A new permutation choice in Halton sequences

Bruno TUFFIN[1]

ABSTRACT This paper has several folds. We make first new permutation choices in Halton sequences to improve their distributions. These choices are multi-dimensional and they are made for two different discrepancies. We show that multi-dimensional choices are better for standard quasi-Monte Carlo methods. We also use these sequences as a variance reduction technique in Monte Carlo methods, which greatly improves the convergence accuracy of the estimators. For this kind of use, we observe that one-dimensional choices are more efficient.

1 Introduction

Quasi-Monte Carlo methods are deterministic analogs of Monte Carlo ones. For the latters, convergence is in $O(1/\sqrt{N})$ for an approximation with N random points. It is possible to construct a sequence where the points are deterministic and "well distributed" all over the integration space, for which the convergence speed is faster (in $O(N^{-1}(\log N)^s)$ for dimension s). Halton sequences verify this property. In this paper we give new permutation choices for Halton sequences to improve their distribution. Next, as the major problem encountered with quasi-Monte Carlo methods is the error bound evaluation, we use the sequences in Monte Carlo methods to obtain a variance reduction. In this case, we observe that a one-dimensional choice of permutations is more efficient.

This paper is organized as follows: Section 2 describes quasi-Monte Carlo methods and Halton sequences and section 3 proposes a new choice of permutations in Halton sequences. Section 4 describes the use of such sequences as a variance reduction in Monte Carlo methods and explain why a one-dimensional choice is better for this kind of use. Finally we conclude in Section 5.

[1]IRISA, Campus de Beaulieu, 35042 Rennes cdex, France

2 Quasi-Monte Carlo methods

2.1 General method

Let us consider the integration of functions on $[0,1)^s$. The objective of the method is to approximate $\int_{[0,1)^s} f(u)du$ by $\frac{1}{N}\sum_{n=1}^{N} f(\xi^{(n)})$, where $(\xi^{(n)})_{n\in\mathbb{N}}$ is a deterministic sequence. Define a measure of uniform distribution over $[0,1)^s$. Let $\mathcal{P} = (\xi^{(n)})_{n\in\mathbb{N}}$ and define $A_N(z,\mathcal{P}) = \sum_{n=1}^{N} 1_{[0,z)}(\xi^{(n)})$ where $z = (z_1,\cdots,z_s) \in [0,1]^s$ and $[0,z) = \prod_{i=1}^{s}[0,z_i)$. The discrepancy in space L^p of the N first terms of $\mathcal{P}$ is defined by

$$T_N^{(p)*}(\mathcal{P}) = \left(\int_{[0,1)^s} \left|\frac{A_N([0,z),\mathcal{P})}{N} - \lambda_s([0,z))\right|^p dz\right)^{1/p}.$$

This discrepancy is an expression of the mean difference between the frequency of points of $\mathcal{P}$ and the measure of each interval of form $[0,x)$. The sequence is uniformly distributed if and only if $\lim_{n\to+\infty} T_N^{(p)*}(\mathcal{P}) = 0$. An expression of $T_N^{(2)*}(\mathcal{P})$ (see [3]) is given by

$$(T_N^{(2)*}(\mathcal{P}))^2 = \frac{1}{N^2}\sum_{k,m=1}^{N}\prod_{i=1}^{s}(1-M_i^{k,m}) - \frac{2^{1-s}}{N}\sum_{k=1}^{N}\prod_{i=1}^{s}(1-\xi_i^{(k)2}) + 3^{-s} \quad (1.1)$$

where $\xi_i^{(n)}$ is the i^{th} coordinate of the vector $\xi^{(n)}$ and $M_i^{k,m} = \max(\xi_i^{(k)}, \xi_i^{(m)})$.

There exist error bounds involving this discrepancy (see [12], [3]).

Sequences with $T_N^{(p)*}(\mathcal{P}) = O(N^{-1}(\log N)^s)$ are called *low discrepancy sequences*.

