Good permutations for scrambled Halton sequences in terms of L_2-discrepancy

Bart Vandewoestyne
Ronald Cools

Report TW 406, September 2004

Katholieke Universiteit Leuven
Department of Computer Science
Celestijnenlaan 200A – B-3001 Heverlee (Belgium)

Good permutations for scrambled Halton sequences in terms of L_2-discrepancy

Bart Vandewoestyne
Ronald Cools

Report TW406, September 2004

Department of Computer Science, K.U.Leuven

Abstract

One of the best known low-discrepancy sequences, used by many practitioners, is the Halton sequence. Unfortunately, there seems to exist quite some correlation between the points from the higher dimensions. A possible solution to this problem is the so-called *scrambling*.

In this paper, we give an overview of known scrambling methods, and we propose a new way of scrambling which gives good results compared to the others in terms of L_2-discrepancy. On top of that, our new scrambling method is very easy to generate.

Keywords : Halton sequence, low-discrepancy sequences, numerical integration, (quasi)-Monte Carlo, scrambling.
AMS(MOS) Classification : 11K06, 11K36, 11K38.

Good permutations for scrambled Halton sequences in terms of L_2-discrepancy

Bart Vandewoestyne [a] and Ronald Cools [a]

[a] *Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan 200A, B-3001 Heverlee, Belgium*

Abstract

One of the best known low-discrepancy sequences, used by many practitioners, is the Halton sequence. Unfortunately, there seems to exist quite some correlation between the points from the higher dimensions. A possible solution to this problem is the so-called *scrambling*.

In this paper, we give an overview of known scrambling methods, and we propose a new way of scrambling which gives good results compared to the others in terms of L_2-discrepancy. On top of that, our new scrambling method is very easy to generate.

Key words: Halton sequence, low-discrepancy sequences, numerical integration, (quasi)-Monte Carlo, scrambling.

1 Introduction

The problem we consider is the numerical calculation of the integral of a function f over the unit cube:

$$I = \int_{[0,1)^s} f(\mathbf{u})\, \mathrm{d}\mathbf{u}. \tag{1}$$

For not so small values of s, this problem is often tackled by a Monte Carlo method. An approximation of the integral is obtained from the values of the function at random points $\mathbf{x}_i$ that lie in $[0, 1)^s$:

Email addresses: `Bart.Vandewoestyne@cs.kuleuven.ac.be` (Bart Vandewoestyne), `Ronald.Cools@cs.kuleuven.ac.be` (Ronald Cools).

URLs: `http://www.cs.kuleuven.ac.be/~bartv/` (Bart Vandewoestyne), `http://www.cs.kuleuven.ac.be/~ronald/` (Ronald Cools).

$$\hat{I} = \frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i). \tag{2}$$

This estimator $\hat{I}$ of I is unbiased if the $\mathbf{x}_i$ are independent samples of the uniform distribution over $[0,1)^s$. Its variance is

$$\mathrm{Var}(\hat{I}) = \frac{\sigma^2}{N} = \frac{1}{N}\left(\int_{[0,1)^s} [f(\mathbf{x}) - I]^2 \, \mathrm{d}\mathbf{x}\right) = \frac{1}{N}\left(\int_{[0,1)^s} (f(\mathbf{x}))^2 \, \mathrm{d}\mathbf{x} - I^2\right),$$

which leads to the well-known probabilistic error bound of $O(N^{-1/2})$ for square integrable functions f.

Quasi-Monte Carlo methods provide an alternative to the Monte Carlo methods by using deterministic point sets instead of random ones. These so-called *quasi-random point sets* are specially 'crafted' to fill the s-dimensional unit cube in a more uniform way. When using estimator (2) for a function f with bounded variation $V_{HK}(f)$ in the sense of Hardy and Krause, a classical error bound is given by the *Koksma-Hlawka inequality* [7]:

$$|\hat{I} - I| \leq V_{HK}(f) D_N^*(\mathbf{x}_1, \ldots, \mathbf{x}_N). \tag{3}$$

Here $D_N^*(\mathbf{x}_1, \ldots, \mathbf{x}_N)$ represents the *star-discrepancy* of the point set. This will be considered in more detail in the following section.

The above inequality suggests that sequences for which the discrepancy is lower than that of a random sequence are likely to give better integration results than plain Monte Carlo.

In the next section, we explain the concept of discrepancy and we show how it can be calculated. Section 3 then introduces the Halton sequence as one particular type of quasi-random point sequence. Scrambling, as a technique to increase the properties of the Halton sequence, will be explained in section 4. We also give an overview of existing scrambling techniques, compare their effectiveness and introduce a new type of scrambling. Conclusions are formulated in section 5.

2 Discrepancy

Let P be a point set $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \subset [0,1)^s$. For an arbitrary subset B of $[0,1)^s$, define $A(B;P)$ as the function that counts the number of points from P that fall inside B and put

$$R(B;P) = \frac{A(B;P)}{N} - \lambda_s(B),$$

where λ_s is the s-dimensional Lebesgue measure. A general notion of discrepancy is then given by

$$D_N(\mathcal{B}; P) = \sup_{B \in \mathcal{B}} |R(B; P)|. \tag{4}$$

Depending on the family $\mathcal{B}$ of subsets of $[0,1)^s$, several types of discrepancies can be considered. In this text we will use D_N^*, D_N, T_N^* and T_N defined below.

Definition 1
The L_∞ star discrepancy $D_N^*(P) = D_N(\mathcal{J}^*; P)$, *where* $\mathcal{J}^*$ *is the family of all subintervals of* $[0,1)^s$ *of the form* $\prod_{i=1}^s [0, u_i)$.

