On the Xorshift Random Number Generators

FRANÇOIS PANNETON and PIERRE L'ECUYER
Université de Montréal

G. Marsaglia recently introduced a class of very fast *xorshift* random number generators, whose implementation uses three "xorshift" operations. They belong to a large family of generators based on linear recurrences modulo 2, which also includes shift-register generators, the Mersenne twister, and several others. In this article, we analyze the theoretical properties of xorshift generators, search for the best ones with respect to the equidistribution criterion, and test them empirically. We find that the vast majority of xorshift generators with only three xorshift operations, including those having good equidistribution, fail several simple statistical tests. We also discuss generators with more than three xorshifts.

Categories and Subject Descriptors: G.4 [**Mathematical Software**]—*Algorithm design and analysis*; I.6 [**Simulation and Modeling**]

General Terms: Algorithms

Additional Key Words and Phrases: Random number generation, xorshift, linear feedback shift register, linear recurrence modulo 2

1. INTRODUCTION

Marsaglia [2003] proposed a class of very fast uniform random number generators (RNGs) called *xorshift*. The state of a xorshift generator is a vector of bits. At each step, the next state is obtained by applying a certain number of xorshift operations to w-bit blocks in the current state, where $w = 32$ or 64; a *xorshift operation* is defined as follows: replace the w-bit block by a bitwise xor (exclusive or) of the original block with a shifted copy of itself by a positions either to the right or to the left, where $0 < a < w$.

Xorshifts are linear operations. The left shift of a w-bit vector $\mathbf{x}$ by one bit, $\mathbf{x} \ll 1$, can also be written as $\mathbf{Lx}$ where $\mathbf{L}$ is the $w \times w$ matrix with ones on its main superdiagonal and zeros elsewhere. Similarly, the right shift $\mathbf{x} \gg 1$ can be

This work has been supported by the National Sciences and Engineering Research Council of Canada (NSERC) Grant Number ODGP0110050, NATEQ-Québec Grant Number 02ER3218, and a Canada Research Chair to the second author. The first author benefited from NSERC and NATEQ scholarships.

Authors' address: Département d'Informatique et de Recherche Opérationnelle, Université de Montréal, H3C 3J7, Canada; email: {panneton,lecuyer}@iro.umontreal.ca.

written as $\mathbf{Rx}$ where $\mathbf{R}$ has ones on its main subdiagonal and zeros elsewhere. Matrices of the forms $(\mathbf{I} + \mathbf{L}^a)$ and $(\mathbf{I} + \mathbf{R}^a)$, where $a \in \{1, \ldots, w-1\}$, are called *left* and *right xorshift matrices*, respectively. They represent *left* and *right a-bit xorshift operations*.

A *xorshift generator* is defined by a recurrence of the form

$$\mathbf{v}_i = \sum_{j=1}^{p} \tilde{\mathbf{A}}_j \mathbf{v}_{i-m_j} \bmod 2 \tag{1}$$

where p is a positive integer, the $\mathbf{v}_i$'s are w-bit vectors, the m_j's are positive integers, and $\tilde{\mathbf{A}}_j$ is either the identity or the product of ν_j xorshift matrices for some $\nu_j \geq 0$, for each j ($\tilde{\mathbf{A}}_j$ is the zero matrix if $\nu_j = 0$). If we define $r = \max_{1 \leq j \leq p} m_j$, the generator's state at step i can be written as $\mathbf{x}_i = (\mathbf{v}_{i-r+1}^{\mathsf{T}}, \ldots, \mathbf{v}_i^{\mathsf{T}})^{\mathsf{T}}$ and the output is $u_i = \sum_{\ell=1}^{w} v_{i,\ell-1} 2^{-\ell}$ where $\mathbf{v}_i = (v_{i,0}, \ldots, v_{i,w-1})^{\mathsf{T}}$. The output sequence $\{u_i,\, i \geq 0\}$ is supposed to imitate independent random variables uniformly distributed over the interval [0, 1]. We can rewrite (1) as

$$\mathbf{v}_i = \sum_{j=1}^{r} \mathbf{A}_j \mathbf{v}_{i-j} \bmod 2 \quad \text{where } \mathbf{A}_j = \sum_{\{l : m_l = j\}} \tilde{\mathbf{A}}_l. \tag{2}$$

This is a special case of the *multiple recursive matrix method* defined in Niederreiter [1995].

Marsaglia [2003] considers three types of xorshift generators, mostly with $\nu_1 + \cdots + \nu_r = 3$ (i.e., exactly three xorshift operations). The type I generators have $r = 1$ and $\mathbf{A}_1$ is the product of three xorshift matrices. For example, one may have $\mathbf{A}_1 = (\mathbf{I} + \mathbf{L}^{13})(\mathbf{I} + \mathbf{R}^{17})(\mathbf{I} + \mathbf{L}^{5})$, which Marsaglia says is "one of my favorite choices." For the type II generators, $r > 1$, $\mathbf{A}_r \neq 0$ and only one other matrix $\mathbf{A}_j$ is nonzero. For example, one could have $r = 4$, $\mathbf{A}_4 = (\mathbf{I} + \mathbf{R}^7)(\mathbf{I} + \mathbf{L}^{11})$, and $\mathbf{A}_1 = (\mathbf{I} + \mathbf{L}^{20})$. The type III generators have $r > 1$, $\mathbf{A}_r \neq 0$; exactly three matrices $\mathbf{A}_j$ are xorshift matrices, and the others are zero. For example, one may have $r = 12$, $\mathbf{A}_2 = (\mathbf{I} + \mathbf{L}^7)$, $\mathbf{A}_3 = (\mathbf{I} + \mathbf{R}^{11})$, and $\mathbf{A}_{12} = (\mathbf{I} + \mathbf{L}^{21})$.

These generators are extremely fast and it is easy to find parameter values for which they have full period length $2^{rw} - 1$. This period length can be made very large by selecting a large r and (say) $w = 32$. But as we all know, a long period does not suffice to have a high-quality generator. In his article, Marsaglia says: "Although I have only tested a few of them, any one of the 648 choices above is likely to provide a very fast, simple, high quality RNG." So, how robust and reliable are they?

To answer the question, in this article, we study the xorshift generators from both the theoretical and empirical perspectives. These generators form a subclass of a large and well-known family of RNGs based on linear recurrences modulo 2, as described in Section 2. In Section 3, we recall standard equidistribution criteria that are widely used to measure the uniformity and independence of these linear generators. In Section 4, we establish some properties of xorshift matrices and generators. In particular, we identify classes of generators having the same period length, others having the same equidistribution, and others that are completely equivalent. In Section 5, we assess the equidistribution of specific instances proposed by Marsaglia and submit them to statistical tests. Many of them have very bad equidistribution and fail some tests spectacularly. In Section 6, we perform a search for the best generators of types I, II, and III with respect to equidistribution criteria, under the constraint that they use only three xorshift operations. We test the best ones and find that they are statistically weak. In Section 7, we briefly look at xorshift generators with more than three xorshifts.

2. MATRIX LINEAR RECURRENCES MODULO 2

The xorshift generators belong to a large class of RNGs based on the following type of linear recurrences modulo 2:

$$\mathbf{x}_i = \mathbf{A}\mathbf{x}_{i-1} \bmod 2, \tag{3}$$

$$\mathbf{y}_i = \mathbf{B}\mathbf{x}_i \bmod 2, \tag{4}$$

$$u_i = \sum_{\ell=1}^{w} y_{i,\ell-1} 2^{-\ell} = .y_{i,0}\ y_{i,1}\ y_{i,2}\ \cdots, \tag{5}$$

where $\mathbf{x}_i = (x_{i,0}, \ldots, x_{i,k-1})^{\mathsf{T}}$ and $\mathbf{y}_i = (y_{i,0}, \ldots, y_{i,w-1})^{\mathsf{T}}$ are the k-bit *state* and the w-bit *output vector* at step i, $\mathbf{A}$ is a $k \times k$ binary *transition matrix*, $\mathbf{B}$ is a $w \times k$ binary *output transformation matrix*, k and w are positive integers, and $u_i \in [0, 1)$ is the *output* at step i. By appropriate choices of $\mathbf{A}$ and $\mathbf{B}$, several types of generators can be obtained as special cases of this general class; for instance the Tausworthe, linear feedback shift register (LFSR), generalized feedback shift register (GFSR), twisted GFSR, Mersenne twister, WELL, and so forth (see, e.g., L'Ecuyer 2004; Panneton and L'Ecuyer 2004, and Panneton 2004 for the details). Important advantages of these generators are that fast implementations are available (they exploit the binary nature of computers) and that their mathematical properties are well understood from a theoretical perspective.