A new notion of discrepancy, $T_N^{(2)}(\mathcal{P})$, is described by Morokoff and Caflish in [3]. It is shown that the computation of $T_N^{(2)*}(\mathcal{P})$ takes more into account the points $x \in [0,1)^s$ near the origin $0 = (0,\cdots,0)$ than the others. The definition is the following: let $x < y$ denote the inequality for each coordinate of vectors, i.e. $x_i < y_i$ $(i = 1,\cdots s)$. Then

$$T_N^{(2)}(\mathcal{P}) = \left[\int_{x,y\in[0,1)^s; x<y} \left|\frac{A_N([x,y),\mathcal{P})}{N} - \lambda_s([x,y))\right|^2 dxdy\right]^{1/2}.$$

With this new definition, all the points of the space $[0,1)^s$ have the same importance. It is proven in [3] that, if $m_i^{k,m} = \min(\xi_i^{(k)}, \xi_i^{(m)})$,

$$(T_N^{(2)}(\mathcal{P}))^2 = \frac{1}{N^2}\sum_{k,m=1}^{N}\prod_{i=1}^{s}(1-M_i^{k,m})m_i^{k,m} - \frac{2^{1-s}}{N}\sum_{k=1}^{N}\prod_{i=1}^{s}(1-\xi_i^{(k)})\xi_i^{(k)} + 12^{-s}. \quad (1.2)$$

Unfortunately, even if this definition is more representative of equi-distribution, there exists yet no error bound using this discrepancy.

There exist many examples of low discrepancy sequences [4]. In this paper, we focus on an important family, Halton sequences, and their properties.

2.2 Halton sequences and improvements

Let $p \in \mathbb{N}$. Let us denote the digit expansion of $n \in \mathbb{N}$ in base p by

$$n = a_j p^j + \cdots + a_1 p + a_0 \text{ with } a_i \in \{0, \cdots, p-1\}.$$

The radical-inverse function of n is then defined by

$$\Phi_p(n) = a_0/p + a_1/p^2 + \cdots + a_j/p^{j+1}.$$

If $p_1, \cdots, p_s$ are s mutually prime integers, the sequence $\mathcal{P} = (\xi^{(n)})_{n \in \mathbb{N}}$ defined by

$$\xi^{(n)} = (\Phi_{p_1}(n), \cdots, \Phi_{p_s}(n)),$$

is a Halton sequence and it verifies

$$T_N^{(2)*}(\mathcal{P}) = O(N^{-1}(\log(N))^s). \tag{1.3}$$

In practice, we will always let p_i equal to the i^{th} prime number. In spite of the asymptotic low discrepancy of such sequences, it is observed in [1], [3], [8] that, in a large dimension, a good distribution needs many iterations to occur (i.e. we obtain bad distributions for a small number of iterations). As a matter of fact, the monotone cycles of length p_i for the i^{th} projection introduce a great regularity between coordinates. An important research effort has been made to improve this. In [3] the sequence is scrambled independently for each coordinate: for the s sequences (corresponding to the s coordinates), the N points are ranged from the smallest to the largest and then they are randomly permuted. It is possible to show that this operation do not change the bound on the discrepancy given by (1.3).

Braaten and Weller [1] have improved the distribution of the sequence with the introduction, for each prime number p_i $(1 \leq i \leq s)$, of a permutation π_{p_i} on $\{0, \cdots, p_i - 1\}$ satisfying $\pi_{p_i}(0) = 0$ and leading to a more chaotic sequence. In this case again, for each choice of permutation the convergence speed (1.3) remains unchanged. This scrambling gives sequences that are the most efficient for certain dimensions and for certain applications [6].

If, from the digit expansion of n in base p_i, we set

$$S_{p_i}(n) = \pi_{p_i}(a_0)/p_i + \pi_{p_i}(a_1)/p_i^2 + \cdots + \pi_{p_i}(a_j)/p_i^{j+1},$$

the new Halton sequence is $\mathcal{P} = (\xi^{(n)})_{n \in \mathbb{N}}$ with $\xi^{(n)} = (S_{p_1}(n), \cdots, S_{p_s}(n))$. The problem is to choose good permutations. Braaten and Weller [1] have built the permutations π_{p_i} as follows. For each p_i, π_{p_i} is chosen in an unidimensional way: if we know $\pi_{p_i}(1), \cdots, \pi_{p_i}(j)$, we choose $\pi_{p_i}(j+1)$ as the element minimizing the mean square discrepancy $T_{j+1}^{(2)*}$ of the $j+1$ points $\{\pi_{p_i}(1)/p_i, \cdots, \pi_{p_i}(j)/p_i, \pi_{p_i}(j+1)/p_i\}$.