Definition 2
The L_∞ (extreme) discrepancy $D_N(P) = D_N(\mathcal{J}; P)$, *where* $\mathcal{J}$ *is the family of all subintervals of* $[0,1)^s$ *of the form* $\prod_{i=1}^s [u_i, v_i)$.

Unfortunately, both these discrepancies are hard to calculate and even trying to compute bounds for them requires a lot of work [12,9]. A more practical type of discrepancy appears if the L_∞-norm in (4) is replaced by the L_2-norm:

$$T_N(\mathcal{B}; P) = \left[\int_{[0,1)^s} R(\mathcal{B}; P)^2 \, \mathrm{d}\mathbf{u} \right]^{\frac{1}{2}}.$$

Again, depending on the family $\mathcal{B}$, we can define several types of discrepancy.

Definition 3
The L_2 star discrepancy $T_N^*(P) = T_N(\mathcal{J}^*; P)$, *where* $\mathcal{J}^*$ *is the family of all subintervals of* $[0,1)^s$ *of the form* $\prod_{i=1}^s [0, u_i)$.

Definition 4
The L_2 (extreme) discrepancy $T_N(P) = T_N(\mathcal{J}; P)$, *where* $\mathcal{J}$ *is the family of all subintervals of* $[0,1)^s$ *of the form* $\prod_{i=1}^s [u_i, v_i)$.

Denoting by $x_k^{(i)}$ the i-th component of the point $\mathbf{x}_k$, Warnock [15] showed that

$$[T_N^*(P)]^2 = \frac{1}{N^2} \sum_{k=1}^N \sum_{m=1}^N \prod_{i=1}^s \left(1 - \max(x_k^{(i)}, x_m^{(i)})\right) - \frac{2^{1-s}}{N} \sum_{k=1}^N \prod_{i=1}^s \left(1 - {x_k^{(i)}}^2\right) + 3^{-s}. \tag{5}$$

He also proposed a way to calculate $T_N^*(P)$, starting from $T_{N-1}^*(P)$.

Following the same approach as Warnock, Morokoff and Caflish [6] show that

$$
\begin{aligned}
[T_N(P)]^2 = & \frac{1}{N^2} \sum_{k=1}^{N} \sum_{m=1}^{N} \prod_{i=1}^{s} \left[1 - \max(x_k^{(i)}, x_m^{(i)})\right] \cdot \min(x_k^{(i)}, x_m^{(i)}) \\
& - \frac{2^{1-s}}{N} \sum_{k=1}^{N} \prod_{i=1}^{s} \left(1 - x_k^{(i)}\right) x_k^{(i)} + 12^{-s}. \qquad (6)
\end{aligned}
$$

In the same paper, they also give the root-mean-square (rms) expectation for T_N and T_N^* as

$$\langle T_N^*(P)^2 \rangle = ((2^{-s} - 3^{-s})/N)^{1/2}, \qquad (7)$$

$$\langle T_N(P)^2 \rangle = (6^{-s}(1 - 2^{-s})/N)^{1/2}. \qquad (8)$$

We will use formulas (5) and (6) to compare the quality of different low-discrepancy sequences. The reader should be aware that the computation of $T_N^*(P)$ and $T_N(P)$ for a set of N points in s dimensions requires $O(sN^2)$ operations. Furthermore, both formulas are ill-conditioned [16].

3 The Halton sequence

One of the best known low-discrepancy sequences, used by many practitioners, is the Halton sequence. Let $b \geq 2$ be an integer, then any integer $n \geq 0$ can be written in the form

$$n = d_0 + d_1 b + d_2 b^2 + \cdots + d_j b^j, \qquad 0 \leq d_i < b. \qquad (9)$$

The radical inverse function $\phi_b(n)$ for base b is defined by

$$\phi_b(n) = \frac{d_0}{b} + \frac{d_1}{b^2} + \cdots + \frac{d_j}{b^{j+1}}. \qquad (10)$$

The *van der Corput* sequence in base b is defined as the one-dimensional point set $\{\phi_b(n)\}_{n=0}^{\infty}$. Halton [3] extends this definition to the s-dimensional sequence $\{\mathbf{x}_i\}$, defining

$$\mathbf{x}_n = (\phi_{b_1}(n), \ldots, \phi_{b_s}(n)), \qquad n = 0, 1, \ldots$$

The integers $b_1, \ldots, b_n$ are greater than one and pairwise prime. Most of the time, they are chosen as the first s primes.

One of the advantages of the Halton sequence is its ease of implementation. Depending on the practical efficiency and accuracy needed, a practitioner can choose from a variety of existing implementations, e.g. [4,5,11].

A disadvantage of the Halton sequence can easily be seen from figures 1 and 2 which show 100 points of a 20-dimensional Halton sequence, projected onto

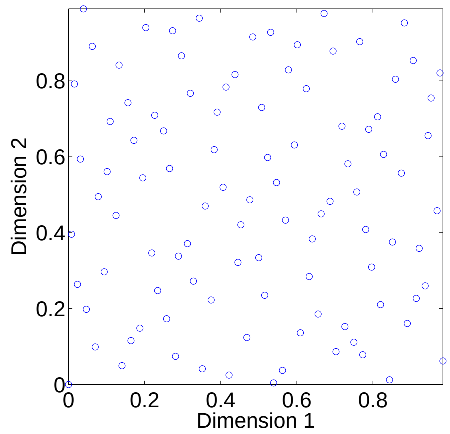


Fig. 1. 1st and 2nd dimension

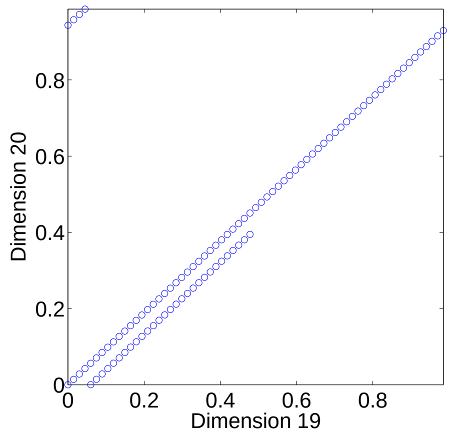


Fig. 2. 19th and 20th dimension

different coordinate planes. The projection onto the plane belonging to dimensions 1 and 2 (primes 2 and 3) is much more uniform than the one belonging to dimensions 19 and 20 (primes 67 and 71). Halton sequences are indeed known to have more correlation between points generated from higher primes. This may deteriorate the performance dramatically in applications.

