GERAD report G-96-41, 1996
This version: March 1997

ENTROPY TESTS FOR RANDOM NUMBER GENERATORS

Pierre L'Ecuyer

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`

Aaldert Compagner

Faculty of Applied Physics, Delft University of Technology
P.O. Box 5046, 2600 GA Delft, The Netherlands

Jean-François Cordeau

Département d'Informatique et de Recherche Opérationnelle
Université de Montréal

Keywords: Random number generators; uniform distribution; goodness-of-fit; entropy.

ABSTRACT

Uniformity tests based on a discrete form of entropy are introduced and studied in the context of empirical testing of uniform random number generators. Numerical results are provided and several currently used generators fail the tests. The linear congruential and nonlinear inversive generators with power-of-two modulus perform especially badly.

1 Introduction

Random number generators should generally be built based on proper *theoretical* analysis and understanding of their structural properties, and then tested *empirically* to further improve one's confidence in them. Different statistical tests are sensitive to different types of deficiencies in generators, so it is useful to apply a wide range of tests. For background on random number generators and statistical testing, see for example [19, 22, 23, 31, 36]. In this paper, we study uniformity and independence tests based on the concept of entropy for discrete uniform distributions, following the suggestion in [2] that entropy might provide a useful testing ground.

Statistical tests based on the entropy of a *continuous* distribution have already been proposed and applied [4, 5, 14, 45]. These tests are based on (continuous) entropy estimators related to sums of logarithms of m-spacings, and have little in common with those proposed here. They are also discussed in [27].

For a discrete random variable Y taking its values in a (discrete) set $\mathcal{S}$, with probability mass function $p_j = P[Y = j]$ for $j \in \mathcal{S}$, the *entropy* of Y (in the sense of Shannon) is defined by

$$-\sum_{j\in\mathcal{S}} p_j \log_2(p_j). \tag{1}$$

In the *equiprobable* case, where Y is uniform over the set $\mathcal{S} = \{0, \ldots, k-1\}$, this becomes:

$$-\sum_{j=0}^{k-1}(1/k)\log_2(1/k) = \log_2(k). \tag{2}$$

This is the maximal entropy for a discrete random variable with values in $\mathcal{S}$. Entropy can be viewed as a measure of *randomness* or *unpredictability* of Y.

Now, let $Y_1, \ldots, Y_n$ be random variables with values in $\mathcal{S}$ and consider the statistical hypothesis $\mathcal{H}_0$: "The Y_i are independent and uniformly distributed over $\mathcal{S}$". To test $\mathcal{H}_0$, we can compute the sample entropy of the Y_i (as follows) and compare it with (2). For each $j \in \mathcal{S}$, let X_j be the number of times the value j is obtained:

$$X_j = \sum_{i=1}^{n} I[Y_i = j] \tag{3}$$

where I denotes the indicator function. Under $\mathcal{H}_0$, the vector $(X_0, \ldots, X_{k-1})$ has the multinomial distribution with parameters $(n, 1/k, \ldots, 1/k)$. Define the *sample entropy* as:

$$H = -\sum_{j=0}^{k-1}(X_j/n)\,\log_2(X_j/n), \tag{4}$$

where $0\log(0) = 0$ by convention. The random variable H should be close to $\log_2(k)$ under $\mathcal{H}_0$, and usually smaller if $\mathcal{H}_0$ is grossly violated. There are other goodness-of-fit statistics for testing this multinomial distribution. The best known is certainly Pearson's chi-square:

$$X^2 = \sum_{j=0}^{k-1}\frac{(X_j - n/k)^2}{n/k}. \tag{5}$$

In this paper, we examine a few ways of using the statistic H to test $\mathcal{H}_0$, and we apply this to random number generators testing. In the next section, we recall some properties of H. Section 3 explains how we extract bit strings of length L from the output sequence of a generator to apply entropy tests with $k = 2^L$. This section then introduces different test variants, based on the linear correlation between successive values of the entropy, and on the entropy of a set of overlapping bit strings. In testing, one can either compute a single value of H with a large n and reject $\mathcal{H}_0$ if H is too far in the tail of the distribution (a *single-level* or *first-order* test), or use a *two-level* or *second-order* testing procedure: compute N values of H from disjoint subsequences, then compare their empirical distribution to the theoretical one, test the correlation between the successive values of H, and so on. This two-level approach is commonly used in random number testing. It permits to test the sequence not only at the global level, but also at a local level (see [11, 19, 22, 31]). The memory size of the computer usually limits the values of n and/or k that can be handled in a first-order test, and second-order tests allow larger sample sizes. Section 4 gives specific illustrations showing that H has better efficiency than X^2 against certain alternative hypotheses. It is not universally better, though. In Section 5, we apply a selection of tests to a set of random number generators that are either commonly used or have been recently proposed. The entropy tests find significant defects in several of these generators. In some cases, they quickly pick up problems that other classical tests miss when considering an equivalent total sample size. Entropy tests thus constitute a useful addition to the toolbox.

2 Tests based on the empirical entropy

2.1 Testing the equiprobable multinomial distribution

The *loglikelihood ratio* statistic for testing the hypothesis of equal probabilities in the multinomial distribution is defined by (see [41])

$$G^2 = 2\sum_{j=0}^{k-1} X_j \ln(X_j/(n/k)). \tag{6}$$

The statistics G^2 and H are related by the simple linear transformation

$$H = \log_2(k) - \frac{G^2}{2n\ln(2)}, \tag{7}$$

so the known theory for G^2 can be used for H as well. Under $\mathcal{H}_0$, when $n \to \infty$ for fixed k (the *fixed-cells case*), G^2 converges in distribution to a chi-square random variable with $k-1$ degrees of freedom. On the other hand, if $n \to \infty$ and $k \to \infty$ simultaneously so that $n/k \to \nu$ for some constant $\nu > 0$ (the *sparse case*), then

$$\frac{G^2 - \mu_\nu}{\sigma_\nu} \Rightarrow N(0,1)$$

for some asymptotic mean and variance constants μ_ν and σ_ν^2 that depend on ν, where $\Rightarrow$ denotes convergence in distribution and $N(0,1)$ is a standard normal random variable. For more details, see, e.g., [41], where general expressions for μ_ν and σ_ν are given in terms of integrals involving Poisson random variables. The asymptotic mean and variance of G^2 in the sparse case are *different* than in the fixed-cells case.

