

Super Mersenne プロジェクト

日本 IBM 東京基礎研究所 手塚 集 (Shu Tezuka)

IBM Tokyo Research Laboratory

1 擬似乱数の生成 — Mersenne Twister

単位区間 $[0, 1)$ 内に一様に分布する乱数を一様乱数とよんでいる。実際の応用では種々の分布に従う乱数が必要となるが、ほとんどの場合一様乱数にある種の変換を施すことにより生成されているので、一様乱数の生成がとりわけ重要になる。シフトレジスター法と呼ばれる一様乱数生成法（以下、GFSR (Generalized Feedback Shift Register) 乱数とよぶことにする）がある。それは $c_1, \dots, c_{p-1}$ を 0 または 1 とした時の線形漸化式

$$a_{i+p} = c_{p-1}a_{i+p-1} + \dots + c_1a_{i+1} + a_i \pmod{2}$$

により生成される 2 進数列 $a_i (i = 1, 2, \dots)$ を使うものである。ここで、特性多項式

$$f(x) = x^p + c_{p-1}x^{p-1} + \dots + c_1x + 1$$

が $GF(2)$ 上の原始多項式の時には 2 進数列は最大周期 $2^p - 1$ をもつことが知られている。この 2 進最大周期列は「M 系列」ともよばれ、現在では、カーナビ、無線 LAN など私たちの日常生活の様々な場面で使われている。

GFSR 乱数は、この M 系列を用いて一様乱数を

$$u_i = 0.a_{j_1+i}a_{j_2+i} \dots a_{j_L+i}$$

という風に生成するものである。ここで、 L は計算機の語長であり、通常は 32 である。また $j_1, \dots, j_L$ はシフトレジスター乱数の位相パラメーターと呼ばれるもので、これが GFSR 乱数のよしあしを決定している。

GFSR 乱数では p が大きいほど周期も大きくなる。が当然、配列のサイズも増える。実際 $p = 19937$ では、単純な実装をしてしまうと配列が大きくなりすぎて問題が生じる。これを一気に解決して、 $\lceil 19937/32 \rceil = 624$ の単精度配列で十分としたのが MT で、次のような 4 つの大きな特徴

- 周期が巨大である
- 623 次元一様性が保証されている

- 効率的にメモリーを使用している
- 非常に高速に乱数が生成される

をもっている。Mersenne Twister は次のようなアルゴリズムである [2]。まず、

$$X_{i+n} = X_{i+m} \text{ XOR } (X_i^u | X_{i+1}^l) A, \quad i = 1, 2, 3, \dots \quad (1)$$

なる漸化式により最大周期 $2^{19937} - 1$ をもつ GFSR 乱数をつくる。ここで A は

$$A = \begin{pmatrix} & 1 & & \cdots & 0 \\ & & 1 & & \cdots & 0 \\ & & & & \cdots & \\ & 0 & & & \cdots & 1 \\ c_{L-1} & c_{L-2} & c_{L-3} & \cdots & c_0 \end{pmatrix}$$

の形をしたある特殊な行列であり、 $(X_i^u | X_{i+1}^l)$ の意味は X_i の上位 $L-r$ ビットと X_{i+1} の下位 r ビットをつないで得られる L ビットの行ベクトルを意味している。しかし、これだけだと $X_i, i = 1, 2, \dots$ にはまだ規則性が残るので、さらにある特殊な線形変換 T を施して得られる GFSR 乱数

$$u_i = T(X_i) \quad i = 1, 2, 3, \dots$$

が MT である。変換 T もシフトと XOR を組み合わせた簡単なものなので計算時間はほとんどかからない。また、 $p = 19937$ なので、乱数の周期は巨大である。Ferrenberg らがテストしたのが $p = 1000$ 程度なので、その巨大さがよくわかる。さらに、メモリーの使い方も 624 個の単精度の配列で済むようになっている。また、全周期でみると 623 次元空間に $L = 32$ ビットの精度で一様分布している。ただここで指摘しなければならないのは、式 (1) の数列 X_i はすでに、上に述べた MT の「4 つの大きな特徴」を満足していることだ。それなら X_i を使えばいいのにということになるが、実際には X_i は乱数としてはよくないことがわかっている。だからこそ変換 T を施したのである。そうすると、あの「4 つの大きな特徴」は「質のいい乱数」を保証していたわけではなかったのか？ ということになる。

実際、MT は、乱数 1 個当たり生成するスピードを高速化することに重点がおかれて設計されたので、「乱数の質」についてはあまり考慮されていない。大きな問題は、わずか 624 個の配列で生成できるようにしたことで、位相パラメータ $j_1, \dots, j_L$ がある特殊な値に固定されてしまったことである。MT が選べる初期値の自由度は先に述べた 2 つのうち第 2 番目の「一周期のなかのどこからスタートするかを変えられる」ことだけである。そして、この固定されてしまった $j_1, \dots, j_L$ が乱数の質を決めるのだが、これがただけでない。実際、表 1 に示すように MT の解像度は最大可能な解像度と比べるとかなり悪くなっているのである。非常に面白い事実として、「位相パラメータ $j_1, \dots, j_L$ をランダムに選ぶと「ほぼ漸近的にランダムな GFSR 乱数」が生成される」ことが知られている [3]。それ

表 1: 1248 次元までの MT の解像度

次元	1~623	624~643	644~664	665~687	688~712	713~738
最大解像度	32	31	30	29	28	27
MT の解像度	32	16	16	16	16	16
次元	739~766	767~797	798~830	831~866	867~906	907~949
最大解像度	26	25	24	23	22	21
MT の解像度	16	16	16	16	16	16
次元	950~996	997~1049	1050~1107	1108~1172	1173~1246	1247~1248
最大解像度	20	19	18	17	16	15
MT の解像度	16	16	16	16	16	11

なのに、このMTの位相パラメーターはわざわざ少数派の悪いところを選んで使っているということになってしまっているのだ。

モンテカルロシミュレーションでは、乱数生成とそれ以外の計算では、後者にかかる時間がはるかに大きいようなアプリケーションも多く存在する。たとえば金融工学などで用いられるモンテカルロシミュレーションでは全計算時間のなかで乱数生成のしめる割合はほんのわずかである。そのようなアプリケーションでは、生成速度に多少時間がかかっても全計算時間はたいして変わらないので、生成速度を短くするために「乱数の質」を犠牲にするのははなはだ疑問である。そういう意味で、MTは「画龍点睛を欠く」と言わざるを得ない。現在、同じ日本人の手でMTに「目玉」を入れるプロジェクトが進行している。そこでは、MTとほとんど同じ生成速度とメモリーサイズで「漸近的にランダム」なGFSR乱数がすでにいくつか見つかっている [1]。

参考文献

- [1] 原瀬 晋、慶応大学大学院理工学研究科修士論文、2004。
- [2] M. Matsumoto and T. Nishimura, Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator, *ACM Trans. Model. Comput. Simul.*, **8** (1998), 3-30.
- [3] S. Tezuka, A Heuristic Approach for Finding Asymptotically Random GFSR Generators, *Journal of Information Processing*, **10**, 3 (1987), 178-182.