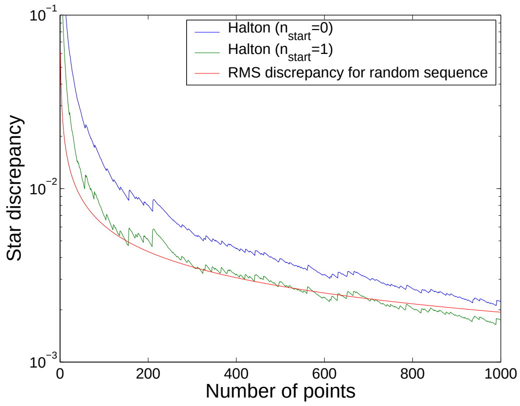


Fig. 3. L_2 star discrepancy ($s = 8$)

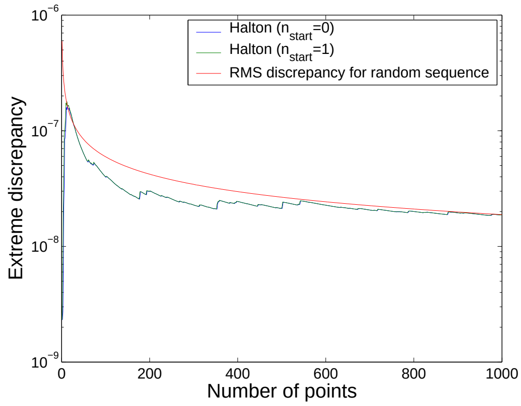


Fig. 4. L_2 extreme discrepancy ($s = 16$)

The L_2-star discrepancy of the first 1000 points of the Halton sequence is shown in figure 3. The smooth curve represents the expected root-mean-square discrepancy for a random sequence (7). The upper line is the Halton sequence starting at index $n_{\text{start}} = 0$ and thus including the origin, while the lower line represents the Halton sequence starting at $n_{\text{start}} = 1$. The figure clearly illustrates the influence of the origin on the star discrepancy. Excluding the origin gives better results. We have also plotted the extreme discrepancy in figure 4. Again here, the smooth curve represents the root-mean-square discrepancy for a random sequence. We can see that the extreme discrepancy seems not really influenced by the origin as both curves almost fall on top of each other. Therefore, some authors state that it might also be good to consider the extreme discrepancy instead of the star discrepancy when comparing sequences.

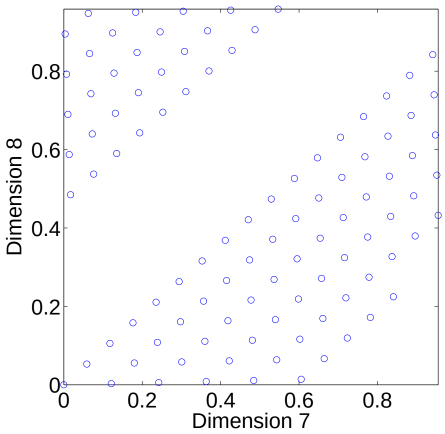


Fig. 5. 100 Halton points

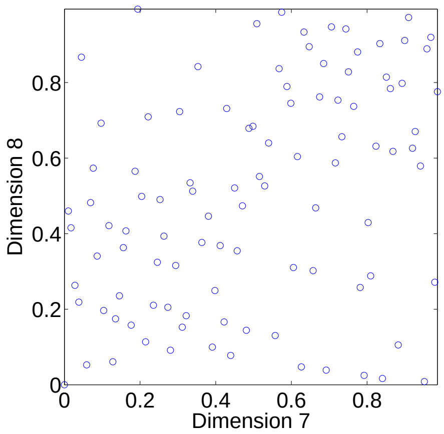


Fig. 6. 100 points from the BW-sequence

In our further comparisons concerning the discrepancy of point sets, we excluded the origin.

4 Scrambling the Halton sequence

The correlation between points of the Halton sequence can be broken by scrambling the digits of the sequence in a way that preserves the low-discrepancy properties. This was first formally described by Braaten and Weller [1], who defined the *scrambled radical inverse function* $S_b(n)$ in analogy with (10) as:

$$S_b(n) = \frac{\pi_b(d_0)}{b} + \frac{\pi_b(d_1)}{b^2} + \cdots + \frac{\pi_b(d_j)}{b^{j+1}}. \tag{11}$$

Here π_b is a permutation on the digits $(0, 1, \ldots, b-1)$ which holds 0 fixed. Their scrambled Halton sequence is then given by

$$\mathbf{x}_n = (S_{b_1}(n), \ldots, S_{b_s}(n)), \qquad n = 0, 1, \ldots \tag{12}$$

Figures 5 and 6 show how the permutations break up the correlation between the coordinates of the Halton sequence. The first figure shows the unscrambled projection of dimension 7 and 8 of an 8-dimensional Halton sequence, while the second figure shows the scrambled version.

In the next subsections, we give a historical overview of the different scramblings that have been proposed in the literature. We also present a new choice for the permutations which gives good results compared to the others and is far more easier to generate.