2.2 Exact Moments

The quality of the approximation can be improved if one replaces the asymptotic mean and variance by their exact values for the finite sample size. Under $\mathcal{H}_0$, each X_j is a binomial random variable, with

$$P[X_j = x] = \binom{n}{x}\left(\frac{1}{k}\right)^x\left(1-\frac{1}{k}\right)^{n-x}, \quad 0 \le x \le n,$$

and for $1 \le j < \ell \le k$, one has

$$P[X_j = x, X_\ell = y] = \binom{n}{x}\binom{n-x}{y}\left(\frac{1}{k}\right)^{x+y}\left(1-\frac{2}{k}\right)^{n-x-y}, \quad 0 \le x+y \le n.$$

Replacing these expressions in the definitions of expectation and variance, one obtains, after some algebraic manipulations,

$$\begin{aligned}
E[H] &= -\sum_{x=0}^{n}\frac{x}{n}\binom{n}{x}\frac{(k-1)^{n-x}}{k^{n-1}}\log_2\left(\frac{x}{n}\right), && (8)\\
\mathrm{Var}[H] &= E\left[\left(\sum_{j=0}^{k-1}\left(\frac{X_j}{n}\log_2\left(\frac{X_j}{n}\right)+\mu\right)\right)^2\right] \\
&= k\,E\left[\left(\frac{X_1}{n}\log_2\left(\frac{X_1}{n}\right)+\mu\right)^2\right] \\
&\quad + k(k-1)\,E\left[\left(\frac{X_1}{n}\log_2\left(\frac{X_1}{n}\right)+\mu\right)\left(\frac{X_2}{n}\log_2\left(\frac{X_2}{n}\right)+\mu\right)\right] \\
&= \sum_{x=0}^{n}\binom{n}{x}\frac{(k-1)^{n-x}}{k^{n-1}}\left(\frac{x}{n}\log_2\left(\frac{x}{n}\right)+\mu\right)^2 \\
&\quad + \sum_{x=0}^{\lfloor n/2\rfloor}\binom{n}{x}\binom{n-x}{x}\frac{(k-1)(k-2)^{n-2x}}{k^{n-1}}\left(\frac{x}{n}\log_2\left(\frac{x}{n}\right)+\mu\right)^2 \\
&\quad + 2\sum_{x=0}^{n}\sum_{y=0}^{\min(n-x,x-1)}\binom{n}{x}\binom{n-x}{y}\frac{(k-1)(k-2)^{n-x-y}}{k^{n-1}} \\
&\qquad \cdot\left(\frac{x}{n}\log_2\left(\frac{x}{n}\right)+\mu\right)\left(\frac{y}{n}\log_2\left(\frac{y}{n}\right)+\mu\right), && (9)
\end{aligned}$$

where $\mu = E[H]/k$. These formulas are not simple, but are nevertheless practical as long as n/k is not too large. If n is large and n/k is, say, no more than 10 or so, the terms of the sums in (8–9) decrease exponentially fast with x and y, and an excellent approximation can be obtained by truncating the sums after a small number of terms. For example, with $n = k = 1000$, the relative errors on $E[H]$ and $\mathrm{Var}[H]$ are less than 10^{-10} if the sums are stopped at $x, y = 14$ instead of 1000, and less than 10^{-15} if the sums are stopped at $x, y = 18$.

For $k/n \ll 1$, one has the following approximations, which can be derived from the moment approximations of G^2 given in [43]:

$$E[H] = \log_2(k) - \frac{(k-1)(1+(k+1)/(6n) - k^2/(6n^2))}{2n\ln(2)} + O((k/n)^4), \tag{10}$$

$$\mathrm{Var}[H] = \frac{(k-1)(1+(k+1)/(3n) - 2k^2/(3n^2))}{2(n\ln(2))^2} + O((k/n)^4). \tag{11}$$

In the sparse case, one would use the standard normal approximation for the statistic $(H - E[H])/(\mathrm{Var}[H])^{1/2}$. In the fixed-cells case, one can define a *moment-corrected* version of H, whose mean and variance are exactly $k-1$ and $2(k-1)$, i.e., are equal to the mean and variance of the asymptotic chi-square distribution with $k-1$ degrees of freedom. Read and Cressie [41] recommend this type of procedure. This moment-corrected statistic is

$$H_C = \frac{H - E[H] + (k-1)\sigma_H}{\sigma_H}, \tag{12}$$

where $\sigma_H^2 = \mathrm{Var}[H]/(2(k-1))$. The mean and variance of H can be replaced by their approximations (10)–(11). Another possibility would be to correct the distribution itself, e.g., using Edgeworth-type expansions [41, p.68]. This gives extremely complicated expressions, due in part to the discrete nature of the multinomial distribution, and the gain is small.

2.3 Efficiency comparisons

Why use H and not only X^2 to test $\mathcal{H}_0$? Because H is more efficient against certain types of alternatives. The statistics X^2 and G^2 have already been compared in the literature in terms of their Pitman and Bahadur asymptotic relative efficiencies (ARE) (see, e.g., [40, 41] for definitions and details). Consider a class of alternatives $\{\mathcal{H}'_c, c > 0\}$, where the parameters of the multinomial are perturbed from $(n, 1/k, \ldots, 1/k)$ to $(n, c\delta_0 + 1/k, \ldots, c\delta_{k-1} + 1/k)$, where $c > 0$, $c\delta_j > -1/k$ for each j, and $\sum_{j=0}^{k-1} \delta_j = 0$. For this specific class of alternatives, in terms of Pitman ARE, it is known that X^2 and G^2 are equivalent in the fixed-cell case, whereas X^2 dominates in the sparse case. In terms of Bahadur ARE, on the other hand, G^2 (i.e., H) wins in both cases.

The Bahadur ARE compares the rates at which the p-values of the tests converge to 1 for the two statistics, as $n \to \infty$, for a fixed alternative hypothesis (e.g., fixed c, in the above example). When testing a specific random number generator, the alternative is fixed, and one usually looks for a very small p-value to reject the generator. In case of doubt, the sample size can be increased as needed. The best test in terms of Bahadur ARE should require a smaller sample size to convince us that there is indeed a problem with a given not-so-good generator. So, Bahadur efficiency seems a sensible concept in this case and this may justify the use of H instead of (or in addition to) X^2.

Let us now consider a different family of alternatives, as follows. Under $\mathcal{H}_c$, we assume a multinomial distribution with k cells, where c cells have probability 0 and the remaining $k_0 = k - c$ cells are equiprobable with probability $1/k_0$. This type of alternative is a good representation of what happens with certain random number generators. We are interested in the probability $p_H(\epsilon)$ (resp., $p_X(\epsilon)$) that the p-value of H (resp., X^2) is less than ϵ, under $\mathcal{H}_c$. One can verify that

$$X^2 = \sum_{j=0}^{k-1} \frac{(X_j - n/k)^2}{n/k} = \frac{k}{k_0} X_0^2 + \frac{nk}{k_0} - n,$$

where

$$X_0^2 = \sum_{j=0}^{k_0-1} \frac{(X_j - n/k_0)^2}{n/k_0}.$$

In the fixed-cells case, X_0^2 is approximately $\chi^2(k_0 - 1)$, a chi-square distribution with $k_0 - 1$ degrees of freedom. (We could also take the moment-corrected versions, but here we ignore this for simplification.) Define $Q(\ell, x) = P[\chi^2(\ell) \le x]$. Then,

$$\begin{aligned} p_X(\epsilon) &= P[X^2 > Q^{-1}(k-1, 1-\epsilon)] \\ &= P[(k/k_0)X_0^2 + (nk/k_0) - n > Q^{-1}(k-1, 1-\epsilon)] \\ &\approx 1 - Q(k_0 - 1, (k_0/k)[Q^{-1}(k-1, 1-\epsilon) - (nk/k_0) + n]). \end{aligned}$$

In the sparse case,

$$p_X(\epsilon) = P[(X^2 - k + 1)\sqrt{n/(2(k-1)(n-1))} > \Phi^{-1}(1-\epsilon)]$$

can be estimated by assuming that $(X_0^2 - k_0 + 1)\sqrt{n/(2(k_0-1)(n-1))}$ has the standard normal distribution. Similar arguments can be made with

$$H = -\sum_{j=0}^{k-1} (X_j/n) \log_2(X_j/n) = -\sum_{j=0}^{k_0-1} (X_j/n) \log_2(X_j/n),$$

yielding

$$\begin{aligned} p_H(\epsilon) &= P[G^2 > Q^{-1}(k-1, 1-\epsilon)] \\ &\approx 1 - Q(k_0 - 1, Q^{-1}(k-1, 1-\epsilon) + 2n \ln 2(\log_2 k_0 - \log_2 k)) \end{aligned}$$

in the fixed-cells case and

$$p_H(\epsilon) \approx \Phi\left(\frac{\Phi^{-1}(\epsilon)\sigma_{H,k} + \mu_{H,k} - \mu_{H,k_0}}{\sigma_{H,k_0}}\right)$$

in the sparse case, where $\mu_{H,k}$ and $\sigma_{H,k}^2$ are the expressions (10)–(11), while μ_{H,k_0} and σ_{H,k_0}^2 are the corresponding expressions for k_0. With these formulas in hand, one can compare the efficiencies of H and X^2 against $\mathcal{H}_c$.

