Kangweon-Kyungki Math. Jour. **14** (2006), No. 1, pp. 85–93

ON THE INITIAL SEED OF THE RANDOM NUMBER GENERATORS

Tae-Soo Kim and Young-Kyun Yang*

Abstract. A good arithmetic random number generator should possess full period, uniformity and independence, etc. To obtain the excellent random number generator, many researchers have found good parameters. Also an initial seed is the important factor in random number generator. But, there is no theoretical guideline for using the initial seeds. Therefore, random number generator is usually used with the arbitrary initial seed. Through the empirical tests, we show that the choice of the initial values for the seed is important to generate good random numbers

1. Introduction

The ability to generate satisfactory sequences of random numbers is one of the key links between Computer Science and Statistics. Standard methods may no longer be suitable for increasingly sophisticated uses, such as in precision simulation studies. A simulation of any system or process in which there are inherently random components requires a method of generating or obtaining numbers that are random, in some sense. All the randomness required by the simulation model is simulated by various random number generators whose output is assumed to be a sequence of independent uniform random variables, which is denoted "$U(0,1)$". These random numbers are then transformed as needed to simulate random variables from different probability distributions. But, the random variable in $U(0,1)$ is an mathematical abstraction. In practice, there are no true random variables. As of today, from a prescribed mathematical formula but satisfy different requirements as if they were

Received April 10, 2006.

2000 Mathematics Subject Classification: 68U20, 65C05, 62G10.

Key words and phrases: random number, random number generator, empirical tests.

* Corresponding author.

true random numbers, we gain the sequence. Such a sequence is called the pseudo-random and the program or the procedure that produce such a sequence is called pseudo-random number generator. The study of the methodology of pseudo-random numbers has a long history. The most popular algorithm for generating pseudo-random numbers was suggested by Lehmer in 1949. It is called the congruential method. The method relies on a sequence of integers that are computed by one formula

$$m_i = g(m_{i-1}, m_{i-2}, \cdots)(modM) \tag{1}$$

where a fixed deterministic function g of previous given elements $m_{i-1}, m_{i-2}, \cdots$, the modulo M are prescribed integers. As pseudo-random numbers, the fractions m_i/M are used. In particular, if g is a linear function of $m_{i-1}, m_{i-2}, \cdots$, we called it as a linear congruential generator (LCG). In general the LCG is probably the most widely used and best understood kind of random-number generator. Turning to small M the length of period reduces. On the other hand, if a long period generator is implemented, then the generation is slow. So there are many alternative types. In order to the formula (1.1) have the full period and good statistical properties, the values of the parameters in a function g must be carefully chosen [1,4,7]. To generate pseudo-random numbers of long period and good statistical properties, methods recommended by many scholars are the Multiple Recursive Generator [3, 4, 8, 9] and the Combined Generator [5,8,10]. In particular, we studied two combined multiple recursive generators which were designed by L'Ecuyer[9]. We have interest to the statistical properties of generators.

In the formula (1.1), when $g(m_{i-1}, m_{i-2}, \cdots, m_{i-q}) = a_1 m_{i-1} + a_2 m_{i-2} + \cdots + a_q m_{i-q}$, where a_i's are constants and the initial values $m_{i-1}, m_{i-2}, \cdots, m_{i-q}$ are not all zero. We called them the qth-order multiple recursive generators (MRGs). From the finite field theory, the qth-order MRGs can produce random numbers of full period $M^q - 1$ if and only if the polynomial $f(x) = x^q - a_1 x^{q-1} - \cdots - a_q$ is a primitive polynomial modulus M. Knuth [6] describes the following conditions for testing the primitiveness modulo M :

(i)$(-1)^{q-1} a_q$ is a primitive root modulo M,

(ii) $[x^r \, mod \, f(x)] \; mod \; M = (-1)^{q-1} a_q$,

(iii) degree$\{[x^{r/s} \; mod \; f(x)] \; mod \; M\} > 0$, for each prime factor s of r, where $r = \dfrac{M^q - 1}{M - 1}$.

Theoretically, there are exactly choices of $(a_1, a_2, \cdots, a_q)$ which satisfy these conditions, where $\phi(M^q{-}1)$ is the Euler function defined as number of integers which is smaller than and relatively prime to $M^k - 1$. For the simplest case of $q = 2$ and the very popular modulus $M = 2^{31} - 1$, there are around $5.74E17$'s candidates [4]. Hence a significant amount of computation is involved in searching for $(a_1, a_2, \cdots, a_q)$ which are able to produce random numbers of full period.

