On the Scrambled Sobol' Sequence

Hongmei Chi[1], Peter Beerli[2], Deidre W. Evans[1], and Micheal Mascagni[2]

[1] Department of Computer and Information Sciences,
Florida A&M University, Tallahassee, FL 32307-5100
hchi@cis.famu.edu
[2] School of Computational Science and Information Technology,
Florida State University, Tallahassee, FL 32306-4120

Abstract. The Sobol' sequence is the most popular quasirandom sequence because of its simplicity and efficiency in implementation. We summarize aspects of the scrambling technique applied to Sobol' sequences and propose a new simpler modified scrambling algorithm, called the multi-digit scrambling scheme. Most proposed scrambling methods randomize a single digit at each iteration. In contrast, our multi-digit scrambling scheme randomizes one point at each iteration, and therefore is more efficient. After the scrambled Sobol' sequence is produced, we use this sequence to evaluate a particular derivative security, and found that when this sequence is numerically tested, it is shown empirically to be far superior to the original unscrambled sequence.

1 Introduction

The use of quasirandom, rather than random, numbers in Monte Carlo methods, is called quasi–Monte Carlo methods, which converge much faster than normal Monte Carlo. Quasi–Monte Carlo methods are now widely used in scientific computation, especially in estimating integrals over multidimensional domains and in many different financial computations.

The Sobol' sequence [21, 22] is one of the standard quasirandom sequences and is widely used in quasi–Monte Carlo applications. The efficient implementation of Sobol' sequence uses Gray codes. We summarize aspects of this technique applied to Sobol' sequences and propose a new scrambling algorithm, called a multiple digit scrambling scheme. Most proposed scrambling methods [1, 8, 16, 19] randomized a single digit at each iteration. In contrast, our multi-digit scrambling scheme, which randomizes one point at each iteration, is efficient and fast because the popular modular power-of-two pseudorandom number generators are used to speed it up.

The construction of the Sobol' sequence uses linear recurrence relations over the finite field, $\mathbb{F}_2$, where $\mathbb{F}_2 = \{0, 1\}$. Let the binary expansion of the nonnegative integer n be given by $n = n_1 2^0 + n_2 2^1 + ... + n_w 2^{w-1}$. Then the nth element of the jth dimension of the Sobol' sequence, $x_n^{(j)}$, can be generated by

$$x_n^{(j)} = n_1 \nu_1^{(j)} \oplus n_2 \nu_2^{(j)} \oplus ... \oplus n_w \nu_w^{(j)} \tag{1}$$

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3516, pp. 775–782, 2005.

where $\nu_i^{(j)}$ is a binary fraction called the ith direction number in the jth dimension. These direction numbers are generated by the following q-term recurrence relation

$$\nu_i^{(j)} = a_1\nu_{i-1}^{(j)} \oplus a_2\nu_{i-2}^{(j)} \oplus ...a_q\nu_{i-q+1}^{(j)} \oplus \nu_{i-q}^{(j)} \oplus (\nu_{i-q}^{(j)}/2^q). \tag{2}$$

We have $i > q$, and the bit, a_i, comes from the coefficients of a degree-q primitive polynomial over $\mathbb{F}_2$. Note that one should use a different primitive polynomial to generate the Sobol' direction numbers in each different dimension. Another representation of $\nu_i^{(j)}$ is to use the integer $m_i^{(j)} = \nu_i^{(j)} * 2^i$. Thus, the choice of q initial direction numbers $\nu_i^{(j)}$ becomes the problem of choosing q odd integers $m_i^{(j)} < 2^i$. The initial direction numbers, $\nu_i^{(j)} = \frac{m_i^{(j)}}{2^i}$, in the recurrence, where $i \leq q$, can be decided by the $m_i^{(j)}$'s, which can be arbitrary odd integers less than 2^i. The Gray code is widely used in implementations [4, 11] of the Sobol' sequence.

The direction numbers in Sobol' sequences come recursively from a degree-q primitive polynomial; however, the first q direction numbers can be arbitrarily assigned for the above recursion (equation (2)). Selecting them is crucial for obtaining high-quality Sobol' sequences. The top pictures in both Fig. 1 and Fig. 2 show that different choices of initial direction numbers can make the Sobol' sequence quite different. The initial direction numbers for the top picture in figure (1) is from Bratley and Fox's paper [4]; while top picture in figure (2) results when the initial direction numbers are all ones.