For a concrete illustration, let $\epsilon = 10^{-4}$, $n = 2^{15}$, $k_0 = 1024$, and $c = 2$. Using the chi-square approximation, one obtains $p_X(\epsilon) = 0.0124$ and $p_H(\epsilon) = 0.1731$. With the same values but with $c = 4$, one has $p_X(\epsilon) = 0.1850$ and $p_H(\epsilon) = 0.9693$. This is typical: H clearly dominates X^2 when c is small relative to k. If $n \approx k_0$, the tests get significant power only for reasonably large c. Then, one may still have $p_H(\epsilon) > p_X(\epsilon)$, but H usually wins by a smaller margin. For example, let $\epsilon = 10^{-4}$, $n = k_0 = 2^{16}$, and $c = 1024$. With the normal approximation, one obtains $p_X(\epsilon) = 0.1826$ and $p_H(\epsilon) = 0.3079$.

The better performance of H over X^2 for the alternative $\mathcal{H}_c$ carries from the first-order to the second-order tests. As an illustration, we made a simulation experiment with $n = 2^{13}$, $k_0 = 1024$, $c = 2$, and with $N = 8$, using the chi-square approximation to the moment-corrected versions of X^2 and H. For X^2, we obtained 8 p-values ranging from 0.290 to 0.996 at the first level, and p-values $\delta^+ = 0.995$ and $\delta^- = 0.151$ for the KS statistics D_N^+ and D_N^- (defined in the next section) at the second level. For H, the p-values at the first level ranged from 0.721 to 0.999, and we had $\delta^+ = 0.9994$ and $\delta^- = 0.000049$ at the second level. Then we took $n = 2^{15}$, again with $k_0 = 1024$, $c = 2$, and $N = 8$. We obtained $\delta^+ = 0.998$ and $\delta^- = 0.005$ for X^2, and $\delta^+ = 0.999983$ and $\delta^- = 0.000000062$ for H. In both cases, H wins. Generator CMRG96 of section 4 was used for these experiments.

It is important to recall that defects in random number generators are not necessarily well-captured by the class of alternatives $\mathcal{H}'_c$ and $\mathcal{H}_c$ described above. For example, the alternative could be that the vectors $(X_0, \ldots, X_{k-1})$ are not multinomially distributed because of some type of correlation between the Y_i. For the tests based on the correlation between successive entropy values and on the overlapping entropy, to be introduced in the next section, the above theoretical results on power comparison do not apply and the question remains open. Empirical comparisons could be useful.

3 Multinomial sampling from the output of a generator

3.1 Extracting the bits

Let $U_1, U_2, U_3, \ldots$ be the successive output values of a random number generator and write the binary expansion of U_i as

$$U_i = \sum_{j=1}^{\infty} b_{i,j} 2^{-j}.$$

We model the U_i as random variables and consider the hypothesis $\tilde{\mathcal{H}}_0$ that these U_i are independent $U(0,1)$. Equivalently, $\tilde{\mathcal{H}}_0$ says that the $b_{i,j}$ are mutually independent Bernoulli random variables, with $P[b_{i,j} = 1] = P[b_{i,j} = 0] = 1/2$.

To construct a multinomial test as described in the previous sections, with $k = 2^L$ for some integer L, extract n disjoint blocks of L bits each from the array of bits $(b_{i,j})$, as follows. Choose two integers $r \geq 0$ and $s \geq 1$. Extract from each U_i the bits $b_{i,r+1}, \ldots, b_{i,r+s}$ and (conceptually) put them in a long string:

$$b_{1,r+1}, \ldots, b_{1,r+s}, b_{2,r+1}, \ldots, b_{2,r+s}, b_{3,r+1}, \ldots. \tag{13}$$

Partition this string into n substrings (or blocks) of L consecutive bits, without overlap. The ith substring is the binary representation of Y_i. The choice of r and s would depend on the testing strategy that one has in mind. For example, to test only the most significant bits of the U_i's, one would take $r = 0$ and a small value of s, whereas to test some least significant bits, one would take $r > 0$ (i.e., throw away the r most significant bits).

From these Y_i, compute $(X_0, \ldots, X_{k-1})$, H, and its p-value. In this setup, $\tilde{\mathcal{H}}_0$ implies $\mathcal{H}_0$, and the latter hypothesis is the one we test. In case of a two-level test, this procedure is repeated N times, with disjoint parts of the sequence. Let $H_1, \ldots, H_N$ denote the N values of H thus obtained. Here, H_1 is computed from $U_1, \ldots, U_{nL/s}$, H_2 from $U_{1+nL/s}, \ldots, U_{2nL/s}$, and so on. We consider the following two ways of testing $\mathcal{H}_0$: (a) construct the empirical distribution of $H_1, \ldots, H_N$ and compare it with the theoretical distribution of H under $\mathcal{H}_0$ and (b) test for significant correlation between the pairs (H_i, H_{i+1}) of successive values of the entropy.

3.2 The Entropy Distribution Test

For the two-level tests, to compare the empirical distribution of $H_1, \ldots, H_N$ with their asymptotic chi-square or normal distribution under $\mathcal{H}_0$, we use *Kolmogorov-Smirnov* (KS) tests as follows. We first compute the p-value $1 - V_i$ of each H_i. In the case of the chi-square approximation (the fixed-cells case), this means first transforming each H_i into the corresponding moment-corrected $H_{C,i}$ by (12), then computing $V_i = Q(k-1, H_{C,i})$ where $Q(k-1, \cdot)$ is the chi-square distribution function with $k-1$ degrees of freedom. In the sparse case, let $Z_i = (H_i - E[H_i])/(\mathrm{Var}[H_i])^{1/2}$ and let $V_i = \Phi(Z_i)$, where Φ is the standard normal distribution function. In both cases, let $V_{(1)}, \ldots, V_{(N)}$ be the values of $V_1, \ldots, V_N$ sorted by increasing order, and define

$$\begin{aligned} D_N^+ &= \max_{1 \le j \le N} (j/N - V_{(j)}), \\ D_N^- &= \max_{1 \le j \le N} (V_{(j)} - (j-1)/N). \end{aligned}$$

The distribution of D_N^+ and D_N^- under $\mathcal{H}_0$ is given, e.g., in [6]. Let d^+ and d^- be the values taken by D_N^+ and D_N^- in a given experiment and define the corresponding (second-order) *p-values* as $\delta^+ = P[D_N^+ > d^+]$ and $\delta^- = P[D_N^- > d^-]$, respectively. The hypothesis $\mathcal{H}_0$ is *rejected* if δ^+ or δ^- is extremely close to 0 or 1. In case of doubt, one replicates the entire test (independently) and rejects $\mathcal{H}_0$ if the p-values are consistently too close to zero or one. This is the *entropy distribution test*. Note that for $N = 1$, δ^+ and δ^- are the same as the p-values of H_1 on the two sides, i.e., $\delta^+ = 1 - V_1$ and $\delta^- = V_1$. Therefore, the p-value δ^+ of a second-order test with $N = 1$ is exactly the same as the p-value $1 - V_1$ of the first order test. The best choice of N for a given total sample size Nn depends on the test and on the type of defect that is targeted; in many cases it is $N = 1$.

