

Combined Generators with Components from Different Families

Pierre L'Ecuyer and Jacinthe Granger-Piché

lecuyer@iro.umontreal.ca, granger@iro.umontreal.ca

<http://www.iro.umontreal.ca/~lecuyer>

*Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville
Montréal, H3C 3J7, Canada*

Abstract

Most random number generators used in practice are based on linear recurrences, with linear output transformations. This gives long periods, fast implementations, and structures that are easy to analyze. But the points produced by these generators have very regular structures. Nonlinear generators can have less regular structures, but they are generally slower and much harder to analyze when their period is long.

In this paper, combined generators with one large linear component, and a second component of a different type (nonlinear or linear), are proposed and studied. The structure of vectors of successive and non-successive output values produced by the combined generators is analyzed. Under mild conditions, these vector sets are proved to have at least as much uniformity than the corresponding sets for the linear component alone. In empirical statistical tests, these combined generators perform better than simple linear generator of comparable period lengths, because of their less regular structure. Efficient implementation methods are suggested.

Key words: random numbers, uniformity, combined generators, statistical tests

1 Introduction

Combined random number generators with components from the same family have been studied extensively. Specific generators of this form are now available in software packages. Examples include linear congruential generators

¹ This work has been supported by NSERC-Canada Grant no. ODGP0110050 and FCAR-Québec grant no. ER-3218 to the first author, and by an NSERC scholarship to the second author

(LCGs) and multiple recursive generators (MRGs) combined with a modulo 1 addition [6] and Tausworthe or linear feedback shift register (LFSR) generators combined via a bitwise exclusive-or [5,8,18]. The theoretical properties of these linear generators are easy to analyze because they have the same type of highly regular structure as their components. Having a lot of structure is convenient from the analysis viewpoint but becomes a drawback from the “apparent randomness” or “unpredictability” viewpoint. It is then interesting to explore how much must be given away in terms of our understanding of the structure and in the “guaranteed uniformity” of the point set produced by the generator over its full period, in order to obtain a more complicated (or less regular) structure.

In this paper, we consider combined generators with components taken from different families, from the theoretical and empirical viewpoints. Our goal is to construct combined generators with good (and guaranteed) uniformity properties, less regular structure than purely linear generators, and fast implementations. We concentrate on the following two main classes of combinations: (a) a LCG or MRG combined with another type of generator by adding the outputs modulo 1, and (b) a LFSR generator combined with another type of generator via a bitwise exclusive-or. The second type of generator used in the combination can be nonlinear, with a complicated structure. Its purpose is to scramble the regularity and increase the apparent randomness.

We analyze the structural properties of sets of t -tuples formed by successive output values, or by non-successive output values at specified lags, for these combined generators. Measuring the uniformity of such sets is a standard way of assessing the approximate uniformity and independence of the output values [11]. We show that a certain level of uniformity for these sets can be guaranteed if the corresponding point sets produced by the LCG, MRG, or LFSR are well distributed. On the other hand, these point sets have a much less regular structure than the corresponding point sets for LCG, MRG, or LFSR. This shows up in empirical testing: Standard statistical tests require a larger sample size to detect the structure, or to find statistical deficiencies of the generator, for a mixed combined generator than for a simple linear generator of comparable period length. In our experiments, this trend is remarkably systematic.

In the next section, we recall properties of two classes of linear generators. In Sections 3 and 4, we analyze the structural properties of combined generators with one MRG component and with one LFSR component, respectively. Section 5 summarizes our statistical experiments and Section 6 discusses implementation issues. Further details can be found in [3].

2 Linear Generators and their Measures of Uniformity

A MRG is defined by the recurrence

$$x_n = (a_1x_{n-1} + \cdots + a_kx_{n-k}) \bmod m; \quad (1)$$

$$u_n = x_n/m, \quad (2)$$

where the *modulus* m and the *order* k are positive integers, the *coefficients* a_i belong to $\mathbb{Z}_m = \{0, 1, \dots, m-1\}$, and $u_n \in [0, 1)$ is the output at step n . For prime m and properly chosen a_i 's, the recurrence has period length $\rho = m^k - 1$ [4]. We have a LCG when $k = 1$.

