数理解析研究所講究録
1032巻 1998年 109-117

近似的な漸近的ランダム性をもつ一様乱数発生法の設計

諸星 穂積　　伏見 正則
東京大学大学院工学系研究科

Designing a Uniform Random Number Generator with Approximate Asymptotic Rondomness

Hozumi Morohosi　　Masanori Fushimi
Graduate School of Engineering, University of Tokyo

Abstract A method for designing a uniform random number generator based on M-sequence is presented. The sequence generated by the method $\{x_t; t = 0, 1, 2 \ldots\}$ as well as its properly decimated sequence $\{x_{nt}; t = 0, 1, 2 \ldots\}$ for several values of n have the property of approximate asymptotic randomness. A key idea of the method is to iterate a permutation of the bits in M-sequence random numbers so that leading bits may become linearly independent. Since there are computational difficulties in finding the condition for the linear independence of bits and in solving the optimization problem of bit permutation, we adopt some heuristic methods for finding approximate solutions.

1　はじめに

擬似乱数列の生成法については，きわめて多くの研究と提案がされている．一方，乱数列とは何かという根源的な問題に対しても，いくつかの研究があり，乱数列の定義が与えられている．しかし，これらの2種類の研究の間には，ほとんど何の関係もなく，これが種々の擬似乱数生成法に対する評価が人によって異なるひとつの原因となっている．そこで，ここでは，これらの2種類の研究の間のギャップを少しでも縮めることをめざす試みを紹介する．

Kolmogorov[5] は，von Mises のコレクティフ（部分列抽出）の概念に基づいて，有限長の数列が乱数列と見なせるための条件と，そのような条件を満たす数列の存在について論じている．しかし，そのような数列の構成法については，何の議論もされていない．Kolmogorov が挙げている部分列抽出規則のいずれを適用しても，得られた部分列がもとの数列と同じ性質を持つような数列を構成することは至難の業であろう．そこで，ここでは最も単純ではあるが，応用上重要である抽出規則，すなわち等間隔の抽出のみを考えることにする．より詳しく言えば，一様擬似乱数列 $\{x_t; t = 0, 1, 2, \ldots\}$ に対してその n 番目ごとの等間隔抽出による部分列 $\{x_{nt}; t = 0, 1, 2, \ldots\}$ (n は正整数) が再び一様分布するように，もとの乱数列を設計したい．例えば，n 個の互いに独立な確率変数 $X_1, \ldots, X_n$ を扱うようなシミュレーションでは，擬似乱数列 $\{x_t\}$ の相続く n 個を各確率変数に割り当てて用いること

が多いであろう．この場合，$\{x_{nt}; t = 0, 1, 2, \cdots\}$ が一様乱数列となることが必要になる．様々な n についてそのような性質をもつ乱数を設計することは，実用上重要であると考えられる．

[2] では，このような等間隔の部分列抽出規則を考えた場合，分布の一様性に関して k 次均等分布の意味で良い乱数列を M 系列をもとにして設計する方法を提案している．一方，M 系列をもとにして生成する乱数列に関しては，[8] が漸近的ランダム性 (asymptotic randomness) という概念を提案している．ただし，そこでは部分列についての考慮はなされていない．

本論文では，M 系列を用いて構成する擬似乱数列に [2] の設計法を繰り返し適用することによって，部分列が近似的に漸近的ランダム性を有する一様乱数の発生アルゴリズムを設計する方法を述べ，実際の適用例を示す．以下，第 2 節で M 系列乱数に関する定義を述べ，第 3 節で漸近的ランダム性を実現するためのアルゴリズムを記述する．第 4 節でアルゴリズムに基づいた計算結果を紹介する．

