available online 30 July 2016

Abstract

We review the Array-RQMC method, its variants, sorting strategies, and convergence results. We are interested in the convergence rate of measures of discrepancy of the states at a given step of the chain, as a function of the sample size n, and also the convergence rate of the variance of the sample average of a (cost) function of the state at a given step, viewed as an estimator of the expected cost. We summarize known convergence rate results and show empirical results that suggest much better convergence rates than those that are proved. We also compare different types of multivariate sorts to match the chains with the RQMC points, including a sort based on a Hilbert curve.

Keywords: Low discrepancy; Quasi-Monte Carlo; Markov chain; Variance reduction; Array-RQMC

1. Introduction

Array-RQMC is a method to simulate an array of n dependent realizations of a Markov chain in a way that each chain is generated from its exact probability law, and with the aim that the empirical distribution of the states at a given step of the chain provides a "low-discrepancy" approximation of the theoretical distribution of the state at that step. At each step, the n copies of the chain are sorted in a particular order and then moved forward by one step using a randomized quasi-Monte Carlo (RQMC) point set of cardinality n. If the state space has more than one dimension, the sort can be multidimensional and the performance may depend strongly on the sorting method. More details on the method, intuitive justifications, convergence results, applications, and empirical evaluations, can be found in [4,5,7,8,13,15–18,20,25].

The aim of this paper is to review briefly what is known and what has been done so far with this method, report new experimental results on convergence rates, and compare different types of multivariate sorting methods, including

* Corresponding author at: DIRO, Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal H3C 3J7, Canada.
E-mail addresses: lecuyer@iro.umontreal.ca (P. L'Ecuyer), mungerd@iro.umontreal.ca (D. Munger), Christian.Lecot@univ-savoie.fr (C. Lécot), bruno.tuffin@inria.fr (B. Tuffin).

http://dx.doi.org/10.1016/j.matcom.2016.07.010

sorts based on space-filling curves. Those types of curves are widely used to map points from the multivariate unit cube $[0, 1]^\ell$ to the unit interval $[0, 1]$ in various areas of applications such as sorting multivariate objects, mapping them to memory addresses in database systems and multiprocessing computers, storing and reconstructing images in computer graphics, etc., see [1]. They also provide one of the most effective heuristics to quickly obtain a good solution for a traveling salesman problem with a large number of points in two or more dimensions [2,3,23]. Thus, their use to reduce the dimension and order the states in Array-RQMC seems natural, and was suggested in [25]. A Hilbert sort to map two-dimensional points to $[0, 1)$ is also proposed in [24] for QMC sampling in computer graphics. Recently, Gerber and Chopin [9] proposed to use the Hilbert space-filling curve to sort multidimensional states in a variant of Array-RQMC combined with particle filtering, which they named *sequential quasi-Monte Carlo*, and proved that under some conditions, the variance of the filtering estimator converges as $o(n^{-1/2})$, i.e., faster than for Monte Carlo. He and Owen [11] study the Hilbert curve as a way of reducing the dimension from $d > 1$ to 1 in the context of estimating a d-dimensional integral by QMC or RQMC (not for Markov chains). They prove convergence rate bounds on the MSE under different sets of conditions on the integrand and the points. We discuss this type of sort and compare it empirically to other multivariate sorts proposed previously. We also survey currently known convergence rate results for Array-RQMC, and show examples in which the observed convergence rates are much better than those that are proved.

The remainder is organized as follows. We review the Array-RQMC algorithm in Section 2, multivariate sorts in Section 3, and theoretical convergence results in Section 4. In Section 5, we compare the convergence rates observed empirically in some examples with those that are proved, for the mean square L_2-discrepancy and the variance of cost estimators.

Note: It is customary to define space-filling curves over the closed hypercube $[0, 1]^s$, QMC points over the semi-open hypercube $[0, 1)^s$ in s dimensions, and uniform random variables over the open interval $(0, 1)$, because the inverse cdf may be infinite at 0 or 1. We follow these conventions as much as we can in the paper, but there are inevitable occasional clashes (when two different conventions would apply to the same box). This may appear inconsistent, but in our practical implementations, no coordinate will ever equal 1, so it does not matter if the interval is open or closed.

2. Array-RQMC

We consider a discrete-time Markov chain whose state evolves in a measurable space $\mathcal{X}$ according to a stochastic recurrence defined by

$$X_0 = x_0 \in \mathcal{X} \quad \text{and} \quad X_j = \varphi_j(X_{j-1}, \mathbf{U}_j) \quad \text{for } j \geq 1,$$

in which $\mathbf{U}_1, \mathbf{U}_2, \ldots$ are independent and identically distributed (i.i.d.) uniform random variables over the unit hypercube $(0, 1)^d$, and each $\varphi_j : \mathcal{X} \times (0, 1)^d \to \mathcal{X}$ is a measurable mapping. Suppose we want to estimate

$$\mu = \mathbb{E}[Y], \quad \text{where } Y = \sum_{j=1}^{\tau} g_j(X_j)$$

for some measurable *cost (or reward) functions* $g_j : \mathcal{X} \to \mathbb{R}$, where τ is a fixed time horizon (a positive integer). The methods we describe also work if τ is a random stopping time, as explained in [18].

The *standard Monte Carlo* (MC) method estimates μ by $\bar{Y}_n = \frac{1}{n}\sum_{i=0}^{n-1} Y_i$ where $Y_0, \ldots, Y_{n-1}$ are n independent realizations of Y. One has $\mathbb{E}[\bar{Y}_n] = \mu$ and $\mathrm{Var}[\bar{Y}_n] = \mathrm{Var}[Y]/n = \mathcal{O}(n^{-1})$ if $\mathbb{E}[Y^2] < \infty$.

A naive way of using RQMC in this setting would be to replace the n independent $d\tau$-dimensional uniformly distributed vectors $\mathbf{V}_i = (\mathbf{U}_{i,1}, \ldots, \mathbf{U}_{i,\tau}) \in (0, 1)^{d\tau}$ by a $d\tau$-dimensional RQMC point set, in $(0, 1)^{d\tau}$. However, when $d\tau$ increases, this RQMC scheme typically becomes ineffective, because $\mathbb{E}[Y]$ is a high-dimensional integral.

The Array-RQMC method was designed to address this issue. To provide an intuitive explanation of how it works, let us assume that we have a one-to-one mapping $h : \mathcal{X} \to [0, 1)^c$, for some small integer $c \geq 1$ (to implement the method, it is not essential to define such a mapping h explicitly, and h does not have to be one-to-one). If h is one-to-one, $\tilde{X}_j = h(X_j)$ contains all the information in X_j that is relevant for the probability law of the future evolution of the chain. At each step j, the n realizations of the chains are "sorted" in some order, based on the realizations of the transformed state $\tilde{X}_j$, and then matched to n RQMC points based on that order. In [18], it was assumed that $c = 1$, so h maps the states to the interval $[0, 1)$, and the states were sorted by increasing order of $\tilde{X}_j$.