4.1 Warnock's folded radical inverse

Warnock [15] used the *folded radical inverse* function

$$\psi_b(n) = \frac{(d_0 + 0) \bmod b}{b} + \frac{(d_1 + 1) \bmod b}{b^2} + \cdots + \frac{(d_j + j) \bmod b}{b^{j+1}} + \cdots$$

instead of (10) to define a scrambled version of the Halton sequence. For example $\psi_3(11) = 0.210\overline{012}_3 = 551/702$ where the overlined part represents an infinitely repeated sequence of digits. His version of the Halton sequence, which we will denote by 'WA1-sequence', then becomes

$$\mathbf{x}_n = (\psi_{b_1}(n), \ldots, \psi_{b_s}(n)), \qquad n = 0, 1, \ldots$$

4.2 The permutations of Braaten and Weller

Braaten and Weller [1] used algorithm 1 to define the π_b from (11). They tabulated their permutations up to the first 16 primes. Figure 6 was obtained using these permutations. We call the resulting sequence 'BW-sequence'. In [1], the discrepancy plots for 8, 12 and 16-dimensional sequences demonstrate that the BW-sequence has a lower star-discrepancy than the Halton sequence for the first 1000 points. The higher the dimension, the better the effect of the BW-sequence seems to be. Braaten and Weller did not show however what happens beyond these 1000 points.

Figure 7 shows the discrepancy for 10000 points in 8, 12 and 16 dimensions. For 8 dimensions, we see that the discrepancy is indeed lower for about the first 5000 points. From then on, both discrepancies are comparable. For 12 and 16 dimensions, the BW-sequence performs better, but the difference between the discrepancies decreases with the number of points.

From figure 7 and similar experiments, we learn that the advantage of the BW-sequence increases with the dimension. For dimensions lower than about 8 its L_2-star discrepancy is quite comparable to that of the Halton sequence.

Algorithm 1 Permutation search of Braaten and Weller

$\pi_b(0) = 0$
ChoiceSet $= \{1, \ldots, b-1\}$
for $i = 1$ to $b - 1$ **do**
  choose $\pi_b(i)$ from ChoiceSet so that it minimizes the one-dimensional discrepancy $T_N^*\left(\{\frac{\pi_b(1)}{b}, \ldots, \frac{\pi_b(i)}{b}\}\right)$.
  ChoiceSet := ChoiceSet$\setminus\{\pi_b(i)\}$
end for

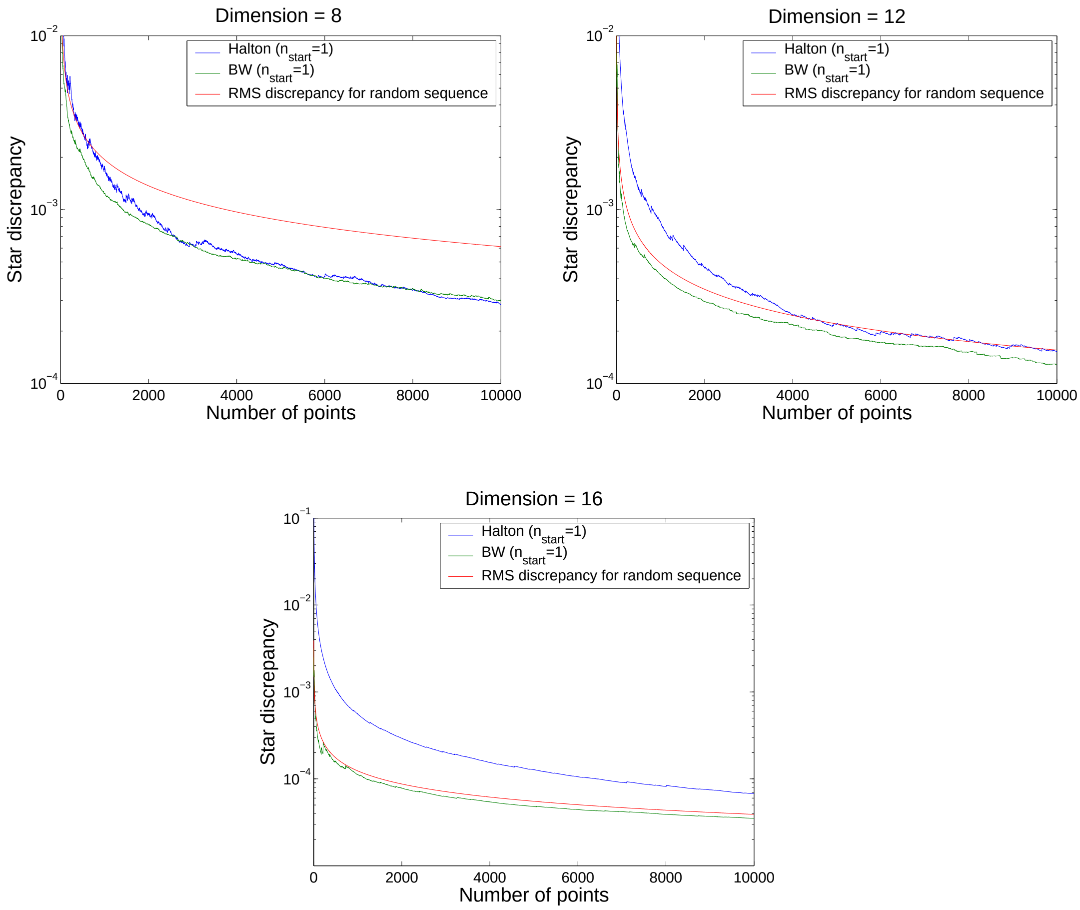


Fig. 7. L_2 star discrepancy of BW in 8, 12 and 16 dimensions for 10000 points

Furthermore, the larger N, the more the discrepancy of the BW-sequence approaches the discrepancy of the Halton sequence.

