On the Deng-Lin Random Number Generators and Related Methods

Pierre L'Ecuyer and Renée Touzin
Département d'informatique et de recherche opérationnelle,
Université de Montréal, C.P. 6128, Succ. Centre-Ville, Montréal, H3C 3J7, Canada.
`email: lecuyer@iro.umontreal.ca`
`http://www.iro.umontreal.ca/∼lecuyer`

May 2, 2003

Abstract. We study the structure and point out weaknesses of recently-proposed random number generators based on special types of linear recurrences with small coefficients, which allow fast implementations. Our theoretical analysis is complemented by the results of simple empirical statistical tests that the generators fail decisively. Directions for improvement and alternative generators are also pointed out.

KEY WORDS: multiple recursive generator (MRG); lattice structure; efficient generator; statistical test of randomness.

1 Introduction

A *random number generator* (RNG) is a small deterministic computer program whose aim is to produce a sequence of numbers $\{u_i,\ i \geq 0\}$ that behaves somewhat like a typical realization of a sequence of independent random variables uniformly distributed over the interval $(0, 1)$ (i.i.d. $U(0, 1)$, for short). A popular type of RNG is the *multiple recursive (linear congruential) generator* (MRG), based on the linear recurrence

$$x_i \;=\; (a_1 x_{i-1} + \cdots + a_k x_{i-k}) \bmod m, \tag{1}$$

for some positive integers m (the modulus), k (the order), and a_j's in $\{0, 1, \ldots, m-1\}$ (the coefficients). The state at step i is $s_i = (x_{i-k+1}, \ldots, x_i)$ and the output can be defined by $u_i = x_i/m$. Usually, m is chosen as a prime number and the coefficients a_j's are selected so that the characteristic polynomial of the recurrence (1) is a primitive polynomial; then, if $s_0 \neq 0$, the output sequence is periodic with period length $\rho = m^k - 1$, its largest possible value (Knuth 1998). The MRG has been studied by several authors, including Grube (1973), L'Ecuyer, Blouin, and Couture (1993), and L'Ecuyer (1996).

Deng and Lin (2000) have proposed a special case which they call *fast* MRG (FMRG), where (1) has the form

$$x_i \;=\; ((m-1)x_{i-1} + ax_{i-k}) \bmod m \;=\; (-x_{i-1} + ax_{i-k}) \bmod m, \tag{2}$$

as RNG "for the new century." The generator is fast because it requires a single multiplication (of x_{i-k} by a). They provide explicit values of a for $m = 2^{31} - 1$ and $k = 2$, 3, and 4. Deng and Xu (2002) proposed another variant called HELP-k, where all nonzero coefficients a_j's in (1) are equal to some integer a (so a single multiplication is required). They provide tables of values of a for $m = 2^{31} - 1$, $k = 102$ and $k = 120$, and recurrences of the special forms:

$$x_i \;=\; a(x_{i-1} + x_{i-k}) \bmod m, \tag{3}$$

$$x_i \;=\; a(x_{i-1} + x_{\lfloor i-k/2\rfloor} + x_{i-k}) \bmod m, \tag{4}$$

$$x_i \;=\; a(x_{i-1} + x_{\lfloor i-k/3\rfloor} + x_{\lfloor i-2k/3\rfloor} + x_{i-k}) \bmod m, \tag{5}$$

and (2). For all the parameters they give, the period length is $m^k - 1$ and $a < \sqrt{m}$ (to make the implementation faster). Marsaglia (1996) proposed similar generators, based on the recurrences

$$x_i \;=\; 2^{10}(x_{i-1} + x_{i-2} + x_{i-3}) \bmod (2^{32} - 5) \tag{6}$$

and

$$x_i \;=\; 2^{20}(x_{i-1} + x_{i-2} + x_{i-3}) \bmod (2^{32} - 209). \tag{7}$$

Here the product can be implemented simply via a binary shift, because a is a power of 2.

In the remainder of this paper, we study the structure and point out weaknesses of these types of generators, both theoretically and empirically. In Section 2, we discuss their lattice structures and provide bounds on their best possible performance in the spectral test, showing that they cannot perform well, especially when a is small. In Section 3, we apply well-known empirical statistical tests to selected instances of these generators. They fail the tests decisively. Improvements and alternatives are pointed out in Section 4.

