Twisted GFSR Generators

Makoto Matsumoto
Research Institute for Mathematical Sciences
Kyoto University, Kyoto 606 Japan
and
Yoshiharu Kurita
National Research Laboratory of Metrology
Tsukuba 305 Japan

April 27, 1992 (Revised Nov. 15)

Abstract

The generalized feedback shift register (GFSR) algorithm suggested by Lewis and Payne is a widely used pseudorandom number generator, but has the following serious drawbacks: 1. An initialization scheme to assure higher order equidistribution is involved and is time-consuming. 2. Each bit of the generated words constitutes an m-sequence based on a primitive trinomial, which shows poor randomness with respect to weight distribution. 3. Large working area is necessary. 4. The period of sequence is far shorter than the theoretical upper bound. This paper presents the twisted GFSR (TGFSR) algorithm, a slightly but essentially modified version of the GFSR, which solves all the above problems without loss of merit. Some practical TGFSR generators were implemented and they passed strict empirical tests. These new generators are most suitable for simulation of a large distributive system, which requires a number of mutually independent pseudorandom number generators with compact size.

1 Introduction

The generalized feedback shift register (GFSR) generator[12] is a widely used pseudorandom number generator based on the linear recurring equation

$$\mathbf{x}_{l+n} := \mathbf{x}_{l+m} \oplus \mathbf{x}_l, \quad (l = 0, 1, \ldots),$$

where each $\mathbf{x}_l$ is a word with components 0 or 1 of size w, and $\oplus$ denotes bitwise exclusive-or operation. Thereby, this algorithm generates the same number of m-sequences as the word length in parallel. Thus, by regarding one word as a real number between 0 and 1, the algorithm generates [0,1] uniform pseudorandom real number sequences. This algorithm has the following merits:

(m1) The generation is very fast. Generation of one pseudorandom number requires only three memory references and one exclusive-or operation.

(m2) The sequence has an arbitrarily long period independent of the word size of the machine.

(m3) The implementation is independent of the word size of the machine.

CR categories and Subject Descriptors: G. 2.1. [Discrete Mathematics] : Combinatorics–*recurrences and difference equations*; G. 3. [Probability and Statistics]–*random number generation.*

General Term: Algorithms, Experimentation.

Additional Key Words and Phrases: generalized feedback shift registers, matrix linear congruential generators, m-sequences, TLP-sequences, GF(2^m)

However, the algorithm has the following serious drawbacks:

(d1) The selection of initial seeds is very critical and influential in the randomness, and good initialization is rather involved and time-consuming.

(d2) Each bit of a GFSR sequence can be regarded as an m-sequence based on the trinomial t^n+t^m+1, which is known to have poor randomness [3][4][14][15] (see also the results of weight distribution tests in Section 4.5).

(d3) The period of a GFSR sequence 2^n-1 is far smaller than the theoretical upper bound; i.e., the number of possible states 2^{nw}.

(d4) The algorithm requires n words of working area, which is memory-consuming if a large number of generators are implemented simultaneously.

This paper describes a new generator named the *twisted GFSR generator* (TGFSR for short), which solves the above four drawbacks. The TGFSR generator is the same as the GFSR generator, except that it is based on the linear recurrence

$$\mathbf{x}_{l+n} := \mathbf{x}_{l+m} \oplus \mathbf{x}_l A, \quad (l = 0, 1, \ldots),$$

where A is a $w \times w$ matrix with 0-1 components and $\mathbf{x}_l$ is regarded as a row vector over GF(2). With suitable choice of n, m, and A, the TGFSR generator attains *the maximal period* $2^{nw}-1$; i.e., it assumes all possible states except the zerostate in a period.

Because of this maximality, the TGFSR generator improves on the above four drawbacks of the GFSR generator as follows:

(Tm1) The initialization is carefree. Any initial seed except for all zero produces the identical sequence disregarding the phase, and hence no special initialization technique, as shown in [5], is necessary. Moreover, since the working area is far smaller, initialization is very fast.

(Tm2) Recurrence is based not on a trinomial, but on a primitive polynomial with many terms.

(Tm3) The period of the generated sequence attains the theoretical upper bound $2^{nw}-1$.

(Tm4) The working area is one-w-th the size of that of the GFSR required to attain the same period. For example, assume that the word size of the machine is 32. Then, the TGFSR generator requires only 25 words for the working area to attain the period $2^{25*32}-1$, while the GFSR generator requires 800 words for the working area for the same period.

(Tm5) The sequence is n-equidistributed, which is best possible.

Below, we explain the *k-equidistribution* property[19], which is an indispensable condition for a good pseudorandom number sequence. For a positive integer k, a sequence $\mathcal{X} = \mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \cdots$ is said to be *k-equidistributed* if any nonzero k-tuple $(\mathbf{a}_0, \mathbf{a}_1, \ldots, \mathbf{a}_{k-1})$ (there are $2^{kw}-1$ such tuples) equally likely occurs as k-tuple $(\mathbf{x}_l, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_{l+k-1})$ in the sequence $\mathcal{X}$ for $l = 0, 1, 2, \ldots$. Now, our TGFSR sequence is n-equidistributed as shown in Section 2.2, which is best possible.

We can choose A so that $\mathbf{x}A$ can be computed with a few machine instructions, as shown in Section 2.1. Thus, the TGFSR generator is a fast algorithm generating highly equidistributed optimally long period pseudorandom sequences consuming only a small area of memory with a very easy and fast initialization scheme.

The TGFSR generators are most suitable for simulation of a large distributive system, which requires a large number of mutually independent pseudorandom number generators, for the following two reasons: (1) The TGFSR generators consume a far smaller amount of working area than do the GFSR. (2) A large number of distinct TGFSR generators can be implemented by merely changing the parameters m and A. This is different from the case of the GFSR, where the basic recurrence is determined by a primitive trinomial, which is rarely found.

In Section 2.1, the TGFSR algorithm is described, and compared with the original GFSR algorithm. In Section 2.2, the algebraic definition and some underlying theory are investigated. In Section 2.3, the merits and drawbacks of the GFSR and the TGFSR generators are compared. A comparison with a recent nice algorithm is also stated. In Section 3, a method for finding suitable values for parameters n, m, and A, is discussed. In Section 4, several TGFSR generators are listed, and intensive tests of randomness are done for several kinds of pseudorandom number generators. Some generators having previously been believed to provide good pseudorandom numbers are rejected. Section 5 offers our conclusions. Appendix A contains proofs of theorems, and Appendix B supplements the detail of the empirical tests.