As Braaten and Weller have only published their permutations up to the first 16 primes, it is not really known how well their scrambled sequence behaves for dimensions over 16. Therefore, we have implemented the algorithm of Braaten and Weller in Matlab in order to compute permutations for dimensions higher than 16. The first 16 permutations found by our implementation are listed in table 1.

The differences between our table and the one in [1] are underlined and can be explained by the fact that sometimes in algorithm 1, a tie arises between the discrepancies of $\{\frac{\pi_b(1)}{b}, \ldots, \frac{\pi_b(i)}{b}\}$ for the choice of $\pi_b(i)$. If such a tie arises, we just picked the smallest $\pi_b(i)$ that was left in ChoiseSet. Another reason for the differences comes from the note below the table of permutations in [1] which reminds the reader to the fact that for each permutation, one can generate another permutation with the same minimal discrepancy by replacing each nonzero integer n by $p - n$. We did not underline this last kind of differences

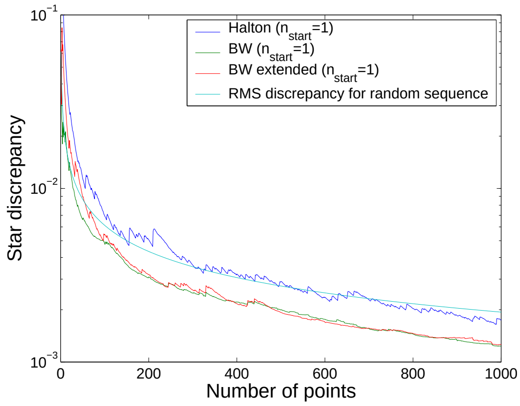
Table 1
Permutations found by our own implementation of Braaten and Weller's algorithm

π_2	(0 1)
π_3	(0 2 1)
π_5	(0 2 4 1 3)
π_7	(0 3 5 1 6 2 4)
π_{11}	(0 5 8 2 10 3 6 1 9 4 7)
π_{13}	(0 6 10 2 8 4 12 1 9 5 11 3 7)
π_{17}	(0 8 13 3 11 5 16 1 10 7 14 4 12 2 15 6 9)
π_{19}	(0 9 14 3 17 6 11 1 15 7 12 4 18 8 2 16 10 5 13)
π_{23}	(0 11 17 4 20 7 13 2 22 9 15 5 18 1 14 10 21 6 16 3 19 8 12)
π_{29}	(0 14 22 5 18 9 27 2 20 11 25 7 16 3 24 13 19 6 28 10 1 23 15 12 26 4 17 8 21)
π_{31}	(0 16 8 26 4 22 13 29 2 19 11 24 6 20 14 28 1 17 9 30 10 23 5 21 15 3 27 12 25 7 18)
π_{37}	(0 18 28 6 23 11 34 3 25 14 31 8 20 36 1 16 27 10 22 13 32 4 29 17 7 35 19 2 26 12 30 9 24 15 33 5 21)
π_{41}	(0 20 31 7 26 12 38 3 23 34 14 17 29 5 40 10 24 1 35 18 28 9 33 15 21 4 37 13 30 8 39 19 25 2 32 11 22 36 6 27 16)
π_{43}	(0 21 32 7 38 13 25 3 35 17 28 10 41 5 23 30 15 37 1 19 33 11 26 42 8 18 29 4 39 14 22 34 6 24 12 40 2 31 20 16 36 9 27)
π_{47}	(0 23 35 8 41 14 27 3 44 18 31 11 37 5 25 39 16 21 33 1 46 12 29 19 42 7 28 10 36 22 4 43 17 32 13 38 2 26 45 15 30 6 34 20 40 9 24)
π_{53}	(0 26 40 9 33 16 49 4 36 21 45 12 29 6 51 23 38 14 43 1 30 19 47 10 34 24 42 3 27 52 15 18 39 7 46 22 32 5 48 13 35 25 8 44 31 17 50 2 37 20 28 11 41)

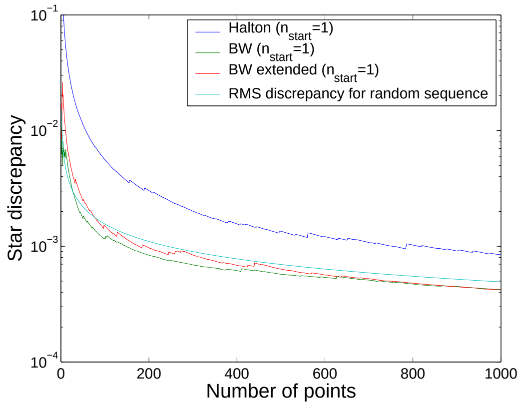
in our table.

Figures 8a, 8b and 8c show how a sequence with our permutations has a similar performance as the BW-sequence. In figures 8d and 8e, we see that the sequence with permutations as being generated by our implementation of algorithm 1 remains better than the original Halton sequence, but the discrepancy is still worse than the rms discrepancy for a random sequence (7). The curve with lowest discrepancy comes from a new permutation choice and will be explained in section 4.6.