Let $P(z) = \det(\mathbf{A} - \mathbf{I}z)$ be the characteristic polynomial of $\mathbf{A}$. It is well known that the recurrence (3) has period length $2^k - 1$ (its maximal possible value) if and only if $P(z)$ is a primitive polynomial modulo 2 [Niederreiter 1992; Knuth 1998].

Xorshift generators fit this setting by taking $k = rw$,

$$\mathbf{A} = \begin{pmatrix} 0 & \mathbf{I} & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{I} \\ \mathbf{A}_r & \mathbf{A}_{r-1} & \cdots & \mathbf{A}_1 \end{pmatrix},$$

$\mathbf{x}_i = (\mathbf{v}_{i-r+1}^{\mathsf{T}}, \ldots, \mathbf{v}_i^{\mathsf{T}})^{\mathsf{T}}$, and $\mathbf{y}_i = \mathbf{v}_i$. The matrix $\mathbf{B}$ contains a $w \times w$ identity $\mathbf{I}$ in its upper left corner and zeros elsewhere. The characteristic polynomial of this $\mathbf{A}$ is $P(z) = \det(z^r\mathbf{I} + \sum_{j=1}^{r} z^{r-j}\mathbf{A}_j)$. While completing this article, we found a related article by Brent [2004a], who also reports the similitudes between the xorshift and LFSR generators.

Note that replacing the identity $\mathbf{I}$ in the upper left corner of $\mathbf{B}$ by an arbitrary invertible $w \times w$ matrix $\tilde{\mathbf{B}}$, defining $\mathbf{y}_i = \tilde{\mathbf{B}}\mathbf{v}_i$, is equivalent to replacing each

$\mathbf{A}_j$ in (2) by $\tilde{\mathbf{B}}\mathbf{A}_j\tilde{\mathbf{B}}^{-1}$, which gives the recurrence

$$\mathbf{y}_i = \sum_{j=1}^{r} \tilde{\mathbf{B}}\mathbf{A}_j\tilde{\mathbf{B}}^{-1}\mathbf{y}_{i-j} \bmod 2. \tag{6}$$

This does not change the characteristic polynomial $P(z)$.

3. EQUIDISTRIBUTION

For an arbitrary integer $t > 0$, a vector of t successive output values of an RNG would ideally behave as a random vector uniformly distributed over the t-dimensional unit hypercube $[0, 1]^t$. But for an RNG defined by (3)–(5), all these vectors must belong to the finite set

$$\Psi_t = \{(u_0, u_1, \ldots, u_{t-1}) : \mathbf{x}_0 \in \{0, 1\}^k\}.$$

It is customary to require that Ψ_t covers $[0, 1]^t$ very evenly for all t up to some large integer, so that drawing a point uniformly from the "sample space" Ψ_t gives a good approximation of a uniform random variable over $[0, 1]^t$ [L'Ecuyer 2004]. For this class of RNGs, a standard way of assessing the uniformity of Ψ_t is the following.

Recall that Ψ_t has cardinality 2^k. If we divide the interval $[0, 1)$ into 2^ℓ equal segments for some positive integer ℓ, this determines a partition of the unit hypercube $[0, 1)^t$ into $2^{t\ell}$ cubic cells of equal size, called a *(t, ℓ)-equidissection* in base 2. The set Ψ_t is said to be *(t, ℓ)-equidistributed* if each cell contains exactly $2^{k-t\ell}$ of its points. This can be verified by expressing the $t\ell$ bits that determine the cell number of a point as a linear combination of the k bits of the state, and checking if the corresponding matrix has full rank [Fushimi and Tezuka 1983; L'Ecuyer 1996]. For a fixed resolution ℓ, let t_ℓ denote the largest value of t such that Ψ_t is (t, ℓ)-equidistributed. A theoretical upper bound on t_ℓ is $t_\ell^* \stackrel{\text{def}}{=} \lfloor k/\ell \rfloor$. We define the *dimension gap* in resolution ℓ as $\delta_\ell = t_\ell^* - t_\ell$. As measures of uniformity, we consider the *worst-case dimension gap* and the *sum of dimension gaps*, defined as

$$\Delta_\infty = \max_{1\le\ell\le w} \delta_\ell \quad \text{and} \quad \Delta_1 = \sum_{\ell=1}^{w} \delta_\ell.$$

A small value of Δ_1 or Δ_∞ indicates good uniformity. A generator is called *maximally equidistributed* (ME) if $\Delta_1 = 0$ [L'Ecuyer 1996]. ME generators have the best possible equidistribution properties in terms of cubic equidissections.

4. PROPERTIES OF XORSHIFT GENERATORS

In this section, we examine some theoretical properties of xorshift matrices and show how these properties affect the period length and the equidistribution of xorshift generators.

4.1 General Properties

Our first proposition implies that to reach the maximal period $2^{rw} - 1$, a xorshift generator must contain *both* left and right xorshift matrices.

PROPOSITION 4.1. *If the nonzero matrices A_i are products and/or sums of either all left or all right xorshift matrices, then $P(z)$ cannot be irreducible, and therefore cannot be primitive.*

PROOF. If the nonzero $\mathbf{A}_i$'s are all products and/or sums of xorshift matrices on the same side, then the matrix $z^r\mathbf{I} + \sum_{j=1}^{r} z^{r-j}\mathbf{A}_j$ is triangular, hence $P(z)$ is the product of its main diagonal—a product of z's and $(1 - z)$'s. □

Consider a generator defined by (3)–(5) where $\mathbf{B}$ has an arbitrary $w \times w$ matrix $\tilde{\mathbf{B}}$ in its upper left corner and zeros elsewhere. Let $\Psi_t(\mathbf{A}, \tilde{\mathbf{B}})$ denote the corresponding set Ψ_t. Our next result says that adding a right xorshift to the output of such a generator preserves its equidistribution, and therefore does not change its values of Δ_1 and Δ_∞.

PROPOSITION 4.2. *For $\ell \le w$, the set $\Psi_t(\mathbf{A}, (\mathbf{I} + \mathbf{R}^a)\tilde{\mathbf{B}})$ is (t, ℓ)-equidistributed for $0 < a < w$ if and only if $\Psi_t(\mathbf{A}, \tilde{\mathbf{B}})$ is.*

PROOF. The $t\ell$ bits that determine in which box of the (t, ℓ)-equidissection the point $(u_0, \ldots, u_{t-1})$ will fall are the ℓ most significant bits of each of its coordinates. Let $\mathbf{y}(t, \ell) = (y_{0,0}, \ldots, y_{0,\ell-1}, \ldots, y_{t-1,0}, \ldots, y_{t-1,\ell-1})^{\mathsf{T}}$ be the vector of those bits and observe that $\mathbf{y}(t, \ell) = \mathbf{M}(t, \ell)\mathbf{x}_0$ where $\mathbf{M}(t, \ell)$ is the $t\ell \times k$ binary matrix whose row $\ell i + j$ is row j of $\mathbf{BA}^i$, for $1 \le j \le \ell$ and $0 \le i \le t - 1$. The set Ψ_t is (t, ℓ)-equidistributed if and only if each possibility for $\mathbf{y}(t, \ell)$ occurs the same number of times when $\mathbf{x}_0$ runs through all of its 2^k possibilities. This happens if and only if the matrix $\mathbf{M}(t, \ell)$ has full rank, $t\ell$.

If we consider $\Psi_t(\mathbf{A}, \tilde{\mathbf{T}}\tilde{\mathbf{B}})$ instead of $\Psi_t(\mathbf{A}, \tilde{\mathbf{B}})$, where $\tilde{\mathbf{T}} = \mathbf{I} + \mathbf{R}^a$, we must redefine row $\ell i + j$ of $\mathbf{M}(t, \ell)$ as row j of $\mathbf{TBA}^i$, where $\mathbf{T}$ is a matrix with $\tilde{\mathbf{T}}$ in its upper-left corner and zeros elsewhere. This amounts to left-multiplying $\mathbf{M}(t, \ell)$ by a block-diagonal $t\ell \times t\ell$ matrix $\mathbf{D}$ comprised of t diagonal blocks that are all equal to the $\ell \times \ell$ upper-left corner of $\tilde{\mathbf{T}}$. But for any $a > 0$, the matrix $\tilde{\mathbf{T}}$ is lower-triangular with ones all over its main diagonal, so $\mathbf{D}$ also has this property and is thus invertible. Therefore, left-multiplying $\mathbf{M}(t, \ell)$ by $\mathbf{D}$ does not change its rank and this completes the proof. □

Two matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ are *similar* if there exists an invertible matrix $\mathbf{C}$ such that $\mathbf{M}_1 = \mathbf{C}\mathbf{M}_2\mathbf{C}^{-1}$. Similarity is an equivalence relationship, denoted $\mathbf{M}_1 \sim \mathbf{M}_2$. Similar transition matrices $\mathbf{A}$ have the same characteristic polynomial and define generators with the same period length. The next proposition exploits this fact to derive several useful properties of xorshift matrices.

