Proceedings of the 1998 Winter Simulation Conference
D.J. Medeiros, E.F. Watson, J.S. Carson and M.S. Manivannan, eds.

EFFICIENCY IMPROVEMENT BY LATTICE RULES FOR PRICING ASIAN OPTIONS

Christiane Lemieux
Pierre L'Ecuyer

Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville
Montréal, H3C 3J7, CANADA

ABSTRACT

This paper compares Monte Carlo methods, lattice rules, and other low-discrepancy point sets on the problem of evaluating asian options. The combination of these methods with variance reduction techniques is also explored.

1 INTRODUCTION

For the approximation of multidimensional integrals, two types of methods are widely used. Monte Carlo (MC) methods are the best known and require the use of a pseudorandom generator. Quasi-Monte Carlo (QMC) methods use low-discrepancy point sets and are deterministic: The idea is to use points that are more regularly distributed over the integration space than random points. The best known methods to achieve this are the lattice rules and (t, s)-sequences (or (t, m, s)-nets); see, e.g., Owen (1998), Niederreiter (1992) and Sloan and Joe (1994).

When MC methods are used to estimate the value of an integral, the central limit theorem allows the calculation of an error estimate that gives an idea of the quality of the estimator. This probabilistic error is in $O(N^{-1/2})$, where N is the sample size, independently of the dimension s of the integrand. For QMC methods, a wide body of literature exists on how to derive deterministic error bounds (Niederreiter 1992; Hickernell 1998; Hickernell 1999). These bounds are the product of two quantities: The *discrepancy*, which measures the uniformity of the point set, and a measure of *variation* of f, which tells about the roughness of the integrand. There are several ways of defining the discrepancy, each one coming with its corresponding definition for the variation of f and with a class of functions for which the error bound applies. For low-discrepancy point sets, these bounds are in $O((\log N)^s/N)$, a better asymptotic rate than $O(N^{-1/2})$. In practice, however, these bounds are often almost impossible to calculate and are also extremely loose.

A simple way of getting an error estimate in the QMC case is to randomly shift the point set, modulo 1 coordinate-wise, and repeat this independently, say, m times. This gives m i.i.d. unbiased integral estimates, from which a variance estimate and a confidence interval can be computed in the usual way. This randomization technique was suggested by Cranley and Patterson (1976).

In this paper, we compare lattice rules with MC methods, on the financial problem of pricing asian options. We find that lattice rules easily win over MC and still dominate in dimensions as large as 120. These results indicate that the class of functions for which these low-discrepancy point sets give efficient estimators is not as restricted as what is usually suggested in the literature (Sloan and Joe 1994; Owen 1997). We also compare the lattice rules with (t, s)-sequences, such as Sobol and generalized Faure sequences, on the same problem, and find that the lattice rules are competitive.

The paper is organized as follows. Section 2 summarizes basic definitions and facts about lattice rules and criteria to select them, for a given number of points. In section 3, we state the problem of asian options pricing. We give numerical results in section 4. Finally, we conclude in section 5.

2 LATTICE RULES

To estimate the integral

$$If = \int_{[0,1)^s} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x},$$

we choose a point set $P = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ in $[0, 1)^s$ and compute the estimator

$$Qf = \frac{1}{N}\sum_{i=1}^{N} f(\boldsymbol{x}_i).$$

In the MC method, the points in P come from a pseudorandom generator. For the QMC method, P is a low-discrepancy point set, coming from either a (t, s)-sequence or a lattice rule.

We recall the following standard definitions regarding lattice rules (Hickernell 1999; Sloan and Joe 1994).