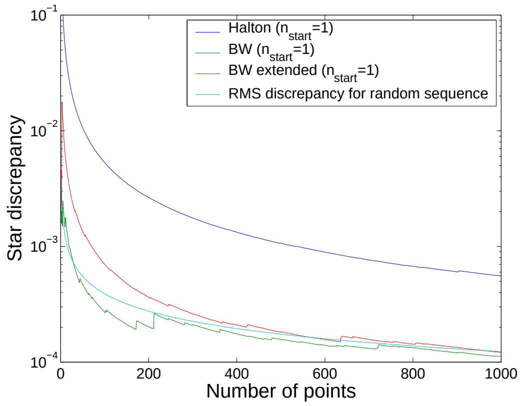
Finally, note that the technique of Braaten and Weller requires a large computational effort for determining the permutations because of the calculation



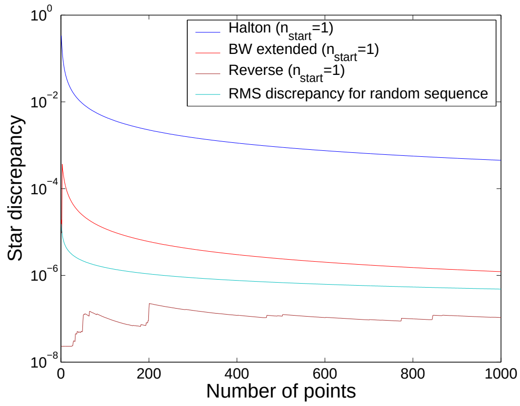
(a) 8 dim



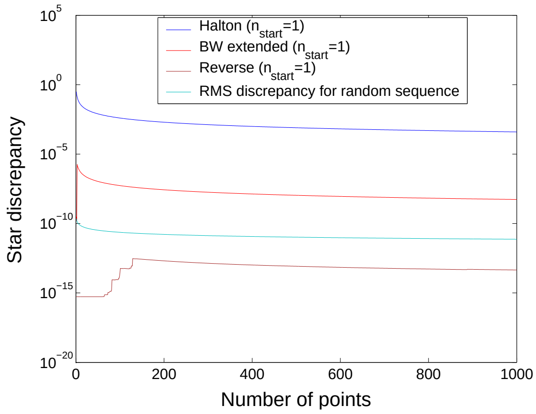
(b) 12 dim



(c) 16 dim



(d) 32 dim



(e) 64 dim

Fig. 8. T_N^* of our own implementation of Braaten and Weller their algorithm for several dimensions.

of many one-dimensional discrepancies. A simple calculation shows that in order to calculate the permutation for prime b, the number of maxima (max), multiplications (M) and additions (A) to compute is

$$\sum_{k=2}^{b-1} k\left(\left[(b-k)^2\right] \max + \left[(b-k)+4\right] M + \left[(b-k)^2 + 4(b-k) - 1\right] A\right).$$

4.3 Faure's algorithm for constructing the permutations

In a search for good permutations for the one-dimensional Van der Corput sequence, Faure [2] proposed algorithm 2 to create the permutations π_b. Although his paper only mentions the one-dimensional case, practitioners are also experiencing good results in using his permutations to scramble the Halton sequence [8,10]. Using Faure's algorithm, one can easily find the permutations from table 2. In this paper, we denote this type of scrambling 'FAU-scrambling' and call the scrambled Halton sequence obtained with it 'FAU-sequence'.

Algorithm 2 Faure's algorithm to determine π_b

```
b = 2
π_b = (0, 1)
repeat
  b = b + 1
  if b is even then
    π_b = (2π_{b/2}, 2π_{b/2} + 1)
  else
    η = π_{b−1}
    k = (b−1)/2
    add 1 to each element of η which is ≥ k
    π_b = η with k added in the middle
  end if
until all necessary permutations are found
```

4.4 Warnock's PhiCf sequence

More recently, Warnock [16] combined the initial behaviour of the Weyl sequence [17] with the asymptotic behaviour of the Halton sequence to construct what he called the *PhiCf sequence.* He replaces each d_i in (9) with $S(b)d_i \bmod b$ to obtain a new kind of radical inverse function

$$\omega_b(n) = \frac{(S(b)d_0) \bmod b}{b} + \frac{(S(b)d_1) \bmod b}{b^2} + \cdots + \frac{(S(b)d_j) \bmod b}{b^{j+1}},$$

Table 2
Permutations by Faure

π_2	(0 1)
π_3	(0 1 2)
π_4	(0 2 1 3)
π_5	(0 3 2 1 4)
π_6	(0 2 4 1 3 5)
π_7	(0 2 5 3 1 4 6)
π_8	(0 4 2 6 1 5 3 7)
$\vdots$	$\vdots$

where $S(b)$ is defined to be a number such that $S(b)/b$ is close to the fractional part of $\sqrt{b}$. The details are given in algorithm 3.

Algorithm 3 Warnock's algorithm to determine $S(b)$

$X_U = \lceil b\{\sqrt{b}\}\rceil$ (where $\{x\}$ denotes the fractional part of x)
$X_L = \lfloor b\{\sqrt{b}\}\rfloor$
$\frac{X_U}{b} = /d_1, d_2, d_3, \ldots, d_k/$ (continued fraction expansion of $\frac{X_U}{b}$)
$\frac{X_U}{b} = /e_1, e_2, e_3, \ldots, e_m/$ (continued fraction expansion of $\frac{X_L}{b}$)
Choose $S(b)$ as either X_U or X_L according to:
(1) The smaller sum of partial quotients
(2) The smallest largest partial quotient
(3) The nearest to the fractional part of $\sqrt{b}$

4.5 Tuffin's permutations

Tuffin [13] introduced four new multi-dimensional algorithms (MCL, MCT, MCL* and MCT*) to determine permutations for scrambling the Halton sequence. We refer to [13,14] for the details of these algorithms. Because they require a lot of computing time and because [14] only lists the first 16 permutations for MCL and MCL*, we will only compare with these two in this paper.