2 M 系列に基づく一様乱数の発生

2.1 M 系列乱数の構成

M 系列は，ガロア体 GF(2) 上の原始多項式

$$f(x) = 1 + c_1 x + \cdots + c_p x^p, \quad c_p = 1, \tag{2.1}$$

を特性多項式とする漸化式

$$a_t = c_1 a_{t-1} + c_2 a_{t-2} + \cdots + c_p a_{t-p} \pmod 2 \tag{2.2}$$

（初期値 $a_1, \ldots, a_p$ はすべてが 0 の場合以外は何でもよい）によって定義される 0-1 系列でその周期は $2^p - 1$ である．M 系列をもとにして，2 進数で表された l ビットの数の系列を

$$x_t = 0.a_{\sigma t+1} a_{\sigma t+2} \cdots a_{\sigma t+l} \tag{2.3}$$

のように構成する方法が Tausworthe [7] によって提案された．ここで，パラメータ σ は $\sigma \geq l$ で周期 $2^p - 1$ と互いに素となる整数である．一方 [6] では，特性多項式として特に 3 項原始多項式 $f(x) = 1 + x^q + x^p$ を用いた上で，GFSR といわれる系列を

$$y_t = 0.a_t a_{t+\tau} a_{t+2\tau} \cdots a_{t+(l-1)\tau} \tag{2.4}$$

のように構成した．ここでパラメータ τ は適当に選んだ正整数である．GFSR の有利な点は，系列 $\{y_t\}$ がビット毎の排他的論理和 $\oplus$ によって $y_t = y_{t-q} \oplus y_{t-p}$ と計算できるため，系列の高速な発生が可能なことである．

これら 2 種類の系列は，一見するとまったく別のもののように思われるが，実は密接な関係がある．いまそれぞれの条件を幾分緩めて，$\{x_t\}$ については $\sigma \geq l$ なる条件をはずし，$\{y_t\}$ については $f(x)$ が 3 項とは限らず一般の原始多項式であるとする．また，それぞれの系列を，特性多項式とパラメータを明示する形で，$\{x_t(f;\sigma)\}$, $\{y_t(f;\tau)\}$ と書くことにする．このとき，2 種類の系列が位相のずれを除いて完全に一致するという意味の同値な関係を “$\simeq$” で表すことにすれば，以下の定理が成り立つ（例えば [3]）．

定理 1　p 次の原始多項式 f と，それぞれ 2^p-1 と互いに素な σ, τ について，次の同値関係が成り立つ．

$$\{x_t(f;\sigma)\} \simeq \{y_t(f_\sigma;\sigma^{-1})\},$$
$$\{y_t(f;\tau)\} \simeq \{x_t(f_\tau;\tau^{-1})\}.$$

ここで，σ^{-1} は 2^p-1 を法とする乗算における σ の逆元である．また，f_σ は f によって生成される M 系列から σ 番目ごとの等間隔抽出によって得られる系列（これも周期 2^p-1 の M 系列となる）の特性多項式である．f_τ についても同様である．

$\{x_t\}$ に対して，2^p-1 と互いに素な整数 n によって $\{x_{nt}\}$ を構成すれば，やはり，周期 2^p-1 の M 系列乱数になる．これを，系統サンプリングによる部分列と呼ぶことにする．

2.2　k 次元均等分布と漸近的ランダム性

本論文では，k 次均等分布 [4] の考え方によって乱数列の設計を行う．M 系列乱数は，l ビットの 2 進小数であり，かつ周期性をもつ．このことを考慮し，k 次均等分布を以下のように定義する．

定義 1　周期 2^p-1 の M 系列乱数 $\{x_t\}$ が，任意の l ビット 2 進小数の組 $w_1,\ldots,w_k$（ただしすべてが 0 の場合は除く）に対して，

$$\Pr\{x_t = w_1, x_{t+1} = w_2, \ldots, x_{t+k-1} = w_k\} = \frac{2^{p-kl}}{2^p-1} \tag{2.5}$$

を満たし，かつ

$$\Pr\{x_t = 0, x_{t+1} = 0, \ldots, x_{t+k-1} = 0\} = \frac{2^{p-kl}-1}{2^p-1} \tag{2.6}$$

が成り立つとき，$\{x_t\}$ は k 次均等分布をするという．

ここで，“確率” Pr は 1 周期についての相対頻度の意味とする．M 系列乱数が k 次均等分布をするための必要十分条件は，以下のようになる [3]．

定理 2　M 系列乱数 $\{x_t\}$ が k 次均等分布するための必要十分条件は，$x_t, x_{t+1}, \ldots, x_{t+k-1}$ の各ビットを構成する kl 個の M 系列の要素全体 $\{a_{\sigma(t+i)+j}\}$ $(i = 0,\ldots,k-1;\ j = 1,\ldots,l)$, が GF(2) 上で線形独立となることである．