2 Lattice structure of MRGs

Select a finite set of integer indices $I = \{i_1, \ldots, i_t\}$, where $0 \le i_1 < \cdots < i_t$, for some positive integer t. For a given MRG, consider the multiset $\Psi_t(I)$ of all t-dimensional output vectors $(u_{i_1}, \ldots, u_{i_t})$ obtained when the initial state $s_0 = (x_0, \ldots, x_{k-1})$ runs over all its m^k possibilities. If s_0 is picked at random (uniformly), then $(u_{i_1}, \ldots, u_{i_t})$ has the uniform distribution over $\Psi_t(I)$. So, roughly, $u_{i_1}, \ldots, u_{i_t}$ will be approximately i.i.d. $U(0,1)$ if and only if $\Psi_t(I)$ covers the unit hypercube $(0,1)^t$ very uniformly. It is well known that $\Psi_t(I)$ is the intersection of a *lattice* in $\mathbb{R}^t$ with the unit hypercube $[0,1)^t$ (Knuth 1998; L'Ecuyer 1997). This implies that there are families of equidistant parallel hyperplanes in $\mathbb{R}^t$ such that each family covers $\Psi_t(I)$, and one can actually compute the distance $d_t(I)$ between the hyperplanes for the family for which this distance is largest. This is called the *spectral test* (Knuth 1998). More specifically, $1/d_t(I)$ is the Euclidean length of a shortest nonzero vector in the *dual* of the lattice generated by $\Psi_t(I)$. Computing this length can be formulated as a quadratic integer programming problem, which can be solved by a branch-and-bound algorithm (L'Ecuyer and Couture 1997). To avoid thick slices of empty space untouched by $\Psi_t(I)$, we want $d_t(I)$ to be as small as possible. On the other hand, there is a minimal possible value of this distance for a general lattice in $\mathbb{R}^t$ with N points per unit of volume. The exact minimum is known for $t \leq 8$ and tight lower bounds are available for $t > 8$ (Conway and Sloane 1999; L'Ecuyer 1999b). Denote these values by $d_t^*(N)$. For $t \leq 8$, one has $1/d_t^*(N) = \gamma_t N^{1/t}$ where $\gamma_2 = (4/3)^{1/4}$, $\gamma_3 = 2^{1/6}$, $\gamma_4 = 2^{1/4}$, $\gamma_5 = 2^{3/10}$, etc. (Knuth 1998). The number of points per unit of volume in $\Psi_t(I)$ is at most $N = m^k$ if $t \geq k$ and $N = m^t$ if $t < k$. To measure the quality of $\Psi_t(I)$ for $t \geq k$, it is customary to use the standardized number $S_t(I) = d_t^*(N)/d_t(I)$, which lies between 0 and 1, and should be as large as possible. Good MRGs have been constructed for which $S_t(I)$ is large for all sets $I \in \mathcal{I}$, where $\mathcal{I}$ is a selected family of index sets; e.g., the set of all $I = \{0, \ldots, t-1\}$ for $t \leq t_0$ for some t_0 which may go up to 40 or more (L'Ecuyer, Blouin, and Couture 1993; L'Ecuyer 1999a). For example, the generator MRG32k3a proposed by L'Ecuyer (1999a) has period length near 2^{191} and $\min_{t \leq 45} S_t(\{0, \ldots, t-1\}) \approx 0.6225$.

Proposition 2 of L'Ecuyer (1997) gives the following lower bound on $d_t(I)$ when I contains k and all j's such that $a_{k-j} \neq 0$:

$$d_t(I) \geq (1 + a_1^2 + \cdots + a_k^2)^{-1/2}. \tag{8}$$

This implies that to have a good quality MRG, it is *necessary* (but not sufficient) that the sum of squares of the coefficients be large.

