論文の内容の要旨

論文題目 Fast lattice reduction algorithms for optimizing $\mathbf{F}_2$-linear pseudorandom number generators
($\mathbf{F}_2$-線形擬似乱数発生法の最適化のための高速格子簡約アルゴリズム)

氏名　　原瀬晋

本論文は，二元体 $\mathbf{F}_2$ 上の線形擬似乱数発生法に対して，その評価方法の一つとして知られている高次元均等分布性における均等分布の次元計算アルゴリズムの高速化について論じる．近年，大規模なコンピュータ・シミュレーションを行う際，並列化・分散化が行われるようになり，複数の CPU 上に，異なるパラメータセットを有する擬似乱数発生法を割り当てて使用する機会が増加している．そこで，非常に多数のパラメータ探索を行い，それぞれのパラメータに対して，乱数性を評価する必要が生じる．既存の均等分布の次元計算アルゴリズムは計算時間を要し，探索のボトルネックとなるため，高速化が望まれていた．

本論文では，$\mathbf{F}_2$-線形擬似乱数発生法と呼ばれる次のモデルを扱う．状態空間を $S := \mathbf{F}_2^p$，状態遷移関数を $f : S \to S$，出力の集合を $O := \mathbf{F}_2^w$ (w はコンピュータのワードサイズ)，出力関数を $o : S \to O$ とおく．f と o は高速に計算可能な $\mathbf{F}_2$-線形写像を熟考の上に選ぶ．初期状態 $s_0 \in S$ が与えられたとき，単位時間ごとに漸化式

$$s_{i+1} = f(s_i) \quad (i = 0, 1, 2, \ldots)$$

を計算して，出力列

$$o(s_0), o(s_1), o(s_2), \ldots \in O$$

を得るものとする．これらの出力を符号なし w ビット二進整数と同一視し，擬似乱数として用いる．

f と o を決める際には，擬似乱数発生法の一様性の評価規準として知られている高次元均等分布性が用いられる．この概念は，出力列の上位 v ビットのみに着目して得られる，v ビット精度の均等分布の次元 $k(v)$ の大きさに基づく．$k(v)$ は次に述べる二元体係数形式的べき級数体 $K := \mathbf{F}_2((t^{-1}))$ 上の簡約基底法を用いて，具体的に計算することができる．状態から出力の上位 v ビットのみの出力を与える出力関数を $o_v : S \xrightarrow{o} O \to \mathbf{F}_2^v$ とし，上位 v ビット出力列 $o_v(s_0), o_v(s_1), o_v(s_2), \ldots \in \mathbf{F}_2^v$ に対して，以下の $\mathbf{F}_2$-線形生成母関数 χ_v: $S \to K^v$ を考える：

$$\chi_v(s_0) := \sum_{j=0}^{\infty} o_v(f^j(s_0))t^{-1-j} = o_v(s_0)t^{-1} + o_v(s_1)t^{-2} + \cdots \in K^v.$$

この $\chi_v(s_0)$ と v 次元単位ベクトル $e_1, \ldots, e_v$ を生成集合とする $\mathbf{F}_2[t]$ 上 の加群

$$\Lambda_v := \langle e_1, e_2, \ldots, e_v, \chi_v(s_0) \rangle_{\mathbf{F}_2[t]}$$

を考えると，v 次元 $\mathbf{F}_2[t]$-格子となる．ここで，$\mathbf{F}_2[t]$-格子の基底として，t の次数によって定義されたウルトラノルムに関して最短のものを簡約基底と呼ぶ．状態遷移関数 f の特性多項式が既約の場合，Λ_v の簡約基底における最長ベクトルのノルムが $k(v)$ を与えることが知られている．

擬似乱数発生法の評価のためには，すべての $k(v)$ $(v = 1, \ldots, w)$ を計算することが必要となる．Couture-L'Ecuyer(2000) は，Λ_v の双対格子と Lenstra(1985) による格子基底簡約アルゴリズムに基づいて，$k(v-1)$ を計算する際に求めた簡約基底を利用して $k(v)$ を効率的に計算する方法を提案した．この方法は既存の最も高速な計算法として知られていたが，次数の高い多項式を成分に持つベクトルを扱う必要があり，改善の余地があった．