PROPOSITION 4.3. *Let $\mathbf{H}, \mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, \mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3 \in \{\mathbf{L}, \mathbf{R}\}$ and such that $\mathbf{H}_i \ne \mathbf{K}_i$ for $i = 1, 2, 3$. We have that:*

(a) $(\mathbf{I} + \mathbf{L}^a) \sim (\mathbf{I} + \mathbf{R}^a)$;
(b) $(\mathbf{I} + \mathbf{H}^a)(\mathbf{I} + \mathbf{H}^b) = (\mathbf{I} + \mathbf{H}^b)(\mathbf{I} + \mathbf{H}^a)$;
(c) $(\mathbf{I} + \mathbf{H}_3^c)(\mathbf{I} + \mathbf{H}_2^b)(\mathbf{I} + \mathbf{H}_1^a) \sim (\mathbf{I} + \mathbf{H}_1^a)(\mathbf{I} + \mathbf{H}_3^c)(\mathbf{I} + \mathbf{H}_2^b)$;
(d) $(\mathbf{I} + \mathbf{H}_3^c)(\mathbf{I} + \mathbf{H}_2^b)(\mathbf{I} + \mathbf{H}_1^a) \sim (\mathbf{I} + \mathbf{K}_3^c)(\mathbf{I} + \mathbf{K}_2^b)(\mathbf{I} + \mathbf{K}_1^a)$;
(e) $(\mathbf{I} + \mathbf{H}_3^c)(\mathbf{I} + \mathbf{H}_2^b)(\mathbf{I} + \mathbf{H}_1^a) \sim (\mathbf{I} + \mathbf{H}_3^c)(\mathbf{I} + \mathbf{H}_1^a)(\mathbf{I} + \mathbf{H}_2^b)$.

PROOF. Let $\mathbf{P}$ denote the square matrix with ones on its main antidiagonal and zeros elsewhere, that is, the identity matrix with its columns in reverse order. We have $(\mathbf{I} + \mathbf{L}^a) = \mathbf{P}(\mathbf{I} + \mathbf{R}^a)\mathbf{P}^{-1}$ and this implies (a) and (d). Part (b) can be verified by multiplying the terms on each side and using the fact that $\mathbf{H}^a\mathbf{H}^b = \mathbf{H}^{a+b} = \mathbf{H}^b\mathbf{H}^a$. For part (c), we have $(\mathbf{I} + \mathbf{H}_3^c)(\mathbf{I} + \mathbf{H}_2^b)(\mathbf{I} + \mathbf{H}_1^a) \sim (\mathbf{I}+\mathbf{H}_1^a)(\mathbf{I}+\mathbf{H}_3^c)(\mathbf{I}+\mathbf{H}_2^b)(\mathbf{I}+\mathbf{H}_1^a)(\mathbf{I}+\mathbf{H}_1^a)^{-1} \sim (\mathbf{I}+\mathbf{H}_1^a)(\mathbf{I}+\mathbf{H}_3^c)(\mathbf{I}+\mathbf{H}_2^b)$. Part (e) is left as an exercise to the reader. □

By using the properties listed in Proposition 4.3, one can easily show the following proposition.

Proposition 4.4. *The eight matrices*

$$\begin{aligned}
\mathbf{X}_1 &= (\mathbf{I}+\mathbf{L}^c)(\mathbf{I}+\mathbf{R}^b)(\mathbf{I}+\mathbf{L}^a), & \mathbf{X}_2 &= (\mathbf{I}+\mathbf{L}^a)(\mathbf{I}+\mathbf{R}^b)(\mathbf{I}+\mathbf{L}^c),\\
\mathbf{X}_3 &= (\mathbf{I}+\mathbf{R}^c)(\mathbf{I}+\mathbf{L}^b)(\mathbf{I}+\mathbf{R}^a), & \mathbf{X}_4 &= (\mathbf{I}+\mathbf{R}^a)(\mathbf{I}+\mathbf{L}^b)(\mathbf{I}+\mathbf{R}^c),\\
\mathbf{X}_5 &= (\mathbf{I}+\mathbf{R}^b)(\mathbf{I}+\mathbf{L}^c)(\mathbf{I}+\mathbf{L}^a), & \mathbf{X}_6 &= (\mathbf{I}+\mathbf{L}^b)(\mathbf{I}+\mathbf{R}^a)(\mathbf{I}+\mathbf{R}^c),\\
\mathbf{X}_7 &= (\mathbf{I}+\mathbf{L}^c)(\mathbf{I}+\mathbf{L}^a)(\mathbf{I}+\mathbf{R}^b), & \mathbf{X}_8 &= (\mathbf{I}+\mathbf{R}^a)(\mathbf{I}+\mathbf{R}^c)(\mathbf{I}+\mathbf{L}^b),
\end{aligned}$$

where a, b, c are in $\{1, \ldots, w-1\}$, are all similar.

Proof. Let $\mathbf{P}$ be as in the proof of Proposition 4.3 and $\mathbf{C}_j$, $1 \le j \le 8$, be the matrices such that $\mathbf{X}_1 = \mathbf{C}_j\mathbf{X}_j\mathbf{C}_j^{-1}$. Using Proposition 4.3, we show that: $\mathbf{C}_2 = (\mathbf{I} + \mathbf{L}^a)^{-1}(\mathbf{I} + \mathbf{L}^c)$, $\mathbf{C}_3 = \mathbf{P}$, $\mathbf{C}_4 = \mathbf{P}(\mathbf{I} + \mathbf{R}^a)^{-1}(\mathbf{I} + \mathbf{R}^c)$, $\mathbf{C}_5 = (\mathbf{I} + \mathbf{L}^c)$, $\mathbf{C}_6 = (\mathbf{I}+\mathbf{L}^c)\mathbf{P}$, $\mathbf{C}_7 = (\mathbf{I}+\mathbf{L}^a)^{-1}$ and $\mathbf{C}_8 = (\mathbf{I}+\mathbf{L}^a)^{-1}\mathbf{P}$. □

We will make use of this last proposition in our examination of the three types of xorshift generators proposed by Marsaglia [2003].

4.2 Generators of Type I

Because the matrices $\mathbf{X}_1, \ldots, \mathbf{X}_8$ are all similar, by finding a triple (a, b, c) for which the generator based on $\mathbf{X}_1$ has full period, we in fact identify eight full-period generators. Marsaglia [2003] recognizes this fact (without providing a proof), but only for the matrices $\mathbf{X}_1, \ldots, \mathbf{X}_6$, and he includes the matrices $(\mathbf{I} + \mathbf{R}^b)(\mathbf{I}+\mathbf{L}^a)(\mathbf{I}+\mathbf{L}^c)$ and $(\mathbf{I}+\mathbf{L}^b)(\mathbf{I}+\mathbf{R}^c)(\mathbf{I}+\mathbf{R}^a)$, which are in fact a repetition $\mathbf{X}_5$ and $\mathbf{X}_6$.

The next proposition implies that to verify the equidistribution of all type I generators defined by a triple (a, b, c), we only need to verify the equidistribution of the generators based on $\mathbf{X}_1$, $\mathbf{X}_2$, $\mathbf{X}_3$, and $\mathbf{X}_5$.