Although this choice gives good results, we construct here four multi-dimensional algorithms which should give better results than Braaten and Weller's one, which is built uni-dimensionally.

3 New choice of permutations

Two algorithms, MCT* and MCL*, based on $T_N^{(2)*}(\mathcal{P})$, are already described in [8] (where they are called respectively MC1 and MC2). We recall them here and give two new choices, called MCT and MCL, based on $T_N^{(2)}(\mathcal{P})$. We will use expression (1.2), instead of (1.1) in [8], to make the choice of permutations in Halton sequences. The algorithms we generate are the same as those in [8], but for $T_N^{(2)}(\mathcal{P})$ instead of $T_N^{(2)*}(\mathcal{P})$.

3.1 Algorithm MCL

As for Braaten and Weller's one, this method for the choice of permutations gives a table which is available for any dimension. But in our case, the table is generated line per line instead of element per element. Thus the whole $(j+1)^{th}$ line is chosen knowing the j previous lines. As the only restriction for the j^{th} permutation is $\pi_{p_j}(0) = 0$, there are $(p_j - 1)!$ possibilities. For example, for $j = 10$, we have $28! = 3.049e29$ choices. Since it is impossible to compute at each time the discrepancy, we make our choice in a random manner.

Given a K-sample $(\pi^{(1)}, \cdots, \pi^{(K)})$ of permutations π of $\{0, \cdots, p_{j+1} - 1\}$ such that $\pi(0) = 0$, an estimator of $\pi_{p_{j+1}}$ is the permutation which minimizes $\{T^{(2)}_{p_{j+1}-1}(\mathcal{P}_{\pi^{(k)}}) | 1 \leq k \leq K\}$ for $p_{j+1}-1$ points. That is, $T^{(2)}_{p_{j+1}-1}(\mathcal{P}_\pi) = \min_{1\leq k\leq K} T^{(2)}_{p_{j+1}-1}(\mathcal{P}_{\pi^{(k)}})$ where $\mathcal{P}_\pi$ is the sequence in dimension $j+1$ issued from π (with permutations $\pi_{p_1}, \cdots, \pi_{p_j}$ fixed for the j first coordinates). We call this algorithm MCL. The algorithm with the same technique, but for $T_N^{(2)*}(\mathcal{P})$ is called MCL* [8].

3.2 Algorithm MCT

Let us show now an algorithm creating a whole table for each dimension: in a fixed dimension s, we search for a table $(\pi_{p_1}, \cdots, \pi_{p_s})$, with π_{p_i} permutation of $\{0, \cdots, p_i - 1\}$ such that $\pi_{p_i}(0) = 0$ and π_{p_i} minimizes $T^{(2)}_{p_s-1}(\mathcal{P}_{p_1,\cdots,p_s})$ where $\mathcal{P}_{p_1,\cdots,p_s}$ is the sequence in dimension s associated with permutations $(\pi_{p_1}, \cdots, \pi_{p_s})$. As the number of possible permutations is even larger than before, we use again a random approach.

Let

$$\Omega_s = \{(\pi_1, \cdots, \pi_s) \mid \quad \forall 1 \leq i \leq s \ \pi_i \text{ is a permutation of } \{0, \cdots, p_i - 1\} \text{ satisfying } \pi_i(0) = 0\}.$$

Let Π be a random variable uniformly distributed on Ω_s and

$$(\pi_1^{(k)}, \cdots, \pi_s^{(k)})_{1 \leq k \leq K}$$

a K-sample from Π, with $\pi_i^{(k)}$ i^{th} coordinate (i.e. permutation) of k^{th} variable $\pi^{(k)}$. An estimator of $(\pi_{p_1}, \cdots, \pi_{p_s})$ is the table of permutations $(\pi_1^{(j)}, \cdots, \pi_s^{(j)})$ such that

$$T^{(2)}_{p_s-1}(\mathcal{P}^j_{p_1,\cdots,p_s}) = \min_{1 \leq k \leq K} T^{(2)}_{p_s-1}(\mathcal{P}^k_{p_1,\cdots,p_s}),$$

with $\mathcal{P}^k_{p_1,\cdots,p_s}$ the sequence corresponding to permutations $(\pi_1^{(k)}, \cdots, \pi_s^{(k)})$. We call this algorithm MCT. The algorithm with the same techniques, but for $T_N^{(2)*}(\mathcal{P})$ is called MCT* in [8].