For the FMRG (2), if we assume that $a^2 < m$ as did Deng and Lin (2000), (8) gives the bounds $d_t(I) \geq (2 + a^2)^{-1/2} \geq (1 + m)^{-1/2}$ and $S_t(I) \leq (1 + m)^{1/2} m^{-\min(k,t)/t}/\gamma_t$ whenever $\{0, k-1, k\} \subseteq I$. For $I = \{0, k-1, k\}$, this gives $S_3(I) \leq 2^{-1/6}(1 + m)^{1/2} m^{-\min(k,3)/3}$. For $m = 2^{31} - 1$, this upper bound evaluates to $S_t(\{0, 1, 2\}) \leq 0.024803$ and $S_t(\{0, k-1, k\}) \leq 1.9224 \times 10^{-5}$ for $k \geq 3$. In other words, an MRG defined by (2) *cannot* perform well in the spectral test for $\Psi_t(I)$. For successive indices, we have $S_4(\{0, 1, 2, 3\}) \leq 0.003906$ for $k = 3$ and $S_5(\{0, 1, 2, 3, 4\}) \leq 0.001289$ for $k = 4$.

The actual values of $S_3(\{0, 1, 2\})$ for the values of a given by Deng and Lin (2000) for $k = 2$

in their Table 1 are all equal to $2^{-1/6}(2+a^2)^{1/2}m^{-2/3}$. They go from $S_3(\{0,1,2\}) = 0.01413$ for $a = 26403$ to $S_3(\{0,1,2\}) = 0.02480$ for $a = 46338$. For $k = 3$ and $k = 4$, the values of $S_t(\{0, k-1, k\})$ are all smaller than 1.9224×10^{-5}.

Deng and Lin (2000) also propose random *vector* generators based on matrix linear recurrences. These generators are in fact equivalent to running in parallel several copies of the same MRG of the form (2), one for each vector coordinate, with different initial states. Thus, these matrix generators have the same weaknesses as the FMRGs.

Consider now an MRG with recurrence

$$x_i \quad = \quad a(x_{i-i_2} + \cdots + x_{i-i_t}) \bmod m, \tag{9}$$

where $0 < i_2 < \cdots < i_t = k$ (hence $t \le k$), and let $I = \{0, k-i_{t-1}, \ldots, k-i_2, k\}$. Applying (8) to (9) yields

$$d_t(I) \ge (1 + (t-1)a^2)^{-1/2} \tag{10}$$

and then

$$S_t(I) \le (1 + (t-1)a^2)^{1/2}/(m\gamma_t). \tag{11}$$

For the special cases of (3) to (5) with $k = 102$ or $k = 120$, and $a^2 < m$, this gives

$$S_t(I) \le \frac{(1 + (t-1)(m-1))^{1/2}}{m\gamma_t} < \frac{\sqrt{t-1}}{\gamma_t\sqrt{m}} \tag{12}$$

where $t = 3$ and $I = \{0, k-1, k\}$ for (3), $t = 4$ and $I = \{0, \lfloor k/2 \rfloor, k-1, k\}$ for (4), $t = 5$ and $I = \{0, \lfloor k/3 \rfloor, \lfloor 2k/3 \rfloor, k-1, k\}$ for (5). For $m = 2^{31} - 1$, this inequality becomes $S_3(I) \le 2.719 \times 10^{-5}$ for $t = 3$, $S_4(I) \le 3.143 \times 10^{-5}$ for $t = 4$, and $S_5(I) \le 3.505 \times 10^{-5}$ for $t = 5$. If a is much smaller than $\sqrt{m}$, the bound (11) is much tighter. For example, the smallest value of a suggested by Deng and Xu (2002) for (3) with $k = 102$ is $a = 23$. In this case, (11) gives $S_3(I) \le 1.4 \times 10^{-8}$, which means that the lattice structure is extremely poor.

Equation (9) can also be written as

$$a_* x_i \bmod m \quad = \quad (x_{i-i_2} + \cdots + x_{i-i_t}) \bmod m, \tag{13}$$

where a_* is the inverse of a modulo m, i.e., the integer in $\{1, \ldots, m-1\}$ such that $a_*a - 1$ is a multiple of m, which exists when m and a have no common factor. Then, based on a similar argument as in the proof of Proposition 1 of L'Ecuyer (1997), one can easily show that the bound (10) can be complemented by

$$d_t(I) \ge (a_*^2 + t - 1)^{-1/2} \tag{14}$$

(by noticing that the vector $(a_*, 1, \ldots, 1)$ belongs to the dual lattice). This bound would be large whenever a_* happens to be small. For example, for $a = 840319688$, $a_* = 23$, a small value, which illustrates the fact that a large a is *not sufficient*.