Proposition 4.5. *For a given triple (a, b, c), the type I generators with $\mathbf{A}_1 = \mathbf{X}_7$ and $\mathbf{A}_1 = \mathbf{X}_5$ have exactly the same equidistribution properties, and the type I generators with $\mathbf{A}_1$ equal to either $\mathbf{X}_3$, $\mathbf{X}_4$, $\mathbf{X}_6$, or $\mathbf{X}_8$ also have exactly the same equidistribution properties.*

Proof. By (6), taking $\mathbf{A}_1 = \mathbf{X}_7$ and $\mathbf{y}_i = (\mathbf{I}+\mathbf{R}^b)\mathbf{v}_i$ is equivalent to using the recurrence $\mathbf{y}_i = (\mathbf{I} + \mathbf{R}^b)\mathbf{X}_7(\mathbf{I} + \mathbf{R}^b)^{-1}\mathbf{y}_{i-1} = (\mathbf{I} + \mathbf{R}^b)(\mathbf{I} + \mathbf{L}^c)(\mathbf{I} + \mathbf{L}^a)(\mathbf{I} + \mathbf{R}^b)(\mathbf{I} + \mathbf{R}^b)^{-1}\mathbf{y}_{i-1} = \mathbf{X}_5\mathbf{y}_{i-1}$. This means that $\Psi_t(\mathbf{X}_5, \mathbf{I}) = \Psi_t(\mathbf{X}_7, \mathbf{I} + \mathbf{R}^b)$ for all t. It then follows from Proposition 4.2 that the generators with $\mathbf{X}_7$ and $\mathbf{X}_5$ have identical equidistribution properties. A similar argument applies to the type I generators produced by $\mathbf{X}_3$, $\mathbf{X}_4$, $\mathbf{X}_6$, and $\mathbf{X}_8$. □

Table I. List of Possible Combinations $(\mathbf{G}, \mathbf{H})$, Regrouped in Classes of Generators having the Same Period Length

i	$\mathbf{G}_i$	$\mathbf{H}_i$	$\mathbf{C}$	Equidistribution
1	$(\mathbf{I}+\mathbf{R}^b)(\mathbf{I}+\mathbf{L}^a)$	$(\mathbf{I}+\mathbf{R}^c)$	$\mathbf{I}$	same as $i = 3$
2	$(\mathbf{I}+\mathbf{L}^b)(\mathbf{I}+\mathbf{R}^a)$	$(\mathbf{I}+\mathbf{L}^c)$	$\mathbf{P}$	
3	$(\mathbf{I}+\mathbf{L}^a)(\mathbf{I}+\mathbf{R}^b)$	$(\mathbf{I}+\mathbf{R}^c)$	$(\mathbf{I}+\mathbf{R}^b)$	same as $i = 1$
4	$(\mathbf{I}+\mathbf{R}^a)(\mathbf{I}+\mathbf{L}^b)$	$(\mathbf{I}+\mathbf{L}^c)$	$(\mathbf{I}+\mathbf{L}^b)\mathbf{P}$	
5	$(\mathbf{I}+\mathbf{R}^c)$	$(\mathbf{I}+\mathbf{R}^b)(\mathbf{I}+\mathbf{L}^a)$	$\mathbf{I}$	same as $i = 7$
6	$(\mathbf{I}+\mathbf{L}^c)$	$(\mathbf{I}+\mathbf{L}^b)(\mathbf{I}+\mathbf{R}^a)$	$\mathbf{P}$	
7	$(\mathbf{I}+\mathbf{R}^c)$	$(\mathbf{I}+\mathbf{L}^a)(\mathbf{I}+\mathbf{R}^b)$	$(\mathbf{I}+\mathbf{R}^b)$	same as $i = 5$
8	$(\mathbf{I}+\mathbf{L}^c)$	$(\mathbf{I}+\mathbf{R}^a)(\mathbf{I}+\mathbf{L}^b)$	$(\mathbf{I}+\mathbf{L}^b)\mathbf{P}$	
9	$(\mathbf{I}+\mathbf{R}^a)(\mathbf{I}+\mathbf{R}^b)$	$(\mathbf{I}+\mathbf{L}^c)$	$\mathbf{I}$	
10	$(\mathbf{I}+\mathbf{L}^a)(\mathbf{I}+\mathbf{L}^b)$	$(\mathbf{I}+\mathbf{R}^c)$	$\mathbf{P}$	
11	$(\mathbf{I}+\mathbf{L}^c)$	$(\mathbf{I}+\mathbf{R}^a)(\mathbf{I}+\mathbf{R}^b)$	$\mathbf{I}$	
12	$(\mathbf{I}+\mathbf{R}^c)$	$(\mathbf{I}+\mathbf{L}^a)(\mathbf{I}+\mathbf{L}^b)$	$\mathbf{P}$	
13–14	$\mathbf{I}$	$\mathbf{X}_j,\ j \in \{1, 2\}$		
15–16	$\mathbf{I}$	$\mathbf{X}_j,\ j \in \{5, 7\}$	See Section 4	same for $i = 15, 16$
17–20	$\mathbf{I}$	$\mathbf{X}_j,\ j \in \{3, 4, 6, 8\}$		same for $17 \le i \le 20$

4.3 Generators of Type II

A generator of type II is implemented via a special version of (2), which can be written as:

$$\mathbf{v}_i = \mathbf{G}\mathbf{v}_{i-m} \oplus \mathbf{H}\mathbf{v}_{i-r} \tag{7}$$

where $\mathbf{G}$ and $\mathbf{H}$ are $w \times w$ matrices and $\oplus$ denotes the bitwise exclusive-or operation. We denote this generator by the triple $(\mathbf{G}, \mathbf{H}, m)$. Property (6) implies that for a given nonsingular $w \times w$ matrix $\mathbf{C}$, $(\mathbf{C}\mathbf{G}\mathbf{C}^{-1}, \mathbf{C}\mathbf{H}\mathbf{C}^{-1}, m)$ provides a full-period generator if and only if $(\mathbf{G}, \mathbf{H}, m)$ provides one.

Table I lists all possibilities of $\mathbf{G}$ and $\mathbf{H}$. They are grouped in a way that within each group, they all have the same characteristic polynomial through the similarity matrix $\mathbf{C}$. The last column indicates the generators having the same equidistribution. Marsaglia [2003] only mentions generators for which $\mathbf{G} = \mathbf{G}_1 = (\mathbf{I} + \mathbf{R}^b)(\mathbf{I} + \mathbf{L}^a)$, $\mathbf{H} = \mathbf{H}_1 = (\mathbf{I} + \mathbf{R}^c)$, and $m = 1$. The following theorem tells us that it is worthless to search for full-period generators of type II with $\mathbf{H} = \mathbf{I}$.

THEOREM 4.6. *Let* $\mathbf{G}$ *be any binary* $w \times w$ *matrix where* $w > 1$. *The recurrence* (7) *with* $\mathbf{H} = \mathbf{I}$ *cannot have period length* $2^{rw} - 1$.

PROOF. For n equal to any power of 2, we denote by $\mathbb{F}_n$ the finite field with n elements, $\mathbb{F}_n^w$ its w-fold cartesian product, and $\mathbb{F}_n[z]$ the space of polynomials with coefficients in $\mathbb{F}_n$. Let $Q(z) = \det(\mathbf{G} - \mathbf{I}z)$ be the characteristic polynomial of $\mathbf{G}$ over $\mathbb{F}_2$ and suppose that $Q(z)$ is irreducible over $\mathbb{F}_2$. Let $\eta \in \mathbb{F}_{2^w}$ be a root of $Q(z)$. The elements $1, \eta, \ldots, \eta^{w-1}$ form a basis of $\mathbb{F}_{2^w}$ over $\mathbb{F}_2$.

We can show (see the proof of Theorem 2 in Matsumoto and Kurita [1992]) that for a nonzero vector in $\mathbf{t} \in \mathbb{F}_2^w$, the homomorphism

$$\begin{aligned} \phi : \mathbb{F}_{2^w} &\to \mathbb{F}_2^w \\ \eta^l &\mapsto \mathbf{G}^l\mathbf{t}, \end{aligned}$$

where $0 \neq \mathbf{t} \in \mathbb{F}_2^w$ is fixed, is also an isomorphism. By applying the inverse of ϕ

to recurrence (7), we obtain the linear recurrence

$$\phi^{-1}(\mathbf{x}_i) = \phi^{-1}(\mathbf{G}\mathbf{x}_{i-m}) + \phi^{-1}(\mathbf{x}_{i-r}),$$

which can be rewritten as the following linear recurrence in $\mathbb{F}_{2^w}$:

$$x_i = \eta x_{i-m} + x_{i-r}, \tag{8}$$

where $x_i = \phi^{-1}(\mathbf{x}_i) \in \mathbb{F}_{2^w}$, because $\phi^{-1}(\mathbf{G}\mathbf{x}_{i-m}) = \eta x_{i-m}$ (see Theorem 2 in Matsumoto and Kurita [1992]). Notice that (7) and (8) have the same period length. The characteristic polynomial over $\mathbb{F}_{2^w}$ of (8) is

$$P(z) = z^r - \eta z^{r-m} - 1 \in \mathbb{F}_{2^w}[z].$$

By Theorem 3.18 of Lidl and Niederreiter [1994], because $(-1)^r P(0) = (-1)^{r+1}$ is not a primitive element of $\mathbb{F}_{2^w}$, the polynomial $P(z)$ is not primitive over $\mathbb{F}_{2^w}$. This implies that the period of (8) cannot be $2^{rw} - 1$, and similarly for (7).