3.3 Results

Dim	Halton	B W	MCT* (#iter.)	MCL* (#iter.)
2	4.34e-2	4.86e-2	4.34e-2	4.34e-2
3	1.67e-2	9.72e-3	8.33e-3	1.35e-2
4	6.97e-3	6.15e-3	3.02e-3	4.51e-3
5	2.49e-3	1.42e-3	9.54e-4 (5 10^6)	1.16e-3 1.79e-3
6	1.91e-3	6.79e-4	4.58e-4 (10^6)	5.85e-4 (10^6)
7	1.18e-3	2.78e-4	1.79e-4 (10^6)	1.82e-4 (10^6)
8	9.86e-4	1.81e-4	7.95e-5 (10^6)	8.06e-5 (10^6)
9	6.76e-4	1.03e-4	3.13e-5 (10^6)	3.10e-5 (10^6)
10	4.18e-4	1.76e-5	1.16e-5 (10^6)	1.09e-5 (10^6)
11	3.63e-4	1.13e-5	4.59e-6 (5 10^5)	4.25e-6 (5 10^5)
12	2.46e-4	5.34e-6	1.75e-6 (2 10^5)	1.64e-6 (5 10^5)
13	1.94e-4	2.34e-6	6.43e-7 (10^5)	6.16e-7 (5 10^5)
14	1.71e-4	1.60e-6	2.46e-7 (6 10^5)	2.27e-7 (2 10^5)
15	1.38e-4	4.07e-7	8.87e-8 (2 10^5)	7.87e-8 (2 10^5)
16	1.04e-4	2.11e-7	3.31e-8 (3.5 10^5)	2.73e-8 (2 10^5)

TABLE I. $(T^{(2)*}_{p_s-1})^2$ for the different algorithms in dimension s.

The square of discrepancies $(T^{(2)*}_{p_s-1}(\mathcal{P}))^2$ obtained for each method and each dimension are given in Table I and for $(T^{(2)}_{p_s-1}(\mathcal{P}))^2$ in Table II. The number of iterations used (K in the previous subsections) is indicated after the value of discrepancy when we are not sure to obtain the real minimum. For algorithms MCL* and MCL, it indicates the number of iterations for the choice of the line. We can make the following remarks on both Tables: the improvements of our algorithms increases with the dimension with respect to standard Halton and Braaten and Weller permutations (due to the multi-dimensional heuristic). Moreover, the best results are given by algorithm

Dim	Halton	MCT (#iter.)	MCL (#iter.)
2	1.39e-2	1.39e-2	1.39e-2
3	1.15e-3	9.93e-4	1.13e-3
4	1.39e-4	1.16e-4	1.22e-4
5	1.13e-5	8.85e-6 ($5\ 10^6$)	9.16e-6
6	1.62e-6	1.15e-6 (10^6)	1.11e-6 (10^6)
7	1.83e-7	1.15e-7 (10^6)	1.11e-7 (10^6)
8	2.81e-8	1.55e-8 (10^6)	1.36e-8 (10^6)
9	3.59e-9	1.89e-9 (10^6)	1.78e-9 (10^6)
10	4.03e-10	2.08e-10 (10^6)	1.75e-10 (10^6)
11	6.26e-11	3.09e-11 ($5\ 10^5$)	2.60e-11 (10^6)
12	7.57e-12	3.48e-12 ($2\ 10^5$)	2.83e-12 ($5\ 10^5$)
13	1.07e-12	4.28e-13 ($2\ 10^5$)	3.26e-13 ($5\ 10^5$)
14	1.75e-13	6.96e-14 ($6\ 10^5$)	4.37e-14 ($2\ 10^5$)
15	2.66e-14	9.34e-15 ($3\ 10^5$)	5.25e-15 ($2\ 10^5$)
16	3.73e-15	1.31e-15 ($3.5\ 10^5$)	6.46e-16 ($2\ 10^5$)

TABLE II. $(T^{(2)}_{p_s-1})^2$ for the different algorithms.

