On the Performance of Birthday Spacings Tests with Certain Families of Random Number Generators

Pierre L'Ecuyer and Richard Simard

Département d'Informatique et de Recherche Opérationnelle
Université de Montréal, C.P. 6128, Succ. Centre-Ville
Montréal, H3C 3J7, Canada
e-mail: `lecuyer@iro.umontreal.ca`

Abstract

We examine how a statistical test based on discrete spacings between points, in 1 or more dimensions, detects the regularities in certain popular classes of random number generators. We provide a rule of thumb giving the minimal sample size for the test to reject the generator systematically, as a function of the generator's size (or period length), for generator families such as the linear congruential, Tausworthe, nonlinear inversive, etc. Full period linear congruential generators with a good behavior in the spectral test, for example, start to fail the 2-dimensional test decisively at sample sizes approximately equal to the cubic root of their period length (or modulus).

1 Introduction

A uniform random number generator (RNG) is designed much like a gigantic roulette wheel. It produces a periodic sequence of numbers between 0 and 1, viewed as a realization of a sequence of i.i.d. $U(0,1)$ random variables $U_0, U_1, \ldots$ (that is, independent and uniformly distributed over the interval $[0,1)$). The users normally *assume* the null hypothesis $\mathcal{H}_0$: "The U_i's are i.i.d. $U(0,1)$". Several goodness-of-fit tests for $\mathcal{H}_0$ have been proposed and studied in the past (see, e.g., [5,9,15,16,18,20] and references therein). No statistical

[1] This work has been supported by the National Science and Engineering Research Council of Canada grant no. ODGP0110050 and by the FCAR-Québec grant no. ER-3218 to the first author.

test can *prove* that an RNG is faultless. However, the tests often point out deficiencies or regularities, and when none of the applied tests detects anything, this is somewhat reassuring.

Suppose the initial state (seed) of the RNG is selected at random, uniformly over the state space, and the first t output values $u_0, \ldots, u_{t-1}$ are taken from the RNG. This means spinning the wheel and taking the first t values starting from its stopping position. Ideally, we would like the random vector $(u_0, \ldots, u_{t-1})$ to be uniformly distributed over the t-dimensional unit hypercube $[0,1)^t$. To approximate this, good RNGs are constructed (via a theoretical analysis) so that the multiset Ψ_t of all vectors of successive values $(u_0, \ldots, u_{t-1})$, obtained from all possible initial states, is evenly distributed over $[0,1)^t$ for $1 \leq t \leq t_1$, where t_1 is an arbitrary constant (e.g., 30 or so). This Ψ_t can be viewed in a sense as the *sample space* from which points are chosen at random to approximate the uniform distribution over $[0,1)^t$. Background on the construction of RNGs can be found in [5,6,8,15,19] and other references given there.

Since Ψ_t is typically (by design of the RNG) much more uniform and regular than a random point set [6,8,15,12,10], problems with the statistical behavior are expected if one uses nearly as many random numbers as the period length. What fraction of the period length can be used safely? It depends on the problem (or statistical test) and on the RNG. Here, we try to answer this question for variants of the birthday spacings test proposed in [18]. Systematic analyses with other tests can be found in [13,15,16].

To assess the interaction between the structure of the test and the structure of the set Ψ_t for specific RNG families, we take from each family considered an RNG with period length near 2^e, chosen on the basis of the usual theoretical quality criteria, for integers e ranging from 10 to 50 or so. We then examine how the p-value of the test evolves as a function of the sample size n. The typical behavior is that the p-values remain "reasonable" for n up to some threshold n_0, then converges to 0 exponentially fast with n. To examine the relationship between n_0 and e, we fit (crudely) a regression model of the form $\log_2 n_0 = \gamma e + \nu + \epsilon$ where γ and ν are constants and ϵ is the noise (or residual). The result gives an idea of the minimal period length, within a given RNG family, to be safe with respect to the test for the sample sizes that are practically feasible.

In the next section we define the test statistic and give its approximate distribution. In section 3 we discuss (heuristically) the power of the test for a certain class of alternatives and what we expect from it. The empirical results are given in section 4.