4.6 A new and easy permutation

Consider the permutations for the prime bases from table 3, which we will call *reverse permutations* for the Halton sequence. The generation of these permutations is by far the most simple one among all algorithms mentioned in this paper. Strangely enough, to our knowledge nothing has been published

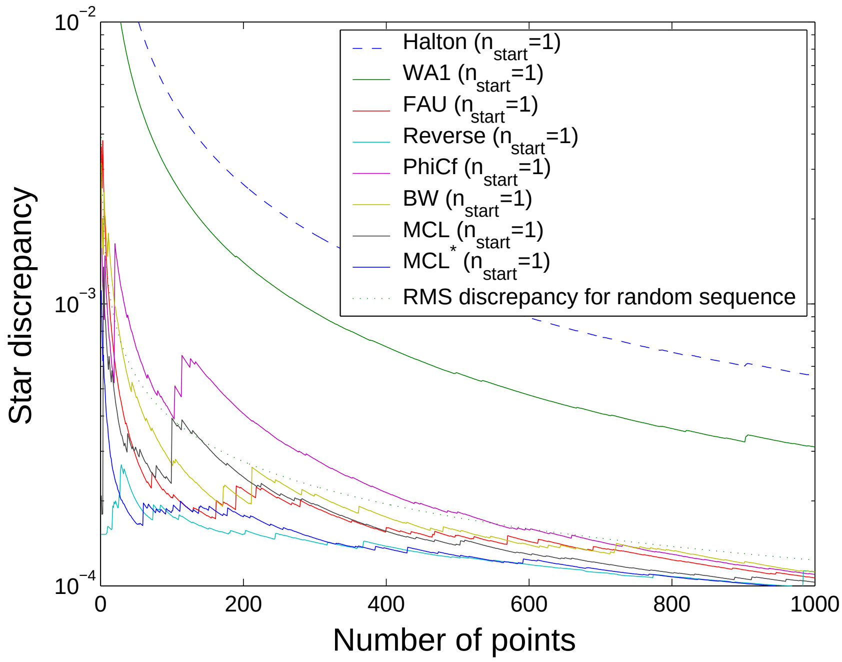


Fig. 9. T_N^* for several 16-dimensional sequences.

about it in the context of scrambling the Halton sequence. Figure 9 shows that in 16 dimensions, the discrepancy of a scrambled Halton sequence with reverse permutations is lower than almost all mentioned Halton scramblings. Only the discrepancy of the sequence that uses permutations obtained with MCL is comparable. Furthermore, the discrepancy of the WA1-sequence seems to behave quite bad compared to the others and even to the root-mean-square discrepancy of a random sequence.

In 32 dimensions, figure 10 shows how the sequence with our *reverse* permutations has a lower star-discrepancy than the WA1 and FAU-sequences for about the first 3000 points. For more points, the discrepancy is comparable. We obtained similar results in other dimensions, but only show the ones for the 16 and 32 dimensional case because these were best for visualization.

Table 3
Our reverse permutations

π_2	$(0\ 1)$
π_3	$(0\ 2\ 1)$
π_5	$(0\ 4\ 3\ 2\ 1)$
$\vdots$	$\vdots$
π_b	$(0\ b-1\ b-2\ \ldots\ 1)$
$\vdots$	$\vdots$

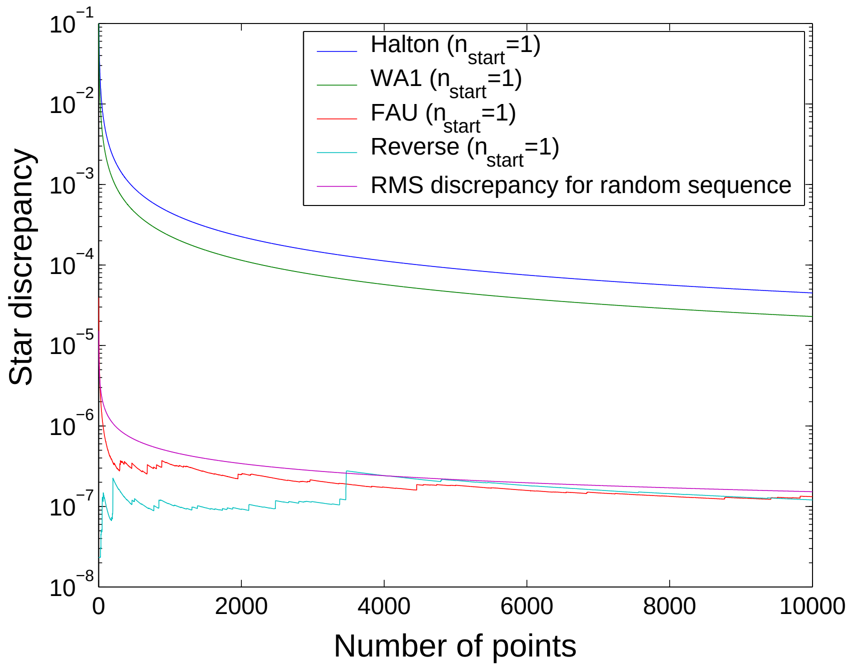


Fig. 10. T_N^* for several 32-dimensional sequences.

The 32 and 64-dimensional data in figure 8 also shows how the discrepancy of a sequence with reverse permutations remains below the expected pseudorandom curve, while a sequence generated by our implementation of Braaten and Weller's algorithm does not.

The initial discrepancy behaviour of a sequence with reverse permutations might look strange at first sight, but when we consider only the initial point $(1/2, 2/3, 4/5, \ldots, b-1/b)$, we see that of all possible first scrambled Halton points, this point is the closest to $(1, 1, 1, \ldots, 1)$. This means that for this initial point, the first two terms in formula (5) will be the smallest when reverse permutations are being used and T_1^* will thus be close to $3^{-s/2}$, which can be observed in figures 9 and 10.