MCL* for $(T^{(2)*}_{p_s-1}(\mathcal{P}))^2$ and MCL for $(T^{(2)}_{p_s-1}(\mathcal{P}))^2$, although it should be given by respectively MCT* and MCT (as they are algorithms which, for a sufficiently large number of iterations, give the smallest discrepancy). Then it should be better to use the permutations given by MCL* or MCL, and it should be more practical because we get only one table for every dimension. Furthermore, the larger the dimension, the longer the cycles dependence between successive coordinates in standard Halton sequences [8]. If we compare the discrepancies for large dimensions (in this way, if the number of points $p_s - 1$ is large and the dimension s is small, standard Halton gives good results), then our improvements are significant.

Let us compare in Table III, on the example of the function in dimension 16 defined by $\prod_{i=1}^{16} 12(x_i - 1/2)^2$, the approximation of the integral $\int_{[0,1)^{16}} \prod_{i=1}^{16} 12(x_i - \frac{1}{2})^2 dx_i$, which value is 1, given by $\frac{1}{N}\sum_{n=1}^{N}\prod_{i=1}^{16} 12(\xi_i^{(n)} - \frac{1}{2})^2$. We see that Braaten and Weller's permutation choice improves the quality of the approximation in comparison with Halton's one, and that ours improve the one of Braaten and Weller. In this case, for a standard quasi-Monte Carlo method, the best choices are given by algorithms MCL and MCT.

It.	Halton	B W	MCT*	MCL*	MCT	MCL
10^3	0.48939	1.69686	0.76792	0.18115	0.29996	0.65553
10^4	0.42182	2.95772	0.53146	1.11605	0.64943	1.29340
10^6	0.90964	1.06985	0.95899	1.05608	1.00286	1.02135

TABLE III. Test on function $\prod_{i=1}^{16} 12(x_i - 0.5)^2$ for the different algorithms.

4 On the use of low discrepancy sequences in Monte Carlo methods

Unfortunately, the known error bounds are generally impossible to evaluate in practice. Then, to obtain a useful error bound, we use low discrepancy sequences to reduce variance in Monte Carlo methods.

Let X be a random variable uniformly distributed on $[0,1)^s$ and $(\xi^{(k)})_{k\in\mathbb{N}}$ a low discrepancy sequence as described in the previous section. Instead of simulating the random variable $f(X)$, we study

$$Z = \frac{1}{n}\sum_{k=1}^{n} f(\{X + \xi^{(k)}\}), \tag{1.4}$$

where $\{x\}$ is the fractional part for each coordinate of $x \in \mathbb{R}^s$. To our knowledge, this type of technique has been used for the first time in [2] where $(\xi^{(k)})_{k\leq n}$ is a lattice developed by Korobov and then for Bayesian integration in [7] with low discrepancy sequences. Owen [5] uses a slightly different technique, where the randomness is introduced on permutations for Niederreiter sequences. In our applications, we will use the permuted Halton sequences described in the previous section.

Theorem 1 *[9] If $(\xi^{(k)})_{k\in\mathbb{N}}$ is a low discrepancy sequence and f a (bounded) Riemann integrable function, we have*

$$\sigma^2\left(\frac{1}{n}\sum_{k=1}^{n} f(\{X + \xi^{(k)}\})\right) = O(n^{-2}(\log n)^{2s}).$$

Then, for a sufficiently large n, we are sure to obtain a variance reduction with respect to n independent random variables $f(X)$, which variance is in n^{-1}. Efficient applications of this method to the analysis of product-form multi-class queuing networks and of a cellular system with dynamic resource sharing can be found respectively in [11] and in [10].