2 Birthday Spacings

We consider the following *discrete spacings* test. Partition the interval $[0, 1)$ into d equal segments. This determines a partition of $[0, 1)^t$ into $k = d^t$ cubic boxes of equal size. Number these boxes from 0 to $k - 1$, in the natural order: The box with lower left corner at $(i_0/d, \ldots, i_{t-1}/d)$ has number $i_{t-1} + di_{t-2} + \cdots + d^{t-1}i_0$. Generate nt random numbers $U_0, \ldots, U_{nt-1}$, construct the points $V_i = (U_{ti}, \ldots, U_{ti+t-1})$, $i = 0, \ldots, n-1$, and let $I_1 \leq I_2 \leq \cdots \leq I_n$ be the numbers of the boxes where these points have fallen, sorted by increasing order. Compute the *spacings* $S_j = I_{j+1} - I_j$, for $j = 1, \ldots, n-1$, and let Y be the number of collisions between these spacings, i.e, 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 sorted spacings. The n points can be viewed as the birthdays of n people in a world where years have k days, whence the name *birthday spacings* suggested in [18].

The distribution of Y is studied in Exercises 3.3.2-28–30 of [5]. If n is large and $\lambda = n^3/(4k)$ is small, Y follows approximately the Poisson distribution with mean λ, under $\mathcal{H}_0$. Knuth [5] suggests repeating the test several times and comparing the empirical distribution of the observed values of Y with its theoretical distribution via a chi-square test. Here we use (instead) a single value of Y with large values of n and k. In our examples, this gives a more sensitive test than following Knuth's suggestion with N replications of a test based on n/N points (i.e., the same total sample size). Of course, the size of the computer memory imposes practical limits on the size of k that one can handle. If the S_j's are stored in a hashing table, the limit is on n instead of k. If one wants $N > 1$ to obtain a total sample size that exceeds this limit, or for some other reason, we recommend adding the N values of Y: Under $\mathcal{H}_0$, the sum is approximately Poisson distributed with mean $N\lambda$. Unless specifically indicated, we assume $N = 1$. If Y takes the value y, the right p-value is

$$p_{\mathrm{R}} = P[Y \geq y \mid \mathcal{H}_0] \approx 1 - \sum_{x=0}^{y-1} \frac{e^{-\lambda}\lambda^x}{x!}.$$

The error of approximation of the distribution of Y by the Poisson(λ) is (asymptotically) $O(\lambda^2/n)$ [5]. We made empirical studies to assess the effect of this error on our test results, and found that the effect is negligible (for the types of experiments that we do here) when $k \geq (4N\lambda)^4$, i.e., $Nn^3 \leq k^{5/4}$. This rule of thumb was found by extensive simulations with *different* robust RNGs proposed in [12,10]. The results were very consistent across the different RNGs.

3 What to Expect From the Tests?

For typical RNGs, the set Ψ_t has a very regular structure in $[0,1)^t$ [6,8,16]. This is especially true for RNGs based on a linear recurrence, such as linear congruential and multiple recursive generators, which have a well-known lattice structure [5,14]. If the structure is too regular, the spacings S_j will tend to be concentrated on a small subset of the non-negative integers. Suppose for example that all the points of Ψ_t are in cells whose number is a multiple of some integer $b > 1$. All the spacings are then necessarily multiples of b as well, and the number of collisions remains the same if we divide all the cell numbers and the spacings by b. Then, with k boxes and under the hypothesis $\mathcal{H}_1$: "The points are i.i.d. uniformly distributed over the boxes whose numbers are multiples of b", the number of collisions has the same distribution as under $\mathcal{H}_0$ with k/b boxes. So Y is approximately Poisson with mean $\lambda_1 = b\lambda$ under $\mathcal{H}_1$. For a concrete illustration, if $\lambda = 1$ and Y takes the value b (its mean under $\mathcal{H}_1$), the right p-value is

$$p_{\mathrm{R}}(b) = P[Y \geq b \mid \mathcal{H}_0] \approx 1 - \sum_{x=0}^{b-1} e^{-1}/x! = \frac{1}{e}\sum_{x=b}^{\infty} 1/x!.$$