ここで，M 系列の要素が線形独立であるという用語は次の意味で使う：任意の t に対して $a_{\sigma(t+i)+j}$ は必要ならば式 (2.2) を繰返し使うことで，$a_t,\ldots,a_{t+p-1}$ の線形結合で表せる．

$$a_{\sigma(t+i)+j} = \sum_{n=0}^{p-1} e_{jn}^{(i)} a_{t+n} \quad (i = 0,\ldots,k-1;\ j = 1,\ldots,l). \tag{2.7}$$

$e_{jn}^{(i)}$ を並べたベクトルの組

$$\boldsymbol{e}_j^{(i)} = (e_{j0}^{(i)}, \ldots, e_{j,p-1}^{(i)})^{\mathrm{T}} \quad (i = 0, \ldots, k-1;\ j = 1, \ldots, l) \tag{2.8}$$

が GF(2) 上で線形独立であるとき，kl 個の M 系列の要素は線形独立であるという．$\boldsymbol{e}_j^{(i)}$ は t とは無関係に決まるので，k 次均等分布という性質を考えるときは，t を適当に（例えば 0 に）固定して考えればよい．$\{x_t\}$ のみでなく，その系統サンプリングによる部分列 $\{x_{nt}\}$ についても，同様の定理が成り立つことに注意しておく．定理より，p 次の原始多項式によって生成される M 系列を用いて構成される l ビットの M 系列乱数の均等分布の最大次数 m について

$$m = \lfloor p/l \rfloor \tag{2.9}$$

が成り立つ．$\lfloor x \rfloor$ は x を超えない最大の整数を表す．ここで，l ビット中の上位 l' ビット $(1 \leq l' \leq l)$ に着目した場合，達成できる均等分布の最大次数は，$\lfloor p/l' \rfloor$ となる．l ビットの系列 $\{x_t\}$ が，$1 \leq l' \leq l$ の範囲の任意の l' について，上位 l' ビットに着目したとき $\lfloor p/l' \rfloor$ 次の均等分布をしているならば，漸近的ランダム性 (asymptotic randomness) を満たす系列であるという [8]．このような性質を満たす系列は，乱数列として望ましいものであると考えられる．次節以降で，M 系列乱数に対して，系統サンプリングによる部分列を考慮に入れた，漸近的ランダム性を近似的に有する系列の設計を考える．

3 漸近的ランダム性の実現

3.1 ビット置換による乱数の設計

定理 2 によれば，p 次の原始多項式によって生成される l ビットの M 系列乱数が，均等分布の最大次数 $m = \lfloor p/l \rfloor$ を達成するかどうかは，原始多項式と σ の値によって決まってしまう．しかし最下位ビットまで見た場合には最大次数が達成できない場合でも，上位 l' ビットだけを見れば，m 次均等分布をすることは可能でありうる．そして乱数を構成する各ビットの置換を行うことで，このような l' を大きくし，乱数列を改良できる可能性がある．すなわち，$\{\pi(1), \pi(2), \ldots, \pi(l)\}$ を $\{1, 2, \ldots, l\}$ の置換として，

$$x'_t = 0.a_{\sigma t+\pi(1)} a_{\sigma t+\pi(2)} \cdots a_{\sigma t+\pi(l)} \tag{3.10}$$

とする．置換 π を $\{a_{\sigma(t+i)+\pi(j)}\}$ $(i = 0, \ldots, m-1;\ j = 1, \ldots, l')$ が独立になるように決めれば $\{x'_t\}$ の上位 l' ビットは m 次均等分布をする．[2] では，この考えによって，与えられた原始多項式に対して，元の M 系列乱数とともに，系統サンプリングによる部分列 $\{x_{nt}\}$ についても同時に，m 次均等分布する上位ビット数 l' を最大にする置換を求めるという問題を扱っている．そのアルゴリズムは以下のようなものである．