Sobol' [22] realized the importance of initial direction numbers, and published an additional property (called Property A) for direction numbers to produce more uniform Sobol' sequences; but implementations [11] of Sobol' sequences showed that Property A is not really that useful in practice. Cheng and Druzdzel [5, 20] developed an empirical method to search for initial direction numbers, $m_i^{(j)}$, in a restricted space. Their search space was limited because they had to know the total number of quasirandom numbers, N, in advance to use their method. Jackel [10] used a random sampling method to choose the initial $m_i^{(j)}$ with a uniform random number u_{ij}, so that $m_i^{(j)} = \lfloor u_{ij} \times 2^{i-1} \rfloor$ for $0 < i < q$ with the condition that $m_i^{(j)}$ is odd.

Owing to the arbitrary nature of initial direction numbers of the sequence, poor two-dimensional projections frequently appear in the Sobol' sequence. Morokoff and Caflisch [18] noted that poor two-dimensional projections for the Sobol' sequence can occur anytime because of the improper choices of initial direction numbers. The bad news is that we do not know in advance which initial direction numbers cause poor two-dimensional projections. In other words, poor two-dimensional projections are difficult to prevent by trying to effectively choose initial direction numbers. Fortunately, scrambling Sobol' sequences [8, 19] can help us improve the quality of the Sobol' sequence having to pay attention to the proper choice of the initial direction numbers.

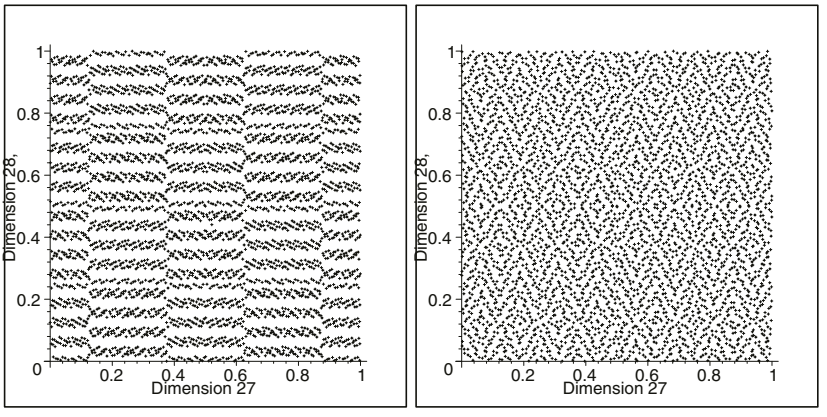

Fig. 1. Left: 4096 points of the original Sobol' sequence and the initial direction numbers are from Bratley and Fox's paper [4]; right: 4096 points of the scrambled version of the Sobol' sequence

2 Scrambling Methods

Recall that Sobol' sequence is defined over the finite field, $\mathbb{F}_2$ [13]. Digit permutation is commonly thought effective in the finite field, $\mathbb{F}_p$. When digit permutation is used to scramble a quasirandom point over $\mathbb{F}_p$, the zero is commonly left out. The reason is that permuting zero (assuming an infinite string of trailing zeros) leads to a biased sequence in the sense that zero can be added to the end of any sequence while no other digit can. So this strategy for pure digital permutation, where zero is not changed, is not suitable for the Sobol' sequence because the Sobol' sequence is over $\mathbb{F}_2$. For example, we could write 0.0101 as 0.01010000 if we want to scramble 8 digits. If zero is left out, the scrambled results for 0.0101 and 0.01010000 are same. Otherwise, the bias may be introduced.

The linear permutation [8] is also not a proper method for scrambling the Sobol' sequence. Let $x_n = (x_n^{(1)}, x_n^{(2)}, \ldots, x_n^{(s)})$ be any quasirandom number in $[0,1)^s$, and $z_n = (z_n^{(1)}, z_n^{(2)}, \ldots, z_n^{(s)})$ be the scrambled version of the point x_n. Suppose that each $x_n^{(j)}$ has a b-ary representation as $x_n^{(j)} = 0.x_{n1}^{(j)} x_{n2}^{(j)} ... x_{nK}^{(j)} ...$, where K defines the number of digits to be scrambled in each point. Then we define

$$z_n^{(j)} = c_1 x_n^{(j)} + c_2, \text{for } j = 1, 2, .., s, \tag{3}$$

where $c_1 \in \{1, 2, ..., b-1\}$ and $c_2 \in \{0, 1, 2, ..., b-1\}$. Since the Sobol' sequence is built over $\mathbb{F}_2$, one must assign 1 to c_1 and 0 or 1 to c_2. Since the choice of c_1 is crucial to the quality of the scrambled Sobol' sequence, this linear scrambling method is not suitable for the Sobol' sequence or any sequence over $\mathbb{F}_2$.