3.3 Accuracy of the asymptotic distribution for finite n

The chi-square and normal distributions that we compare with are only asymptotic approximations. We may check the approximation error for finite n and k, to make sure that it is not a significant source of false rejections. Since we use KS statistics to compare the distributions, the following error measures are appropriate:

$$\Delta^+(k,n) \stackrel{\text{def}}{=} \sup_{0\le u\le 1} (P[V_i \le u] - u),$$
$$\Delta^-(k,n) \stackrel{\text{def}}{=} \sup_{0\le u\le 1} (u - P[V_i \le u]).$$

These quantities appear hard to compute exactly, but can be estimated by $\hat\Delta^+(k,n) = D_N^+$ and $\hat\Delta^-(k,n) = D_N^-$, respectively, for N as large as possible, where D_N^+ and D_N^- are computed by Monte Carlo simulation. As an illustration, we computed these estimators for the normal approximation of H, with $N = 10^6$ and $n = k = 2^L$, for $L = 6, 7, 8, \ldots, 13$, and found the rough but conservative approximation: $\hat\Delta^-(k,n) \le \hat\Delta^+(k,n) \le 0.16/\sqrt{n}$. To see the practical effect of this error, suppose we apply a test with parameters $N = 256$ and $k = n = 1024$. Then, to get $\delta^+ < 0.01$ (say) we need $D_N^+ > 0.10$ (approximately) and similarly for D_N^-. In this case, an error smaller than $0.16/\sqrt{n} = 0.005$ in the value of D_N^+ is deemed negligible. Therefore, these parameters, as well as larger n and k with the same N, or smaller N with the same k and n, are safe to use in actual testing. We made a similar check for all our tests.

To estimate these errors we used the generator CMRG96 defined in Section 5.1. So far, this generator has been extremely reliable in all our experiments. We also double checked with other generators of different types, including CLCG88 and the combined Tausworthe generator proposed in [25], and the results agreed. These results also agree with the theory.

3.4 Correlation between successive entropy values

A second type of test of $\mathcal{H}_0$ verifies whether the pairs (H_i, H_{i+1}), for $1 \le i \le N-1$, are significantly correlated. If they are, it means that a low [high] entropy in one part of the sequence (13) tends to be followed by a low [high] entropy in the next part. Equivalently, one may test the correlation between the pairs of standardized observations (Z_i, Z_{i+1}). The sample correlation is

$$\hat\rho_N = \frac{1}{N-1} \sum_{i=1}^{N-1} Z_i Z_{i-1}. \tag{14}$$

Under $\mathcal{H}_0$, as $N \to \infty$, $\sqrt{N}\hat\rho_N \Rightarrow N(0,1)$, so a statistical test readily follows: Take a large N, and reject $\mathcal{H}_0$ if the p-value $\delta_c = \Phi(\sqrt{N}\hat\rho_N)$ is too close to 0 or 1. We call this an *entropy correlation test*. Here, n can be small but N must be large, the opposite as for the entropy distribution test.

This correlation test differs significantly from one that would compute the sample correlation between the successive U_i. For instance, the latter test gives more importance to the most significant bits of the U_i's, whereas in the entropy-based test, all the extracted bits have the same importance in the test.

3.5 Overlapping bit strings

In Section 3, the n bit strings of length L were constructed without overlap; that is, from disjoint parts of the sequence (13). We now consider a setup in which they are constructed with overlap. Take the first n bits of the sequence (13), relabel them as $b_1, \ldots, b_n$, and put them in a circle (i.e., define $b_0 = b_n$ and $b_j = b_{j \bmod n}$ for all integers j). For $i = 1, \ldots, n$, let Y_i be the integer represented by the bit string of length L starting at position i:

$$Y_i = \sum_{j=0}^{L-1} b_{i+L-j-1} 2^j. \tag{15}$$

Define the X_j and H as in (3) and (4), and let H_1 be the value of H thus obtained. Repeat the same procedure with the bits $n+1$ to $2n$ of the sequence (13), yielding an entropy value H_2, and so on. So, for $i \geq 1$, H_i is the value of H obtained by putting the bits $(i-1)n+1, \ldots, in$ in a circle and looking at all n strings of L consecutive bits over that circle.

A possible advantage of this overlapping variant is that it squeezes more information from the bit string (13) compared to the non-overlapping case. However, in the overlapping case, $(X_1, \ldots, X_k)$ is no longer a multinomial random vector, so the mean and variance formulæ and the limiting distribution of H that we gave for the multinomial case no longer apply. For small values of n, one can compute the exact mean and variance directly from their definitions (which involves a sum of 2^n terms corresponding to the 2^n possibilities for $\{b_1, \ldots, b_n\}$). The exact values for some pairs (L, n) are reported in Table I. With these values in hand, one can compute the standardized values $Z_1, \ldots, Z_N$, defined as before. For large N,

$$\sqrt{N}\bar{Z}_N = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} Z_i$$

is approximately $N(0,1)$ under $\mathcal{H}_0$. The *overlapping average entropy test* computes this statistic to test the empirical mean of the entropy against its theoretical value. It rejects $\mathcal{H}_0$ if $\delta_a = \Phi(\sqrt{N}\bar{Z}_N)$ is too close to 0 or 1.

A linear correlation test can also be applied as in the previous subsection; we call this version the *overlapping entropy correlation test*. For large n (say, $n > 30$), the exact mean and variance of H_i are unknown, but they can be replaced by their sample counterparts, yielding:

$$\hat{\rho}_N = \frac{1}{\hat{\sigma}_T^2} \left[\frac{1}{N-1} \sum_{i=1}^{N-1} H_i H_{i+1} - (\bar{H})^2 \right] \tag{16}$$

where

$$\bar{H} = \frac{1}{N} \sum_{i=1}^{N} H_i$$

Table I
Mean and variance of the overlapping entropy for some pairs (L, n).

L	n	$E[\hat{H}_d(k,n)]$	$\mathrm{Var}[\hat{H}_d(k,n)]$
2	4	1.375000	0.3593750
3	8	2.299772	0.1867293
4	16	3.238725	0.1007388
5	20	3.817000	0.0815392
5	25	4.014291	0.0694637
5	30	4.160005	0.0591489

and

$$\hat{\sigma}_H^2 = \frac{1}{N-1}\sum_{i=1}^{N}(H_i - \bar{H})^2.$$

Under $\mathcal{H}_0$, $\sqrt{N}\hat{\rho}_N \Rightarrow N(0,1)$ and (for large N) the test reject $\mathcal{H}_0$ if $\delta_c = \Phi(\sqrt{N}\hat{\rho}_N)$ is too close to 0 or 1.