If we remove the condition $a^2 < m$, the "fast" implementation method suggested by Deng and Lin (2000) no longer applies. It is nevertheless instructive to examine what happens to the quality of the lattice structure. The lower bound on $d_t(I)$ obtained via (8) can obviously be much smaller and the corresponding upper bound on $S_t(I)$ can be much larger. For example, for the FMRG with $m = 2^{31} - 1$, this gives a bound on $S_t(\{0, k-1, k\})$ larger than 1 (useless) for $k = 2$ and near 0.89 for $k \geq 3$. However, the following more detailed analysis provides tighter bounds and shows that the quality cannot be good even for larger a.

Consider the lattice structure of the FMRG (2) with $t = 3$, and $I = \{0, k-1, k\}$. In this case, we recall (see, e.g., L'Ecuyer and Couture 1997) that the dual lattice is the set $\{\mathbf{w} = z_1\mathbf{w}_1 + z_2\mathbf{w}_2 + z_3\mathbf{w}_3$ such that each z_j is an integer$\}$, where $\mathbf{w}_1 = (m, 0, 0)$, $\mathbf{w}_2 = (0, m, 0)$, and $\mathbf{w}_3 = (1, -a, 1)$. This implies in particular that (putting $z_1 = 0$) every vector of the form $\mathbf{w} = (z_3, z_2m - z_3a, z_3)$ belongs to the dual lattice. This vector has square length $2z_3^2 + (z_2m - z_3a)^2$. Therefore, $1/d_t^2(I) \leq 2z_3^2 + (z_2m - z_3a)^2$ for every pair of integers $(z_2, z_3) \neq (0, 0)$. Taking $(z_2, z_3) = (0, 1)$ gives the bound (8). Other choices of (z_2, z_3) give the following bounds:

$$\begin{aligned}
(z_2, z_3) = (0, 1) \quad &\rightarrow \quad 1/d_t^2(I) \leq a^2 + 2, \\
(z_2, z_3) = (1, 1) \quad &\rightarrow \quad 1/d_t^2(I) \leq (m - a)^2 + 2, \\
(z_2, z_3) = (1, 2) \quad &\rightarrow \quad 1/d_t^2(I) \leq (m - 2a)^2 + 8, \\
(z_2, z_3) = (1, 3) \quad &\rightarrow \quad 1/d_t^2(I) \leq (m - 3a)^2 + 18, \\
(z_2, z_3) = (2, 3) \quad &\rightarrow \quad 1/d_t^2(I) \leq (2m - 3a)^2 + 18, \\
(z_2, z_3) = (1, 4) \quad &\rightarrow \quad 1/d_t^2(I) \leq (m - 4a)^2 + 32, \\
(z_2, z_3) = (3, 4) \quad &\rightarrow \quad 1/d_t^2(I) \leq (3m - 4a)^2 + 32,
\end{aligned}$$

and so on. These bounds indicate that d_t will be large if a is close to 0, or m, or $m/2$, or $m/3$, or $2m/3$, etc. By taking the minimum of all these bounds, over all pairs (z_2, z_3), one obtains a much tighter bound than (8). For $m = 2^{31} - 1$, this gives approximately $d_t \geq 1.689 \times 10^{-5}$ and, for $k = 2$, $S_3(\{0, 1, 2\}) \leq 0.03170$. This value is approximately reached if we take $a = 10934394$. For $k \geq 3$, we obtain $S_3(\{0, k-1, k\}) \leq 2.457 \times 10^{-5}$. These bounds are almost the same as when $a^2 < m$. In other words, even if we relax the condition $a^2 < m$, finding a FMRG with a good lattice structure remains hopeless.

For the MRGs proposed by Marsaglia, with $I = \{0, 1, \ldots, t-1\}$, one has $d_4(I) = 5.638 \times 10^{-4}$ and $S_4(I) = 8.890 \times 10^{-5}$ for (6), whereas $d_4(I) = 2.432 \times 10^{-4}$ and $S_4(I) = 2.061 \times 10^{-4}$ for (7). Thus, these two generators have a poor lattice structure in four dimensions.