2 TGFSR generators

2.1 GFSR and TGFSR algorithms

As a preliminary step, we describe the GFSR algorithm based on a primitive trinomial $t^n + t^m + 1$. Let $x[n]$ be an array of n integers of word size and l be an integer variable. The GFSR algorithm is described as follows:

```
Step 1. $l \leftarrow 0$
Step 2. Set $x[0], x[1], \ldots, x[n-1]$ to some suitable initial values.
Step 3. Output $x[l]$.
Step 4. $x[l] \leftarrow x[(l+m) \bmod n] \oplus x[l]$
Step 5. $l \leftarrow (l+1) \bmod n$
Step 6. Goto Step 3.
```

The TGFSR algorithm is obtained by replacing Step 4 with

Step 4′. $x[l] \leftarrow x[(l+m) \bmod n] \oplus \text{shiftright}(x[l]) \oplus \begin{cases} 0 & \text{if LSB of } x[l]{=}0 \\ \mathbf{a} & \text{if LSB of } x[l]{=}1, \end{cases}$

where $\mathbf{a}$ is a fixed constant word and shift means the one-bit shift operation. This apparently *adhoc* and minor but essential change leads to great improvements, as shown in Section 2.3. Some of the suitable parameters are listed in Table 1 (see Section 4.1).

2.2 Theory of TGFSR generators

This section provides a definition of the TGFSR sequence and some algebraic theory. For readability, proofs of theorems will be postponed to Appendix A. Let $\mathrm{GF}(2^k)$ be the 2^k-element field. Any algebraic operation is regarded as being in the field $\mathrm{GF}(2^k)$ unless stated otherwise. Other algebraic terminology follows [13].

Hereafter, a word $\mathbf{x}$ is regarded as a row vector $\mathbf{x} \in \mathrm{GF}(2)^w$ with word size w.

Definition. Let n, m, and w be positive integers with $n > m$, let A be a $w \times w$ matrix over GF(2), and let $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n-1}$ be vectors in $\mathrm{GF}(2)^w$. The *TGFSR sequence* is the sequence $\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots$ of $\mathrm{GF}(2)^w$ defined by the linear recurring equation

$$\mathbf{x}_{l+n} = \mathbf{x}_{l+m} + \mathbf{x}_l A$$

for $l = 0, 1, 2, \cdots$. The ordered-tuple $(\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{n-1})$ is called the *initial value.* At least one of these vectors is assumed to be nonzero. The generated sequence is denoted by $\mathcal{X}(n, m, A) = \mathbf{x}_0, \mathbf{x}_1, \cdots$.

The next theorem provides the definition of *maximal* TGFSR sequences.

Theorem 1. Let $\varphi_A(t)$ be the characteristic polynomial of the matrix A. The period of $\mathcal{X}(n, m, A)$ is $2^{nw} - 1$ if and only if $\varphi_A(t^n + t^m)$ is a primitive polynomial of degree nw. In this case, each bit of the $\mathcal{X}(n, m, A)$ constitutes an m-sequence based on $\varphi_A(t^n + t^m)$. We call such $\mathcal{X}(n, m, A)$ a *maximal TGFSR sequence* (m-TGFSR sequence for short).

The m-TGFSR sequence can be regarded as producing an m-sequence over $\mathrm{GF}(2^w)$ as follows.

Theorem 2. Let η be a root of a polynomial $\varphi_A(t)$ of degree w with coefficients in GF(2). Then, $\varphi_A(t^n + t^m)$ is primitive if and only if the following conditions hold.

(i) The polynomial $\varphi_A(t)$ is irreducible.

(ii) The polynomial $t^n + t^m + \eta$ is primitive over $\mathrm{GF}(2^w)$.

In this case, there is a GF(2) linear isomorphism between $\mathcal{X}(n, m, A)$ and an m-sequence over $\mathrm{GF}(2^w)$ based on $t^n + t^m + \eta$.

Corollary. An m-TGFSR sequence $\mathcal{X}(n, m, A)$ is n-equidistributed. Any initial value except for zero provides the identical sequence disregarding the phase.

When we find a tuple (n, m, A) with $\varphi_A(t^n + t^m)$ primitive, the implementation of the algorithm is quite easy. It is sufficient to replace Step 4 of the original algorithm with
Step 4″. $x[l] \leftarrow x[(l + m) \bmod n] \oplus x[l]A$.

The matrix A should be chosen so that $x[l]A$ can be calculated quickly. For example, one may choose A of the form

$$\begin{pmatrix} & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \\ a_0 & a_1 & \cdots & \cdots & a_{w-1} \end{pmatrix}.$$

It is wellknown that $\varphi_A(t) = t^w + \sum_{i=0}^{w-1} a_i t^i$(see, for example, [7]). Let $\mathbf{x}$ be a register with content $(x_0, x_1, \ldots, x_{w-1})$, $x_i = 0$ or 1. Then, $\mathbf{x} \leftarrow \mathbf{x}A$ can be executed by a statement

$$\begin{array}{lll} \text{if} \quad x_{w-1} = 0 & \text{then} & \mathbf{x} \leftarrow \text{shiftright}(\mathbf{x}), \\ & \text{else} & \mathbf{x} \leftarrow \text{shiftright}(\mathbf{x}) \oplus \mathbf{a}, \end{array}$$

where $\mathbf{a} = (a_0, a_1, \ldots, a_{w-1})$. Now we obtain Step 4′ in Section 2.1. This restriction on the form of A does not affect the generality in the following sense. In view of Theorem 2, an m-TGFSR sequence is determined uniquely by (n, m, φ_A) up to GF(2) linear isomorphism. Hence any m-TGFSR sequence is essentially isomorphic to this type of TGFSR sequence.

From now on, we consider only matrices A of the above type, and denote $\mathcal{X}(m, n, \mathbf{a})$ instead of $\mathcal{X}(m, n, A)$. Thus, we will not mention the matrix A but consider only φ or $\mathbf{a}$.

2.3 Merits of TGFSR generators

In this section we investigate how the TGFSR algorithm turns drawbacks (d1)–(d4) of the GFSR algorithm into merits (Tm1)–(Tm5) without loss of the original merits (m1)–(m3), as stated in the Introduction.