An additional advantage of our scrambling, is that it also has a lower or comparable extreme discrepancy than all other sequences considered here. Figure 11 shows this. Also note that the BW-sequence appears to have the highest extreme discrepancy of all sequences considered.

5 Conclusions

We emphasized that the higher the dimension gets, the better the performance of the BW-sequence is compared to a standard unscrambled Halton sequence. The advantage of permuting the digits does seem to decrease with increasing

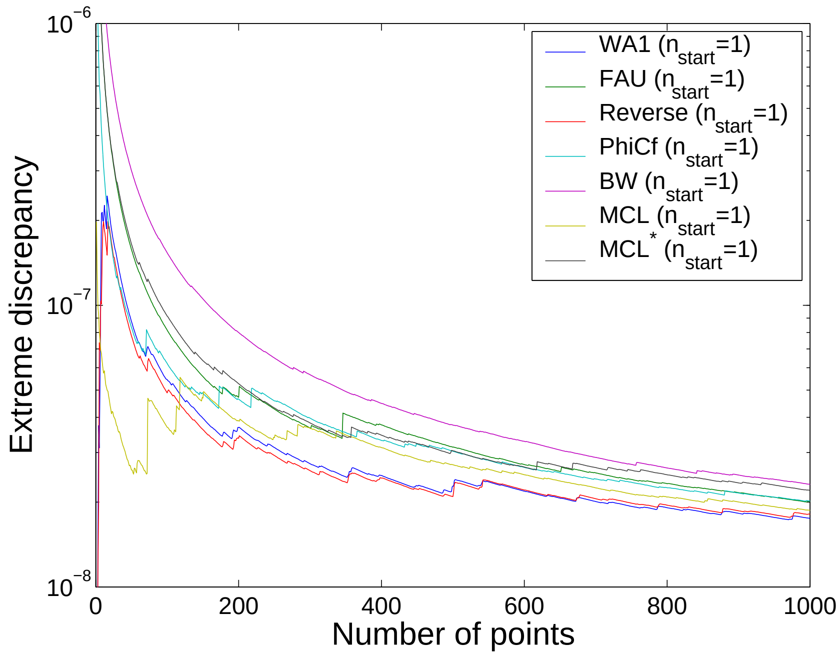


Fig. 11. T_N for several 16-dimensional sequences.

N. In addition, for higher dimensions – say about 32 and more – the L_2 star discrepancy of a scrambled sequence with permutations generated by a straightforward implementation of algorithm 1 is higher than the expected discrepancy of a random sequence.

Good new permutations for scrambling the Halton sequence are the main result in this paper. These permutations are by far the most easiest to generate and, compared to all earlier known methods, they perform equally well or even better in terms of L_2 star and extreme discrepancy.

References

[1] Eric Braaten and George Weller. An improved low-discrepancy sequence for multidimensional quasi-Monte Carlo integration. *Journal of Computational Physics*, 33:249–258, 1979.

[2] Henri Faure. Good permutations for extreme discrepancy. *Journal of Number Theory*, 42:47–56, 1992.

[3] John H. Halton. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. *Numerische Mathematik*, 2:84–90, 1960.

[4] John H. Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Communcations of the ACM*, 7(12):701–702, 1964.

[5] Miroslav Kolar and Seamus F. O'Shea. Fast, portable, and reliable algorithm for the calculation of Halton numbers. *Computers and Mathematics with Applications*, 25(7):3–13, April 1993.

[6] William J. Morokoff and Russel E. Caflish. Quasi-random sequences and their discrepancies. *SIAM Journal on Scientific Computing*, 15(6):1251–1279, 1994.

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

[8] Giray Ökten and Ashok Srinivasan. Parallel quasi-monte carlo methods on a heterogeneous cluster. In Harald Niederreiter, Kai-Tai Fang, and Fred J. Hickernell, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 406–421, Berlin Heidelberg, 2002. Springer-Verlag.

[9] Tim Pillards. A note on E. Thiémard's algorithm to compute bounds for the star discrepancy. *Journal of Complexity*, 2004. To appear.

[10] Ashok Srinivasan. Parallel and distributed computing issues in pricing financial derivatives through quasi Monte Carlo. In *Proceedings of the Sixteenth International Parallel and Distributed Processing Symposium*, 2002.

[11] Jens Struckmeier. Fast generation of low-discrepancy sequences. *Journal of Computational and Applied Mathematics*, 61(1):29–41, 1995.

[12] Eric Thiémard. An algorithm to compute bounds for the star discrepancy. *Journal of Complexity*, 17(4):850–880, December 2001.

[13] Bruno Tuffin. A new permutation choice in Halton sequences. In *Monte Carlo and Quasi-Monte Carlo 1996*, volume 127, pages 427–435, New-York, 1997. Springer-Verlag.

[14] Bruno Tuffin. *Simulation accélérée par les méthodes de Monte Carlo et quasi-Monte Carlo: théorie et applications*. PhD thesis, École Doctorale de Mathématiques de l'Ouest, Université de Rennes, October 1997.

[15] Tony T. Warnock. Computational investigations of low-discrepancy point sets. In S.K. Zaremba, editor, *Applications of Number Theory to Numerical Analysis, (Proc. Sympos., Univ. Montreal, Que., 1971)*, pages 319–343. Academic Press, New York, 1972.

[16] Tony T. Warnock. Computational investigations of low-discrepancy point sets II. In Harald Niederreiter and Peter J.-S. Shiue, editors, *Monte Carlo and quasi-Monte Carlo methods in scientific computing*. Springer, 1995.

[17] Hermann Weyl. Über die gleichverteilung von Zahlen mod. Eins. *Mathematische Annalen*, 77:313–352, 1916.