As we will see later on, several generators fail these overlapping entropy tests with relatively small sample sizes. In particular, some generators fail unequivocally a test that makes 10000 calls to the generator. The test finds that the successive entropies are clearly correlated and have the wrong average. For comparison, we also tried standard tests on the average of the U_i and on the correlation between the successive U_i, for $i = 1, \ldots, N$, for the same generators with the same seeds. These tests detected nothing significant. These generators also pass other standard tests that we have tried (e.g., the serial test, etc.) for that sample size. Thus, the overlapping entropy tests appear useful.

4 Experimental results with random number generators

4.1 A selection of random number generators

We selected a few popular or recently-proposed random number generators, listed in Table II, and submitted them to entropy tests. The list is far from exhaustive. The aim here is not to recommend any specific generator, but rather to investigate the power of entropy-based tests to detect their deficiencies.

Generators LCG1 to LCG7 are well-known linear congruential generators (LCGs), based on a recurrence of the form $x_i = (ax_{i-1} + c) \bmod m$, with output $u_i = x_i/m$ at step i. LCG1 and LCG2 are recommended by Fishman and Moore [12] and a FORTRAN implementation of LCG1 is given in [11]. LCG3 is recommended by Law and Kelton [20], among others, and is used in the SIMSCRIPT II.5 and INSIGHT simulation languages. LCG4 is in numerous software systems, including the IBM and Macintosh operating systems, the SIMAN and SLAM II simulation languages, MATLAB, the IMSL library, the Numerical Recipes [39],

Table II

List of selected generators.

LCG1.	LCG with $m = 2^{31} - 1$ and $a = 950706376$.
LCG2.	LCG with $m = 2^{31} - 1$ and $a = 742938285$.
LCG3.	LCG with $m = 2^{31} - 1$ and $a = 630360016$.
LCG4.	LCG with $m = 2^{31} - 1$ and $a = 16807$.
LCG5.	LCG with $m = 2^{32}$, $a = 69069$, and $c = 1$.
LCG6.	LCG with $m = 2^{31}$, $a = 1103515245$, $c = 12345$.
LCG7.	LCG with $m = 2^{48}$, $a = 25214903917$, $c = 11$.
RAN1.	Shuffled LCG ran1 in [39].
INV1.	Implicit inversive with $m = 2^{31} - 1$ and $a_1 = a_2 = 1$.
INV2.	Explicit inversive with $m = 2^{31} - 1$ and $a = b = 1$.
INV3.	Implicit inversive with $m = 2^{32}$, $a = b = 1$, and $z_0 = 5$.
INV4.	Explicit inversive of [10] with $m = 2^{32}$, $a = 6$, and $b = 1$.
INV5.	Modified explicit inversive of [8] with $m = 2^{32}$, $a = 6$, and $b = 1$.
GFSR1.	GFSR-521 in the Appendix of [42].
GFSR2.	GFSR proposed in [13].
GFSR3.	GFSR proposed in [18].
GFSR4.	Twisted GFSR T800 proposed in [33].
RLUX1.	RANLUX with $L = 24$ (see [17]).
WEY1.	Nested Weyl with $\alpha = \sqrt{2}$ (see [16]).
WEY2.	Shuffled nested Weyl with $\alpha = \sqrt{2}$ (see [16]).
CLCG88.	Combined LCG in Fig. 3 of [21].
CMRG96.	Combined MRG in Fig. 1 of [24].

etc., and is suggested in several books and papers (e.g., [1, 37, 42]). LCG5 is used in the VAX/VMS operating system and on Convex computers. LCG6 and LCG7 are the `rand` and `rand48` functions in the libraries of the C programming language [38, 44]. RAN1 is a shuffled version of LCG4, proposed in [39]. The next five are inversive generators modulo m. Their output at step i is always $u_i = z_i/m$. INV1 is an *implicit* inversive generator of the form $z_i = (a_1 + a_2 z_{i-1}^{-1}) \bmod m$, where $0^{-1} \bmod m$ is defined as 0 (see [7]). INV2 is an *explicit* inversive generator of the form $x_i = (ai + b) \bmod m$, $z_i = x_i^{-1} \bmod m = x_i^{m-2} \bmod m$ [7, 15]. INV3 is an implicit inversive generator with power-of-two modulus $m = 2^e$, based on the recurrence: $z_i = T(z_{i-1})$ where $T(2^\ell z) = (a_1 + 2^\ell a_2 z^{-1}) \bmod 2^e$ for odd z (see [9]). INV4 and INV5 are explicit inversive generators with power-of-two modulus; INV4 is defined in [10] and INV5 is defined as in [8], with the recurrence: $z_i = i(ai + c)^{-1} \bmod 2^e$. GFSR1 is the GFSR generator based on the recurrence $x_i := x_{i-p} \oplus x_{i-q}$, with $p = 512$ and $q = 32$, where $\oplus$ denotes the bitwise exclusive-or, and with the initialization procedure given in the appendix of Ripley [42]. GFSR2 is another GFSR, with $p = 521$ and $q = 32$, with the initialization procedure of Fushimi [13], and GFSR3 is the GFSR generator given in Kirkpatrick and Stoll [18]. GFSR4 is a twisted GFSR generator proposed in [33]. RLUX1 is the RANLUX generator implemented by James [17], with luxury level $L = 24$. At this luxury level, RANLUX is equivalent to a subtract-with-borrow generator proposed in [32] and used, for example, in MATHEMATICA. WEY1 is a generator based on the nested Weyl sequence defined by $u_i = i^2\alpha \bmod 1$, where $\alpha = \sqrt{2}$ (see [16]). WEY2 implements the shuffled nested Weyl sequence proposed in [16], defined by $u_i = ((Mi^2\alpha \bmod 1) + 1/2)^2\alpha \bmod 1$, with $\alpha = \sqrt{2}$ and $M = 12345$. CLCG88 and CMRG96 are the combined LCG of L'Ecuyer [21] and the combined MRG given in Figure 1 of [24].

4.2 Results of entropy distribution tests

We now report on experiments with the selected generators, using the entropy distribution test based on H. To the generators of Table II, we applied the 14 tests whose parameters are given in Table III. The last column of the table gives the total number of calls to the generator for each test (L/s calls were made for each U_i). For the tests Q1 to Q3, we used a chi-square approximation to the moment-corrected version of H defined in (12), while for N1 to N11, we used a $N(0,1)$ approximation for H standardized by its exact mean and variance.

Table III

Parameters for entropy distribution tests

Test	N	n	k	r	s	L/s	Nb. Calls
Q1	1	2^{20}	2^{12}	0	12	1	2^{20}
Q2	1	2^{20}	2^{12}	10	12	1	2^{20}
Q3	1	2^{20}	2^{12}	10	4	3	$3 \cdot 2^{20}$
N1	1	2^{16}	2^{20}	0	20	1	2^{16}
N2	1	2^{16}	2^{20}	10	10	2	2^{17}
N3	1	2^{20}	2^{30}	0	30	1	2^{20}
N4	1	2^{20}	2^{30}	0	15	2	2^{21}
N5	1	2^{20}	2^{30}	10	15	2	2^{21}
N6	1	2^{20}	2^{30}	0	10	3	$3 \cdot 2^{20}$
N7	1	2^{20}	2^{30}	0	6	5	$5 \cdot 2^{20}$
N8	32	2^{20}	2^{30}	0	15	2	2^{26}
N9	32	2^{20}	2^{30}	10	15	2	2^{26}
N10	32	2^{20}	2^{30}	10	10	3	$3 \cdot 2^{25}$
N11	32	2^{20}	2^{30}	0	6	5	$5 \cdot 2^{25}$

Tables IV and V report the p-values for these tests. When $N = 1$, we report the p-value of the statistic H, whereas for the tests with $N = 32$, we report the p-values δ^+ and δ^- of the KS statistics at the second level. The negative values should be interpreted "modulo 1" and represent the significance levels close to 1. For example, -4.3E-5 means $1 - 4.3 \times 10^{-5}$. Values smaller than 10^{-15} are denoted by ϵ. Thus, a p-value of ϵ for $N = 1$ means that the entropy H is much too large (i.e., that the points fall in the cells in a much too regular manner), whereas $-\epsilon$ means that the entropy is much too small (the points fall too often in the same cells).