For (m1), the number of memory references, which has the greatest effect on the speed, is unchanged. Only one bit operation, one conditional operation, and one exclusive-or operation are required to be added. Thus, the operation $\mathbf{x} \leftarrow \mathbf{x}A$ can be implemented with a few machine instructions using an assembler language. One can also write an efficient and portable code in any language possessing bitwise exclusive-or operation.

The merit (m2) is virtually preserved. In fact, in this paper we list some useful and practical examples of sufficiently long periods. The merit (m3) is partially affected in the following sense. The algorithm itself is valid for any word size, but one must find an irreducible polynomial $\varphi(t)$ whose degree is near to the word size, and find numbers n and m such that $\varphi(t^n + t^m)$ is primitive. This paper contains a list of such $\varphi(t)$, n and m which provides enough examples for the implementation.

Now we investigate how the drawbacks of the GFSR algorithm stated in the Introduction are solved. In view of the Corollary of Theorem 2, drawback (d1) is transformed into merits (Tm1) and (Tm5). From Theorem 1, (d3) and (d4) become (Tm3) and (Tm4). Drawback (d2) becomes (Tm2) as follows. Since the number of terms of $\varphi(t)$ does not influence the speed of generation, we may choose $\varphi(t)$ of arbitrarily many terms. Primitive polynomials of the form $\varphi(t^n + t^m)$ in general have many terms. Lindholm[14] showed that if φ divides a trinomial of low degree, then the distribution of 0 and 1 in the m-sequence does not conform to the theoretical binomial distribution because the third moment assumes a large negative value. Strictly speaking, there is no assurance that $\varphi(t^n + t^m)$ will not divide a trinomial of relatively low degree. However, empirical tests on the third moment, stated in Section 4, show good randomness of the TGFSR sequences. *Pentanomial* versions of GFSR also seem to be effective (see Section 4; *cf* [15]), but there is an objection as follows. A. Compagner[3][4] introduced the concept of the *hierarchy of correlation coefficients*, which measures the randomness of a binary sequence from the theoretical and philosophical point of view. According to his observation, the number of terms of the characteristic polynomial should be at least the square root of the degree. Parameters of TGFSR generators satisfying this condition are easily found. (In fact, all parameters we could find by far satisfy this.)

The quality and performance of TGFSR are comparable to those of ADD–WITH–CARRY (AWC) and SUBTRACT–WITH–BORROW (SWB) algorithms recently proposed by Marsaglia and Zaman[17]. The period of AWC (or SWB) is nearly maximal with respect to the memory area, hence these generators have similar properties to (Tm3)–(Tm5). Strictly saying, the period of TGFSR generators is nearer to the maximal, hence TGFSR is a bit superior to AWC (SWB) from the view point of the easy initialization and the uniformity, but there seems to be no significant difference. As for the generation speed, there would be no great difference. From the view point of portability, AWC (SWB) is slightly superior to TGFSR, since the former does not use exclusive-or.

Marsaglia rejected some of GFSR generators through his stringent tests[16]. He compared GFSR with the other generators (such as additive generators) so that they have about the same size of working area, and as a consequence, those GFSR have too short periods. (Of course this is one of the fair ways of comparison.) This seems to be the main reason why those GFSR showed far poorer randomness than the others.

TGFSR generators will not suffer from such way of comparison due to (Tm3).

3 How to find n, m, and $\varphi(t)$

We want to find n, m, and $\varphi(t)$ so that the degree of φ is w and $\varphi(t^n + t^m)$ is primitive. First, designate w near the word size, and n as moderately large. Note that n is the order of equidistribution.

Generate polynomials $\varphi(t)$ of degree w, and check their irreducibility. Next, for $m = 1$ to $n-1$, calculate $\varphi(t^n + t^m)$ and test its primitivity. The test on irreducibility and primitivity is the same as that described in [20]. For the primitivity test, the complete factorization of $2^{nw} - 1$ is required. This paper utilizes the list by Brillhart *et al.*[2]. For large nw, namely $nw > 700$, $2^{nw} - 1$ is often not factorized. In fact, whether the factorization of $2^{nw} - 1$ is known or not limits n, the order of equidistribution. This fact indicates the importance of extending such mathematical tables.

On searching for such n, m, and φ, the authors discovered an important phenomenon: if $n \equiv \pm 3 \bmod 8$ and w is odd, the corresponding m and φ are rarely found. This phenomenon resembles the case of usual trinomials $t^n + t^m + 1$, where Swan's Theorem solves this question [1]. Part of the argument in [1] holds for $t^n + t^m + \eta$, as shown in the following proposition.

Proposition 1. Let η be an element of $\mathrm{GF}(2^w)$. The polynomial $t^n + t^m + \eta$ over $\mathrm{GF}(2^w)$ has an even number of irreducible factors, and hence is not irreducible if one of the following holds:

1. $n \equiv \pm 3 \bmod 8$, w:odd, m:even, and $m \not| 2n$.

2. $n \equiv \pm 3 \bmod 8$, w:odd, m:odd, and $n - m \not| 2n$.

3. n:even, m:even.

4. n:even, $n \neq 2m$, and $nm \equiv 0, 2 \bmod 8$.

5. n:even, $n \neq 2m$, $nm \equiv -2, 4 \bmod 8$, and w is even.

Since the irreducibility of $t^n + t^m + \eta$ is a necessary condition for $\mathcal{X}(n, m, \varphi)$ to be maximal, where η is a root of φ, this proposition serves as a powerful sieve in searching for n and m. For example, suppose that both w and n are even. Then, by 4 and 5 above, $\mathcal{X}(n, m, \varphi)$ can never be an m-TGFSR sequence unless $n = 2m$.

4 Statistical Tests

4.1 Generators

In this section, several practical TGFSR generators and the results of intensive statistical tests are given.

In the upper part of Table 1, the authors propose five m-TGFSR generators named T400, T403, T775, T800 and T1600 (The initial T denotes "twisted") with the values of w, n, m and $\mathbf{a} = (a_0, a_1, ..., a_{w-1})$ in hexadecimal form for which $\varphi(t) = t^w + \sum_{i=0}^{w-1} a_i t^i$. The first one is designed for 16-bit machines, the middle three are for 32-bit machines, and the last one is for 64-bit machines. These generators were found by the method described in Section 3. The other entries in Table 1 are generators for use as comparative references for the statistical test. The L521 is the original GFSR generator based on the trinomial m-sequence[12]. The F521 and G607 are also GFSR generators of which initialization algorithms have been proposed by Fushimi[6] and Tootill[19], respectively. Thus, both of these are carefully designed to have higher order equidistribution, differently from those tested by Marsaglia[16].