This expression decreases with b at an exponential rate. For example, one has $p_{\mathrm{R}}(4) = 1.9 \times 10^{-2}$, $p_{\mathrm{R}}(8) = 1.0 \times 10^{-5}$, $p_{\mathrm{R}}(16) = 1.9 \times 10^{-14}$, and $p_{\mathrm{R}}(32) = 1.4 \times 10^{-36}$. A test with $\lambda = 1$ thus has enormous power as soon as b exceeds a few units. On the other hand, in order to get a powerful test, λ and λ_1 should not be too close to 0, otherwise Y will be 0 with significant probability, and $\mathcal{H}_0$ will not be rejected even if it is grossly violated. To maintain $\lambda \approx 1$ when k and n increase, we must have $n^3 \approx 4k$. If we take k proportional to the period length ρ of the RNG (a sensible heuristic) and want to keep $\lambda = O(1)$, we need a sample size $n = O(\rho^{1/3})$.

We conclude from this heuristic reasoning that for very regular RNGs, the test should require a minimal sample size proportional to the cubic root of the period length to get systematic rejection. Our empirical results in the next section show that this is exactly what happens for certain types of generators. This kind of *interaction* between the test and the structure of the RNG occurs in particular for LCGs, because of their lattice structure.

4 Empirical Evaluation for RNG Families

4.1 Selected Families of RNGs

We now report systematic experiments to assess the effectiveness of birthday spacings tests for detecting the regularities in specific families of *small* RNGs (we denote each family by an acronymn, as defined below). Within each family, we constructed a list of specific RNG instances with period lengths near 2^e, for e ranging from 10 to 50. These RNGs are too small to be considered for serious general purpose softwares, but they are good testbeds for studying the behavior of larger instances from the same families. At step ℓ, a generator outputs a number $u_\ell \in [0, 1)$.

Combined linear feedback shift register (or Tausworthe) generators with three components (*LFSR3*) are defined by $x_{j,\ell} = (a_{r_j} x_{j,\ell-r_j} + a_{k_j} x_{j,\ell-k_j}) \bmod 2$, $u_{j,\ell} = \sum_{i=1}^{32} x_{j,\ell s_j+i-1} 2^{-i}$, for $1 \leq j \leq 3$, and $u_\ell = u_{1,\ell} \oplus u_{2,\ell} \oplus u_{3,\ell}$, where $\oplus$ means bitwise exclusive-or, and the (k_j, r_j, s_j) are constant parameters selected so that the k_j are reasonably close to each other, and the sequence $\{u_\ell\}$ has period length $(2^{k_1} - 1)(2^{k_2} - 1)(2^{k_3} - 1)$ and is maximally equidistributed (see [7] for the definition and further details about these generators). *GoodLCG* denotes linear congruential generators (LCGs): $x_\ell = a x_{\ell-1} \bmod m$ and $u_\ell = x_\ell / m$, where m is a prime near 2^e, and a is selected so that the LCG has period length $m - 1$ and an excellent behavior with respect to the spectral test (i.e., excellent lattice structure) in up to at least 8 dimensions [11]. The *BadLCG2*'s have the same structure, except that their a is chosen so that they have a mediocre lattice structure in 2 dimensions. More details and the values of a and m can be found in [11] and [15]. *MRG2* refers to multiple recursive generators of order 2, of the form $x_\ell = (a_1 x_{\ell-1} + a_2 x_{\ell-2}) \bmod m$ and $u_\ell = x_\ell / m$, period length $m^2 - 1$, and excellent lattice structure [6,8]. The *CombLCG2*'s combine two LCGs via $x_{j,\ell} = a_j x_{j,\ell-1} \bmod m_j$ for $1 \leq j \leq 2$ and $u_\ell = ((x_{1,\ell} - x_{2,\ell}) \bmod m_1)/m_1$, so that the combined generator has period length $(m_1 - 1)(m_2 - 1)/2$ and a good lattice structure (see [17] for details). *CombCubic2* refers to combined cubic generators with 2 components, defined in [3,15]. *LFSR-LCG2* and *Cub-LCG* denote an LCG combined with an LFSR generator, and with a cubic generator [3,15], respectively. *InvExpl*, *InvImpl*, *InvExpl2a*, *InvExpl2b*, *InvImpl2a*, and *InvImpl2b* are inversive congruential generator families proposed by Eichenauer-Hermann and his co-authors (see [1,4] and the references given there). The first two are (explicit and implicit, respectively) inversive modulo a prime, whereas the other four are inversive modulo a power of 2.