As stated previously, the quality of the Sobol' sequence depends heavily on the choices of initial direction numbers. The correlations between different dimensions are due to improper choices of initial direction numbers [5]. Many

methods [5, 10] to improve the Sobol' sequence focus on placing more uniformity into the initial direction numbers; but this approach is difficult to judge by any measure. We concentrate on improving the Sobol' sequence independent of the initial direction numbers. This idea motivates us to find another approach to obtain high-quality Sobol' sequences by means of scrambling each point.

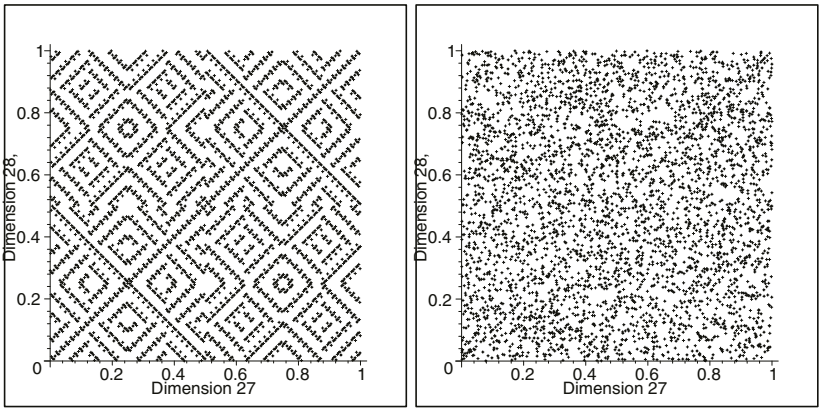

Fig. 2. Left: 4096 points of the original Sobol' sequence with all initial direction numbers ones [23], right: 4096 points of the scrambled version of the Sobol' sequence

3 An Algorithm for Scrambling the Sobol' Sequence

We provide a new approach for scrambling the Sobol' sequence, and measure the effectiveness of this approach with the number theoretic criterion that we have used in [6]. Using this new approach, we can now scramble the Sobol' sequence in any number of dimensions.

The idea of our algorithm is to scramble k bits of the Sobol' sequence instead of scrambling one digit at a time. The value of k could be any positive integer as long as we could find a suitable Linear Congruential Generators (LCG) for it. Assume x_n is nth Sobol' point, and we want to scramble first k bits of x_n. Let z_n be the scrambled version of x_n. Our procedure is described as follows:

1. $y_n = \lfloor x_n * 2^k \rfloor$, is the k most-significant bits of x_n, to be scrambled.
2. $y_n^* = ay_n \pmod m$ and $m \geq 2^k - 1$, is the linear scrambling, applied to this integer.
3. $z_n = \frac{y_n^*}{2^k} + (x_n - \frac{y_n}{2^k})$, is the reinsertion of these scrambled bits into the Sobol' point.

The key step of this approach is based on using LCGs as scramblers. LCGs with both power-of-two and prime moduli are common pseudorandom number generators. When the modulus of an LCG is a power-of-two, the implementation is cheap and fast due to the fact that modular addition and multiplication are

just ordinary computer arithmetic when the modulus corresponds to a computer word size. The disadvantage, in terms of quality, is hard to obtain the desired quality of pseudorandom numbers when using a power-of-two as modulus. More details are given in [14, 15]. So LCGs with prime moduli are chosen in this paper.

The rest of our job is to search for a suitable and reliable LCG as our scrambler. When the modulus of a LCG is prime, implementation is more expensive. A special form of prime, such as a Merssene[1] or a Sophie-Germain prime[2], can be chosen so that the costliest part of the generation, the modular multiplication, can be minimized [15].

To simplify the scrambling process, we look to LCGs for guidance. Consider the following LCG:

$$y_n^* = ay_n \pmod{m}, \tag{4}$$

where m is chosen to be a Merssene, 2^k-1, or Sophie-Germain prime in the form of $2^{k+1}-k_0$, k is the number of bits needed to "scramble", and a is a primitive root modulo m [12, 7]. We choose the modulus to be a Merssene or Sophie-Germain [15] because of the existence of a fast modular multiplication algorithms for these primes. The optimal a should generate the optimal Sobol' sequence, and the optimal a's for modulus $2^{31}-1$ are tabulated in [7]. A proposed algorithm for finding such optimal primitive root modulus m, a prime, is described [6].

Primarily, our algorithm provides a practical method to obtain a family of scrambled Sobol' sequences. Secondarily, it gives us a simple and unified way to generate an optimal Sobol' sequence from this family. According to Owen's proof [19], after scrambling, the Sobol' sequence is still a (t, s)-net with base 2. However, using our algorithm, we can begin with the worse choices for initial direction numbers in the Sobol' sequence: all initial direction numbers are ones. The results are showed in Fig.2. The only unscrambled portion is a straight line in both pictures. The reason is that the new scrambling algorithm cannot change the point with the same elements into a point with different elements.