The PF89 and PF521 are *penta*-nomial versions of GFSR generators using Fushimi's initialization algorithm. These two primitive pentanomials on GF(2) are obtained by the authors[10]. The last generator named LM is Lehmer's congruential type. The multiplier of the LM was selected after a computer search with primitive root tests, spectral tests of dimensions 2 through 6 and serial correlation tests[9]. Our policy is to choose the known best parameters in each algorithm (except for L521).

4.2 Triple Kolmogorov-Smirnov test

The triple Kolmogorov-Smirnov (KS for short) test is the overall test for uniformity of random numbers in the (0,1) interval. The *double* KS test which is suggested by Knuth[8, p.49] means that (i) several KS tests for moderately large N to detect local behavior, (ii) combining r observations of (i) in another KS test to detect global behavior. To detect overall (mean) behavior in the entire period, the *triple* test adds furthermore, (iii) t repetitions of the procedures (i) and (ii), where t subsequences of length Nr are randomly sampled. This triple KS test is performed with the following procedures.

```
{
Set parameters of the test; $N$(length of a sample), $r$(number of samples), and $t$(number of tests).
$for(\tau = 1;\ \tau \leq t;\ \tau++)$ {
    Generate a sequence $(x_0, x_1, ..)$, using randomly chosen initial seeds $\pi_\tau$ (see Appendix B-1).
    $for(\rho = 1;\ \rho \leq r;\ \rho++)$ {
        Generate the subsequence of length $N$; $(x_{(\rho-1)N},\ x_{(\rho-1)N+1}, ...,\ x_{\rho N-1})$ as the $\rho$-th sample.
        Consider these $N$ values as $N$ realizations of random variables $X$, thereby obtaining the values $X_1,\ X_2, .., X_N$.
        Form the empirical distribution function of $X$ by up-sorting these $N$ values.
        Measure the difference between this empirical and the expected distribution function of $X$; $F_0(x) = x$, using KS statistics $K_N^+$ and $K_N^-$ defined as follows:

            $$K_N^+ = \sqrt{N} \max_{1 \leq j \leq N} (\frac{j}{N} - F_0(x_j)),$$

            $$K_N^- = \sqrt{N} \max_{1 \leq j \leq N} (F_0(x_j) - \frac{j-1}{N}).$$

        Denote these two statistics by $(K_N^+)_\rho$ and $(K_N^-)_\rho$ .
        }
    Form the empirical distribution function by up-sorting these $r$ statistics $(K_N^+)_\rho$, $\rho = 1, 2, .., r$.
    Measure the difference between the empirical and the expected distribution function: $G_0(x; r)$, using KS statistics, where $G_0(x; r)$ is the KS probability distribution function, (see Appendix B-2).
    Denote these two statistics by $(K_r^{++})_\tau$ and $(K_r^{-+})_\tau$.
    Similarly, obtain two statistics $(K_r^{+-})_\tau$ and $(K_r^{--})_\tau$ using $(K_N^-)_\rho$, $\rho = 1, 2, .., r$.
    }
Form the empirical distribution function by up-sorting these $t$ statistics $(K_r^{++})_\tau, \tau = 1, 2, .., t$.
Measure the difference between the empirical and the expected distribution function: $G_0(x; t)$, using KS statistics.
Denote these two statistics by $K_t^{+++}$ and $K_t^{-++}$.
Similarly, obtain three pairs of statistics $(K_t^{+-+},\ K_t^{--+})$, $(K_t^{++-},\ K_t^{-+-})$ and $(K_t^{+--},\ K_t^{---})$ using $(K_r^{-+})_\tau$, $(K_r^{+-})_\tau$ and $(K_r^{--})_\tau$, respectively, where $\tau = 1, 2, .., t$.
As the final result, convert these eight KS statistics into the probability value $K^{ijk}$, $i, j$ and $k = \pm$, using $G_0(x; t)$.
}
```

Table 1 lists one of the results when $N = 2048$, $r = 512$, $t = 64$. The column $K_t^{\pm\ \pm\ \pm}$ indicates the corresponding percentage value. The column 5% indicates the number of KS values with probability $\geq$ 95% or $\leq$ 5% of these eight values. The number in the parenthesis corresponds to that with probability $\geq$ 99% or $\leq$ 1%. According to this result, L521 is rejected.

4.3 Run test

The run test is the test for independency among subsequent elements of random numbers in the (0,1) interval, by means of examining the lengths of subsequences that are increasing or decreasing. The basic algorithm adopted here closely follows Knuth[8, pp.65-68]. Since this algorithm is rather complicated, complete rewriting is omitted here. This test is performed with the following procedures.

```
{
Set parameters of the test; $N$(length of a sample), $r$(number of samples), $t$(number of tests), and $k$(rounded length of up and down observations).
$for(\tau = 1;\ \tau \leq t;\ \tau{+}{+})$ {
    Generate sequence $(x_0,\ x_1, ..)$, using randomly chosen initial seeds $\pi_\tau$ (see Appendix B-1).
    $for(\rho = 1;\ \rho \leq r;\ \rho{+}{+})$ {
        Generate the subsequence of length $N$; $(x_{(\rho-1)N},\ x_{(\rho-1)N+1}, ...,\ x_{\rho N-1})$ as the $\rho$-th sample.
        Store the number of up sequences of length $r(1 \leq r < k)$ in $u(r)$ and length $\geq k$ in $u(k)$
        .
        Calculate the following statistics $V$ according to the empirical frequency distribution.

            $$V = \frac{1}{N} \sum_{1 \leq p,q \leq k} (u(p) - Nb_p)(u(q) - Nb_q) \cdot C^{-1}_{pq},$$

        where $b_j = \mathrm{mean}(R_j)$ , $R_j$ is the number of up runs of length $j$ exactly when all permutations of $N$ elements are given, $j = p, q$; $C_{pq}$ is the $(p, q)$ element of the covariance matrix $\mathrm{cov}(R_p, R_q)$.
        Do the same for the down sequences.
        Denote these two statistics by $(V_{up})_\rho$ and $(V_{dn})_\rho$ .
        }
    Form the empirical distribution function with these $r$ statistics $(V_{up})_\rho$, $\rho = 1, 2, .., r$.
    Measure the difference between $G_r(x)$ and the expected distribution function: ($\chi^2-$ distribution with $d.f. = k + 1$), using KS statistics,
    Denote these two statistics by $(K^+_{r,up})_\tau$ and $(K^-_{r,up})_\tau$.
    Similarly, obtain two statistics $(K^+_{r,dn})_\tau$ and $(K^-_{r,dn})_\tau$ using $(V_{dn})_\rho$, $\rho = 1, 2, .., r$.
    }
Form the empirical distribution function with these $t$ statistics $(K^+_{r,up})_\tau, \tau = 1, 2, .., t$.
Measure the difference between this empirical and the expected distribution function $G_0(x; t)$, using KS statistics (see Appendix B-2).
Denote these two statistics by $K^{++}_{t,up}$ and $K^{-+}_{t,up}$.
Similarly, obtain three pairs of statistics $(K^{+-}_{t,up},\ K^{--}_{t,up})$, $(K^{++}_{t,dn},\ K^{-+}_{t,dn})$ and $(K^{+-}_{t,dn},\ K^{--}_{t,dn})$ using $(K^-_{r,up})_\tau$, $(K^+_{r,dn})_\tau$, and $(K^-_{r,dn})_\tau$, respectively, where $\tau = 1, 2, .., t$.
As the final result, convert these eight KS statistics into probability value $K^{ij}_l$, $i, j = \pm$, $l = up,\ dn$ using $G_0(x; t)$.
}
```