For any fixed set of non-negative integers $I = \{i_1, i_2, \dots, i_t\}$, define

$$\Psi_I = \{\mathbf{u}_I = (u_{i_1}, \dots, u_{i_t}) : (x_0, \dots, x_{k-1}) \in \mathbb{Z}_m^k\}, \quad (3)$$

the set of all vectors $(u_{i_1}, \dots, u_{i_t})$ produced by the generator from its m^k possible initial states. (Formally speaking, Ψ_I is a *multiset* in the sense that duplicate vectors are counted as many times as they appear in it, so Ψ_I always has cardinality m^k .) It is well known that in the case of a MRG, this set is the intersection of a lattice L_I with the t -dimensional unit hypercube $[0, 1)^t$, where L_I contains the set \mathbb{Z}^t of all integer vectors [4,10]. This implies that Ψ_I lies on a limited number of equidistant parallel hyperplanes, at a distance (say) d_I apart. A standard way of measuring the uniformity of Ψ_I is to compute this d_I , which must be as small as possible for all index sets I that one wishes to consider. This is called the *spectral test*. It is customary to normalize d_I by computing d_t^*/d_I , where d_t^* is an absolute lower bound on d_I given m , k , and t , and to select MRGs based on a figure of merit defined as the worst case (smallest) of these ratios over a specified class of sets I [6,7,12]. A common choice for this class is to consider the sets $I = \{0, \dots, t-1\}$ of *successive* indices, for $t \leq t_1$, where t_1 is an arbitrary constant.

The LFSR (or Tausworthe) generators considered here are defined by the recurrence

$$x_n = (a_1x_{n-1} + \cdots + a_kx_{n-k}) \bmod 2; \quad (4)$$

$$u_n = \sum_{i=1}^w x_{n+s+i-1} 2^{-i}, \quad (5)$$

for some positive integers s and w [5,16–18]. The maximal period length is $\rho = 2^k - 1$. Specific parameter sets and implementations are given, e.g., in [8,18] and some references therein.

Let B be an arbitrary set of selected bits of the output values. More specifically, consider the bit string formed by concatenating the bits $b_{0,1}, \dots, b_{0,s_0}$ in the

expansion of u_0 , the bits $b_{1,1}, \dots, b_{1,s_1}$ in the expansion of u_1, \dots , and the bits $b_{t-1,1}, \dots, b_{t-1,s_{t-1}}$ in the expansion of u_{t-1} , where $s_0 + \dots + s_{t-1} \stackrel{\text{def}}{=} k' \leq k$. Let B denote the set of these k' bit indices and let Φ_B be the set of all bit strings thus formed when the *initial state* of the generator runs over its 2^k possible values. (Φ_B is called a *set*, but like Ψ_I , it should be interpreted as a *multiset*, in the sense that each bit string must be counted as many times as it occurs.) We say that the generator is *B-equidistributed* if Φ_B is *equidistributed*, i.e., if it contains every bit string of length k' exactly $2^{k-k'}$ times. This equidistribution can be verified by checking if the linear transformation that expresses these k' bits in terms of (x_0, \dots, x_{k-1}) has full rank [5,13]. In (5), the bit $b_{n,i}$ of u_n is x_{ns+i-1} , but our development easily extends to more general classes of “LFSRs”, such as those considered in [13].