- An s-dimensional **integration lattice**, L, is a discrete subset of $\mathbb{R}^s$ that is closed under addition and substraction and which contains the integer vectors $\mathbb{Z}^s$ as a subset.
- A **shifted lattice** with shift $\Delta \in \mathbb{R}^s$ is the set $L + \Delta = \{\boldsymbol{z} + \Delta : \boldsymbol{z} \in L\}$ for some lattice L.
- The **node set** for a shifted integration lattice, $L + \Delta$, is the set of points in the lattice that fall inside the unit cube, that is, $P = (L + \Delta) \cap [0, 1)^s$.
- The **dual lattice** of L is defined as $L^\perp = \{\boldsymbol{k} \in \mathbb{R}^s : \boldsymbol{k} \cdot \boldsymbol{z} \in \mathbb{Z}$ for all $\boldsymbol{z} \in L\}$.
- A **rank-1 lattice** is a lattice whose node set may be expressed as

  $$P = \{\{i\boldsymbol{h}/N\} : i = 0, \ldots, N-1\},$$

  for some generating vector $\boldsymbol{h} \in \mathbb{Z}^s$, where $\{\boldsymbol{x}\} = \boldsymbol{x} - \lfloor \boldsymbol{x} \rfloor = \boldsymbol{x} \bmod 1$.

It is interesting to note that for a rank-1 lattice with N points and generating vector $\boldsymbol{h}$ of the form $\boldsymbol{h} = (1, a, \ldots, a^{s-1})$, the node set is the set of all s-tuples formed by successive output values of a linear congruential generator (LCG) with modulus N and multiplier a, from all possible initial seeds (including 0). We use this connection to choose our lattice rules (as explained later).

If P is the node set of an integration lattice, assuming that the integrand f has an absolutely convergent Fourier series, the integration error is (see. e.g., Sloan and Joe 1994)

$$Qf - If = \sum_{\mathbf{h} \in L^\perp \setminus \{\mathbf{0}\}} \hat{f}(\boldsymbol{h}), \qquad (1)$$

where $\hat{f}(\boldsymbol{h}) = \int_{[0,1)^s} f(\boldsymbol{x}) e^{-2\pi\sqrt{-1}\mathbf{h}\cdot\mathbf{x}} \mathrm{d}\boldsymbol{x}$, is the Fourier coefficient of f evaluated at $\boldsymbol{h}$.

It is common to assume that the Fourier coefficients decrease as $\boldsymbol{h}$ gets away from the origin. This amounts to assuming that f is "sufficiently smooth". In view of (1), we thus want the points of the dual lattice to be far away from the origin, when the distances are measured with an arbitrary norm, say $\|\cdot\|$. A general *figure of merit* (to be minimized) can have the form

$$\sum_{\mathbf{h} \in L^\perp \setminus \{\mathbf{0}\}} \|\boldsymbol{h}\|^{-\alpha} \qquad (2)$$

or perhaps

$$\sup_{\mathbf{h} \in L^\perp \setminus \{\mathbf{0}\}} \|\boldsymbol{h}\|^{-\alpha} \qquad (3)$$

for some constant $\alpha > 0$. If one considers the class of functions f for which $|\hat{f}(\boldsymbol{h})| \leq \|\boldsymbol{h}\|^{-\alpha}$ for all $\boldsymbol{h} \neq \mathbf{0}$, then (2) is the *worst case integration error* for that class of functions, and thus provides an error bound.

A popular norm used in this context is the product norm $\|\boldsymbol{h}\|_\pi = \bar{h}_1 \cdots \bar{h}_s$ where $\bar{h}_j = \max(1, |h_j|)$ for $\boldsymbol{h} = (h_1, \ldots, h_s)$. The figure of merit (2) with this norm is known as P_α, and is often recommended with $\alpha = 2$ (Sloan and Walsh 1990; Sloan and Joe 1994). With this product norm and $\alpha = 1$, (3) is $1/\rho_s$ where ρ_s is known as the *Zaremba index* (Niederreiter 1992). If one uses the Euclidean norm instead, with $\alpha = 1$, (3) becomes the distance between the successive hyperplanes in the lattice L, which is the figure of merit computed by the *spectral test* commonly used to measure the quality of LCGs.

The lattice rules used for the experiments reported in this paper are rank-1 rules that correspond to LCGs with prime moduli and period $N - 1$, and which have been selected based on the spectral test. These LCGs are taken from Table 2 of L'Ecuyer (1998b). They have the convenient property that their lattice structure is good uniformly in s, for $s \leq 32$, and also quite good for *most* $s > 32$. The same LCG (or lattice rule) can thus be used for all s, instead of choosing a different rule for each s as done, e.g., in Sloan and Joe (1994). The lattice rules based on LCGs are also very easy to implement: The points are obtained simply by running the LCG as usual and taking all the overlapping vectors of successive output values over the entire period.