Table 1 lists one of the results when $N = 65536$, $r = 128$, $t = 64$ and $k = 6$. The meaning of the columns is the same as that for the KS-test. This indicates that L521 must be rejected.

4.4 Weight distribution test

This test was originally designed for independency of the subsequent bits of a binary random sequence. Here, this test is applied to the most significant bit of uniform random numbers in the (0,1) interval with the following procedures.

```
{
Set parameters of the test; $N$(length of a sample), $r$(number of samples), and $t$(number of tests).
$for(\tau = 1;\ \tau \le t;\ \tau{++})$ {
    Generate a sequence $(x_0,\ x_1, ..)$, using randomly chosen initial seeds $\pi_\tau$ (see Appendix B-1).
    $for(\rho = 1;\ \rho \le r;\ \rho{++})$ {
        Generate the $N$-tuple as the $\rho$-th sample; $(x_{(\rho-1)N},\ x_{(\rho-1)N+1}, ...,\ x_{\rho N-1})$.
        Calculate the weight of the $N$-tuple, where the weight is the number of $x_i$ such that
        $x_i \ge 1/2$.
        Denote this weight by $w_\rho$.
        }
    Form the empirical distribution function with these $r$ statistics $w_\rho$, $\rho = 1, 2, .., r$.
    Consider the weight as random variable $X$. (Then, $X$ is expected to conform with the binomial
    distribution: $prob(X = k) = \binom{N}{k}(1/2)^N$).
    Measure the difference between the empirical and the expected distribution function using $\chi^2$
    statistics.
    Denote such a probability value by $W_\tau$.
    Calculate furthermore, the third and fifth moments of this empirical distribution, denoted by
    $(M_3)_\tau$ and $(M_5)_\tau$, respectively.
    }
Form the empirical distribution function with $W_\tau$, $\tau = 1, 2, .., t$.
Measure the difference between this empirical distribution function and the expected one, using KS
statistics.
Denote these two statistics by $K_t^+$ and $K_t^-$.
As the final result, convert these two KS statistics, to the probability value $K^+$, $K^-$, using $G_0(x; t)$.
Calculate also, as the other final result, the average value of $(M_3)_\tau$ and $(M_5)_\tau$, where $\tau = 1, 2, .., t$.
Denote these values by $[M_3]$ and $[M_5]$, respectively.
}
```

Table 1 lists one of the results when $N = 1024$, $r = 8192$, $t = 64$. The GFSR generators based on trinomial m-sequences; namely L521, F521, and G607 are definitely rejected.

4.5 Results of test

The authors would like to stress that the extent of the capability of the statistical test, as for selecting generators according to their quality, is limited to the rejection of "definitely bad generators"; the test can neither positively choose the best generators nor determine the rank order of those falling between the two extremes.

From this viewpoint, the five m-TGFSR sequences passed all tests (more exactly, cannot be rejected). The L521 sequence is definitely rejected in all three kinds of tests. Also, the GFSR sequences generated by the F521 and G607 are definitely rejected by the WD-test. In particular, these three trinomial-based GFSR sequences indicate the highly skewed distribution through the WD-test ($[M_3] \ll 0$ and $[M_5] \ll 0$). The pentanomial-based GFSR sequences (PF521 and PF89) passed all tests. Also the congruential sequence LM passed all tests.

5 Conclusions

An m-sequence-based pseudorandom number generator twisted GFSR algorithm was proposed. This algorithm retains most of the merits of the original GFSR algorithm, and shows improvements in four area: (1) fast and carefree initialization, (2) higher order equidistribution, (3) the number of terms of characteristic polynomial, (4) memory efficiency. Some practical TGFSR generators were implemented, and they passed the intensive Kolmogorov-Smirnov test, run test, and weight-distribution test. GFSR sequences with characteristic trinomials failed in the weight-distribution test. GFSR sequences with characteristic pentanomials passed all three tests.

Acknowledgement. The authors would like to express their sincere gratitude to Professor P. L'Ecuyer and Professor S. Tezuka for their invaluable comments. Thanks are also due to the anonymous referees for useful suggestions. The authors are indebted to Professor N. Yoneda who gave us a key to think of the twisted GFSR generator.

A Proofs of Theorems

A.1 Proof of Theorem 1

Let B be the $(nw \times nw)$-matrix

$$\begin{array}{r} \\ \\ m\text{-th block} \rightarrow \\ \\ 0\text{-th block} \rightarrow \end{array}\begin{pmatrix} & \mathrm{I}_w & & & \\ & & \mathrm{I}_w & & \\ \mathrm{I}_w & & & \ddots & \\ & & & & \mathrm{I}_w \\ A & & & & \end{pmatrix},$$

where I_w is the identity matrix of size w. It is clear that

$$(\mathbf{x}_{l+n}, \mathbf{x}_{l+n-1}, \ldots, \mathbf{x}_{l+1}) = (\mathbf{x}_{l+n-1}, \mathbf{x}_{l+n-2}, \ldots, \mathbf{x}_{l})B$$

holds for $l = 0, 1, \ldots$.