1. 原始多項式 $f(x)$，パラメータ σ，乱数列のビット長 l，
   $N = \{$系統サンプリングを行う正整数 n の集合$\}$ を決める．
2. 均等分布の最大次数 $m = \lfloor p/l \rfloor$ を計算する．
3. 各 $n \in N$ に対して，$x_{n(t+i)} = 0.a_{\sigma n(t+i)+1} a_{\sigma n(t+i)+2} \cdots a_{\sigma n(t+i)+l}$ $(i = 0, \ldots, m-1)$ の各ビット，$a_{\sigma n(t+i)+j}$ $(i = 0, \ldots, m-1;\ j = 1, \ldots, l)$ を $a_t, \ldots, a_{t+p-1}$ で表したと

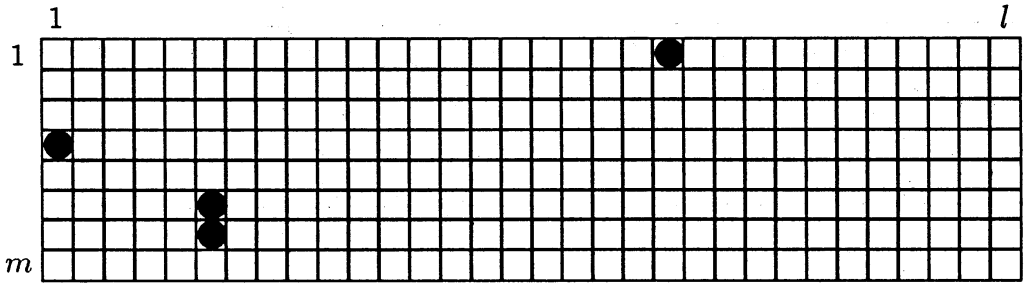

図 1. ビット間の従属関係（上図の • で示した位置の間に従属関係が存在する）と，対応する制約条件の行ベクトル

きの重みベクトル $\boldsymbol{e}_j^{(i)}$ (式 (2.7),(2.8) 参照) を求め，それらを列ベクトルとして並べて $p \times lm$ の行列 E_n を作成する．

4. 各 $n \in N$ に対し E_n の列ベクトルの中で極小な線形従属関係をすべて求め，$\mathcal{C}_n = \{C \mid C$は極小な線形従属関係をもつ列ベクトルの集合$\}$ とする．ここで，極小従属関係をもつ集合とは，その中からどの 1 要素を除いても 残りの要素が独立になってしまう集合のこととする．

5. $\mathcal{C}_n$ より，ビット間の従属関係を表す行列 G を次のように構成する：$C \in \mathcal{C}_n$ が $C = \{\boldsymbol{e}_{j_1}^{(i_1)}, \ldots, \boldsymbol{e}_{j_\nu}^{(i_\nu)}\}$ であったとする．これに対応して l 次元の 0-1 ベクトル $\boldsymbol{g} = (g_1, \ldots, g_l)$ を $g_{j_1} = \cdots = g_{j_\nu} = 1$, その他の成分を 0 として定義する．すべての $n \in N$ と $C \in \mathcal{C}_n$ に対して $\boldsymbol{g}$ をつくり，それらを行ベクトルとして並べて G とする（図 1 参照）．

6. 以下の 0-1 整数計画問題を解く．

$$\text{(3.11)} \qquad \text{ILP:} \quad \text{minimize} \qquad z_0 = \sum_{j=1}^{l} z_j$$

$$\text{(3.12)} \qquad \text{subject to} \qquad G\boldsymbol{z} \geq \mathbf{1},$$

$$\text{(3.13)} \qquad z_j = 0, 1 \quad (j = 1, \ldots, l).$$

ただし，$\boldsymbol{z} = (z_1, \ldots, z_l)^{\mathrm{T}}$, $\mathbf{1} = (1, \ldots, 1)^{\mathrm{T}}$ である．

7. ILP の解として $z_{j_1}, \ldots, z_{j_{l'}}$ が 0，$z_{j'_1}, \ldots, z_{j'_{l-l'}}$ が 1 となるものが得られたとする．$x'_t = 0.a_{\sigma t+j_1} \cdots a_{\sigma t+j_{l'}} a_{\sigma t+j'_1} \cdots a_{\sigma t+j'_{l-l'}}$ とすると，$\{x'_t\}$ の上位 l' ビットについて m 次均等分布が実現している．