3 Statistical testing

We now illustrate to what extent simple empirical statistical tests can detect the weaknesses studied theoretically in Section 2. We have tested 11 generators, defined as follows. DL00a1, DL00a2, DL00a3, DL00b, and DL00c denote the FMRG based on (2) with $m = 2^{31} - 1$ and $(k, a) = (2, 26403)$, $(2, 39613)$, $(2, 46338)$, $(3, 21960)$, and $(4, 22093)$, respectively. DX02a and DX02b represent (3) with $m = 2^{31} - 1$, $k = 102$, and $a = 23$ and 45787, respectively. All these parameters are taken from the tables of Deng and Lin (2000) and Deng and Xu (2002). For the FMRGs with $k = 2$, we took the smallest and largest values of a, as well as the value $a = 39613$ chosen in Eq. (12) and Figure 1 of Deng and Lin (2000). DX02as and DX02bs are versions of DX02a and DX02b for which we first generate 1 number, skip the next 100, then we generate 3 numbers, skip the next 100, generate another 3 numbers, skip the next 100, and so on. By doing this, the three-dimensional vectors produced by the generator will have the form $(u_i, u_{i+101}, u_{i+102})$ in terms of the output sequence $\{u_i,\ i \geq 0\}$ of the original generator. MAR96a and MAR96b are the MRGs (6) and (7).

We report results for two well-known tests, namely the maximum-of-t and the birthday spacings tests (Knuth 1998; L'Ecuyer and Simard 2001). For each test, we select two positive integers n and t, and we generate n points "independently" in the t-dimensional unit hypercube $[0, 1)^t$, by calling the RNG t times for each point (once for each coordinate).

For the *maximum-of-t* test, let X_i be the largest coordinate of point i, for $i = 1, \ldots, n$. The test compares the empirical distribution of $X_1, \ldots, X_n$ with its theoretical distribution $F(x) = x^t$ under the null hypothesis $\mathcal{H}_0$ that the RNG produces i.i.d. $U(0, 1)$ random variables. As suggested by Knuth (1998), page 70, the comparison is made via a chi-square test, by partitioning $[0, 1)$ into d intervals in a way that the expected number of values of X_i in each interval is exactly n/d under $\mathcal{H}_0$, and comparing the expected numbers with the observed numbers in all intervals. For all the tests reported here, we chose d so that $n/d = 16$.

For the *birthday spacings test*, we partition $[0, 1)^t$ into $k = 2^{bt}$ cubic boxes of equal size by dividing the interval $[0, 1)$ into 2^b equal parts, for some integer b. These boxes are numbered from 0 to $k - 1$, in lexicographic order of the coordinates. Let $I_{(1)} \leq I_{(2)} \leq \cdots \leq I_{(n)}$ be the box numbers where the n points fall, sorted by increasing order, and define the *spacings* $S_j = I_{(j+1)} - I_{(j)}$, for $j = 1, \ldots, n-1$. Let Y be the number of values of $j \in \{1, \ldots, n-2\}$ such that $S_{(j+1)} = S_{(j)}$, where $S_{(1)}, \ldots, S_{(n-1)}$ are the spacings sorted by increasing order. This is the number of collisions between the spacings. Under $\mathcal{H}_0$, Y is approximately a Poisson random variable with mean $\lambda_2 = n^3/(4k)$ if k is large while λ_2 is not too large (L'Ecuyer and Simard 2001). If y denotes the observed value of Y, then the right p-value of the test is $p^+ \overset{\text{def}}{=} P[Y \geq y \mid Y \sim \text{ Poisson}(\lambda_2)]$.