To estimate the integration error, we use the randomization technique of Cranley and Patterson (1976), which works as follows. For a given point set $P = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, generate m i.i.d. vectors $\Delta_1, \ldots, \Delta_m$ uniformly distributed in $[0, 1)^s$, and for $i = 1, \ldots, m$, compute the value X_i of Qf for the point set $\{P + \Delta_i\}$, i.e., P shifted by Δ_i modulo 1. These X_i turn out to be i.i.d. random variables with expectation If. Then compute

$$\hat{\mu} \;=\; \frac{1}{m} \sum_{i=1}^{m} X_i \qquad (4)$$

as the estimator of If and

$$\hat{\sigma}^2 \;=\; \frac{1}{m(m-1)} \sum_{i=1}^{m} (X_i - \hat{\mu})^2 \qquad (5)$$

as an estimator of the variance of $\hat{\mu}$. For the MC estimator, we use $m * N$ replications with a pseudorandom number generator.

3 PRICING ASIAN OPTIONS

Finance gives rise to several interesting problems involving the computation of multidimensional integrals. An example of this kind of problems is the pricing of *contingent claims*, which are assets having a price that depends on the value of other assets, called *underlying assets*. An *asian option* is an example of a contingent claim; its price depends on the mean value of an underlying asset during a certain period of time. More precisely, let

$$\begin{aligned}
S(t) &= \text{value of the underlying asset at time } t,\\
s &= \text{number of instants where } S(\cdot) \text{ is sampled},\\
T_1 &= \text{beginning of the period in which}\\
&\ \text{the mean is calculated},\\
T &= \text{expiration date of the option},\\
t_j &= T_1 + j(T-T_1)/s,\ j=1,\ldots,s,\\
K &= \text{strike price of the option.}
\end{aligned}$$

If $C_A(t)$ denotes the value of the asian call option at time t, then by definition its final value is

$$C_A(T) = \max\left(0, \frac{1}{s}\sum_{j=1}^{s} S(t_j) - K\right),$$

which is the difference between the average value of the underlying asset sampled at s points equally spaced over the period (T_1, T) and the strike price K, provided that this average exceeds K, otherwise the option has no value.

To model the evolution of the underlying asset, we use the model of Black and Scholes (1973), which says that

$$dS(t) = \mu S(t)dt + \sigma S(t)dB(t),$$

where μ is the mean return parameter of the asset, σ is its volatility parameter and $B(\cdot)$ is a standard Brownian motion. This model involves other assumptions about the market implying the existence of a unique risk-neutral measure under which

$$dS(t) = rS(t)dt + \sigma S(t)d\tilde{B}(t),$$

where r is the risk-free interest rate and $\tilde{B}(\cdot)$ is a standard Brownian motion under the risk-neutral measure. The value at time 0 of the asian option is then given by

$$C_A(0) = \tilde{\mathrm{E}}(e^{-rT}C_A(T)), \tag{6}$$

where $\tilde{\mathrm{E}}(\cdot)$ is the expectation under the risk-neutral measure.

Even for this simple model, one cannot compute $C_A(0)$ analytically (this would amount to compute the distribution of a sum of lognormal random variables). So one would rely on MC or QMC methods to estimate the s-dimensional integral (6). To compute each replicate of $C_A(T)$, a point $\boldsymbol{x} \in [0,1)^s$ is transformed into a vector of normal random variables which are used to generate the values of the Brownian motion at the observation points t_j. The paths of $S(\cdot)$ are simulated as follows:

$$S(t_j) = S(0)\exp\Big((r - 0.5\sigma^2)t_j + \sigma\tilde{B}(t_j)\Big)$$

for $j = 1, \ldots, s$, where $\tilde{B}(t) \sim N(0, t^2)$, a Gaussian random variable with mean 0 and variance t.