上記のアルゴリズムに基づいて，近似的に漸近的ランダム性を有する乱数列の設計を以下のように行う：上記のアルゴリズムで得られた系列 $\{x'_t\}$ の上位 l' ビットからなる系列について，再びアルゴリズムを適用すれば，上位 l'' ビットが $m'(= \lfloor p/l' \rfloor)$ 次均等分布する系列 $\{x''_t\}$ が得られる．以下同様に最上位の 1 ビットに到達するまで繰返す．このようにして得られた M 系列乱数列は，上位ビットに着目するほど均等分布の次数が大きくなっており，近似的には，漸近的ランダム性を満たしているということが出来るだろう．

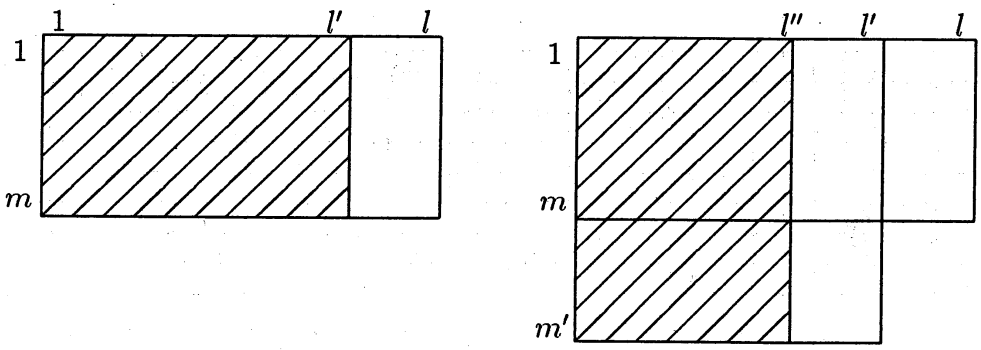

図 2. 漸近的ランダム性の近似的実現

実際にアルゴリズムを適用するときに問題となるのは，$\mathcal{C}_n$ を求める部分と，ILP を解く部分である．ILP については [2] で既に提案されている近似解法を用いる．$\mathcal{C}_n$ を求める部分については，次節で考察をする．

3.2 極小従属集合の数え上げ

GF(2) 上の行列 E_n が与えられたときに，極小な従属関係にある列ベクトルの集合をすべて求めたい．一般に有限集合上の要素の独立／従属性は，マトロイドの理論を用いて解析することができる．マトロイドの理論では，極小な従属集合をサーキットと呼ぶ．また，極大な独立集合のことを基と呼ぶ．ある 1 つの基 B に，基以外の 1 つの要素 x を付け加えた集合には x を含むサーキットが ただ 1 つ存在する．これを x による基 B に関する基本サーキットと呼び $C(x)$ で表す．われわれの問題は，マトロイドの言葉でいえば，行列 $E_n = (\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{lm})$ が与えられたときに，その列ベクトルを台集合とするマトロイドにおいて，すべてのサーキットを求める問題ということができる．

GF(2) 上の行列で表現されるマトロイドは 2 値マトロイド (binary matroid) と呼ばれる．さて 2 値マトロイドについては次の事実が知られている (例えば [1] 参照)．

命題 1　マトロイドが 2 値マトロイドであるための必要十分条件は，任意の基 B と任意のサーキット C に対して，$C(x)$ を $x \in C \setminus B$ なる要素と B によって決まる基本サーキットとすると，

(3.14)
$$C = \triangle_{x \in C \setminus B} C(x)$$

が成り立つことである．

ここで，$\triangle$ は集合に対する対称差を表す*．この命題により，マトロイドのある基 B に対する基本サーキットを求め，それらのすべての部分集合に対する対称差 (それらすべてがサーキットであるとは限らない) を求めれば，その中にすべてのサーキットが含まれていることになるので，すべての対称差の中でサーキットのチェックをすれば，全サーキットを数え上げたことになる．基本サーキットを求めるためには，列ベクトルの従属関係を調べればよい．そのために，行列のある行を定数倍して他の行に加えても列ベクトル間の独立／従属