4 Geometric Asian Options

Here, we present the valuation of a complex option, which has a simple analytical solution. The popular example for such problems is a European call option on the geometric mean of several assets, sometimes called a geometric Asian option. Let K be the strike price at the maturity date, T. Then the geometric mean of N assets is defined as

$$G = (\prod_{i=1}^{N} S_i)^{\frac{1}{N}},$$

where S_i is the ith asset price. Therefore the payoff of this call option at maturity can be expressed as $\max(0,\ G-K)$. Boyle [3] proposed an analytical solution

[1] If 2^q-1 and q are primes, then 2^q-1 is a Merssene prime.

[2] If $2q+1$ and q are primes, then $2q+1$ is a Sophie-Germain prime.

for the price of a geometric Asian option. The basic idea is that the product of lognormally distributed variables is also lognormally distributed. This property results because the behavior of an asset price, S_i, follows geometric Brownian motion [2]. The formula for using the Black-Scholes equation [2, 9] to evaluate a European call option can be represented as

$$\begin{aligned} C_T &= S * Norm(d_1) - K * e^{-r(T-t)} * Norm(d_2), \\ \text{with } d_1 &= \frac{ln(S/K) + (r + \sigma^2)(T-t)}{\sigma\sqrt{T-t}}, \\ d_2 &= d_1 - \sigma\sqrt{T-t}, \end{aligned} \tag{5}$$

where t is current time, r is a risk-free rate of interest, which is constant in the Black-Scholes world, and Norm(d_2) is the cumulative normal distribution. Because the geometric Asian option has an analytical solution, we have a benchmark to compare our simulation results with analytical solutions. The parameters used for our numerical studies are as follows:

Number of assets	N
Initial asset prices, $S_i(0)$	100, for $i = 1, 2, ..., N$
Volatilities, σ_i	0.3
Correlations, ρ_{ij}	0.5, for $i < j$
Strike price, K	100
Risk-free rate, r	10%
Time to maturity, T	1 year

The formula for computing the analytic solution for a geometric Asian option is computed by a modified Black-Scholes formula. Using the Black-Scholes formula, we can compute the call price by using equation (5) with modified parameters, S and σ^2, as follows:

$$\begin{aligned} S &= Ge^{(-A/2+\sigma^2/2)T} \\ A &= \frac{1}{N}\sum_{i=1}^{N}\sigma_i^2 \\ \sigma^2 &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{i=j}^{N}\rho_{ij}\sigma_i\sigma_j. \end{aligned} \tag{6}$$

We followed the above formula in equation (5) and (6), computed the prices for different values of $N = 10$ and $N = 30$, with $K = 100$, and computed $p = 12.292$ and $p = 12.631$ respectively. For each simulation, we had an analytical solution, so we computed the relative difference between that and our simulated solution with the formula $\frac{|p_{qmc}-p|}{p}$, where p is the analytical solution and p_{qmc} is the price obtained by simulation. For different N, we computed p_{qmc} by simulating the asset price fluctuations using geometric Brownian motion. The results are shown in Fig.3.

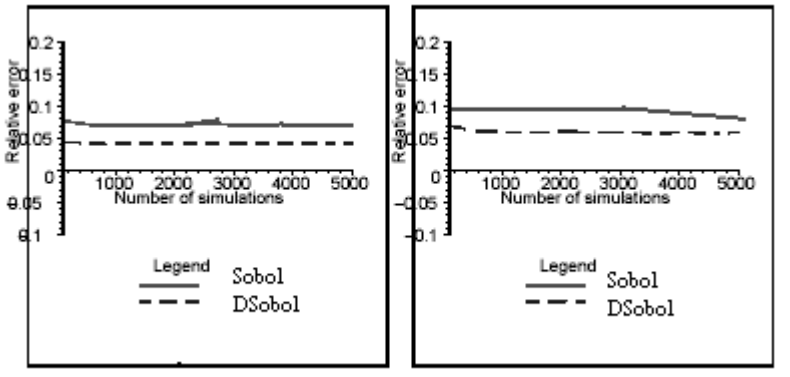

Fig. 3. Left: geometric mean of 10 stock prices; right: geometric mean of 30 stock prices. Here the label "Sobol" refers to the original Sobol' sequence [4], while "DSobol" refers to our optimal Sobol' sequence

From equation (5), we can see that random variables are drawn from a normal distribution. Each Sobol' point must be transformed into a normal variable. The favored transformation method for quasirandom numbers is the inverse of the cumulative normal distribution function. The inverse normal function provided by Moro [17] was used in our numerical studies. From Fig. 3, it is easily seen that the optimal Sobol' sequence performs much better than the original Sobol' sequence.