In practice, it is the option on the arithmetic average that is sold, but the one on the geometric average can be used as a control variable, as explained in Kemna and Vorst (1990). The final value of the asian call option on the geometric average is defined as

$$C_G(T) = \max\left(0, \left(\prod_{j=1}^{s} S(t_j)\right)^{1/s} - K\right).$$

This option can be priced exactly for the Black-Scholes model (a product of lognormal random variables is also a lognormal). The geometric average is always smaller than the arithmetic average, but these two quantities are highly correlated and so are the corresponding option prices. As shown in the next section, this control variable brings a dramatic variance reduction. We also use antithetic variates, which give us a small additional reduction of the variance. For more on variance reduction, see, e.g., L'Ecuyer (1994) and other references given there. Lemieux (1996) studies and compares several approaches for evaluating asian options under the Black-Scholes model, including simulation with the variance reduction techniques described above, and other approximation methods.

4 NUMERICAL RESULTS

4.1 Impact of Variance Reduction Techniques

The following results compare lattice rules with MC simulations on the asian option problem, with and without the control variable and antithetic variates. Four different estimators are calculated for both MC and the lattice rules: A naive estimator (naive), one with antithetic variates (ant.), one with the control variable (c.v.), and one that combines these two variance reduction methods (ant. c.v.). The last 8 columns of Tables 1 and 2 correspond to these 8 cases.

Three values of the strike price K are considered in these tables. The higher the strike price, the smaller the probability of a nonzero final value, which means (intuitively) a larger relative error. In the tables, $\hat{\mu}$ and $\hat{\sigma}$ are the empirical mean and standard error as defined in (4) and (5). For the MC estimators, the number in parentheses under $\hat{\sigma}$ is $\hat{\sigma}^2/\hat{\sigma}_0^2$, where $\hat{\sigma}_0^2$ is the value of $\hat{\sigma}^2$ for the naive estimator. For the lattice rule estimators, this number in parentheses gives the ratio of their estimated variance $\hat{\sigma}^2$ to that of the *corresponding* MC estimator (which uses the same variance reduction strategy). The next line in the tables gives the CPU time in seconds to perform all the mN simulations runs and compute the estimators. The line labelled "eff." gives the (estimated) *efficiency* of the estimator, which is defined as the inverse of the product of the estimated variance by the CPU time. The numbers in parentheses under the efficiency values provide ratios similar to those given for the variance three lines above, but for the efficiency instead of $\hat{\sigma}^2$.

For the examples considered in this paper, the risk-free rate and the volatility in the Black-Scholes model are fixed to $r = \ln 1.09$ and $\sigma = 0.2$. We also take $S(0) = 100$, $T = 120$ days and $m = 100$. The length of the sampling period, $T - T_1$, which is also equal to the number s of dimensions, is 10 in Table 1 and 120 in Table 2. The value of N, which is the number of points as well as the modulus of the corresponding LCG, and its multiplier a, are given at the top of each table.

In Table 1, we see that the control variable increases the efficiency by large factors (around 10^6). When no other variance reduction technique is used, the lattice rule estimator is more efficient than the MC estimator by a factor as large as 500 in one case and at least 100 in all cases. Even when both variance reduction techniques are combined, lattice rules still provide estimators at least 3 times more efficient than the MC estimators. Variance reduction techniques "even" the integrand. For this new integrand, the lattice rules are still doing significantly better than MC.

Table 2 gives the results of a higher-dimensional problem, with $s = 120$, and a smaller number of points, $N = 509$. With these parameters, the lattice rules provide less improvement than in Table 1, but they are still more efficient than the MC methods.

The control variable does not work as well as in Table 1, but still gives estimators significantly more efficient than the naive ones, by factors around 10^3. Increasing s means that we sample $S(\cdot)$ over a longer period (T_1, T). Hence, the probability of observing abnormal prices is greater and this increases the difference between the two averages, so the correlation between $C_A(T)$ and $C_G(T)$ is reduced.

In both tables, the variance ratios usually decrease with K. Notice that f is more irregular for larger K ($f(\boldsymbol{x}) = 0$ for a large set of values of $\boldsymbol{x} \in [0,1)^s$ and then increases abruptly). Also, the correlation between $C_A(T)$ and $C_G(T)$ decreases, and this goes with a larger probability of having the pathological case of $C_A(T) > 0$ and $C_G(T) = 0$.