Thus, from the general theory of a matrix linear congruential generator [18], the period of each bit attains $2^{nw} - 1$ if and only if its characteristic polynomial $\varphi_B(t)$ of B is primitive. In this case, the nw-dimensional vector

$$(\mathbf{x}_{l+n-1}, \mathbf{x}_{l+n-2}, \ldots, \mathbf{x}_{l}) \quad (l = 0, 1, \ldots)$$

runs over all the 2^{nw} vectors except the zero vector. This implies n-equidistribution.

We now show that $\varphi_B(t) = \varphi_A(t^n + t^m)$. By blockwise column transformation, it is easy to see that

$$\begin{aligned} \varphi_B(t) &= \det \begin{pmatrix} t\mathrm{I}_w & \mathrm{I}_w & & & \\ & t\mathrm{I}_w & \mathrm{I}_w & & \\ \mathrm{I}_w & & \ddots & \ddots & \\ & & & t\mathrm{I}_w & \mathrm{I}_w \\ A & & & & t\mathrm{I}_w \end{pmatrix} \\ &= \det \begin{pmatrix} t\mathrm{I}_w & \mathrm{I}_w & & & \\ & & \mathrm{I}_w & & \\ \vdots & & & \ddots & \\ & & & & \mathrm{I}_w \\ A + (t^n + t^m)\mathrm{I}_w & & & & \end{pmatrix} \\ &= \det((t^n + t^m)\mathrm{I} - A). \end{aligned}$$

(q.e.d.)

A.2 Proof of Theorem 2

Assume that $\varphi_A(t)$ is irreducible, and let η be its root. Clearly $(1, \eta, \eta^2, \dots, \eta^{w-1})$ is a basis of $\mathrm{GF}(2^w)$ as a vector space over GF(2). Fix a nonzero vector $\mathbf{i} \in$GF$(2)^w$. The set $\{\phi(t) \in \mathrm{GF}(2)[t] \mid \mathbf{i}\phi(A) = 0\}$ is obviously an ideal of GF(2)$[t]$ containing $\varphi_A(t)$ and not containing 1. Since $\varphi_A(t)$ is irreducible, this ideal is $(\varphi_A(t))$, and consequently, $\mathbf{i}, \mathbf{i}A, \mathbf{i}A^2, \dots, \mathbf{i}A^{w-1}$ are linearly independent and form a basis of GF$(2)^w$. Thus, there is an isomorphism Φ between vector spaces

$$\begin{array}{rccc} \Phi : & \mathrm{GF}(2^w) & \to & \mathrm{GF}(2)^w \\ & \eta^l & \mapsto & \mathbf{i}A^l \end{array}$$

for $l = 0, 1, \dots, w-1$. It can easily be checked that $\Phi(\alpha\eta) = \Phi(\alpha)A$ holds for any $\alpha \in \mathrm{GF}(2^w)$ by rewriting α as a polynomial of η. The sequence $\mathcal{X}(n, m, A) = (\mathbf{x}_0, \mathbf{x}_1, \dots)$ is defined by the recurrence

$$\mathbf{x}_{l+n} = \mathbf{x}_{l+m} + \mathbf{x}_l A$$

for $l = 0, 1, 2 \dots$. By applying Φ^{-1} on both sides, we have

$$\Phi^{-1}(\mathbf{x}_{l+n}) = \Phi^{-1}(\mathbf{x}_{l+m}) + \Phi^{-1}(\mathbf{x}_l)\eta.$$

The sequence $\Phi^{-1}(\mathcal{X}) = (\Phi^{-1}(\mathbf{x}_0), \Phi^{-1}(\mathbf{x}_1), \dots)$ is thus a linear recurring sequence over GF(2^w) with characteristic polynomial $t^n + t^m + \eta$. Since the period of the sequence $\Phi^{-1}(\mathcal{X})$ is the same with $\mathcal{X}$, we find that $t^n + t^m + \eta$ is primitive if and only if the period of $\mathcal{X}$ is $(2^w)^n - 1$, namely, if and only if the TGFSR sequence is maximal. (q.e.d.)

A.3 Proof of Proposition 1

The proof of Proposition 1 is quite similar to the proof of Swan's Corollary 6.696 in [1]. Some propositions in [1] are used with the same numbering as in the book. Such propositions are marked with *.

Lemma 1. Let $f(t) = t^n + \sum_{i=0}^{n-1} f_i t^i$ be a polynomial with free variables f_i as coefficients. Suppose that its discriminant $D(f)$ satisfies $D(f) \equiv [\zeta(f)]^2 - 4\xi(f) \bmod 8$, where $D(f)$, $\zeta(f)$, $\xi(f)$ are represented as polynomials over the rational integers in the variables $f_0, f_1, \dots f_{n-1}$. Let $\widehat{f_0}, \dots, \widehat{f_{n-1}}$ be arbitrary elements in GF(2^w). For a polynomial g with variables $t, f_0, \dots f_{n-1}$, $\widehat{g}$ denotes the polynomial in GF$(2^w)[t]$ obtained by substituting $\widehat{f_i}$ for f_i in g. Let r be the number of irreducible factors of $\widehat{f}(t)$. Then, $\widehat{\xi(f)} = 0$ and $\widehat{\zeta(f)} \neq 0$ imply $r \equiv n \bmod 2$, and $[\widehat{\zeta(f)}]^2 = \widehat{\xi(f)} \neq 0$ implies $r \equiv n + w \bmod 2$.

Proof. Theorem 6.695* asserts that $\mathrm{Tr}[\widehat{\rho(f)}] = \mathrm{Tr}\{\widehat{\xi(f)}/[\widehat{\zeta(f)}]^2\}$, where Tr denotes the trace map to GF(2) and $\rho(f)$ is defined by

$$\rho(f) = \sum_{1 \le i < j \le n} \alpha_i \alpha_j / (\alpha_i + \alpha_j)^2$$

for α_i all roots of f with counting multiplicity. Since $\rho(f)$ is a symmetric expression of α_i, it can be represented as a polynomial with variables t, f_i. Theorem 6.69* proves that $r \equiv n \bmod 2$ if and only if $\mathrm{Tr}[\widehat{\rho(f)}] = 0$. Since Tr[1]$\equiv w \bmod 2$, we have the above consequence. (q. e. d.)