For a given class of RNGs, given t, and a way of choosing k, we apply

Table 1
The suspect p-values $P[Y \geq y \mid \mathcal{H}_0]$ for the GoodLCGs, $t = 2$, $n = 2^\nu m^{1/3}$, $\lambda = 1$.

e	$\nu = 0$	$\nu = 1$	$\nu = 2$	$\nu = 3$
30			8.3E-5	ϵ
31			1E-5	ϵ
32				ϵ
33			1E-5	ϵ
34			1E-5	ϵ
35				ϵ
36				ϵ
37				ϵ
38			1E-5	ϵ
39			8.3E-5	ϵ
40			8.3E-5	ϵ
41			5.9E-4	ϵ
42				ϵ
43				ϵ
44				ϵ
45			1.1E-7	ϵ
46				ϵ
47			1E-5	ϵ
48			8.3E-5	ϵ
49				ϵ

the test for different values of e and with sample size $n = 2^{\gamma e+\nu}$, for $\nu = \ldots, -1, 0, 1, 2, \ldots$, where the constant γ is chosen so that the test starts to fail at approximately the same value of ν for most e. More specifically, we define $\tilde{\nu}$ (resp. ν^*) as the smallest values of ν for which $p_{\mathrm{R}}(y) < 10^{-2}$ (resp. $p_{\mathrm{R}}(y) < 10^{-15}$) for a majority of values of e.

4.2 Test Results: Examples and Summary

Table 1 gives a sample of the right p-values for the class *GoodLCG*, for $t = 2$, $d = \lfloor n^{3/2}/2 \rfloor$ (so $\lambda \approx 1$) and $n = 2^\nu m^{1/3}$. Only the p-values less than 0.01 are displayed. The symbol ϵ means $< 10^{-15}$. The p-values for $e < 30$ (not shown in the table) are very similar to those for $e \geq 30$. The small p-values appear with striking regularity; one has $\tilde{\nu} = 2$, $\nu^* = 3$. The test fails because the (lattice) point structure of the LCGs is too regular and the test interacts with the lattice structure of the LCG as explained in Section 3.

Table 2 gives the values of $\tilde{\nu}$ and ν^* for the selected RNG families, for $t = 2, 4, 8$ dimensions. For certain families, no result is reported for $t = 2$ because $\log_2 d$

Table 2
Test results for RNG families, with $\lambda = 1$, $k \approx 2^e$, $n = 2^{\gamma e + \nu}$.

	t	γ	$\tilde{\nu}$	ν^*
GoodLCG	2		2	3
	4	1/3	3	4
	8		4	5
BadLCG2	2		1	2
	4	1/3	3	4
	8		4	5
LFSR3	2		4	6
	4	2/5	5	6
	8		6	7
MRG2	4	1/3	3	4
	8		4	5
CombLCG2	4	1/3	4	4
	8		4	5
CombCubic2	4	1	1	1
	8	1/2	6	7
LFSR-LCG2	2		2	3
	4	1/2	3	3
	8		4	5

	t	γ	$\tilde{\nu}$	ν^*
Cub-LCG	2		2	3
	4	1/2	3	3
	8		4	5
InvExpl	4	1	1	1
	8		1	1
InvImpl	4	1	1	1
	8		1	1
InvExpl2a	2		1	2
	4	1/2	1	2
	8		1	2
InvExpl2b	2		2	3
	4	1/2	1	2
	8		2	3
InvImpl2a	2		2	3
	4	1/2	1	2
	8		1	2
InvImpl2b	2		0	0
	4	2/3	2	2
	8		1	2