To increase the period and try to get rid of the regular patterns displayed by LCGs, it has often been suggested that different generators be combined to produce a hybrid one. Such combination is often viewed as completely heuristic and is sometimes discouraged. But besides being strongly supported by empirical investigations, combination has some theoretical support. First, in most cases, the period of the hybrid is much longer than that of each of its components, and can be computed. Second, there are theoretical results suggesting that some forms of combined generators generally have better statistical behavior. In this paper, we think about the combination of two MRGs, which w as developed and studied by L'Ecuyer, is defined by

$$m_{1,i} = (a_{1,1}m_{1,i-2} - a_{1,2}m_{1,i-3})[\mod (2^{32} - 209)], \tag{2}$$

$$m_{2,i} = (a_{2,1}m_{2,i-1} - a_{2,2}m_{2,i-3})[\mod (2^{32} - 22853)], \tag{3}$$

$$Y_i = (M_{i,i} - m_{2,i})[\mod (2^{32} - 209)], \tag{4}$$

$$U_i = \frac{Y_i}{2^{32} - 209}, \tag{5}$$

where $a_{1,1} = 1403580$, $a_{1,2} = 810728$, $a_{2,1} = 527612$, $a_{2,2} = 1370589$ and has period of approximately 2^{191} (which is about 3.1×10^{57}) as well as excellent statistical properties through dimension 32 [2]. The advantage of the above generator is a brief program, simple computations and a huge period. In order to use this algorithm, likewise using any other random generators, we need the seed vector with 6-elements $\{m_{1,0}, m_{1,1}, m_{1,2}, m_{2,1}, m_{2,2}, m_{2,3}\}$.

The choice of the initial seed vectors in random number generator could not be determined by the theoretical basis. The recommendation to select initial values at random is doubtful. In general, the initial seed vectors could be chosen by empirical methods. To be sure, the careful selection of the seeds is important to generate the pseudo-random numbers. So, L'Ecuyer gave the 10,000's seeds vector as related header-file and asserted that the results have excellent statistical properties. But, for the empirical test to see the uniformity and independence of the two combined-MRGs, we obtained the different results. The test results will be given in the next section.

2. The Empirical Tests

In general, theoretical test examine global randomness. However, since most of the time, only a small fraction of the whole cycle of random numbers will be used in simulation studies, the local randomness is also very important. The local evaluation is usually performed by statistically testing subsequences of random numbers produced from a generator to see how close those numbers resemble i.i.d. uniform random variable. Some famous statistical tests are the runs and auto-correlation tests for testing independence and the chi-square (or frequency) and serial tests for testing uniformity in different dimensions. In this section, we practice the various simulations to test the uniformity and independence of distribution of the corresponding pseudo-random numbers. And all tests are related to the deterministic interpretation of goodness-of-fit tests. In facts, d-dimensional random points with independent Cartesian coordinates $(\gamma_1, \cdots, \gamma_d), (\gamma_{d+1}, \cdots, \gamma_{2d}), (\gamma_{2d+1}, \cdots, \gamma_{3d}), \cdots$ are uniformly distributed in the d-dimensional unit cube at any d. This property is necessary and sufficient for a successful implementation of Monte Carlo algorithms with constructive dimension d. To test whether the null hypothesis H_0 : the above d-tuples sequences are distributed uniformly on $[0,1]^d$, is true or not, divide $[0,1]$ into k subintervals of equal size and let $f_{j_1,j_2,\cdots,j_d}$ be the number of γ_i's having first component in subinterval j_1, second component in subinterval j_2, etc. If we let $\chi_N^2 = \frac{k^d}{N}\sum_{j_1=1}^{k}\cdots\sum_{j_d=1}^{k}(f_{j_1,j_2,\cdots,j_d} - \frac{N}{k^d})^2$, then χ_N^2 will have an approximate chi-square distribution with degree of freedom $k^d - 1$, under the null hypothesis H_0 is true. The smaller is χ_N^2 the better is the agreement of empirical values with theoretical ones. Large values χ_N^2 correspond to small p-values. So, too small values of p-values indicate that the experimental data contradicts to our uniformity hypothesis. Firstly, for the uniformity, we have tested for the case $d = 1$, which is called the frequency or chi-square test, and $d = 2, 3, 4$, which are called the serial tests. For modelling different problems, different quantities of pseudo-random numbers are necessary. Therefore, we have simulated various initial seeds of a sequence with lengths $N = N_d \times 2^s$, where $s = 0, 1, 2, \cdots, 14$, $600, 300, 250, 150$, according to the $d = 2, 3$ and 4, respectively. And let k the number of subintervals of $[0, 1]$ be as $16, 8, 5$, and 4 with respect to the $d = 1, 2, 3$, and 4.