*2 つの集合 X, Y に対する対称差は $X \triangle Y = (X \setminus Y) \cup (Y \setminus X)$ である．集合 X, Y, Z に対して $\triangle$ は結合則 $X \triangle (Y \triangle Z) = (X \triangle Y) \triangle Z$ を満たすので，3 個以上の集合に対しても対称差は定義可能である．式 (3.14) の右辺は，$C \setminus B = \{x_1, \ldots, x_\nu\}$ とすれば，$\triangle_{x \in C \setminus B} C(x) = C(x_1) \triangle \cdots \triangle C(x_\nu)$ を表している．

表 1. 漸近的ランダム性を近似的に実現するビット配置の算出過程 (A)

l	m	l'	independent bits
32	16	18	1 2 3 4 5 6 7 8 9 19 20 21 22 28 29 30 31 32
18	28	9	1 2 3 4 5 6 7 8 9
9	57	5	2 3 4 5 6
5	104	4	3 4 5 6
4	130	2	3 6

関係は変わらないことを利用し，行列 E_n に対して（行方向に） Gauss-Jordan の消去法を実行すればよい．行，列の入れ換えを許した消去法の結果，

$$\left(\begin{array}{c|c} I & K \\ \hline 0 & 0 \end{array}\right)$$

という形の行列が得られれば，得られた単位行列 I に対応する列が基になる．非基底の各列は，その非零成分をみると，基の中のどの列の集合との間に従属関係をもつかがわかる．この従属関係をなす列が，消去法の結果得られた基に対する基本サーキットである．

すべての極小従属集合を数え上げる方法は上記の通りであるが，この算法を用いて $\mathcal{C}_n$ を構成することは，所要計算時間の観点から一般には難しい．一方，ILP の制約条件を決める行列 G が，ビット間の従属性の記述に関して不十分な形ではあっても，その解に従って上位 l' ビットを選択したものが均等分布を実現している可能性はある．ビット置換した結果の乱数列の上位 l' ビットが均等分布を実現しているかどうは，容易に検証できる（選択した l' ビットに対応する重みベクトル (2.8) の線形独立性をチェックすればよい）．発見的方法であるが，ここでは G を構成する方法として，すべての $\mathcal{C}_n$ を求めるのではなく，各 E_n の基本サーキットのみから G を構成し，求まった解に対して均等分布の検証を行う，という方法を提案する．次節でその有効性を検証する．

4 数値実験例

原始多項式として $f(x) = x^{521} + x^{32} + 1$ を選んで，近似的に漸近的ランダム性を有する M 系列乱数を構成する．この原始多項式によって生成される M 系列の周期は $2^{521} - 1$ であって，これはメルセンヌ素数になり，従って任意の正整数 $n < 2^{521} - 1$ に対して系統サンプリングを考えることができるが，ここでは，$N = \{1, 2, \ldots, 16\}$ とした．また，ビット長は $l = 32$ から計算を始め，パラメータは $\sigma = 32$ とした．各 $n \in N$ について E_n を求め，それらの基本サーキットのみから ILP の制約条件の行列 G を作成し，近似解法によって解いた．$l = 32$ の場合，得られた ILP の解は，$z_0 = 14$ $(l' = 18)$ で，$\{j \mid z_j = 0\} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 19, 20, 21, 22, 28, 29, 30, 31, 32\}$ であった．次に $l = 18$ として，再び ILP を解くと $z_0 = 9$ $(l' = 9)$, $\{j \mid z_j = 0\} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ が得られた．以下同様に計算を進めた結果が，表 1 である．計算の各段階 $(l = 32, 18, 9, 5, 4)$ で，ILP の解の l' ビットが m $(= \lfloor p/l \rfloor)$ 次均等分布をしていることを検証した．

表 2. 漸近的ランダム性を近似的に達成するビット配置

A	3	6	4	5	2	1	7	8	9	19	20	21	22	28	29	30	31	32	10	11	12	13	14	15	16	17	18	23	24	25	26	27
B	6	8	9	15	16	17	18	19	21	24	25	26	27	28	1	2	3	4	5	7	10	11	12	13	14	20	22	23	29	30	31	32
C	24	9	27	22	20	21	26	4	16	17	25	1	2	3	5	6	7	8	10	11	12	13	14	15	18	19	23	28	29	30	31	32
D	8	10	13	14	15	16	17	20	21	22	24	27	28	29	2	3	4	5	6	7	9	11	12	18	19	23	25	26	30	1	31	32
E	10	19	20	21	23	24	26	29	6	8	12	13	18	22	28	4	5	7	9	11	14	15	16	17	25	27	20	31	1	2	3	32
F	25	20	15	16	13	19	21	22	28	17	18	24	8	11	23	27	2	3	4	5	6	7	9	10	12	14	27	29	1	30	31	32