exceeds the number of bits of precision produced by the generator before the test fails (with $\lambda = 1$ and our choice of parameters). In general, the factor γ, which characterizes the speed at which a generator of a given family fails the test, is the same for all t (at least for $t \leq 8$). One remarkable exception is *CombCubic2*, for which a significant defect shows up in 5 (or more) dimensions and becomes increasingly dominant for larger t. (This defect will be explained elsewhere.) The simple or combined LCGs fail with $\gamma = 1/3$. The *BadLCG2* are worse than the other *LCGs* in 2 dimensions, as can be expected from their mediocre lattice structure in 2 dimensions, but have the same γ. Combining two different types of generators may improve their behavior (as can be seen from the *LFSR-LCG2* which fail with $\gamma = 1/2$ as compared to $\gamma = 2/5$ and $\gamma = 1/3$ for their respective components). The two inversive families *InvImpl* and *InvExpl*, with prime modulus m, fail the test only when the sample n is larger than the period ($n = 2^{e+1} \approx 2m$), while the other four families, with modulus $m = 2^e$, fail for n much smaller than the period ($n \approx \sqrt{m}$ for three of the families). The *InvImpl2b* are defined in [2]. In general, nonlinear generators perform better in this test than the linear ones. The *CombCubic2, InvImpl* and *InvExpl* are the only ones which pass the test for n as large as the period.

4.3 Test Results: real-life RNGs

We also tested several well-known and widely used LCGs, and they failed as predicted by the model of Table 2. For example, the LCGs with modulus $m = 2^{31} - 1$ and multipliers $a = 16807$, 950706376, 742938285, and 630360016, all failed with $p_{\mathrm{R}} < 10^{-15}$ at sample size $n = 2^{13}$. The generator `drand48` of the SUN standard C library, an LCG with modulus 2^{48}, multiplier $a = 25214903917$, and additive constant $c = 11$, failed with $p_{\mathrm{R}} < 10^{-15}$ at sample size $n = 2^{19}$.

References

[1] J. Eichenauer-Herrmann. Pseudorandom number generation by nonlinear methods. *International Statistical Reviews*, 63:247–255, 1995.

[2] 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.

[3] J. Eichenauer-Herrmann and E. Herrmann. Compound cubic congruential pseudorandom numbers. *Computing*, 59:85–90, 1997.

[4] J. Eichenauer-Herrmann, E. Herrmann, and S. Wegenkittl. A survey of quadratic and inversive congruential pseudorandom numbers. In P. Hellekalek, G. Larcher, H. Niederreiter, and P. Zinterhof, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, volume 127 of *Lecture Notes in Statistics*, pages 66–97, New York, 1997. Springer.

[5] D. E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms.* Addison-Wesley, Reading, Mass., third edition, 1997.

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

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

[8] P. L'Ecuyer. Random number generation. In Jerry Banks, editor, *Handbook of Simulation*, pages 93–137. Wiley, 1998. chapter 4.

[9] P. L'Ecuyer. Random number generators and empirical tests. In P. Hellekalek, G. Larcher, H. Niederreiter, and P. Zinterhof, editors, *Monte Carlo and Quasi-Monte Carlo Methods in Scientific Computing*, volume 127 of *Lecture Notes in Statistics*, pages 124–138. Springer, New York, 1998.

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

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

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

[13] P. L'Ecuyer, J.-F. Cordeau, and R. Simard. Close-point spatial tests for random number generators. *Operations Research*, 1999. To appear.

[14] 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.

[15] 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.

[16] P. L'Ecuyer, R. Simard, and S. Wegenkittl. Sparse serial tests of uniformity for random number generators. Submitted, 1998.

[17] P. L'Ecuyer and S. Tezuka. Structural properties for two classes of combined random number generators. *Mathematics of Computation*, 57(196):735–746, 1991.

[18] 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.

[19] 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.

[20] M. S. Stephens. Tests for the uniform distribution. In R. B. D'Agostino and M. S. Stephens, editors, *Goodness-of-Fit Techniques*, pages 331–366. Marcel Dekker, New York and Basel, 1986.