Secondly, for the test of independence, we think the run test. Let n_i be the number of runs of length i in a sequence of $N = 600 \times 2^s$, where $s = 0, 1, 2, \cdots, 14$. For an independent sequence, the expected values of n_i for runs up and down are given by
(1)

$$E(n_i) = \begin{cases} \dfrac{2}{(i+3)!}[N(i^2 + 3i + 1) - (i^3 + 3i^2 - i - 4)], & i \leq N - 2 \\ \dfrac{2}{N!}, & i = N - 1 \end{cases}$$

Under the null hypothesis H_0 : the pseudo-random numbers which are generated by the two combined MRGs is distributed independently. We know $\chi^2_N = \sum_{i=1}^{4} \frac{(n_i - np_i)^2}{np_i} + \frac{(n'_5 - np_5)^2}{np_5}$ where n'_5 is the number of run with length larger than 5, $n = n_1 + n_2 + n_3 + n_4 + n'_5$ means the total number of runs, and the probabilities $p_i = E(n_i)$, for $i = 1, 2, \cdots, N-1$, will have an approximate chi-square distribution with degree of freedom 4.

For all tests, we use $\Phi_i = \max_s \chi^2_N$, for $i = 1$, which means the frequency test, for $i = 2, 3$ and 4, which means the 2, 3, and 4 dimensional serial tests, respectively, for $i = 5$, which means the run test as the criteria. When all values of Φ_i are less than the quantiles Φ^*_i for this tests with respect to p-values as 0.1, we will say that the pseudo-random numbers generated by two-combined MRG are distributed uniformly and independently. The recommendation of L'Ecuyer was arbitrarily to select an initial value in $10,000$'s seed vectors was proposed in his header-file. We have tested arbitrary 100 sequences initial seed vectors among $10,000$. And we selected the seed vectors meets criteria in all five tests at the same time. The results of the above tests are terrible. The only one 5230th seed vector $(1338960199, 3947731640, 1058186044, 1875415108, 1948201518, 3217931286)$ passed the all five tests. And the results Φ_i and $P_i = \max_i P(\chi^2_N) = \min_i \int_{\chi^2_N}^{\infty} f(x)dx$, where $f(x)$ is a probability density of χ^2_N with degree of freedom $k^d - 1$, of each tests are described in Table 1.

Table 1. The results of test with the 5230th initial seed vector

	Values of χ^2_N in each tests			
s	Frequency	Serial:2-dim	Serial : 3-dim	Run Test
0	15.6267	57.5467	118.75	1.44922
1	19.1733	56.2133	110	5.62656
2	12.52	69.6533	136.25	1.77436
3	12.1667	57.4933	133.75	4.17822
4	12.7433	46.32	124.922	2.46628
5	7.68667	55.7067	102.852	4.86404
6	7.035	54.9533	98.0469	1.7428
7	10.5175	48.9233	88.8867	1.42989
8	16.8548	72.095	110.542	1.57092
9	17.3196	75.4642	105.469	1.35517
10	19.6557	62.1771	106.177	3.69256
11	11.6118	61.3904	128.611	7.6133
12	15.2261	64.9315	144.329	3.31268
13	11.0268	53.8317	133.254	2.15742
14	13.4993	64.3363	136.213	3.42884
$\Phi_i = \max_s \chi^2_N$	19.6557	75.4642	144.329	7.6133
$\min_i P(\chi^2_N)$	0.19	0.14	0.1	0.11
Φ^*_i with p-value 0.1	22.3	77.7	145	7.78

Continuously, we proceed with the five empirical tests for all given $10,000$'s seed vectors. It required the very enormous test time. We found out the only 44 of $10,000$ passed all five tests. Table 2 and Table 3 at the end of the paper show the result of tests.

3. Conclusions

An ideal random number generator should possess at least the properties of long period, good lattice structure, and sound statistical properties. To generate good pseudo-random numbers, one method recommended by many scholars is the multiple recursive and combined generator. We present empirical tests for two combined MRGs. As L'Ecuyer asserted that the generator has a good theoretical property, but the empirical tests shows the different results.

In simulation studies, the quality of the random number generator adopted has a major effect on the results derived. The arbitrary selections of the initial seed values in the random number generators would be not suitable results. So, we select the initial conditions with attention. As a future theme, we would find the theoretical condition for good random number generator in various cases.

References

[1] Ana Proykova, *How to improve a random number generator*, Computer Physics Communications. **123** (2000), 125-131.

[2] Averill M. Law & W. David Kelton, *Simulation modeling and Analysis, 3rd Ed*, McGraw-Hill. (2000).

[3] Chiang Kao & Huey-Chin Tang, *Upper Bounds in Spectral Test for Multiple Recursive Random Number Generators with Missing Terms*, Computers Math. Applic. **vol. 33**, (1997), 119-125.

[4] Chiang Kao & Hui-Chin Tang, *Several Extensively Tested Multiple Recursive Random Number Generator*, Computers Math. Applic. **Vol 36** (1998), 129-136.