Table 1: The right p-values for some maximum-of-t tests

RNG	t	n	p^+	CPU time (sec)
DL00a1	3	2^{19}	0.29	0.68
DL00a1	3	2^{20}	$< 10^{-15}$	1.34
DL00a2	3	2^{20}	2.8×10^{-4}	1.35
DL00a2	3	2^{21}	$< 10^{-15}$	2.89
DL00a3	3	2^{20}	0.20	1.34
DL00a3	3	2^{21}	$< 10^{-15}$	2.88
DL00b	4	2^{19}	0.88	0.72
DL00b	4	2^{20}	3.8×10^{-5}	1.52
DL00b	4	2^{21}	$< 10^{-15}$	3.20
DL00c	5	2^{21}	0.03	3.40
DL00c	5	2^{22}	$< 10^{-15}$	7.09

Tables 1 and 2 give the right p-values for selected RNGs and parameter sets, as we increase the value of n. We also report the CPU time it took to run each test, on a PC with a 1733 Mhz AMD Athlon processor running Linux. Most of these tests took only a few seconds or less. As predicted by the theory, DL00a1, DL00a2, DL00a3, DX02as, and DX02bs fail some tests in 3 dimensions, DL00b, MAR96a, and MAR96b fail in 4 dimensions, and DL00c fails in 5 dimensions. These failures are clear and spectacular: in each case, we have p-values smaller than 10^{-15}. Generally speaking, whenever $p^+ < 10^{-15}$ at some sample size n, the p-value remains smaller than 10^{-15} for all larger values of n. For example, DL00a1 passes the birthday spacings test with $t = 3$ for sample size $n = 2^{18}$, gives a suspect p-value of $p^+ \approx 5 \times 10^{-6}$ for $n = 2^{19}$, and fails with a p-value smaller than $p^+ < 10^{-15}$ for $n = 2^{20}$ and all larger values of n. DL00a1 also fails the maximum-of-t test with $p^+ < 10^{-15}$ for all $n \geq 2^{20}$. These generators also fail other tests such as the *collision* test and *close-pair* tests, whose results are not reported here, for the same values of t. DX02a and DX02b passed all the tests we tried.

4 Improvements and alternatives

The idea of taking many coefficients a_j equal to a common value in recurrence (1), in order to improve the implementation speed, is certainly valuable. However, this must be done in a way that allows the sum of squares of the a_j's to be large and the resulting generators must be submitted to a lattice structure analysis via the spectral test. Having few nonzero a_j's and taking them small is a bad idea. Taking a *single* large a_j, as in (2), also leads to trouble.

As alternatives to the RNGs examined in this paper, we mention the generators LFSR113

Table 2: The right p-values for some birthday spacings tests

RNG	t	n	b	p^+	CPU time (sec)
DL00a1	3	2^{18}	17	0.32	0.29
DL00a1	3	2^{19}	17	5.0×10^{-6}	0.62
DL00a1	3	2^{20}	17	$< 10^{-15}$	1.20
DL00a2	3	2^{18}	17	0.32	0.26
DL00a2	3	2^{19}	17	1.2×10^{-5}	0.61
DL00a2	3	2^{20}	17	$< 10^{-15}$	1.30
DL00a3	3	2^{18}	17	0.14	0.30
DL00a3	3	2^{19}	17	8.3×10^{-7}	0.59
DL00a3	3	2^{20}	17	$< 10^{-15}$	1.26
DL00b	4	2^{23}	16	0.03	12.75
DL00b	4	2^{24}	16	3.1×10^{-13}	27.06
DL00b	4	2^{25}	16	$< 10^{-15}$	57.11
DX02as	3	2^{10}	10	0.03	0.01
DX02as	3	2^{11}	10	$< 10^{-15}$	0.02
DX02bs	3	2^{18}	18	0.22	2.43
DX02bs	3	2^{19}	18	1.4×10^{-6}	4.90
DX02bs	3	2^{20}	18	$< 10^{-15}$	9.68
MAR96a	4	2^{19}	13	0.06	0.66
MAR96a	4	2^{20}	13	1.5×10^{-13}	1.38
MAR96a	4	2^{21}	13	$< 10^{-15}$	2.92
MAR96b	4	2^{20}	13	0.10	1.36
MAR96b	4	2^{21}	13	$< 10^{-15}$	2.93

of L'Ecuyer (1999c), MRG32k3a of L'Ecuyer (1999a), MRG31k3p of L'Ecuyer and Touzin (2000), and MT19937 of Matsumoto and Nishimura (1998). These generators easily pass all statistical tests discussed above, for the largest sample size n that we can handle. They are also reasonably fast. To give a concrete idea, our fastest implementation of DL00a1 generates one billion (10^9) random numbers is approximately 23 seconds on our 1733 Mhz Athlon PC running Linux, using the gcc compiler with full optimization (option `-O3`) with its fast `fmod` function for the modulo operations. For comparison, LFSR113, MRG32k3a, MRG31k3p, and MT19937 need approximately 7, 79, 59, and 37 seconds, respectively, for the same task.