Now, suppose that $Q(z)$ is not irreducible. Let $q = r - m$. We know that

$$\begin{aligned} P(z) = \det(z^r\mathbf{I} - z^q\mathbf{G} - \mathbf{I}) &= \det(z^q(z^m\mathbf{I} - \mathbf{G} - z^{-q}\mathbf{I})) \\ &= \det(z^q\mathbf{I})\det((z^m - z^{-q})\mathbf{I} - \mathbf{G}) \\ &= z^{wq}Q(z^m - z^{-q}) \in \mathbb{F}_2[z]. \end{aligned}$$

Notice that $Q(z^m - z^{-q})$ is not in $\mathbb{F}_2[z]$, so we do not know yet if $P(z)$ is irreducible over $\mathbb{F}_2$ or not. On the other hand, $Q(z^m - z^{-q}) \in \mathbb{L}_2$ where $\mathbb{L}_2$ is the field of formal Laurent series with coefficients in $\mathbb{F}_2$. Because $Q(z)$ is not irreducible, we can decompose it in $c > 1$ factors $Q_i(z) \in \mathbb{F}_2[z]$, each of degree $d_i > 0$.

Let $h(z) = h_n z^n + h_{n-1}z^{n-1} + \cdots \in \mathbb{L}_2$. We define the function $p(h(z)) = \min\{i : h_i \neq 0\}$. For $h(z) \in \mathbb{L}_2$ to be in $\mathbb{F}_2[z]$, it is necessary that $p(h(z)) \geq 0$. Observe that, because $Q(z)$ is of degree w, we have that $w = d_1 + \cdots + d_c$ and $p(Q_i(z^m + z^{-q})) = -qd_i$. Let $\bar{Q}_i(z) = z^{qd_i}Q_i(z^m + z^{-q})$. Then, $p(\bar{Q}_i(z)) = 0$, which implies that $\bar{Q}_i(z) \in \mathbb{F}_2[z]$. We can develop

$$\begin{aligned} z^{wq}Q(z^m - z^{-q}) &= z^{wq}\prod_{i=1}^{c} Q_i(z^m - z^{-q}) \\ &= \prod_{i=1}^{c} z^{qd_i}Q_i(z^m - z^{-q}) \\ &= \prod_{i=1}^{c} \bar{Q}_i(z). \end{aligned}$$

The last equality implies that $P(z)$ is not irreducible over $\mathbb{F}_2$ because it can be decomposed in $c > 1$ factors $\bar{Q}_i(z) \in \mathbb{F}_2[z]$, $i = 1, \ldots, c$. This shows that if $Q(z)$ is not irreducible over $\mathbb{F}_2$, then neither is $P(z)$, and (8) cannot have the maximal period $2^{rw} - 1$ in that case. □

So far, we have discussed the choice of $\mathbf{H}$ and $\mathbf{G}$, but not the choice of m. For the case where $\mathbf{G} = \mathbf{I}$, conditions on m for the recurrence (7) to have full period $2^{rw} - 1$ can be found in Matsumoto and Kurita [1992]. In the case where both m and r are even, we have never encountered a full-period generator of type II, despite making exhaustive searches for $r = 4$, 8, and 12. So we conjecture that one cannot have full period in this case, but we have no proof.

4.4 Generators of Type III

The generators of type III are based on the recurrence

$$\mathbf{v}_n = \left(\mathbf{I} + \mathbf{H}_1^{a_1}\right)\mathbf{v}_{n-m_1} \oplus \left(\mathbf{I} + \mathbf{H}_2^{a_2}\right)\mathbf{v}_{n-m_2} \oplus \left(\mathbf{I} + \mathbf{H}_3^{a_3}\right)\mathbf{v}_{n-r} \tag{9}$$

where $\mathbf{H}_i \in \{\mathbf{L}, \mathbf{R}\}$, $a_i < w$, and $1 \leq i \leq r$. We denote this variant by $(\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, m_1, m_2, a_1, a_2, a_3)$. If $(\mathbf{H}_1, \mathbf{H}_2, \mathbf{H}_3, m_1, m_2, a_1, a_2, a_3)$ provides a generator of maximal period then so does $(\mathbf{K}_1, \mathbf{K}_2, \mathbf{K}_3, m_1, m_2, a_1, a_2, a_3)$ where $\mathbf{K}_i \neq \mathbf{H}_i$ and $\mathbf{K}_i \in \{\mathbf{L}, \mathbf{R}\}$, by the similarity matrix $\mathbf{C} = \mathbf{P}$.

5. SPECIFIC GENERATORS PROPOSED BY MARSAGLIA

Marsaglia [2003] lists all parameters (a, b, c) that yield full period type-I xorshift generators with $w = 32$ and $w = 64$. We have checked his results and they are correct except for what seems to be a typo: for $w = 32$, $(a, b, c) = (9, 5, 1)$ should read $(a, b, c) = (9, 5, 14)$. For the type II and type III generators, Marsaglia provides only a few sets of parameters. In the following subsections, we analyze the equidistribution and statistical properties of some generators he proposed.

5.1 Equidistribution Properties

We computed the values of Δ_1 for all type I full-period generators with $w = 32$ and $w = 64$; they are listed in Panneton [2004]. These values range from $\Delta_1 = 1$ (good equidistribution) to $\Delta_1 = 153$ (very bad). For example, a type I generator qualified by Marsaglia as one of his "favorites" is based on $\mathbf{X}_2$ with $(a, b, c) = (5, 17, 13)$ and $w = 32$. This generator has $\Delta_1 = 2$, a good value. Another interesting example is the type I generator based on $\mathbf{X}_4$ with $(a, b, c) = (7, 1, 9)$. This generator has $\Delta_1 = 1$, the best value of all type I generators with $w = 32$, but if we change $\mathbf{X}_4$ to $\mathbf{X}_3$ with the same parameters (a, b, c)—simply change the order in which the two right xorshifts are performed—we get $\Delta_1 = 56$, which is the worst possible value when $w = 32$. This illustrates the fact that the behavior of a generator based on a given triple (a, b, c) and given transition matrix $\mathbf{X}_i$ does not help predicting the behavior of another generator with the same triple but a different choice of $\mathbf{X}_i$.

All type II and type III generators proposed by Marsaglia with $w = 32$ have rather poor values of Δ_1. For example, the type II generator $((\mathbf{I}+\mathbf{R}^1)(\mathbf{I}+\mathbf{L}^2), (\mathbf{I}+\mathbf{R}^4), 1)$ with $r = 5$ has $\Delta_1 = 164$, the type II generator $((\mathbf{I}+\mathbf{R}^1)(\mathbf{I}+\mathbf{L}^{10}), (\mathbf{I}+\mathbf{R}^{26}), 1)$ with $r = 3$ has $\Delta_1 = 81$, and the type III generator $(\mathbf{L}, \mathbf{R}, \mathbf{L}, 1, 2, 6, 19, 3)$ with $r = 3$ has $\Delta_1 = 69$.

5.2 Statistical Testing

We have applied the batteries of empirical statistical tests SmallCrush and Crush implemented in the software package TestU01 [L'Ecuyer and Simard 2001b] to several xorshift generators proposed by Marsaglia [2003]. All tests included in these batteries look for evidence against the null hypotheses $\mathcal{H}_0$ that the generator produces independent $U(0, 1)$ random variables. These two batteries run in about one minute and one hour, respectively, on a standard

Table II. List of Tests From TestU01 Used in This Paper

Num.	Name	Parameters
1	Maximum-of-t	$N = 10$, $n = 10^7$, $\tau = 0$, $d = 5 \times 10^5$, $t = 5$
2	Birthday spacings	$N = 5$, $n = 10^7$, $\tau = 14$, $d = 2^8$, $t = 8$, $p = 1$
3	Indep. Hamming weights	$n = 10^8$, $\tau = 20$, $s = 10$, $L = 30$
4	Matrix rank	$n = 10^6$, $\tau = 20$, $s = 10$, $L = 90$
5	Matrix rank	$n = 50000$, $\tau = 20$, $s = 10$, $L = 300$

PC. None of the proposed xorshift generators passed all these tests and most of them even failed the "baby" battery SmallCrush in a spectacular way.

In what follows, we describe a few of these tests and give concrete illustrations of the results.

The *maximum-of-t* test (see Knuth [1998, page 70]) generates n sequences of t values in [0, 1] and computes the maximum X of the t values for each sequence. The interval [0, 1] is partitioned into d segments in a way that under $\mathcal{H}_0$, X falls in any given segment with probability n/d. The empirical and theoretical frequencies of the d segments are then compared via a chi-square test statistic. The *p-value* of this test is defined as the probability that the chi-square statistic takes a value larger or equal to the one observed, under $\mathcal{H}_0$. For a two-level test, this is repeated N times and the empirical distribution of the N p-values thus obtained is compared to the uniform distribution via a Kolmogorov-Smirnov test, whose p-value (at the second level) is taken as the final p-value of the test. In Crush, we find this test with the parameters given in the first row of Table II (test number 1).