Table IV

Results (p-values) of the entropy distribution tests

Test		LCG1	LCG2	LCG3	LCG4	LCG5	LCG6	RAN1
Q1								
Q2						ϵ	ϵ	
Q3						ϵ	ϵ	
N1								
N2						$-\epsilon$	ϵ	
N3		ϵ	ϵ	ϵ	ϵ	3.6E-9	ϵ	ϵ
N4		ϵ	ϵ	ϵ	-5.4E-14	ϵ	ϵ	
N5		ϵ	5.7E-13	ϵ	ϵ	ϵ	ϵ	
N6		3.8E-12	1.0E-10	2.3E-5			1.9E-7	
N7		2.0E-12	1.2E-6	9.2E-7	2.2E-8			
N8	δ^+	ϵ	ϵ	ϵ	$-\epsilon$	ϵ	ϵ	
	δ^-	$-\epsilon$	$-\epsilon$	$-\epsilon$	ϵ	$-\epsilon$	$-\epsilon$	
N9	δ^+	ϵ	ϵ	ϵ	$-\epsilon$	ϵ	ϵ	
	δ^-	$-\epsilon$	$-\epsilon$	$-\epsilon$	ϵ	$-\epsilon$	$-\epsilon$	
N10	δ^+	ϵ	ϵ	-2.9E-11	-8.8E-7	ϵ	ϵ	
	δ^-	$-\epsilon$	$-\epsilon$	ϵ	ϵ	$-\epsilon$	$-\epsilon$	
N11	δ^+	ϵ	ϵ	ϵ	ϵ	ϵ	-1.0E-4	
	δ^-	-1.1E-13	-2.1E-14	-4.2E-14	$-\epsilon$	-2.7E-6	1.7E-9	

Table V

Results (p-values) of the entropy distribution tests (continued)

Test		INV1	INV2	INV3	INV4	INV5	GFSR3	RLUX1	WEY1	WEY2
Q1										
Q2				ϵ	ϵ	ϵ				
Q3				$-\epsilon$		-3.4E-5			$-\epsilon$	
N1										
N2					-6.0E-3	6.0E-3				
N3		ϵ	ϵ	1.6E-9	ϵ	5.9E-7		$-\epsilon$	-4.5E-3	ϵ
N4					-5.8E-3		6.0E-3			
N5				$-\epsilon$		3.8E-7		$-\epsilon$		
N6					$-\epsilon$	1.4E-4	-6.6E-3		$-\epsilon$	
N7									$-\epsilon$	
N8	δ^+				-5.7E-7	-2.6E-5				
	δ^-				ϵ	ϵ				
N9	δ^+			$-\epsilon$	3.6E-8	ϵ		$-\epsilon$	-1.0E-4	
	δ^-			ϵ	2.7E-8	-3.8E-7		ϵ	2.0E-13	
N10	δ^+			$-\epsilon$	$-\epsilon$	3.5E-14			$-\epsilon$	
	δ^-			ϵ	ϵ	3.0E-4			ϵ	
N11	δ^+				-4.1E-5				$-\epsilon$	
	δ^-				2.4E-13		2.6E-3		ϵ	

Only the *suspect* p-values, that is, those smaller than 0.01 or larger than 0.99, are given. The other entries are left blank. The generators or tests not mentioned in the tables had no suspect p-value in this sense, with one exception: CLCG88 had a p-value of 0.0034 for the test N1. We re-ran this test three times for this same generator with different seeds and it passed all three times, so we conclude that this exception was obtained by chance and is not enough to reject the generator. All other generators mentioned in Tables IV and V have at least one p-value smaller than 10^{-15} or larger than $1 - 10^{-15}$, and we believe that they can be rejected on the basis of these entropy tests. We also computed (in parallel) the sample correlation $\hat{\rho}_N$ in (14) for the same sets of parameters and the results of these correlation tests were consistent (in general) with those of the entropy tests.

All the LCGs with modulus $m = 2^{31} - 1$, as well as the five generators with power-of-two modulus 2^{31} or 2^{32} (that is, LCG5, LCG6, INV3, INV4, INV5), fail several tests in a spectacular way. Those with power-of-two modulus also fail with much smaller sample sizes when the tests are based on their middle bits. The explanation is that for these generators, the $(r+1)$th bit follows a periodic pattern with period length at most 2^{-r} times that of the most significant bit. Their middle bits then look less random than the most significant ones (see, for example, the tests Q2, Q3, and N2). The generators LCG1 to LCG6, RAN1, INV1 to INV5, RLUX, WEY1, and WEY2 all fail dramatically at least one test based on a sample size not exceeding 2^{20}, which represents in all these cases less than 1/2000 of the period length of the generator. The test N3, in particular, throws 2^{20} points into 2^{30} cells and many of these generators (e.g., the LCGs, the inversives, RAN1, and WEY2) fail it because the points tend to fall not often enough in the same cells. This is due to the fact that these generators never produce twice the same output value (over 31 or 32 bits) within their period length. The output of RLUX1 happens to have only 24 bits of precision (as in [17]). This explains why it fails the tests N3, N5, and N9, which look at more than the first 24 bits.

We made several other experiments besides those whose results are reported here. In particular, we made experiments to compare (empirically) the efficiencies of tests with the same total sample size Nn, but with different values of N. In general, the tests with $N = 1$ tended to be most sensitive. This indicates that the defects of the generators tend to be of a global nature. It often happens that for a large enough k, a certain fraction of the X_j are always zero, no matter how large is the sample size n. In this case, increasing n instead of N gives more power.

4.3 Results of entropy tests with overlapping

Tables VII and VIII report results of the average entropy test and entropy correlation test with overlapping. The test parameters are given in Table VI. The total number of calls to the generator for some of these tests is smaller than for the entropy distribution tests of Table III. The first four tests look at the 30 most significant bits of each number and compute the corresponding overlapping entropy. The next four look at the bits 11 to 20, using 3 successive numbers to construct each block of 30 bits. The last four tests take the bits 21 to 23 of each number, and 10 numbers are used for each block.