A special case is when B contains the ℓ most significant bits of the first t successive output values u_0, \dots, u_{t-1} . In this case *B-equidistribution*, usually called *(t, ℓ)-equidistribution*, means that if each axis of the unit hypercube $[0, 1]^t$ is partitioned into 2^ℓ equal parts, each of the $2^{\ell t}$ small cubes of volume $2^{-\ell t}$ thus determined contains exactly $2^{k-\ell t}$ points from the set $\Psi_t = \{\mathbf{u}_t = (u_0, \dots, u_{t-1}) : (x_0, \dots, x_{k-1}) \in \{0, 1\}^k\}$. A generator is called *asymptotically random* or *maximally equidistributed* (ME) for the word size w if it is *(t, ℓ)-equidistributed* whenever $\ell \leq \min(\lfloor k/t \rfloor, w)$, for $1 \leq t \leq k$ [5,18,19]. Such generators are listed, e.g., in [8].

For nonlinear generators, the uniformity of Ψ_t is often assessed via discrepancy bounds [16], which are sometimes averages over an entire family of generators. For specific generator instances, they are not tight. No easily computable uniformity measure is currently available for these generators. On the other hand, certain types of nonlinear generators tend to perform better than the linear ones in empirical statistical tests [11]. In the forthcoming sections, we try to team up the guaranteed uniformity of linear generators with the irregularity of the nonlinear ones.

3 Combining a MRG with another generator

In this section and the next one, we combine two generators, G_1 and G_2 . For $j = 1, 2$, let S_j be the state space of G_j , $s_{j,0} \in S_j$ its initial state, $u_{j,n} \in [0, 1)$ its output at step n , and ρ_j its period length. Let G denote the combined generator, with state space S , initial state s_0 , output u_n at step n , and period length ρ .

We suppose in this section that G_1 is a MRG, G_2 any other type of generator, and that the output of G is defined by

$$u_n = (u_{1,n} + u_{2,n}) \bmod 1. \tag{6}$$

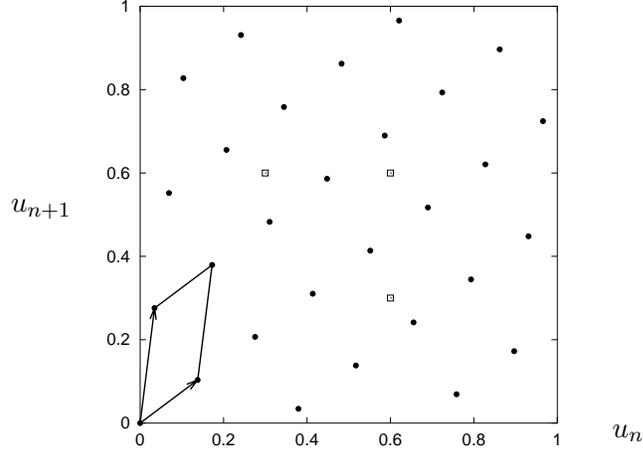


Fig. 1. Pairs (u_n, u_{n+1}) for the LCG with $m = 29$ and $a_1 = 8$

For any $I = \{i_1, i_2, \dots, i_t\}$, let $\Psi_{j,I} = \{\mathbf{u}_{j,I} = (u_{j,i_1}, \dots, u_{j,i_t}) : s_{j,0} \in S_j\}$ and $\Psi_I = \{\mathbf{u}_I = (u_{i_1}, \dots, u_{i_t}) : s_0 \in S\}$. Here, $\Psi_{1,I}$ is the intersection with $[0, 1]^t$ of a lattice L_I that contains \mathbb{Z}^t . We also have $\Psi_I = \{(\mathbf{u}_{1,I} + \mathbf{u}_{2,I}) \bmod 1 : s_{1,0} \in S_1 \text{ and } s_{2,0} \in S_2\} = \cup_{s_{2,0} \in S_2} (\Psi_{1,I} + \mathbf{u}_{2,I}) \bmod 1$, where the “mod 1” reduction is applied coordinatewise. Thus Ψ_I is the superposition of $|S_2|$ different shifts, modulo 1, of the set $\Psi_{1,I}$. The next lemma shows that shifting $\Psi_{1,I}$ modulo 1 preserves its structure, in the sense that it is the same as shifting the lattice L_I and taking the intersection with $[0, 1]^t$. Then, Ψ_I is the intersection with $[0, 1]^t$ of the union of $|S_2|$ shifted copies of L_I , and we can conclude that it covers the unit hypercube $[0, 1]^t$ at least as well as $\Psi_{1,I}$, in the sense that it cannot leave bigger uncovered gaps.