MRG32k3a and MRG31k3p belong to a class of *combined* MRGs designed explicitly to perform very well in the spectral test while allowing efficient and portable implementations. These generators are actually MRGs whose coefficients a_j are large and are selected in a way that the MRGs can be decomposed into components having a fast implementation. There are certainly other efficient ways of implementing MRGs with large coefficients. Each method usually imposes some special conditions on these coefficients. Good parameter sets can be found by making computer searches, within the class of MRGs that satisfy these conditions, to find instances having maximal period and good performance in the spectral test. This was done for combined MRGs in L'Ecuyer (1999a) and could be done in a similar way for other types of implementations or special cases, e.g., for MRGs whose nonzero coefficients a_j are all equal. This is an interesting avenue for further work.

Acknowledgements

This work has been supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Grant No. ODGP0110050, NATEQ-Québec grant No. 02ER3218, and a Killam Research Fellowship to the first author. The work of R. Touzin was supported by a NSERC scholarship. The authors thank Richard Simard, who ran the statistical tests, and an anonymous reviewer whose comments helped improving the paper.

REFERENCES

Conway, J. H., and N. J. A. Sloane. 1999. *Sphere packings, lattices and groups*. 3rd ed. Grundlehren der Mathematischen Wissenschaften 290. New York: Springer-Verlag.

Deng, L.-Y., and D. K. J. Lin. 2000. Random number generation for the new century. *The American Statistician* 54 (2): 145–150.

Deng, L.-Y., and H. Xu. 2002. Design, search, and implementation of high-dimensional, efficient, long-cycle, and portable uniform random variate generator. Technical report, Department of Statistics, University of California at Los Angeles. Preprint #327.

Grube, A. 1973. Mehrfach rekursiv-erzeugte Pseudo-Zufallszahlen. *Zeitschrift für angewandte Mathematik und Mechanik* 53:T223–T225.

Knuth, D. E. 1998. *The art of computer programming, volume 2: Seminumerical algorithms*. Third ed. Reading, Mass.: Addison-Wesley.

L'Ecuyer, P. 1996. Combined multiple recursive random number generators. *Operations Research* 44 (5): 816–822.

L'Ecuyer, P. 1997. Bad lattice structures for vectors of non-successive values produced by some linear recurrences. *INFORMS Journal on Computing* 9 (1): 57–60.

L'Ecuyer, P. 1999a. Good parameters and implementations for combined multiple recursive random number generators. *Operations Research* 47 (1): 159–164.

L'Ecuyer, P. 1999b. Tables of linear congruential generators of different sizes and good lattice structure. *Mathematics of Computation* 68 (225): 249–260.

L'Ecuyer, P. 1999c. Tables of maximally equidistributed combined LFSR generators. *Mathematics of Computation* 68 (225): 261–269.

L'Ecuyer, P., F. Blouin, and R. Couture. 1993. A search for good multiple recursive random number generators. *ACM Transactions on Modeling and Computer Simulation* 3 (2): 87–98.

L'Ecuyer, P., and R. Couture. 1997. An implementation of the lattice and spectral tests for multiple recursive linear random number generators. *INFORMS Journal on Computing* 9 (2): 206–217.

L'Ecuyer, P., and R. Simard. 2001. 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 R. Touzin. 2000. Fast combined multiple recursive generators with multipliers of the form $a = \pm 2^q \pm 2^r$. In *Proceedings of the 2000 Winter Simulation Conference*, ed. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 683–689. Pistacaway, NJ: IEEE Press.

Marsaglia, G. 1996. The Marsaglia random number CDROM including the DIEHARD battery of tests of randomness. See `http://stat.fsu.edu/pub/diehard`.

Matsumoto, M., and T. Nishimura. 1998. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation* 8 (1): 3–30.