The precision of the estimators $\hat{\sigma}$ can be assessed by constructing confidence intervals (C.I.'s) for σ. These intervals are not given in the tables, but their typical sizes is well illustrated by the following examples. In Table 1, with $K = 100$ and the naive estimator, the 99% C.I.'s are $(0.0122, 0.0123)$ for MC and $(0.0057, 0.0082)$ for the lattice rules. For the MC method, the variance estimator is more accurate because it has more degrees of freedom: MC makes mN independent runs, whereas with the lattice rules we have only m independent groups of runs.

In the same scope, one can construct C.I.'s for the variance ratios, to test whether these ratios are *significantly* different from 1. For $m = 100$ (i.e. for the lattice rules), the variance ratios given in the line below $\hat{\sigma}$ in the tables have to be smaller than 0.699 for the ratios to be significative at the 99% level. This is the case for all 12 entries in Table 1, and for 8 entries in Table 2.

4.2 Comparison With (t, s)-Sequences

We now compare MC with three types of QMC point sets: The lattice rules, and point sets taken from Sobol and generalized Faure (GFaure) sequences. The latter point sets were obtained using the software FINDER of Papageorgiou and Traub (1996), Paskov and Traub (1995). In all cases, we use both the antithetic variates and the control variable. For the MC method, we use the generator MRG32k5a of L'Ecuyer (1998a). Ratios for the variance and the efficiency are given w.r.t. the MC estimator.

To randomize the Sobol and GFaure point sets (for error estimation), we use the same random shift method as for the lattice rules. Shifting randomly a (t, m, s)-net does not preserve the net property, but if $f \in \mathcal{L}^2$, it produces an unbiased estimator with a variance in the order of $N^{-2}(\log N)^{2s}$, as shown in Tuffin (1996): This is the same order of convergence as the *scrambled net* estimator proposed by Owen (1995), which gives another way to randomize (t, m, s)-nets.

For the Sobol method, we use the smallest power of two $\tilde{N}$ greater than the number of points N used by the three other methods, because for Sobol sequences it is better to take N as a power of two. It is often recommended to skip a certain number of points at the beginning of a (t, s)-sequence (Fox 1986; Bratley, Fox, and Niederreiter 1992). We skipped the first $\tilde{N}$ points for Sobol and the first 200000 points for GFaure (as did Acworth, Broadie, and Glasserman 1997).

In Table 3 (where $s = 10$), the lattice rule estimators lose to both Sobol and GFaure in terms of efficiency, but in Table 4 (where $s = 60$), the situation is reversed.

In another experiment with $s = 90$ and $N = 509$, not shown here, the lattice rules still win over MC by non-negligible factors, whereas Sobol and GFaure are

Table 1: $s = T - T_1 = 10$ days. $N = 4093, a = 209$.

		Monte Carlo				Lattice Rule			
K		naive	ant.	c.v.	ant. c.v.	naive	ant.	c.v.	ant. c.v.
90	$\hat{\mu}$	13.001	13.004	13.008	13.008	13.008	13.008	13.008	13.008
	$\hat{\sigma}$	1.60e-2	3.54e-3	1.29e-5	1.18e-5	7.39e-4	5.21e-4	8.00e-6	6.60e-6
		(1.00)	(4.89e-2)	(6.49e-7)	(5.44e-7)	(2.13e-3)	(2.16e-2)	(3.85e-1)	(3.12e-1)
	cpu	47.2	62.4	51.7	69.8	38.0	53.2	42.5	60.8
	eff.	8.27e+1	1.28e+3	1.16e+8	1.03e+8	4.82e+4	6.93e+4	3.67e+8	3.78e+8
		(1.00)	**(15.5)**	**(1410000)**	**(1240000)**	**(582)**	**(54.3)**	**(3.16)**	**(3.68)**
100	$\hat{\mu}$	5.864	5.865	5.863	5.863	5.863	5.863	5.863	5.863
	$\hat{\sigma}$	1.23e-2	5.79e-3	1.18e-5	8.56e-6	6.76e-4	5.50e-4	6.11e-6	4.47e-6
		(1.00)	(2.23e-1)	(9.25e-7)	(4.87e-7)	(3.03e-3)	(9.02e-3)	(2.68e-1)	(2.73e-1)
	cpu	47.1	62.5	51.7	69.9	38.0	53.2	42.6	60.7
	eff.	1.41e+2	4.77e+2	1.39e+8	1.95e+8	5.76e+4	6.21e+4	6.28e+8	8.24e+8
		(1.00)	**(3.38)**	**(986000)**	**(1380000)**	**(408)**	**(130)**	**(4.52)**	**(4.22)**
110	$\hat{\mu}$	1.912	1.913	1.917	1.917	1.917	1.917	1.917	1.917
	$\hat{\sigma}$	7.27e-3	4.67e-3	8.96e-6	6.16e-6	7.39e-4	5.71e-4	6.16e-6	3.62e-6
		(1.00)	(4.13e-1)	(1.52e-6)	(7.18e-7)	(1.03e-2)	(1.49e-2)	(4.73e-1)	(3.45e-1)
	cpu	47.1	62.5	51.8	69.9	38.1	53.2	42.5	60.6
	eff.	4.02e+2	7.34e+2	2.41e+8	3.77e+8	4.80e+4	5.77e+4	6.20e+8	1.26e+8
		(1.00)	**(1.83)**	**(599000)**	**(939000)**	**(120)**	**(78.7)**	**(2.58)**	**(3.34)**