Lemma 1 *Let $\mathbf{w} \in \mathbb{R}^t$ and let $\tilde{\Psi}_{1,I} = (\Psi_{1,I} + \mathbf{w}) \bmod 1$. Then, $\tilde{\Psi}_{1,I} = (L_I + \mathbf{w}) \cap [0, 1]^t$, where $L_I + \mathbf{w}$ is the lattice L_I shifted by \mathbf{w} .*

Proof: Let $\mathbf{x} \in \tilde{\Psi}_{1,I}$. Then, $\mathbf{x} = (\mathbf{y} + \mathbf{w}) \bmod 1$ for some $\mathbf{y} \in \Psi_{1,I}$. That is, $\mathbf{x} = \mathbf{y} + \mathbf{w} - \mathbf{z}$ where $\mathbf{z} \in \mathbb{Z}^t$. This implies that $\mathbf{x} \in (L_I + \mathbf{w}) \cap [0, 1]^t$, because $\mathbf{y} - \mathbf{z} \in L_I$ (recall that L_I contains \mathbb{Z}^t) and $\mathbf{x} \in [0, 1]^t$. Conversely, if $\mathbf{x} \in (L_I + \mathbf{w}) \cap [0, 1]^t$, then $\mathbf{x} = \mathbf{y} + \mathbf{w} \in [0, 1]^t$ for some $\mathbf{y} \in L_I$, and therefore $\mathbf{x} \in \tilde{\Psi}_{1,I}$. ■

Example 1 Let G_1 be a LCG with parameters $m = 29$ and $a_1 = 8$, and let $t = 2$ and $I = \{0, 1\}$. The black disks in Figure 1 are the 29 points $\mathbf{u}_{1,I} = (u_{1,0}, u_{1,1})$ of $\Psi_{1,I}$. The two arrows in the figure represent the vectors $\mathbf{v}_1 = (4/29, 3/29)$ and $\mathbf{v}_2 = (1/29, 8/29)$, which form a basis of L_I ; i.e., L_I is comprised of all integer linear combinations of these two vectors. These vectors determine a parallelogram P , also illustrated in the figure. Suppose now that G_2 has only three states and produces the output sequence 0.3, 0.6, 0.6. Then, $\Psi_{2,I} = \{(0.3, 0.6), (0.6, 0.6), (0.6, 0.3)\}$. These points are marked by three small squares in Figure 1. Figure 2 shows the 87 points of Ψ_I .

Each point $\mathbf{u}_{2,I} \in \Psi_{2,I}$ can be moved to the parallelogram P by adding to it