5 Conclusions

A new algorithm for scrambling the Sobol' sequence is proposed. This approach can avoid the consequences of improper choices of initial direction numbers that negatively impact the quality of this sequence. Therefore, our approach can enhance the quality of the Sobol' sequence without worrying about the choices of initial direction numbers. In addition, we proposed an algorithm and found an optimal Sobol' sequence within the scrambled family. We applied this sequence to evaluate a complex security and found promising results even for high dimensions. We have shown the performance of the Sobol' sequence generated by our new algorithm empirically to be far superior to the original sequence. The promising results prompt us to use more applications to test the sequences, and to reach for more general scrambling techniques for the Sobol' sequence.

References

1. E. Atanassov. A new efficient algorithm for generating the scrambled sobol' sequence. In *Numerical Methods and Applications (LNCS 2542)*, pages 81–90, New York, 2003. Springer-Verlag.

2. F. Black and M. Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, **81**:637–659, 1973.
3. P. Boyle. New life forms on the option landscape. *Journal of Financial Engineering*, **2**(3):217–252, 1992.
4. P. Bratley and B. Fox. Algorithm 659: Implementing soboĺ's quasirandom sequnence generator. *ACM Trans. on Mathematical Software*, **14**(1):88–100, 1988.
5. J. Cheng and M.J. Druzdzel. Computational investigation of low-discrepancy sequences in simulation algorithms for bayesian networks. In *Uncertainty in Artificial Intelligence: Proceedings of the Sixteenth Conference (UAI-2000)*, pages 72–81, San Francisco, CA, 2000. Morgan Kaufmann Publishers.
6. H. Chi, M. Mascagni, and T. Warnock. On the optimal Halton sequences. *Mathematics and Computers in Simulation*, To appear, 2005.
7. G. A. Fishman and L. R. Moore. An exhaustive analysis of multiplicative congruential random number generators with modulus $2^{31} - 1$. *SIAM J. Sci. Stat. Comput.*, **7**:24–45, 1986.
8. H. S. Hong and F. J. Hickernell. Algorithm 823: Implementing scrambled digital sequences. *ACM Transactions on Mathematical Software*, **29**(2):95–109, june 2003.
9. J. Hull. *Options, Future and Other Derivative Secutrities*. Prentice-Hall, New York, 2000.
10. P. Jackel. *Monte Carlo Methods in Finance*. John Wiley and Sons, New York, 2002.
11. S. Joe and F. Y. Kuo. Remark on Algorithm 659: Implementing Soboĺ's quasirandom sequence generator. *ACM Transactions on Mathematical Software*, **29**(1):49–57, March 2003.
12. D. E. Knuth. *The Art of Computer Programming, vol. 2: Seminumerical Algorithms*. Addison-Wesley, Reading, Massachusetts, 1997.
13. R. Lidl and H.Niederreiter. *Introduction to Finite Fields and Their Applications*. Cambridge University Press, Cambridge, 1994.
14. M. Mascagni. Parallel linear congruential generators with prime moduli. *Parallel Computing*, **24**:923–936, 1998.
15. M. Mascagni and H. Chi. Parallel linear congruential generators with Sophie-Germain moduli. *Parallel Computing*, **30**:1217–1231, 2004.
16. J. Matousek. On the l2-discrepancy for anchored boxes. *Journal of Complexity*, **14**:527–556, 1998.
17. B. Moro. The full monte. *Risk*, **8**(2) (February):57–58, 1995.
18. W.J. Morokoff and R.E. Caflish. Quasirandom sequences and their discrepancy. *SIAM Journal on Scientific Computing*, **15**:1251–1279, 1994.
19. A.B. Owen. Randomly permuted(t,m,s)-netsand (t,s)-sequences. *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, **106** in Lecture Notes in Statistics:299–317, 1995.
20. S. H. Paskov and J. F. Traub. Faster valuation of financial derivatives. *J. Portfolio Management*, **22**(1):113–120, Fall 1995.
21. I.M. Soboĺ. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Comput. Math. and Math. Phy.*, **7**(4):86–112, 1967.
22. I.M. Soboĺ. Uniformly distributed sequences with additional uniformity properties. *USSR Comput. Math. and Math. Phy.*, **16**:236–242, 1976.
23. S. Tezuka. *Uniform Random Numbers, Theory and Practice*. Kluwer Academic Publishers, IBM Japan, 1995.