The *birthday spacings* test [Marsaglia 1985; L'Ecuyer and Simard 2001a] partitions $[0, 1]^t$ into d^t subcubes of equal sizes, numbers them from 0 to $k - 1$ in a natural order, and throws n points into $[0, 1]^t$ using t successive uniform random numbers for each point. Let $I_1 \le I_2 \le \cdots \le I_n$ be the (sorted) numbers of cells in which the points fall. The test computes the differences $I_{j+1} - I_j$, for $1 \le j < n$, and counts the number Y of collisions between these differences. Under $\mathcal{H}_0$, Y is approximately Poisson with mean $\lambda = n^3/(4k)$. This process is repeated N times "independently" and the p-value of the test is defined as the probability that a Poisson random variable with mean $N\lambda$ takes a value larger or equal to the sum of the N observed values of Y. In a slight generalization of the test, each output value u of the generator (a real number between 0 and 1) is replaced by $2^\tau u$ mod 1 before these numbers are used to determine the box number. The effect of this is to skip the first τ bits of each output value of the generator and take the following ones. This generalization applies to the other tests of Table II as well. Crush contains this test with $\tau = 14$, $t = 8$, $d = 2^8$, $n = 10^7$, and $N = 5$. So it uses bits 15 to 22 of 8 successive uniforms to determine the box number of each point, assuming that bit 1 is the most significant bit. This is test number 2 in Table II.

The *independence of Hamming weights* test [L'Ecuyer and Simard 1999] takes s successive bits, say bits $\tau + 1$ to $\tau + s$, from each of $2n \lceil L/s \rceil$ successive uniforms, and concatenates these bits to construct $2n$ blocks of L bits. Let X_j be the Hamming weight (the numbers of bits equal to 1) of the jth block, for $j = 1, \ldots, 2n$. Each vector (X_j, X_{j+1}) can take $(L + 1) \times (L + 1)$ possible values. The test counts the number of occurrences of each possibility among the nonoverlapping pairs $\{(X_{2j-1}, X_{2j}),\ 1 \le j \le n\}$, and compares these observations with the expected numbers under $\mathcal{H}_0$, via a chi-square test, after lumping together in a single class all classes for which the expected number is less than 10. This is test number 3 in Table II.

The *matrix rank* test generates n random $L \times L$ binary matrices, computes their ranks, and compares the empirical distribution of these ranks with their theoretical distribution under $\mathcal{H}_0$ via a chi-square test [Marsaglia 1985; L'Ecuyer and Simard 2001a]. Each matrix is filled up line by line by taking s-bit blocks from L^2/s successive uniforms (assuming that s divides L). Tests 4 and 5 in Table II are of this type. These tests skip the $\tau = 20$ most significant bits of each output value and uses the $s = 10$ bits that follow. With a xorshift generator, the rank of the matrix cannot exceed the degree $k = rw$ of $P(z)$, so we expect all xorshift generators to fail this test when $L > rw$. But in many cases, we observed decisive failures for L much smaller than kw.

We tested all full-period type I generators with $\mathbf{A}_1 = \mathbf{X}_1$, for $w = 32$ and $w = 64$. All type I generators with $w = 32$ and about half of those with $w = 64$ had a p-value smaller than 10^{-300} for at least one test of SmallCrush. This can certainly be called a *spectacular failure*! All but two of the generators of type II proposed in Marsaglia [2003] failed a maximum-of-t test found in SmallCrush (with $t = 6, d = 10^5$, $\tau = 0$, and $n = 2 \times 10^6$) with a p-value smaller than 10^{-300}, and the other two failed other tests in SmallCrush or Crush. The generator qualified by Marsaglia as one of his "favorites" had p-values smaller than 10^{-164} for all five tests of Table II.

The type II generator $((\mathbf{I}+\mathbf{R}^1)(\mathbf{I}+\mathbf{L}^2), (\mathbf{I}+\mathbf{R}^4), 1)$ with $r = 5$ has the p-value of $p = 3.6 \times 10^{-9}$ in test 2. The one with $((\mathbf{I}+\mathbf{R}^1)(\mathbf{I}+\mathbf{L}^{10}), (\mathbf{I}+\mathbf{R}^{26}), 1)$ and $r = 3$ gets $p = 1.9 \times 10^{-165}$ for test 1 and $p < 10^{-300}$ for test 3. The type III generator proposed by Marsaglia [2003], which has $(\mathbf{L}, \mathbf{R}, \mathbf{L}, 1, 2, 6, 19, 3)$ and $r = 3$, gets $p = 1.6 \times 10^{-96}$ for test 1.

6. A SEARCH FOR BETTER XORSHIFT GENERATORS

We made a search for good full-period generators of types II and III with respect to the equidistribution criterion Δ_1 with the constraint that the number of xorshifts cannot exceed 3, as for most generators proposed in Marsaglia's article. We tested empirically the best generators we found, by applying SmallCrush and Crush [L'Ecuyer and Simard 2001b].

Table III lists the best generators of type II we found, with respect to Δ_1, in an exhaustive search among those with $w = 32$ and $r = 2, 3, 4, 5, 8, 12, 25$. The best ones have reasonably good values of Δ_1 if we compare them with other well-known generators. For example, for $r = 25$ and $w = 32$, we have one with $\Delta_1 = 123$, which compares advantageously with the TT800 generator of Matsumoto and Kurita [1994], for which $\Delta_1 = 261$.

We applied SmallCrush and Crush to all generators listed in Table III. The number p_3 in the table is the p-value for test number 3 of Table II. The symbol ϵ means "smaller than 10^{-300}" and a blank indicates a p-value between 0.01 and 0.99. For $r \le 5$, all but one generator fail this test spectacularly. This

Table III. Value of Δ_1 for the Best Generators of Type II with $w = 32$ and $r = 2, 3, 4, 5, 8, 12, 25$