Swan's Theorem 6.67*.

$$D(t^n + at^m + b) = (-1)^{n(n-1)/2} b^{m-1} (n^N b^{N-M} - (-1)^N (n-m)^{N-M} m^M a^N)^d,$$

where $d = \gcd(n, m)$, $n = Nd$, and $m = Md$.

Proof of Proposition 1. Let f be $t^n + at^m + b$ with a, b free variables.

Case 1. n is odd.
Suppose m is even. If $M \geq 3$, then $m^M \equiv 0 \bmod 8$, and hence

$$\begin{aligned} D(f) &\equiv (-1)^{n(n-1)/2} b^{m-1} (n^N b^{N-M})^d \bmod 8 \\ &= (-1)^{(n-1)/2} n^n b^{n-1}. \end{aligned}$$

Note that if n is odd, $n^2 \equiv 1 \bmod 8$. Thus, for odd n, $n^n \equiv n \bmod 8$ holds.

If $n \equiv \pm 3 \bmod 8$,

$$D(f) \equiv (-1)^{(n-1)/2}(\pm 3) b^{n-1} = -3b^{n-1} = [b^{(n-1)/2}]^2 - 4b^{n-1}$$

holds, and Lemma 1 implies that $r \equiv n + w \equiv w + 1 \bmod 2$. Thus, if $n \equiv \pm 3 \bmod 8$, w is odd, m is even, and $M \geq 3$, then $t^n + t^m + \eta$ has an even number of irreducible factors. Since n is odd and m is even, M is even, and consequently, $M \geq 3$ is equivalent to $M \neq 2$; in other words, $m \not| 2n$.

Suppose that m is odd. If $N - M \geq 3$, then $(n-m)^{N-M} \equiv 0 \bmod 8$, and the above arguments hold again. In this case, $N - M \geq 3$ is equivalent to $n - m \not| 2n$.

Case 2. n is even.
If m is also even, $t^n + t^m + \eta$ is a square; hence we may assume that m is odd and d is odd. If $N \geq 3$, then $n^N \equiv 0 \bmod 8$, and consequently,

$$\begin{aligned} D(f) &\equiv (-1)^{n(n-1)/2} b^{m-1} (-(-1)^N (n-m)^{N-M} m^M a^N)^d \\ &\equiv (-1)^{n/2} b^{m-1} (-1)^d (n-m)^{n-m} m^m a^n \\ &\equiv -(-1)^{n/2} b^{m-1} (n-m) m a^n \\ &\equiv -(-1)^{nm/2} (nm-1) [a^{n/2} b^{(m-1)/2}]^2 \\ &\equiv \begin{cases} [a^{n/2} b^{(m-1)/2}]^2 & \text{if } nm \equiv 0, 2 \bmod 8 \\ -3[a^{n/2} b^{(m-1)/2}]^2 & \text{if } nm \equiv -2, 4 \bmod 8. \end{cases} \end{aligned}$$

Thus, again by Lemma 1, we find that if $nm \equiv 0, 2 \bmod 8$ then $r \equiv n \equiv 0 \bmod 2$, and that if $nm \equiv -2, 4 \bmod 8$ then $r \equiv n + w \equiv w \bmod 2$. The condition $N \geq 3$ is equivalent to $n \neq 2m$, and this completes the proof of the proposition. (q. e. d.)

B Supplements to the statistical tests

B.1 Initial seeds generator

Initial seeds π_i are generated as follows:
Suppose that n words of k-bit integers are needed as initial seeds. Generate $2n$ random numbers by the recurrence $v_l = 2\,100\,005\,341\, v_{l-1} \bmod (2^{31} - 1)$ with $v_0 = 314\,159\,265$ $(l = 1, \ldots, 2n)$.
Obtain n words of 32-bit integers π_i $(i = 1, \ldots, n)$ by π_i =(1 bit logical shiftleft(v_{2i-1}))$\oplus$ (16 bit logical shiftright(v_{2i})), $i = 1, 2, \ldots, n$.
If $k < 32$, set $\pi_i = (k - 32)$-bit logical shiftright(π_i). If $k = 64$, generate $2n$ words of 32-bit integers as above, and concatenate them pairwise.

B.2 KS statistics

The exact KS distribution is obtained by

$$G_0(s;N) = prob(K_N^{\pm} \leq s) =$$
$$\frac{s}{\sqrt{N}} \sum_{0\leq k\leq\lfloor\sqrt{N}s\rfloor} (-1)^k \binom{N}{k} (\frac{\sqrt{N}s-k}{N})^k (1+\frac{\sqrt{N}s-k}{N})^{N-k-1}. \quad (1)$$

For a large value of N, this can be approximated (see [8, p.48]) by

$$prob(K_N^{\pm} \leq s) = 1 - e^{-2(s+\frac{1}{6\sqrt{N}})^2}. \quad (2)$$

In this report, we calculate (1) directly when $N \leq 99$, and we use (2) when $N \geq 100$. The approximation error of this asymptotic formula (2) is less than 10^{-3} at $N \approx 100$.

References

[1] Berlekamp, E. R. *Algebraic Coding Theory.* McGraw-Hill Book Co., New York, 1968.

[2] Brillhart, J. et al. *Factorizations of* $b^n \pm 1$, *Contemporary Mathematics Vol. 22*(2nd ed.), AMS, Providence, 1988.

[3] Compagner, A. The hierarchy of correlations in random binary sequences. *J. of Stat. Phys.* 63(1991), 883–896.

[4] Compagner, A. Definitions of randomness. *Am. J. Phys.* 59(1991), 700–705.

[5] Fushimi, M., and Tezuka, S. The k-distribution of generalized feedback shift register pseudorandom numbers. *Commun. ACM* 26(1983), 516–523.

[6] Fushimi, M. *Ransuu.* (in Japanese) Applied Mathematics Series Vol.12, Tokyo University Press, Tokyo, 1989.

[7] Golomb, S. W. *Shift Register Sequences.* Holden-Day, San Francisco, 1967.

[8] Knuth, D. E. *The Art of Computer Programming*, Vol 2: *Seminumerical Algorithms*, 2nd ed. Addison-Wesley, Reading, Mass., 1981.

[9] Kurita, Y. Choosing parameters for congruential random numbers generators. *Recent development in statistics.* Barra, J.R., Ed. North-Holland Publishing Company, 1977

[10] Kurita, Y. and Matsumoto, M. Primitive t-nomials ($t = 3, 5$) over GF(2) whose degree is a Mersenne exponent ≤ 44497. *Math. Comp.* 56 (1991), 817–821.