表 3. m 次均等分布をするビット数

A			B			C			D			E			F		
l	m	l'	l	m	l'	l	m	l'	l	m	l'	l	m	l'	l	m	l'
32	16	18	32	16	14	32	16	11	32	18	29	32	18	28	32	18	28
18	28	9				11	47	7	29	20	14	28	21	15	28	21	16
9	57	5				7	74	4				15	40	8	16	37	12
5	104	4				4	130	3							12	50	9
4	130	2				3	173	1							9	67	5
															5	121	4
															4	151	3
															3	202	2
															2	303	1

上に示した方法によって，以下の 6 種類の異なる原始多項式によって生成される 32 ビット長の乱数に対して，漸近的ランダム性を実現するためにビット置換の計算を行った．なお，すべての例で $\sigma = 32$, $N = \{1, 2, \ldots, 16\}$ である．

(A) $x^{521} + x^{32} + 1$ （既出）　(D) $x^{607} + x^{105} + 1$

(B) $x^{521} + x^{48} + 1$　(E) $x^{607} + x^{147} + 1$

(C) $x^{521} + x^{168} + 1$　(F) $x^{607} + x^{273} + 1$

表 2 に，計算によって得られた A ～ F 各々のビット置換した結果を示す．また，表 3 に上位 l ビットの中で m 次均等分布を実現するビット数 l' を示した．表中で，B, D, E の 3 つの場合については，ILP を解いて得られた解のビットが独立性を満たさず（これは，極小従属集合の列挙が不十分で制約条件を表す行列 G が完全でなかったことを意味する），計算を最後まで進めることが出来なかった．

5 まとめ

M 系列乱数に対し，その等間隔抽出法による部分列も含めて，漸近的ランダム性を近似的に実現するための設計方法を提案した．提案した方法に基づいて数値実験を行ない，近似的に漸近的ランダム性を有する M 系列乱数をいくつか得ることができた．A, C, F の中では，漸近的ランダム性に関しては，F が一番好ましいと言えよう．

本論文では，Tausworthe 型の M 系列乱数について議論したが，実際の乱数の発生は定理 1 によって GFSR に変換して行なう．特に，前節の例のように σ が 2 のべき乗の場合には，$f_\sigma = f$, $\sigma^{-1} = 2^p/2^l$ なので，Tausworthe 型から GFSR への変換はきわめて容易である．またビットの置換については，乱数を発生させるための初期設定の中で一度だけやっ

ておけばよいので，毎回の乱数の発生の速度は，通常の M 系列乱数の発生と変わらない．

参考文献

[1] Fournier, J. C., Binary Matroids, in *Combinatorial Geometries* (N. White, ed.) (*Encyclopedia of Mathematics and Its Applications, Vol. 29*), Cambridge University Press, Cambridge, 1987.

[2] Fushimi, M., Designing a Uniform Random Number Generator Whose Subsequences Are k-Distributed, *SIAM Journal on Computing*, Vol. 17(1988), pp. 89–99.

[3] 伏見正則, 乱数, UP 応用数学選書 12, 東京大学出版会, 東京, 1989.

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

[5] Kolmogorov, A. N., On Tables of Random Numbers, *Sankhya*, Vol. 25A(1963), pp. 369–376.

[6] Lewis, T. G. and W. H. Payne, Generalized Feedback Shift Register Pseudorandom Number Algorithms, *Journal of the Association for Computing Machinery*, Vol. 20(1973), pp. 456–468.

[7] Tausworthe, R. C., Random Numbers Generated by Linear Recurrence Modulo Tow, *Mathematics of Computation*, Vol. 19(1965), pp. 201–209.

[8] Tootill, J. P. R., W. D. Robinson, and D. J. Eagle, An Asymptotically Random Tausworthe Sequence, *Journal of the Association for Computing Machinery*, Vol. 20(1973), pp. 469–481.