Table VI

Parameters for the entropy tests with overlapping

Test	N	n	L	r	s	Nb. Calls
C1	10^4	30	5	0	30	10 000
C2	10^5	30	5	0	30	100 000
C3	10^6	30	5	0	30	1 000 000
C4	10^7	30	5	0	30	10 000 000
C5	10^4	30	5	10	10	30 000
C6	10^5	30	5	10	10	300 000
C7	10^6	30	5	10	10	3 000 000
C8	10^7	30	5	10	10	30 000 000
C9	10^4	30	5	20	3	100 000
C10	10^5	30	5	20	3	1 000 000
C11	10^6	30	5	20	3	10 000 000
C12	10^7	30	5	20	3	100 000 000

Table VII

Results of the overlapping average entropy tests

Test	LCG5	LCG6	INV2	INV3	INV4	INV5	GFSR1	RLUX1	WEY1
C1			$-\epsilon$				8.9E-3	$-\epsilon$	
C2			-8.1E-5					$-\epsilon$	
C3								$-\epsilon$	
C4								$-\epsilon$	$-\epsilon$
C5									
C6									7.5E-3
C7				$-\epsilon$	1.1E-4	1.5E-4			
C8				$-\epsilon$	ϵ	ϵ			
C9		2.2E-4		$-\epsilon$	$-\epsilon$				5.3E-11
C10		ϵ		$-\epsilon$	$-\epsilon$	-1.8E-13			ϵ
C11	-9.0E-4	ϵ		$-\epsilon$	$-\epsilon$	$-\epsilon$			ϵ
C12	$-\epsilon$	ϵ		$-\epsilon$	$-\epsilon$	$-\epsilon$			ϵ

Table VIII

Results of the overlapping entropy correlation tests

Test	LCG5	LCG6	INV2	INV3	INV4	INV5	GFSR1	RLUX1	WEY1
C1			ϵ	4.8E-10		1.6E-4			
C2	3.3E-3		4.1E-10	ϵ					
C3	3.0E-6	-2.3E-5	1.8E-5	ϵ	6.8E-5				
C4	ϵ	$-\epsilon$		ϵ	ϵ	1.1E-6			
C5				7.7E-3					
C6									4.9E-4
C7				3.0E-4					ϵ
C8	1.9E-4	-7.1E-3		ϵ			6.9E-3		ϵ
C9				-3.4E-13	ϵ				ϵ
C10	-9.0E-3		3.7E-3	$-\epsilon$	ϵ		3.0E-3		ϵ
C11	-2.0E-12	7.6E-3		$-\epsilon$	ϵ		6.6E-3		ϵ
C12	$-\epsilon$	6.8E-15		$-\epsilon$	ϵ		1.0E-4	-8.8E-3	ϵ

Again, the five generators with modulus 2^{31} or 2^{32} fail decisively many tests based on the middle or low-order bits, sometimes with very small sample sizes (e.g., C1, C2, C9). For the average entropy tests, these five generators only fail the tests based on low-order bits (the average entropy tends to be too low), but for the correlation tests, they also fail for the high-order bits. The correlation tests are more sensitive than the average in some cases, less in others, so both tests seem useful. If we consider only the tests applied to the most significant bits (e.g., C1–C4), the overlapping entropy correlation tests do much better, in some cases, than all the other tests considered in this paper, for an equivalent total sample size. The explicit inversive generator INV2 fails some tests dramatically, but only with small sample sizes for the high order bits. This may appear curious at first sight, but can be explained as follows: the first n values produced by this generator are the inverses (modulo $2^{31}-1$) of the first n positive integers, divided by $2^{31}-1$, and it turns out that the inverses of smaller integers tend to have lower entropy for their high order bits in this case (the reader can easily observe this by writing down the inverses modulo $2^{31}-1$ of the first few positive integers). We applied the same tests to the explicit inverse generator with parameter values $m = 2^{31}-1$, $a = 1$ and $b = 993652$ (the last value was chosen randomly), and it passed. These results illustrate the fact that not all parameters are equally good for inversive generators. Practical guidelines for choosing the parameters are still lacking. The GFSR recommended by Ripley (GFSR1) has some trouble with these tests. We replicated twice, for this generator, the tests which gave p-values less than 0.01, and on one occasion we obtained a p-value of 1.8×10^{-7} for the correlation test C12. The failure of RLUX1 for the tests with $s = 30$ was expected, because RLUX1 has only 24 bits of precision, but note that it fails only for the average entropy and not for the correlation. This indicates again that these two tests are sensitive to different aspects.

We actually made many more experiments with these tests than what is reported here. For example, we tried the same tests as C9–C12, but with $r = 0$ instead of $r = 20$ (i.e., testing the 3 most significant bits), and replicated three times the entire set of tests. The GFSR generators GFSR1 and GFSR3 failed some of the tests (at significance levels less than 10^{-5}) in some of the replications, but passed in others. The test results for these GFSR generators depend very much on the initial seed. For the other generators, the spectacular failures (e.g., significance levels of ϵ or $-\epsilon$) observed for one seed are typically observed for an arbitrary seed. One exception is the generator INV2, for which changing the seed changes the behavior as explained previously.

We now consider some additional overlapping entropy tests based on longer bit strings (see Table IX). For such large n, the overlapping entropy distribution is unknown, so we apply only the correlation tests. As indicated in Table X, GFSR1, GFSR3, RLUX1, and WEY1 fail these tests decisively. This time, the failure of RLUX1 is not due to taking more than the first 24 bits, but is a consequence of a structural defect of this generator in dimension 25 (see, e.g., [3, 26]). The failure of GFSR1 and GFSR3 can be explained in a similar way. For example, the test C16 constructs blocks of 1000 bits by taking the first bit of 1000 successive output values. But these bits follow the linear recurrence $b_j = b_{j-p} \oplus b_{j-q}$, so the blocks have a rather simple structure, which the test can easily detect when n is large enough.

Table IX

Parameters for additional overlapping entropy correlation tests

Test	N	n	L	r	s
C13	10^6	250	10	0	10
C14	10^6	250	10	0	1
C15	10^6	1000	12	0	10
C16	10^6	1000	12	0	1

Table X

Results of additional overlapping entropy correlation tests

Test	LCG3	RAN1	INV2	INV3	INV4	GFSR1	GFSR3	RLUX1	WEY1	WEY2
C13						ϵ		3.3E-8	ϵ	
C14				6.7E-3			1.1E-15		ϵ	
C15	-8.1E-3	2.9E-3	6.8E-3		4.7E-3	ϵ			4.1E-5	-4.3E-3
C16						ϵ	ϵ		ϵ	

5 Conclusion

Only the generators LCG7, GFSR2, GFSR4, CLCG88, and CMRG96 passed all the tests in this paper. The GFSRs fail some other tests described in [34] due to their poor bit-mixing properties [2, 35]. There are of course other generators not mentioned here which pass all these tests and all other tests that we have tried, and are also reasonably fast. Listing them goes beyond the scope of this paper, but the reader can find some in references [28, 29, 34], for example. There is therefore no excuse not to discard those that fail decisively here. Most of them are still used extensively on a daily basis.

Our tests reject several generators after looking at only a small fraction of the period length. The LCGs considered here fail other tests (e.g., [22, 30, 46]). But the usual chi-square tests, as reported in previous papers, need a much larger sample size than what we have observed here before clear failure is observed. The inversive generators failed the chi-square tests in [30, 46] only when the sample size was close to the period length.

Of course, all generators eventually fail if one increases the sample size indefinitely, because they have finite period length and because of the conservation law for the total amount of correlation valid for all finite sequences [2]. But if the generator is well-designed and has long-enough period, a test may require a practically infeasible amount of computing time to detect the structure.

Ideally, meaningful statistical tests should be sensitive to the weaknesses that are regarded most harmful in arbitrary applications. However, without restricting the class of admissible applications, this is an elusive requirement. General purpose random number generators should pass a rich battery of statistical tests of different types. Since entropy is one of the most fundamental measures of randomness, entropy tests are certainly a useful addition to the existing collection of tests for random number generators.