Num.	r	m	**H**	**G**	Δ_1	p_3
1	2	1	$(\mathbf{I}+\mathbf{L}^{11})$	$(\mathbf{I}+\mathbf{R}^{13})(\mathbf{I}+\mathbf{L}^{19})$	4	10^{-6}
2	2	1	$(\mathbf{I}+\mathbf{L}^{11})$	$(\mathbf{I}+\mathbf{L}^{19})(\mathbf{I}+\mathbf{R}^{13})$	7	10^{-9}
3	2	1	$(\mathbf{I}+\mathbf{R}^{8})(\mathbf{I}+\mathbf{L}^{9})$	$(\mathbf{I}+\mathbf{L}^{22})$	7	ϵ
4	2	1	$(\mathbf{I}+\mathbf{L}^{11})(\mathbf{I}+\mathbf{R}^{9})$	$(\mathbf{I}+\mathbf{L}^{17})$	7	ϵ
5	3	1	$(\mathbf{I}+\mathbf{L}^{23})$	$(\mathbf{I}+\mathbf{R}^{4})(\mathbf{I}+\mathbf{L}^{13})$	11	ϵ
6	3	1	$(\mathbf{I}+\mathbf{L}^{23})$	$(\mathbf{I}+\mathbf{L}^{11})(\mathbf{I}+\mathbf{R}^{7})$	12	ϵ
7	3	1	$(\mathbf{I}+\mathbf{L}^{23})$	$(\mathbf{I}+\mathbf{R}^{7})(\mathbf{I}+\mathbf{L}^{11})$	12	ϵ
8	3	1	$(\mathbf{I}+\mathbf{L}^{18})$	$(\mathbf{I}+\mathbf{R}^{5})(\mathbf{I}+\mathbf{L}^{13})$	13	ϵ
9	4	1	$(\mathbf{I}+\mathbf{R}^{7})(\mathbf{I}+\mathbf{L}^{11})$	$(\mathbf{I}+\mathbf{L}^{20})$	13	ϵ
10	4	1	$(\mathbf{I}+\mathbf{R}^{7})(\mathbf{I}+\mathbf{L}^{11})$	$(\mathbf{I}+\mathbf{L}^{19})$	17	ϵ
11	4	1	$(\mathbf{I}+\mathbf{L}^{17})$	$(\mathbf{I}+\mathbf{R}^{5})(\mathbf{I}+\mathbf{L}^{12})$	17	ϵ
12	4	1	$(\mathbf{I}+\mathbf{L}^{19})$	$(\mathbf{I}+\mathbf{R}^{15})(\mathbf{I}+\mathbf{L}^{7})$	19	
13	4	1	$(\mathbf{I}+\mathbf{L}^{11})(\mathbf{I}+\mathbf{R}^{7})$	$(\mathbf{I}+\mathbf{L}^{19})$	19	ϵ
14	5	1	$(\mathbf{I}+\mathbf{R}^{7})(\mathbf{I}+\mathbf{L}^{11})$	$(\mathbf{I}+\mathbf{L}^{20})$	18	ϵ
15	5	1	$(\mathbf{I}+\mathbf{R}^{6})(\mathbf{I}+\mathbf{L}^{11})$	$(\mathbf{I}+\mathbf{L}^{20})$	19	ϵ
16	5	3	$(\mathbf{I}+\mathbf{L}^{9})(\mathbf{I}+\mathbf{R}^{6})$	$(\mathbf{I}+\mathbf{L}^{20})$	25	ϵ
17	5	3	$(\mathbf{I}+\mathbf{R}^{6})(\mathbf{I}+\mathbf{L}^{9})$	$(\mathbf{I}+\mathbf{L}^{20})$	25	ϵ
18	8	3	$(\mathbf{I}+\mathbf{R}^{13})(\mathbf{I}+\mathbf{L}^{19})$	$(\mathbf{I}+\mathbf{L}^{8})$	45	
19	8	3	$(\mathbf{I}+\mathbf{R}^{14})(\mathbf{I}+\mathbf{L}^{17})$	$(\mathbf{I}+\mathbf{L}^{8})$	48	
20	8	1	$(\mathbf{I}+\mathbf{R}^{7})(\mathbf{I}+\mathbf{L}^{15})$	$(\mathbf{I}+\mathbf{L}^{10})$	52	
21	8	1	$(\mathbf{I}+\mathbf{R}^{11})(\mathbf{I}+\mathbf{L}^{8})$	$(\mathbf{I}+\mathbf{L}^{21})$	54	
22	12	5	$(\mathbf{I}+\mathbf{L}^{21})(\mathbf{I}+\mathbf{R}^{11})$	$(\mathbf{I}+\mathbf{L}^{6})$	74	
23	12	5	$(\mathbf{I}+\mathbf{R}^{6})(\mathbf{I}+\mathbf{L}^{7})$	$(\mathbf{I}+\mathbf{L}^{22})$	79	
24	12	1	$(\mathbf{I}+\mathbf{R}^{8})(\mathbf{I}+\mathbf{L}^{7})$	$(\mathbf{I}+\mathbf{L}^{18})$	84	
25	12	5	$(\mathbf{I}+\mathbf{L}^{7})(\mathbf{I}+\mathbf{R}^{6})$	$(\mathbf{I}+\mathbf{L}^{22})$	90	
26	25	9	$(\mathbf{I}+\mathbf{R}^{8})(\mathbf{I}+\mathbf{L}^{11})$	$(\mathbf{I}+\mathbf{L}^{18})$	123	
27	25	9	$(\mathbf{I}+\mathbf{L}^{11})(\mathbf{I}+\mathbf{R}^{8})$	$(\mathbf{I}+\mathbf{L}^{18})$	137	
28	25	7	$(\mathbf{I}+\mathbf{L}^{20})$	$(\mathbf{I}+\mathbf{R}^{13})(\mathbf{I}+\mathbf{L}^{5})$	155	
29	25	2	$(\mathbf{I}+\mathbf{L}^{10})$	$(\mathbf{I}+\mathbf{R}^{13})(\mathbf{I}+\mathbf{L}^{19})$	158	

means that there is significant dependence between the Hamming weights of successive numbers produced by these generators. Many of these generators also failed the matrix rank tests of Table II. For example, two of the generators with $rw = 25 \times 32 = 800$ failed Test 5 and three of those with $rw = 12 \times 32 = 384$ failed Test 4, both with a p-value smaller than 10^{-300}.

Three generators in Table III passed all the tests of SmallCrush and Crush: these are generators 22, 28, and 29. In view of the weaknesses of the other generators of the same family and same (or slightly smaller) period, we cannot recommend them for simulation purposes.

We also made an exhaustive search for good generators of type III with respect to Δ_1, with $w = 32$ and $r = 2, 3, 4, 5, 8, 12, 25$, and tested the best ones with SmallCrush and Crush. Almost all those with $r \leq 8$ have failed at least one of the tests, whereas all those with $r \geq 12$ passed all the tests in these batteries. The parameters of the latter generators are given in Table IV. All generators with $r = 25$ in that table have a smaller value of Δ_1 than TT800, but slightly larger than that for the best generators of type II. Although these generators behave better than those of type II in the statistical tests, we are

Table IV. Value of Δ_1 for the Best Generators of Type III with $w = 32$ and $r = 12, 25$

Num.	r	m_2	m_1	$\mathbf{H}_2^{a_2}$	$\mathbf{H}_1^{a_1}$	$\mathbf{H}_3^{a_3}$	Δ_1
31	12	2	3	$\mathbf{L}^7$	$\mathbf{R}^{11}$	$\mathbf{L}^{21}$	96
32	12	5	11	$\mathbf{L}^5$	$\mathbf{L}^{18}$	$\mathbf{R}^{11}$	100
33	12	7	9	$\mathbf{L}^{18}$	$\mathbf{R}^{11}$	$\mathbf{L}^5$	102
34	12	5	10	$\mathbf{L}^5$	$\mathbf{L}^{18}$	$\mathbf{R}^{11}$	103
35	25	4	10	$\mathbf{L}^{21}$	$\mathbf{R}^{11}$	$\mathbf{L}^7$	186
36	25	5	24	$\mathbf{L}^5$	$\mathbf{R}^{11}$	$\mathbf{L}^{18}$	188
37	25	7	24	$\mathbf{L}^5$	$\mathbf{R}^{11}$	$\mathbf{L}^{18}$	190
38	25	5	16	$\mathbf{L}^{19}$	$\mathbf{R}^{11}$	$\mathbf{L}^5$	219

hesitant to recommend them for simulation, because those of (already large) period lengths $2^{8 \times 32} - 1 = 2^{256} - 1$ fail the tests decisively, which is not reassuring.

7. INCREASING THE NUMBER OF XORSHIFTS

So far we have restricted ourselves to generators with only three xorshift operations, as in Marsaglia's article. An obvious idea for improving the statistical robustness is to increase the number of xorshifts. In this context, it is important to note that an a-bit left [right] xorshift does not modify the a least [most] significant bits. For this reason, if we decide to have several xorshifts and if we care about the least significant bits as well as the most significant ones, there should be a good balance between the numbers of left and right xorshifts. On the other hand, our criterion Δ_1 gives more importance to the most significant bits than to the least significant ones, because of the way equidistribution is defined. For this reason, computer searches based on Δ_1 tend to return generators having mostly left xorshifts and very few right xorshifts. To balance the two types of xorshifts in a search based on Δ_1, we must impose constraints on their respective numbers. Another possibility would be to modify the criterion Δ_1 to take into account the equidistribution of the low-order bits. We leave this as a topic for further investigation.

We performed a computer search for full-period xorshift generators with $w = 32$, order $r = 8$, $s = 7$ xorshifts with at least 3 right ones, and the smallest possible Δ_1. These generators have period length $2^{256}-1$ and the best we found have $\Delta_1 = 9$. One of them has recurrence $\mathbf{v}_n = (\mathbf{I} + \mathbf{L}^9)(\mathbf{I} + \mathbf{L}^{13})\mathbf{v}_{n-1} + (\mathbf{I} + \mathbf{L}^7)\mathbf{v}_{n-4} + (\mathbf{I} + \mathbf{R}^3)\mathbf{v}_{n-5} + (\mathbf{I} + \mathbf{R}^{10})\mathbf{v}_{n-7} + (\mathbf{I} + \mathbf{L}^{24})(\mathbf{I} + \mathbf{R}^7)\mathbf{v}_{n-8}$ and its characteristic polynomial $P(z)$ has 131 nonzero coefficients. A C implementation of this generator is given in Figure 1.