[5] I. M. Sobol & Yu. L. Levitan, *A Pseudo-Random Number Generator for Personal Computers*, Computers and Mathematics with Applications. **Vol 37** (1999), 33-40.

[6] Knuth, D.E, *The Art of Computer Programming, Vol.2 : Seminumerical Algorithms, 3d ed.*, Addison-Wesley, Reading, Massachusetts. (1998).

[7] Pierre L'Ecuyer, *Efficient and Portable Combination Random Number Generators*, Communications of the ACM. **No. 6**, (1988), 742-749.

[8] Pierre L'Ecuyer, *Random Numbers for Simulation*, Communications of the ACM. **Vol. 33, No. 10** (1990), 85-97.

[9] Pierre L'Ecuyer, Francois Blouin & Raymond Couture, *A Search for Good Multiple Recursive Random Number Generators*, ACM Transactions on Modeling and Computer Simulation. **Vol.3, No.2** (1993), 87-98.

[10] Pierre L'Ecuyer & Terry H. Andres, *A random number generator based on the combination of four LCGs*, Mathematics and Computers in Simulation. **44** (1997), 99-107.

School of Liberal Arts
Seoul National University of Technology
Seoul, 139–172, Korea
E-mail: ykyang@snut.ac.kr

Table 2. The list of numbers among 10,000 which passed the all five test in the L'Ecuyer's header-file

Seeds Number	The Seeds Vectors		
	$m_{1,0}$	$m_{1,1}$	$m_{1,2}$
74	3793615118	2750706029	2156058298
256	474425456	4013621006	1047529229
315	778777092	3506874608	886397267
420	2858021237	130793867	2576255171
1007	1357637645	1059427249	800951665
1385	2324283389	3980402648	909451590
1561	4048081921	3484009500	1949064959
2069	2648774766	1836017866	109487550
2744	2045043622	3736990058	2158192863
3139	225105283	1028800446	3475530378
4081	468568482	60748908	3600254120
4415	4250970605	2247141194	4160009317
4950	4001071387	1935425346	2569502716
5147	3458369350	1365751610	1950454722
5230	3217931286	1948201518	1875415108
5376	1869132997	3411217504	4246800601
5798	2534516661	3392823319	2126521932
6020	1462998075	3841141927	815069390
6105	321831138	2513261002	3158817632
6118	1844407534	713506037	3904241368
6154	3331527132	1971780948	951052068
6246	3711507128	3658041075	1732724216
6537	1823785284	3740987246	420862234
6921	2660619449	739866491	1523313346
7389	3370051646	2351773946	3578192525
7900	801753079	1053157281	3143374566
8372	2311663784	635058214	420512396
8983	2330960437	3519068800	264254434
8990	3496226347	2759155171	387573809
9329	2165744782	4129645042	1719314779
9424	3172187716	1889519277	712896719
9542	4220822992	2571281666	615285499
9718	2032201018	1586274056	1588256539
9998	560024289	1830276631	144885590

Table 3. The list of numbers among 10,000 which passed the all five test in the L'Ecuyer's header-file

Seeds	The Seeds Vectors		
Number	$m_{2,0}$	$m_{2,1}$	$m_{2,2}$
74	3079033430	2780569996	3936920391
256	2719529576	739324835	2964280517
315	387258206	219138949	2542372807
420	948143174	3901676992	4087606491
1007	1558460654	1972949201	3182661420
1385	3456420597	566308252	1902340646
1561	1932583407	3634728800	2787358029
2069	2022962442	3129355995	2917956914
2744	3952353473	3553708899	3379872074
3139	341271471	2907536336	2910932183
4081	1480623720	3200597697	3743886328
4415	19851053	3029883115	2473778054
4950	2843613632	1391350969	3143604249
5147	2112776806	1880897145	2809013922
5230	1058186044	3947731640	1338960199
5376	2727299193	2124744171	2208018226
5798	1644640541	2064925845	1553045961
6020	1378992995	787238713	3341259540
6105	548848962	3747010403	4151524440
6118	1863539380	2307868432	3912446738
6154	3071057510	4173447399	1708016892
6246	2827811352	3899843311	3845035395
6537	3065014647	974128584	3925274174
6921	1754164860	656162706	3755724112
7389	2668422752	4168309552	337611966
7900	4201753809	2762737338	3163930922
8372	3997997619	803364095	3678353094
8983	2694918818	3959029062	2393099014
8990	2458849830	3162364581	1962632124
9329	1209022018	2804053529	2557562793
9424	527235853	1060776700	1468758996
9542	689476507	1228137211	3484207157
9718	1860616301	765681796	3206901949
9998	1556615741	1597610225	1856413969