Table 2: $s = T - T_1 = 120$ days. $N = 509$, $a = 35$.

		Monte Carlo				Lattice Rule			
K		naive	ant.	c.v.	ant. c.v.	naive	ant.	c.v.	ant. c.v.
90	$\hat{\mu}$	11.237	11.209	11.207	11.207	11.209	11.208	11.208	11.208
	$\hat{\sigma}$	2.86e-2	3.28e-3	4.00e-4	3.29e-4	4.34e-3	2.96e-3	3.09e-4	2.64e-4
		(1.00)	(1.32e-2)	(1.96e-4)	(1.33e-4)	(2.30e-2)	(8.10e-1)	(5.95e-1)	(6.43e-1)
	cpu	70.1	97.4	70.9	92.5	55.6	80.4	56.5	78.5
	eff.	1.74e+1	9.52e+2	8.80e+4	9.97e+4	9.54e+2	1.42e+3	1.85e+5	1.83e+5
		(1.00)	**(54.6)**	**(5050)**	**(5720)**	**(54.8)**	**(1.50)**	**(2.11)**	**(1.83)**
100	$\hat{\mu}$	3.368	3.365	3.368	3.368	3.378	3.376	3.367	3.368
	$\hat{\sigma}$	1.99e-2	9.40e-3	3.16e-4	2.33e-4	8.44e-3	8.38e-3	2.32e-4	1.83e-4
		(1.00)	(2.23e-1)	(2.52e-4)	(1.36e-4)	(1.79e-1)	(7.94e-1)	(5.39e-1)	(6.20e-1)
	cpu	70.1	97.8	70.9	92.4	55.6	77.7	56.5	78.2
	eff.	3.59e+1	1.16e+2	1.41e+5	2.00e+5	2.53e+2	1.83e+2	3.28e+5	3.81e+5
		(1.00)	**(3.22)**	**(3920)**	**(5560)**	**(7.03)**	**(1.58)**	**(2.33)**	**(1.91)**
110	$\hat{\mu}$	0.394	0.386	0.386	0.386	0.387	0.387	0.386	0.386
	$\hat{\sigma}$	6.92e-3	4.66e-3	2.55e-4	1.79e-4	5.04e-3	3.89e-3	2.47e-4	1.77e-4
		(1.00)	(4.54e-1)	(1.36e-3)	(6.72e-4)	(5.30e-1)	(6.96e-1)	(9.40e-1)	(9.75e-1)
	cpu	70.1	97.7	70.9	92.7	55.6	77.7	56.5	77.9
	eff.	2.98e+2	4.71e+2	2.17e+5	3.35e+5	7.08e+2	8.52e+2	2.89e+5	4.09e+5
		(1.00)	**(1.58)**	**(729)**	**(1130)**	**(2.38)**	**(1.81)**	**(1.34)**	**(1.22)**