Acknowledgments

This work has been supported by NSERC-Canada grants # ODGP0110050 and SMF0169893, FCAR-Québec grant # 93ER1654, and by the Dutch grants NWO B62-424 and STW-DTI66.4085. Richard Simard helped writing the computer programs and Stefan Wegenkittl gave useful comments.

References

[1] P. Bratley, B. L. Fox, and L. E. Schrage. *A Guide to Simulation.* Springer-Verlag, New York, second edition, 1987.

[2] A. Compagner. Operational conditions for random number generation. *Physical Review E*, 52(5-B):5634–5645, 1995.

[3] R. Couture and P. L'Ecuyer. On the lattice structure of certain linear congruential sequences related to AWC/SWB generators. *Mathematics of Computation*, 62(206):798–808, 1994.

[4] E. J. Dudewicz and E. C. van der Meulen. Entropy-based tests of uniformity. *Journal of the American Statistical Association*, 76(376):967–974, 1981.

[5] E. J. Dudewicz, E. C. van der Meulen, M. G. SriRam, and N. K. W. Teoh. Entropy-based random number evaluation. *American Journal of Mathematical and Management Sciences*, 15:115–153, 1995.

[6] J. Durbin. *Distribution Theory for Tests Based on the Sample Distribution Function.* SIAM CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia, 1973.

[7] J. Eichenauer-Herrmann. Inversive congruential pseudorandom numbers: A tutorial. *International Statistical Reviews*, 60:167–176, 1992.

[8] J. Eichenauer-Herrmann. Modified explicit inversive congruential pseudorandom numbers with power-of-two modulus. *Statistics and Computing*, 6:31–36, 1996.

[9] J. Eichenauer-Herrmann and H. Grothe. A new inversive congruential pseudorandom number generator with power of two modulus. *ACM Transactions on Modeling and Computer Simulation*, 2(1):1–11, 1992.

[10] J. Eichenauer-Herrmann and K. Ickstadt. Explicit inversive congruential pseudorandom numbers with power of two modulus. *Mathematics of Computation*, 62(206):787–797, 1994.

[11] G. S. Fishman. *Monte Carlo: Concepts, Algorithms, and Applications.* Springer Series in Operations Research. Springer-Verlag, New York, 1996.

[12] G. S. Fishman and L. S. Moore III. An exhaustive analysis of multiplicative congruential random number generators with modulus $2^{31} - 1$. *SIAM Journal on Scientific and Statistical Computing*, 7(1):24–45, 1986.

[13] M. Fushimi. Random number generation with the recursion $x_t = x_{t-3p} \oplus x_{t-3q}$. *Computational and Applied Mathematics*, 31:105–118, 1990.

[14] D. V. Gokhale. On entropy-based goodness-of-fit tests. *Computational Statistics and Data Analysis*, 1:157–165, 1983.

[15] P. Hellekalek. Inversive pseudorandom number generators: Concepts, results, and links. In C. Alexopoulos, K. Kang, W. R. Lilegdon, and D. Goldsman, editors, *Proceedings of the 1995 Winter Simulation Conference*, pages 255–262. IEEE Press, 1995.

[16] B. L. Holian, O. E. Percus, T. T. Warnock, and P. A. Whitlock. Pseudorandom number generator for massively parallel molecular-dynamics simulations. *Physical Review E*, 50(2):1607–1615, 1994.

[17] F. James. RANLUX: A Fortran implementation of the high-quality pseudorandom number generator of Lüscher. *Computer Physics Communications*, 79:111–114, 1994.

[18] S. Kirkpatrick and E. Stoll. A very fast shift-register sequence random number generator. *Journal of Computational Physics*, 40:517–526, 1981.

[19] D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms.* Addison-Wesley, Reading, Mass., second edition, 1981.

[20] A. M. Law and W. D. Kelton. *Simulation Modeling and Analysis.* McGraw-Hill, New York, second edition, 1991.

[21] P. L'Ecuyer. Efficient and portable combined random number generators. *Communications of the ACM*, 31(6):742–749 and 774, 1988. See also the correspondence in the same journal, 32, 8 (1989) 1019–1024.

[22] P. L'Ecuyer. Testing random number generators. In *Proceedings of the 1992 Winter Simulation Conference*, pages 305–313. IEEE Press, Dec 1992.

[23] P. L'Ecuyer. Uniform random number generation. *Annals of Operations Research*, 53:77–120, 1994.

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

[25] P. L'Ecuyer. Maximally equidistributed combined Tausworthe generators. *Mathematics of Computation*, 65(213):203–213, 1996.

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

[27] P. L'Ecuyer. Tests based on sum-functions of spacings for uniform random numbers. *Journal of Statistical Computation and Simulation*, 59:251–269, 1997.

[28] P. L'Ecuyer. Good parameters and implementations for combined multiple recursive random number generators. *Operations Research*, 47(1), 1999. To appear.

[29] P. L'Ecuyer. Tables of maximally equidistributed combined LFSR generators. *Mathematics of Computation*, 68(225), 1999. To appear.

[30] H. Leeb and S. Wegenkittl. Inversive and linear congruential pseudorandom number generators in empirical tests. *ACM Transactions on Modeling and Computer Simulation*, 7(2):272–286, 1997.

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

[32] G. Marsaglia and A. Zaman. A new class of random number generators. *The Annals of Applied Probability*, 1:462–480, 1991.

[33] M. Matsumoto and Y. Kurita. Twisted GFSR generators. *ACM Transactions on Modeling and Computer Simulation*, 2(3):179–194, 1992.

[34] M. Matsumoto and Y. Kurita. Twisted GFSR generators II. *ACM Transactions on Modeling and Computer Simulation*, 4(3):254–266, 1994.

[35] M. Matsumoto and Y. Kurita. Strong deviations from randomness in m-sequences based on trinomials. *ACM Transactions on Modeling and Computer Simulation*, 6(2):99–106, 1996.

[36] 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.

[37] S. K. Park and K. W. Miller. Random number generators: Good ones are hard to find. *Communications of the ACM*, 31(10):1192–1201, 1988.

[38] P. J. Plauger. *The Standard C Library*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.

[39] W. H. Press and S. A. Teukolsky. Portable random number generators. *Computers in Physics*, 6(5):522–524, 1992.

[40] M. P. Quine and J. Robinson. Efficiencies of chi-square and likelihood ratio goodness-of-fit tests. *Annals of Statistics*, 13:727–742, 1985.

[41] T. R. C. Read and N. A. C. Cressie. *Goodness-of-Fit Statistics for Discrete Multivariate Data*. Springer Series in Statistics. Springer-Verlag, New York, 1988.

[42] B. D. Ripley. Thoughts on pseudorandom number generators. *Journal of Computational and Applied Mathematics*, 31:153–163, 1990.

[43] P. J. Smith, D. S. Rae, R. W. Manderscheid, and S. Silbergeld. Approximating the moments and distribution of the likelihood ratio statistic for multinomial goodness of fit. *Journal of the American Statistical Association*, 76:737–740, 1981.

[44] Sun Microsystems. *Numerical Computations Guide*, 1991. Document number 800-5277-10.

[45] O. Vasicek. A test for normality based on sample entropy. *Journal of the Royal Statistical Society: Series B*, 38:54–59, 1976.

[46] S. Wegenkittl. Empirical testing of pseudorandom number generators. Master's thesis, University of Salzburg, 1995.