To compare permutation choices for this kind of method, we use the function $f(x_1, \cdots, x_{12}) = \prod_{i=1}^{12} \frac{\pi}{2}\sin(\pi x_i)$. We take $n = 10^4$ elements of the low discrepancy sequences and make $I = 100$ independent iterations of the random variable to estimate the variance. We compare in Table IV the variance of this estimator for each permutation choice with the one of the standard Monte Carlo estimator for 10^6 iterations, to have the same number of calls to the function.

Method	Variance
Monte Carlo	1.130e-2
Halton	7.228e-4
B& W	1.645e-5
MCL*	7.188e-5
MCT*	3.361e-5
MCL	1.019e-4
MCT	6.753e-5

TABLE IV. Variance of different estimators of $\int_{[0,1)^s} \prod_{i=1}^{12} \frac{\pi}{2} \sin(\pi x_i) dx_i$: the Monte Carlo one and those using permuted Halton sequences as a variance reduction.

An important remark is that the best choice for this application is the one-dimensional sequence of Braaten and Weller (except for MCT, but our experiments tell us that this is marginal). As a matter of fact, the introduction of an additional term in (1.4) breaks the multi-dimensional building of the sequence. Another remark is that the Braaten and Weller sequence generally outperforms the $(0, s)$ sequence of Niederreiter, usually used in quasi-Monte Carlo methods, for dimensions smaller than 10 (see [9]).

5 Conclusion

We give here new permutation choices in Halton sequences. The new choices are based on $T_N^{(2)}$ or on $T_N^{(2)*}$ and are multi-dimensional, which is an advantage with respect to previous proposals. These choices are better in a standard quasi-Monte Carlo integration. We also use low-discrepancy sequences as variance reduction techniques in Monte Carlo methods. Whereas classical Monte Carlo algorithms do not change the convergence speed, ours does it. For this utilization of Quasi-Monte Carlo in a Monte Carlo scheme (and because of the additional term) a one dimensional choice for the permutations is recommended. It is commonly known that quasi-Monte Carlo techniques give better accuracy than Monte Carlo ones. Nevertheless in practice, to obtain an error bound, a combination of Monte Carlo and quasi-Monte Carlo methods can be efficiently used.

6 References

[1] E. Braaten and G. Weller. An Improved Low-Discrepancy Sequence for Multidimensional Quasi-Monte Carlo Integration. *J. Comput. Phys.*, 33:249–258, 1979.

[2] R. Cranley and T.N.L. Patterson. Randomization of number theoretic methods for multiple integration. *SIAM J. Numer. Anal.*, 13(6):904–

914, December 1976.

[3] W. J. Morokoff and R. E. Caflisch. Quasi-Random Sequences and their Discrepancies. *SIAM Journal on Scientific Computing*, pages 1571–1599, December 1994.

[4] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*. CBMS-SIAM 63, Philadelphia, 1992.

[5] A. B. Owen. Randomly permuted (t, m, s)-nets and (t, s)-sequences. In Harald Niederreiter and Peter Jau-Shyong Shiue, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, volume 106 of *Lecture Notes in Statistics*, pages 299–315. Springer, 1995.

[6] G. Pagès and Y.J. Xiao. Sequences with low discrepancy and pseudo-random numbers: theoretical results and numerical tests. *to appear in J. Statist. Comput. Simulat.*, 1996.

[7] J. E. H. Shaw. A quasirandom approach to integration in Bayesian statistics. *Ann. Statist.*, 16:895–914, 1988.

[8] B. Tuffin. Improvement of Halton sequences distribution. Technical Report 998, IRISA, March 1996.

[9] B. Tuffin. On the Use of Low Discrepancy Sequences in Monte Carlo Methods. *Monte Carlo Methods and Applications*, 2(4):295–320, 1996.

[10] B. Tuffin. Variance reduction technique for a cellular system with dynamic resource sharing. In *Proceedings of the 10th European Simulation Multiconference*, pages 467–471, Budapest, June 1996.

[11] B. Tuffin. Variance Reductions applied to Product-Form Multi-Class Queuing Network: antithetic variates and low discrepancy sequences. Technical Report 1005, IRISA, April 1996.

[12] S. K. Zaremba. Some applications of multidimensionnal integration by parts. *Ann. Pol. Math*, XXI:85–96, 1968.