Table 3: $s = 10,\ N = 4093,\ a = 209.$

K		MC	LR	Sobol	GFaure
90	$\hat{\mu}$	13.008	13.008	13.008	13.008
	$\hat{\sigma}$	1.18e-5	6.07e-6	4.30e-6	4.49e-6
		(1.00)	(0.263)	(0.132)	(0.144)
	cpu	192.0	185.0	183.0	218.0
	$\frac{\text{eff.}}{10^4}$	3718.1	14652	29559	22718
		(1.00)	**(3.94)**	**(7.95)**	**(6.11)**
100	$\hat{\mu}$	5.863	5.863	5.863	5.863
	$\hat{\sigma}$	8.61e-6	4.35e-6	3.42e-6	4.03e-6
		(1.00)	(0.255)	(0.158)	(0.219)
	cpu	191.0	184.0	184.0	218.0
	$\frac{\text{eff.}}{10^4}$	7068.3	28727	46476	28237
		(1.00)	**(4.06)**	**(6.58)**	**(4.00)**
110	$\hat{\mu}$	1.917	1.917	1.917	1.917
	$\hat{\sigma}$	6.18e-6	3.45e-6	2.88e-6	2.74e-6
		(1.00)	(0.312)	(0.217)	(0.197)
	cpu	190.0	185.0	182.0	219.0
	$\frac{\text{eff.}}{10^4}$	13765	45382	66319	60682
		(1.00)	**(3.30)**	**(4.82)**	**(4.41)**

Table 4: $s = 60,\ N = 4093,\ a = 209.$

K		MC	LR	Sobol	GFaure
90	$\hat{\mu}$	12.179	12.179	12.179	12.179
	$\hat{\sigma}$	6.92e-5	3.92e-5	5.83e-5	4.46e-5
		(1.00)	(0.321)	(0.710)	(0.414)
	cpu	963.0	922.0	905.0	1039.0
	$\frac{\text{eff.}}{10^4}$	21.7	70.5	32.5	48.5
		(1.00)	**(3.25)**	**(1.50)**	**(2.24)**
100	$\hat{\mu}$	4.826	4.826	4.826	4.826
	$\hat{\sigma}$	5.04e-5	2.77e-5	4.22e-5	3.50e-5
		(1.00)	(0.301)	(0.699)	(0.481)
	cpu	963.0	921.0	905.0	1055.0
	$\frac{\text{eff.}}{10^4}$	40.8	141.7	62.2	77.4
		(1.00)	**(3.47)**	**(1.52)**	**(1.90)**
110	$\hat{\mu}$	1.192	1.192	1.192	1.192
	$\hat{\sigma}$	3.75e-5	2.48e-5	3.07e-5	2.70e-5
		(1.000)	(0.437)	(0.671)	(0.520)
	cpu	964.0	918.0	904.0	1051.0
	$\frac{\text{eff.}}{10^4}$	73.7	177.1	117.2	130.0
		(1.00)	**(2.40)**	**(1.59)**	**(1.77)**

sometimes even less successful than MC. The advantage of the lattice rules over MC, in this case, decreases more slowly with s than the advantage of Sobol or GFaure.

5 CONCLUSION

We gave numerical examples in which lattice rules are efficient estimators. When no other variance reduction technique is used for the problem of asian options, the lattice rules estimator easily beats the MC estimator, sometimes by factors as large as 500. This advantage decreases as the dimension s increases and N decreases, but is still present for s as large as 120 and $N = 509$.

The use of antithetic variates and of the price of the asian option on the geometric average as a control variable improves the MC estimator by very large factors, but lattice rules still provide an additional improvement. Compared with the Sobol and generalized Faure estimators, the lattice estimator loses in small dimensions and when the number of points is large. But it dominates in larger dimension $(s \geq 60)$. Also, when N is small, the two other QMC estimators hardly bring variance reduction compared with the MC estimator.

The simplicity of implementation of lattice rules estimators and the fact that they seem to suffer less from large dimensions than the Sobol or generalized Faure sequences estimators should encourage adepts of QMC methods to use lattice rules more often to estimate multidimensional integrals.

ACKNOWLEDGMENTS

We thank Anargyros Papageorgiou and Joseph Traub for providing the FINDER software. This work has been supported by NSERC-Canada via a scholarship to the first author and grants # ODGP0110050 and # SMF0169893 to the second author.