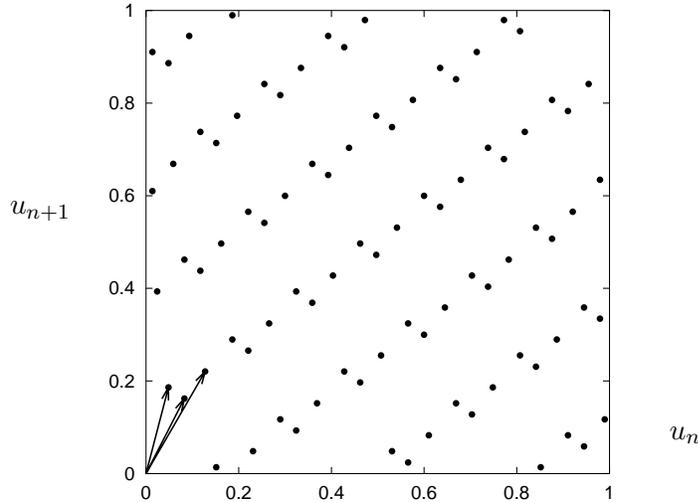


Fig. 2. Pairs (u_n, u_{n+1}) for the combined generator

some integer multiples of \mathbf{v}_1 and \mathbf{v}_2 ; that is, moved to $\tilde{\mathbf{u}}_{2,I} = \mathbf{u}_{2,I} + z_1\mathbf{v}_1 + z_2\mathbf{v}_2 \in P$ for some integers z_1 and z_2 . Let $\tilde{\Psi}_{2,I} \subset P$ be the set of points thus obtained. These points are marked by the three arrows in Figure 2. Note that $L_I + \tilde{\mathbf{u}}_{2,I} = L_I + \mathbf{u}_{2,I} + z_1\mathbf{v}_1 + z_2\mathbf{v}_2 = L_I + \mathbf{u}_{2,I}$, because $z_1\mathbf{v}_1 + z_2\mathbf{v}_2 \in L_I$. Therefore $\Psi_I = (L_I + \Psi_{2,I}) \cap [0, 1]^t = (L_I + \tilde{\Psi}_{2,I}) \cap [0, 1]^t$, which means that moving the points of $\Psi_{2,I}$ to P as we just did does not change Ψ_I .

The three points of $\tilde{\Psi}_{2,I}$ form a pattern in P that is repeated (modulo 1) 29 times in Ψ_I (see Figure 2). The uniformity of the point set Ψ_I is determined by two things: the quality of the lattice L_I and the uniformity of the point set $\tilde{\Psi}_{2,I}$ in P . To improve the quality of Ψ_I here, we could certainly choose G_2 so that the set $\tilde{\Psi}_{2,I}$ covers the parallelogram P more evenly. This could be viewed as a criterion for choosing G_2 , given G_1 . The same idea applies in more than two dimensions, and to other MRGs as well. However, constructing G_2 so that $\tilde{\Psi}_{2,I}$ covers the appropriate parallelogram for several sets I simultaneously seems hard in practice, unless G_2 and the combined generator also have a lattice structure, but this is precisely what we want to get away from. Constructing G_2 to nicely cover these parallelograms seems incompatible with our goal of introducing irregularity in G .

The bottom line: We are sure that Ψ_I covers $[0, 1]^t$ at least as well as $\Psi_{1,I}$. Since Ψ_I contains more points, the coverage could be expected to be *improved*, but it seems hard to *guarantee* an improvement while introducing irregularity at the same time.

4 Combining a LFSR with another generator

Now let G_1 be a LFSR, G_2 another type of generator, and the output of G be defined by

$$u_n = u_{1,n} \oplus u_{2,n}, \quad (7)$$

where \oplus denotes the bitwise exclusive-or of the binary expansions of the two fractions. For any bit selection B of size k' , let $\Phi_{j,B}$ and Φ_B be the corresponding sets of bit strings for G_j and G , respectively.

Proposition 1 *If G_1 is B -equidistributed, then $\Phi_{1,B} \oplus \mathbf{w}$ is equidistributed for each k' -bit vector \mathbf{w} and G is also B -equidistributed.*

Proof: Using the fact that $\mathbf{x} \oplus \mathbf{w} = \mathbf{x}' \oplus \mathbf{w}$ if and only if $\mathbf{x} = \mathbf{x}'$, it is easily seen that $\Phi_{1,B} \oplus \mathbf{w} = \cup_{\mathbf{x} \in \Phi_{1,B}} (\mathbf{x} \oplus \mathbf{w})$ is equidistributed if and only if $\Phi_{1,B}$ is, because the strings that are identical in one set are also identical in the other. When this happens, $\Phi_B = \cup_{\mathbf{w} \in \Phi_{2,B}} (\Phi_{1,B} \oplus \mathbf{w})$ is also equidistributed because the union of equidistributed sets is equidistributed. ■