[11] L'Ecuyer, P. Random numbers for simulation. *Commun. ACM*, 33, 10 (1990), 85–97.

[12] Lewis, T. G., and Payne, W. H. Generalized feedback shift register pseudorandom number algorithms. *J. ACM* 20, 3(July 1973), 456–468.

[13] Lidl, R., and Niederreiter, H. *Finite Fields.* Addison-Wesley, Reading, Mass., 1983.

[14] Lindholm, J. H. An analysis of the pseudo-randomness properties of subsequences of long m-sequences. *IEEE Trans. Inform. Theory,* IT-14(July 1968), 569–576.

[15] Matsumoto, M. and Kurita, Y. The fixed point of an m-sequence and local nonrandomness. *Tech. Rep. of Dep. of Inf. Sci. Univ. Tokyo* **88–027**(1988).

[16] Marsaglia, G. A current view of random numbers. in *Computer Science and Statistics: The Interface.* L. Billard (ed.) Elsevier, 1985.

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

[18] Niederreiter, H. Random Number Generation and Quasi-Monte Carlo Methods. SIAM, Philadelphia, 1992.

[19] Tootill, J. P. R., Robinson, W. D., and Eagle, D. J. An asymptotically random Tausworthe sequence. *J. ACM* 20, 3(July 1973), 469–481.

[20] Zierler, N. and Brillhart, J. On Primitive Trinomials (Mod 2). *Information and Control* **13**, (1968) 541–554.

Generator		KS test $N = 2048,\ r = 512,\ t = 64$					RUN test $N = 65536, r = 128, t = 64, k = 6$					WD test $N = 1024,\ r = 8192,\ t = 64$				
ID	Paramaters (p means period)	K_t^{+++} K_t^{++-}	K_t^{-++} K_t^{-+-}	K_t^{+-+} K_t^{+--}	K_t^{--+} K_t^{---}	5% (1%)	K_{up}^{++} K_{dn}^{++}	K_{up}^{+-} K_{dn}^{+-}	K_{up}^{-+} K_{dn}^{-+}	K_{up}^{--} K_{dn}^{--}	5% (1%)	K^+	K^-	5% (1%)	$[M_3]$	$[M_5]$ $\times 10^{-3}$
T400	$(w, n, m) = (16, 25, 11),\ p = 2^{400} - 1$ $\mathbf{a} = $ A875	43.5 86.4	19.3 5.0	68.1 70.2	33.8 71.0	0(0)	90.8 90.4	45.9 40.7	13.2 38.9	96.9 86.4	1(0)	47.1	73.9	0(0)	−13	−31
T403	$(w, n, m) = (31, 13, 2),\ p = 2^{403} - 1$ $\mathbf{a} = $ 6B5E CCF6	82.7 32.3	40.5 41.7	43.9 8.9	82.6 71.0	0(0)	79.5 42.7	34.3 12.1	3.2 45.6	95.2 71.0	1(0)	81.1	19.8	0(0)	−37	−101
T775	$(w, n, m) = (31, 25, 8),\ p = 2^{775} - 1$ $\mathbf{a} =$6C6C B38C	93.2 18.5	7.5 87.0	35.2 43.0	91.7 15.5	0(0)	33.9 8.5	69.2 54.9	68.4 57.0	39.5 44.0	0(0)	47.7	51.8	0(0)	13	8
T800	$(w, n, m) = (32, 25, 7),\ p = 2^{800} - 1$ $\mathbf{a} =$8EBF D028	12.9 25.5	87.9 52.2	75.7 34.0	35.1 94.8	0(0)	29.3 53.7	95.4 22.3	0.7 20.8	74.6 83.6	2(1)	17.7	75.9	0(0)	−2	−6
T1600	$(w, n, m) = (64, 25, 3),\ p = 2^{1600} - 1$ $\mathbf{a} =$B380 C13A A838 387E	37.2 32.7	57.0 21.1	58.6 66.1	74.3 34.9	0(0)	54.6 82.7	33.4 14.5	34.0 10.3	39.3 72.0	0(0)	71.8	11.1	0(0)	−3	29
L521	$X^{521} + X^{158} + 1,\ p = 2^{521} - 1$ original GFSR[**]	84.5 99.0	76.1 14.7	1.4 43.3	100.0 100.0	4(3)	0 0	100.0 100.0	100.0 100.0	0 0	8(8)	100.0	0	2(2)	−416	−1140
F521	$X^{521} + X^{32} + 1,\ p = 2^{521} - 1$ Fushimi[**]	77.1 37.7	45.7 73.7	48.2 73.5	48.2 66.8	0(0)	88.6 95.3	11.7 2.9	1.6 13.6	85.9 87.3	3(0)	100.0	0.3	2(2)	−373	−927
G607	$X^{607} + X^{273} + 1,\ p = 2^{607} - 1$ Tootill[**]	39.3 75.9	84.4 56.0	48.3 15.4	24.7 96.7	1(0)	49.8 25.9	17.9 97.6	44.1 94.0	79.1 4.7	2(0)	100.0	1.7	2(1)	−338	−840
PF89	$X^{89} + X^{72} + X^{53} + X^{17} + 1$[**] $p = 2^{89} - 1$	34.4 34.0	30.0 50.1	49.0 17.6	75.5 32.0	0(0)	39.8 71.2	37.1 12.0	8.8 7.3	89.5 89.7	0(0)	96.3	11.6	1(0)	−25	−71
PF521	$X^{521} + X^{424} + X^{236} + X^{111} + 1$[**] $p = 2^{521} - 1$	72.9 71.2	80.0 2.0	86.8 5.3	16.5 66.3	1(0)	3.2 71.1	89.8 58.9	76.3 67.7	9.0 63.0	0(0)	31.2	45.9	0(0)	28	65
LM	$X_i = a \cdot X_{i-1} \bmod M,\ M = 2^{31} - 1$ $a = 2\ 100\ 005\ 341$[**], $p = 2^{31} - 1$	67.6 26.9	34.1 68.5	38.7 9.6	35.9 83.7	0(0)	79.7 54.5	11.7 73.1	42.9 50.3	37.9 19.5	0(0)	89.6	14.6	0(0)	21	53

Table 1. Five TGFSR and six other type generators and their results of statistical tests