In a similar search for $s = 13$ and the same values for the other parameters, the best we found also has $\Delta_1 = 9$ and 129 nonzero coefficients in its characteristic polynomial. Its recurrence is: $\mathbf{v}_n = (\mathbf{I} + \mathbf{L}^{17})\mathbf{v}_{n-1} + (\mathbf{I} + \mathbf{L}^{10})\mathbf{v}_{n-2} + (\mathbf{I} + \mathbf{R}^9)(\mathbf{I} + \mathbf{L}^{17})\mathbf{v}_{n-4} + (\mathbf{I} + \mathbf{R}^3)\mathbf{v}_{n-4} + (\mathbf{I} + \mathbf{R}^{12})\mathbf{v}_{n-5} + (\mathbf{I} + \mathbf{R}^{25})\mathbf{v}_{n-5} + (\mathbf{I} + \mathbf{R}^3)(\mathbf{I} + \mathbf{R}^2)\mathbf{v}_{n-6} + (\mathbf{I} + \mathbf{R}^{27})\mathbf{v}_{n-7} + (\mathbf{I} + \mathbf{R}^{22})\mathbf{v}_{n-7} + (\mathbf{I} + \mathbf{L}^{24})(\mathbf{I} + \mathbf{R}^3)\mathbf{v}_{n-8}$. These two generators pass all the tests in Crush. On the other hand, they obviously fail matrix rank tests for $L > 256$ and tests based on linear complexity, because their bit sequences follow linear recurrences of order 256, modulo 2.

```
static unsigned int x[8];    /* Generator's state.*/

/* Initializes state.*/
void initxorshift7 (unsigned int *init) {
   int j;
   for (j=0; j<8; j++) x[j] = init[j];
}

/* Advances by one step and returns a number in [0,1).*/
double xorshift7 (void) {
   static int k = 0;
   unsigned int y, t;
   t = x[(k+7) & 0x7U];   t = t ^ (t<<13);   y = t ^ (t<<9);
   t = x[(k+4) & 0x7U];   y^= t ^ (t<<7);
   t = x[(k+3) & 0x7U];   y^= t ^ (t>>3);
   t = x[(k+1) & 0x7U];   y^= t ^ (t>>10);
   t = x[k];              t = t ^ (t>>7);    y^= t ^ (t<<24);
   x[k] = y;              k = (k+1) & 0x7U;
   return ((double) y  * 2.32830643653869628906e-10  );
}
```

Fig. 1. A C implementation of a seven-xorshift generator.

Table V. Values of Δ_1 for the Type II Generators of the Form (10) Proposed by Brent

r	m	a	b	c	d	Δ_1
2	1	17	14	12	19	7
4	3	15	14	12	17	34
8	3	18	13	14	15	58
16	1	17	15	13	14	142
32	15	19	11	13	16	141
64	59	19	12	14	15	465
128	95	17	12	13	15	845
132	67	15	14	13	18	1838
140	19	17	13	15	16	2038

For comparison, we made a similar search *without* any constraint on the number of right xorshifts, with $s = 13$ and $r = 7$. The best generator we found had $\Delta_1 = 1$ but a single right xorshift for 12 left xorshifts, and it failed test 4 in Table II.

Brent [2004b] recently proposed type II xorshift generators based on the recurrence

$$\mathbf{v}_i = (\mathbf{I} + \mathbf{R}^d)(\mathbf{I} + \mathbf{L}^c)\mathbf{v}_{i-m} \oplus (\mathbf{I} + \mathbf{R}^b)(\mathbf{I} + \mathbf{L}^a)\mathbf{v}_{i-r}, \qquad (10)$$

with four xorshift operations, and the parameters given in Table V. We computed the equidistribution of these generators and their values of Δ_1, given in the table, are not particularly good. We submitted them to the Crush battery, and the first two failed the matrix-rank tests of Table II, but the other ones (with $r \geq 8$ and period length $\rho \geq 2^{256} - 1$) passed all the tests. Brent recognizes the potential weaknesses of these xorshift generators and proposes an implementation that incorporates a combination with a Weyl generator, with the hope that this would improve the quality.

To see how the number of xorshifts affects the speed of the generators, we implemented generators of types I, II, and III with three xorshifts, as well as the generators described above with 4, 7, and 13 xorshifts. To generate 10^9 (one billion) uniform random numbers in $[0, 1)$ and add them up, it took approximately 32 seconds for the fastest generators (type I with three xorshifts) and 38 seconds with the slowest one (13 xorshifts). So adding xorshifts does not slow down the generator significantly in comparison with the time for the function call, transformation into a real number, and adding the numbers. The timings with the Mersenne twister of Matsumoto and Nishimura [1998] and the WELL generators of Panneton and L'Ecuyer [2004] are about the same. These timings are for a 2.8 Ghz Intel Pentium 4 processor running Linux and the `gcc` compiler with the `-O2` optimization flag.

8. CONCLUSION

We have studied, both theoretically and empirically, the xorshift generators proposed by Marsaglia. These generators are fast, but not reliable, according to our analysis. Those of type I are doomed right from the start because of their short period length. The others also have problems when the number of xorshifts is restricted to three. To get over these limitations, one may want to use a larger number of xorshifts together with a long period. To get rid of the linear structure of the output sequence and further improve the statistical robustness, these xorshift generators could be combined with other RNGs from different classes. Preferably, this should be supported by some theoretical analysis of the point set Ψ_t *of the combined generator*, for example, as in L'Ecuyer and Granger-Piché [2003].

ACKNOWLEDGMENTS

We thank Richard Simard for his help in improving the presentation and running the statistical tests.

REFERENCES

BRENT, R. P. 2004a. Note on Marsaglia's xorshift random number generators. *J. Stat. Soft. 11*, 5, 1–4. See `http://www.jstatsoft.org/v11/i05/brent.pdf`.

BRENT, R. P. 2004b. Some uniform and normal random number generators. `http://web.comlab.ox.ac.uk/oucl/work/richard.brent/random.html`.

FUSHIMI, M. AND TEZUKA, S. 1983. The k-distribution of generalized feedback shift register pseudorandom numbers. *Comm. ACM 26*, 7, 516–523.

KNUTH, D. E. 1998. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*, Third ed. Addison-Wesley, Reading, Mass.

L'ECUYER, P. 1996. Maximally equidistributed combined Tausworthe generators. *Mathematics of Computation 65*, 213, 203–213.

L'ECUYER, P. 2004. Random number generation. In *Handbook of Computational Statistics*, J. E. Gentle, W. Haerdle, and Y. Mori, Eds. Springer-Verlag, Berlin, 35–70. Chapter II.2.

L'ECUYER, P. AND GRANGER-PICHÉ, J. 2003. Combined generators with components from different families. *Mathematics and Computers in Simulation 62*, 395–404.

L'ECUYER, P. AND SIMARD, R. 1999. Beware of linear congruential generators with multipliers of the form $a = \pm 2^q \pm 2^r$. *ACM Trans. Math. Soft. 25*, 3, 367–374.

L'ECUYER, P. AND SIMARD, R. 2001a. On the performance of birthday spacings tests for certain families of random number generators. *Mathematics and Computers in Simulation 55*, 1–3, 131–137.

L'ECUYER, P. AND SIMARD, R. 2001b. TestU01: A software library in ANSI C for empirical testing of random number generators. Software User's Guide. Available at: http://www.iro.umontreal.ca/~lecuyer.

LIDL, R. AND NIEDERREITER, H. 1994. *Introduction to Finite Fields and Their Applications*, Revised Ed. Cambridge University Press, Cambridge.

MARSAGLIA, G. 1985. A current view of random number generators. In *Computer Science and Statistics, Sixteenth Symposium on the Interface*. Elsevier Science Publishers, North-Holland, Amsterdam, 3–10.

MARSAGLIA, G. 2003. Xorshift RNGs. *J. Stat. Soft. 8*, 14, 1–6. See `http://www.jstatsoft.org/v08/i14/xorshift.pdf`.

MATOUSĚK, J. 1998. On the L_2-discrepancy for anchored boxes. *J. Complexity 14*, 527–556.

MATSUMOTO, M. AND KURITA, Y. 1992. Twisted GFSR generators. *ACM Trans. Model. Comput. Simul. 2*, 3, 179–194.

MATSUMOTO, M. AND KURITA, Y. 1994. Twisted GFSR generators II. *ACM Trans. Model. Comput. Simul. 4*, 3, 254–266.

MATSUMOTO, M. AND NISHIMURA, T. 1998. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul. 8*, 1, 3–30.

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

NIEDERREITER, H. 1995. The multiple-recursive matrix method for pseudorandom number generation. *Finite Fields and their Applications 1*, 3–30.

OWEN, A. B. 1997. Monte Carlo variance of scrambled equidistribution quadrature. *SIAM J. Numer. Anal. 34*, 5, 1884–1910.

PANNETON, F. 2004. Construction d'ensembles de points basée sur des récurrences linéaires dans un corps fini de caractéristique 2 pour la simulation Monte Carlo et l'intégration quasi-Monte Carlo. Ph.D. thesis, Département d'informatique et de recherche opérationnelle, Université de Montréal, Canada.

PANNETON, F. AND L'ECUYER, P. 2004. Improved long-period generators based on linear recurrences modulo 2. Manuscript.

Received November 2004; revised October 2005; accepted October 2005