As a special case of this proposition, if G_1 is (t, ℓ) -equidistributed, then G also is. The proposition also holds with G_1 replaced by G_2 . This implies, in particular, that if G_1 is B_1 -equidistributed and G_2 is B_2 -equidistributed for some bit sets B_1 and B_2 , then G is both B_1 -equidistributed and B_2 -equidistributed. This does not imply, however, that G is $(B_1 \cup B_2)$ -equidistributed. For example, suppose $B_1 = \{1, 2\}$, $B_2 = \{3\}$, $B = \{1, 2, 3\}$, and $\Phi_B = \{000, 000, 011, 011, 100, 100, 111, 111\}$. Then Φ_{B_1} and Φ_{B_2} are both equidistributed, but Φ_B is not.

5 Statistical Testing

We now report the results of our statistical testing with families of linear and nonlinear generators, and their combinations. For each of the tests that we have applied, n random points are thrown in the unit hypercube $[0, 1]^t$, where n is called the *sample size*. For the *nearest pair* test [9], the test statistic is the distance between the two nearest points. For the *birthday spacings* [14] and the *collision* [15] tests, the hypercube is partitioned into $k = d^t$ cubic cells of equal sizes. The collision test computes the number of times a point falls in a cell already occupied (number of collisions), whereas the birthday spacings test computes the number of collisions in the spacings between the successive numbers of the occupied cells. The reader can consult the references for precise definitions and analyzes of these tests.

These tests have been chosen because they are powerful for detecting regularities in linear generators. They have been applied in a systematic way, as

follows [9,11,14,15]. For each family of generators we seek an approximate relationship of the form $n_0 = K\rho^\gamma$, for some constants γ and K , where ρ is the period length of the generator and n_0 is the minimal sample size for which the p -value of the test is outside the interval $(10^{-15}, 1 - 10^{-15})$. The p -value is defined as

$$p = \begin{cases} P[T \geq t \mid \mathcal{H}_0] & \text{if } P[T \geq t \mid \mathcal{H}_0] < P[T \leq t \mid \mathcal{H}_0], \\ 1 - P[T \leq t \mid \mathcal{H}_0] & \text{otherwise,} \end{cases} \quad (8)$$

where t is the value taken by the test statistic T , and the null hypothesis \mathcal{H}_0 is that the u_n 's are i.i.d. $U(0, 1)$ random variables. It is important to distinguish the two cases in (8) because in some tests (e.g., the birthday spacings and collision tests), T can only take integer values.

To estimate γ and K for any given family, we applied the tests to generators of period length $\rho \approx 2^e$ for all integer values of e in a certain range. After determining γ in a heuristic way, we defined $K = 2^\nu$ where ν is the smallest integer i for which the p -value of the test with sample size $n = 2^{i+\gamma e}$ is outside the interval $(10^{-15}, 1 - 10^{-15})$.

The generator families considered here are LCGs selected via the spectral test in up to 8 dimensions and taken from Table 2 of [7], combined LFSR generators with good equidistribution properties [8], explicit inversive generators [1], defined by $x_n = (an + c) \bmod m$, $z_n = x_n^{-1} \bmod m$ (the inverse of x_n modulo m), and $u_n = z_n/m$, and cubic congruential generators [2,11], defined by $x_n = (ax_{n-1}^3 + 1) \bmod m$ and $u_n = x_n/m$. For the latter two nonlinear generators, the parameters m , a and c for each value of e were taken from [11]. They were selected so that the output sequence has maximal period length m . We also consider the combination of a LCG with each other type of generator by addition modulo 1, and a LFSR combined with each other type of generator by bitwise exclusive-or. In these combinations, the period length of component j is $\rho_j \approx 2^{e_j}$ and that of the combined generator is $\rho \approx 2^e \approx 2^{e_1} 2^{e_2}$.