REFERENCES

Acworth, P., M. Broadie., and P. Glasserman. 1997. A comparison of some Monte Carlo and quasi-Monte Carlo techniques for option pricing. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, ed. P. Hellekalek and H. Niederreiter, Lecture Notes in Statistics 127, 1–18. Springer-Verlag.

Black, F. and M. Scholes. 1973. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81:637–659.

Bratley, P., B. L. Fox., and H. Niederreiter. 1992. Implementation and tests of low-discrepancy sequences. *ACM Transactions on Modeling and Computer Simulation*, 2:195–213.

Cranley, R. and T. N. L. Patterson. 1976. Randomization of number theoretic methods for multiple integration. *SIAM Journal on Numerical Analysis*, 13(6):904–914.

Fox, B. L. 1986. Implementation and relative efficiency of quasirandom sequence generators. *ACM Transactions on Mathematical Software*, 12:362–376.

Hickernell, F. J. 1998. A generalized discrepancy and quadrature error bound. *Mathematics of Computation*, 67:299–322.

Hickernell, F. J. 1999. Lattice rules: How well do they measure up? In *Pseudo- and Quasi-Random Point Sets*, ed. P. Hellekalek and G. Larcher, Lecture Notes in Statistics. New York: Springer. To appear.

Kemna, A. G. Z. and C. F. Vorst. 1990. A pricing method for options based on average asset values. *Journal of Banking and Finance*, 14:113–129.

L'Ecuyer, P. 1994. Efficiency improvement via variance reduction. In *Proceedings of the 1994 Winter Simulation Conference*, 122–132. IEEE Press.

L'Ecuyer, P. 1998a. Good parameters and implementations for combined multiple recursive random number generators. *Operations Research*. To appear.

L'Ecuyer, P. 1998b. A table of linear congruential generators of different sizes and good lattice structure. *Mathematics of Computation*. To appear.

Lemieux, C. 1996. L'évaluation des options asiatiques. Master's thesis, Université de Montréal, Département de Mathématiques.

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

Owen, A. B. 1995. Randomly permuted (t, m, s)-nets and (t, s)-sequences. In *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, ed. H. Niederreiter and P. J.-S. Shiue, Lecture Notes in Statistics 106, 299–317. Springer-Verlag.

Owen, A. B. 1997. Scrambled net variance for integrals of smooth functions. *Annals of Statistics*, 25(4):1541–1562.

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

Papageorgiou, A. and J. Traub. 1996. Beating Monte Carlo. *Risk*, 9:63–65.

Paskov, S. and J. Traub. Fall 1995. Faster valuation of financial derivatives. *Journal of Portfolio Management*, 113–120.

Sloan, I. H. and S. Joe. 1994. *Lattice Methods for Multiple Integration*. Oxford: Clarendon Press.

Sloan, I. H. and L. Walsh. 1990. A computer search of rank 2 lattice rules for multidimensional quadrature. *Mathematics of Computation*, 54:281–302.

Tuffin, B. 1996. On the use of low-discrepancy sequences in Monte Carlo methods. Technical Report No. 1060, I.R.I.S.A., Rennes, France.

AUTHOR BIOGRAPHIES

CHRISTIANE LEMIEUX is currently a Ph.D. student in the “Département d'Informatique et de Recherche Opérationnelle”, at the University of Montreal. She works on efficiency improvement by lattice rules. She completed her M.Sc. in mathematics (actuarial science) at the same university in 1996.

PIERRE L'ECUYER is a professor in the “Département d'Informatique et de Recherche Opérationnelle”, at the University of Montreal. He received a Ph.D. in operations research in 1983, from the University of Montreal. >From 1983 to 1990, he was with the Computer Science Department, at Laval University, Quebec. He obtained the *E. W. R. Steacie* grant from the Natural Sciences and Engineering Research Council of Canada for the period 1995–97. His main research interests are random number generation, efficiency improvement via variance reduction, sensitivity analysis and optimization of discrete-event stochastic systems, and discrete-event simulation in general. He was the Departmental Editor for the Simulation Department of *Management Science* from 1993 to 1998 and is an Area Editor for the *ACM Transactions on Modeling and Computer Simulation*. You can find more at: `www.iro.umontreal.ca/~lecuyer`.