Tables 1 and 2 summarize the results (i.e., give our estimates of γ and K) for a two-dimensional nearest pair test and for an eight-dimensional birthday spacings test. We see in both tables that for the (nonlinear) inversive and cubic generators, failure occurs only after the entire period length is exhausted. On the other hand, LCGs fail with a sample size proportional to the square root of the period length ($\gamma = 1/2$) for the nearest pair test and the cubic root ($\gamma = 1/3$) for the birthday spacings test. When LCGs are combined with a fixed-size nonlinear generator (e.g., $e_2 = 6$ or $e_2 = 10$), this rule (i.e., γ) remains the same, although the constant K increases by a factor that depends on e_2 . This means that the mixed combined generator stays alive longer than the LCG in face of these tests if both have similar period lengths. If the size of the nonlinear component increases with that of the LCG, i.e., if e_2 increases with e (e.g., $e_2 = e/4$ or $e_2 = e/2$), we see an improvement in the value of γ ,

Table 1

Estimates of γ and K for the nearest pair test in $t = 2$ dimensions

Family	e_2	γ	K
LCG		1/2	8
Inversive		1	1
Cubic		1	1
LCG + Inversive	6	1/2	32
LCG + Inversive	10	1/2	256
LCG + Inversive	$e/4$	2/3	2
LCG + Inversive	$e/2$	2/3	32
LCG + Cubic	6	1/2	32
LCG + Cubic	10	1/2	256
LCG + Cubic	$e/4$	2/3	2
LCG + Cubic	$e/2$	2/3	32
LCG + LFSR	10	1/2*	256
LCG + LFSR	$e/2$	2/3*	16

Table 2

Estimates of γ and K for the birthday spacings test, with $t = 8$, $k \approx \rho$, and $n^3/4k \approx 1$

Family	e_2	γ	K
LCG		1/3	8
LFSR		2/5	32
Inversive		1	2
Cubic		1	1
LCG + Inversive	6	1/3	16
LCG + Inversive	10	1/3	64
LCG + Inversive	$e/4$	2/5	8
LCG + Inversive	$e/2$	1/2	8
LCG + LFSR	10	1/3	64
LCG + LFSR	$e/2$	3/5	1
LFSR \oplus Inversive	6	2/5	128
LFSR \oplus Inversive	10	1/2	128
LFSR \oplus Inversive	$e/4$	3/5	4
LFSR \oplus Inversive	$e/2$	3/5	32
LFSR \oplus LCG	10	1/2	64
LFSR \oplus LCG	$e/2$	3/5	16

from 1/2 to 2/3 in Table 1 and from 1/3 to 1/2 in Table 2. Similar results are obtained in Table 2 when the LCG is combined with a LFSR. We also obtain the same types of results for a LFSR combined with another type of generator via bitwise exclusive-or (the bottom part of Table 2).

Other tests were made, for other values of t and e_2 , and the results were similar. See [3] for more details. The “correct” values of γ and K were not always clear from the results, especially for the collision tests. However, the following was observed systematically: For comparable period lengths, the mixed combined generators always did better in the tests than the linear ones, there was

more improvement when the value of e_2 was larger, and γ was improved only when e_2 was increased together with e . This means that to destroy the regularity, combining a large linear generator with a very small nonlinear one does not suffice. The size of the nonlinear component must be significant enough. Moreover, combining two different types of linear generators, such as a LCG or MRG with a LFSR, seems to do as well as the linear-nonlinear combinations, at least from the empirical perspective.

6 Implementations

One might argue that a combined generator with a large nonlinear component would be unacceptably slow for many applications. However, nonlinear generators with arbitrarily long periods can be constructed by combining small nonlinear generators with relatively prime period lengths, and very fast implementations of these small nonlinear generators can be obtained simply by precomputing their output sequences and storing them in tables. Moreover, once it is decided to implement a generator component via a table, its definition can be made very complicated and this has no effect on the running speed. In fact, for a component of period ρ_j , one can simply generate a random permutation of the ρ_j output values, by any method, and store it in the table.

For a concrete illustration, we took the generator `lfsr113` of [8], with period length near 2^{113} , and combined it by bitwise exclusive-or with a nonlinear generator of period length 2^{18} implemented in a table. The code was written in C and compiled by the `gcc` compiler with full optimization options. The CPU time needed to generate and add 10 million random numbers was 0.7 seconds with this combined generator, compared with 0.6 seconds for `lfsr113` alone, on an AMD Athlon processor at 750 MHz. On a Pentium III at 500 MHz and with the `g++` compiler, the numbers were 1.9 and 1.3 seconds, respectively.

References

- [1] J. Eichenauer-Herrmann. Inversive congruential pseudorandom numbers: A tutorial. *International Statistical Reviews*, 60:167–176, 1992.
- [2] J. Eichenauer-Herrmann and E. Herrmann. Compound cubic congruential pseudorandom numbers. *Computing*, 59:85–90, 1997.
- [3] J. Granger-Piché. Générateurs pseudo-aléatoires combinant des récurrences linéaires et non linéaires. Master’s thesis, Département d’informatique et de recherche opérationnelle, Université de Montréal, 2001.
- [4] D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley, Reading, Mass., third edition, 1998.

- [5] P. L'Ecuyer. Maximally equidistributed combined Tausworthe generators. *Mathematics of Computation*, 65(213):203–213, 1996.
- [6] P. L'Ecuyer. Good parameters and implementations for combined multiple recursive random number generators. *Operations Research*, 47(1):159–164, 1999.
- [7] P. L'Ecuyer. Tables of linear congruential generators of different sizes and good lattice structure. *Mathematics of Computation*, 68(225):249–260, 1999.
- [8] P. L'Ecuyer. Tables of maximally equidistributed combined LFSR generators. *Mathematics of Computation*, 68(225):261–269, 1999.
- [9] P. L'Ecuyer, J.-F. Cordeau, and R. Simard. Close-point spatial tests and their application to random number generators. *Operations Research*, 48(2):308–317, 2000.
- [10] P. L'Ecuyer and R. Couture. An implementation of the lattice and spectral tests for multiple recursive linear random number generators. *INFORMS Journal on Computing*, 9(2):206–217, 1997.
- [11] P. L'Ecuyer and P. Hellekalek. Random number generators: Selection criteria and testing. In P. Hellekalek and G. Larcher, editors, *Random and Quasi-Random Point Sets*, volume 138 of *Lecture Notes in Statistics*, pages 223–265. Springer, New York, 1998.
- [12] P. L'Ecuyer and C. Lemieux. Variance reduction via lattice rules. *Management Science*, 46(9):1214–1235, 2000.
- [13] P. L'Ecuyer and F. Panneton. Construction of equidistributed generators based on linear recurrences modulo 2. In K.-T. Fang, F. J. Hickernell, and H. Niederreiter, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2000*, pages 318–330, Berlin, 2002. Springer-Verlag.
- [14] P. L'Ecuyer and R. Simard. On the performance of birthday spacings tests for certain families of random number generators. *Mathematics and Computers in Simulation*, 55(1–3):131–137, 2001.
- [15] P. L'Ecuyer, R. Simard, and S. Wegenkittl. Sparse serial tests of uniformity for random number generators. *SIAM Journal on Scientific Computing*. To appear.
- [16] H. Niederreiter. *Random Number Generation and Quasi-Monte Carlo Methods*, volume 63 of *SIAM CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, Philadelphia, 1992.
- [17] R. C. Tausworthe. Random numbers generated by linear recurrence modulo two. *Mathematics of Computation*, 19:201–209, 1965.
- [18] S. Tezuka. *Uniform Random Numbers: Theory and Practice*. Kluwer Academic Publishers, Norwell, Mass., 1995.
- [19] J. P. R. Tootill, W. D. Robinson, and D. J. Eagle. An asymptotically random Tausworthe sequence. *Journal of the ACM*, 20:469–481, 1973.