そこで，本論文では，第 I 部と第 II 部とに分けて，$\mathbf{F}_2$-線形擬似乱数発生法に対して，均等分布の次元 $k(v)$ の計算アルゴリズムの高速化を提案する．

第 I 部では，双対格子に変換せず，Λ_v の簡約基底を直接求める方針を取った．そして，従来用いられていた Lenstra アルゴリズムの代わりに，Schmidt(1991) による格子基底簡約アルゴリズムを採用し，入力が生成集合に対応するように修正し，SGR アルゴリズムと名づけた．これは，双対格子の場合の計算順序とは逆に，$k(v)$ を $v = w, w-1, \ldots, 1$ の順に帰納的に計算するためである．すなわち，$\rho : K^v \to K^{v-1}$ を v 番目の座標を削除する射影とすると，$\rho(\Lambda_v) = \Lambda_{v-1}$ が示されている (Couture-L'Ecuyer, 2000)．そこで，Λ_v の簡約基底を求めておき，各々のベクトルに ρ を施すと

Λ_{v-1} の生成集合が得られる．入力が基底に制限されている Lenstra アルゴリズムでは適用できないが，SGR アルゴリズムではこの生成集合から簡約基底を求めることが可能となる．この射影を用いた計算法をインダクティブ・プロジェクションと名付けた．

更に，Λ_v の点を状態空間 S の元を用いて表現する方法を提案した．状態遷移関数 f の特性多項式が既約で，χ_v が非零であると仮定する．このとき，$\mathbf{F}_2$-ベクトル空間の直和として $\Lambda_v = \mathbf{F}_2[t]^v \oplus \chi_v(S)$ と分解でき，任意の格子点 $\omega \in \Lambda_v$ は，$\omega = poly + \chi_v(s)$ なる一意的な表現 $(poly, s) \in \mathbf{F}_2[t]^v \times S$ を有する．この $(poly, s)$ を ω の状態表現と呼ぶ．SGR アルゴリズムで必要となる格子点同士の和は状態表現の和に，t 倍の操作 $\omega \cdot t$ は $(poly \cdot t + o_v(s), f(s))$ と状態表現され，状態表現内部の代数的な演算に帰着可能であることを示した．この表現により，格子点の成分に現れるべき級数を回避し，結果として，格子簡約アルゴリズム内で扱う Λ_v の生成集合に対して，各ベクトルのメモリ使用量を $\dim(S)$ ビットに抑えることに成功した．

以上，SGR アルゴリズム，インダクティブ・プロジェクション，状態表現の一連の手法を SIS 法と名づけ，ビット演算量の最悪値解析を行った．その結果，Couture-L'Ecuyer(2000) よりも弱い仮定の下で，すべての $k(v)$ $(v = w, \ldots, 1)$ を求めるために要するビット演算量の上限

$$2w \dim(S)^2 + \frac{4}{3}w^3 \dim(S) + \frac{1}{4}w^4$$

を得た．Couture-L'Ecuyer(2000) の方法と比較し，$\dim(S)^2$ の項における w のオーダーが減少しており，実用上，$\dim(S)$ が w に対して非常に大きいため，SIS 法の方が高速であると予想される．実際，上述のアルゴリズムを代表的な $\mathbf{F}_2$-線形擬似乱数発生法である MT 法 (松本-西村, 1997) と WELL 法 (Panneton-L'Ecuyer-松本, 2006) に適用し，計算機実験を行った．すべての $k(v)$ を求めるのかかった CPU 時間を測定したところ，既存の双対格子による方法よりも 10 倍程度の高速化に成功した．

第 II 部では，SIS 法の改良を行い，更なる高速化を図った．SIS 法において採用した SGR アルゴリズムは，係数ベクトル間の $\mathbf{F}_2$-線形関係式を求めるために，行列の掃き出し計算を繰り返す必要が生じる．格子の次元が大きくなると，この部分に起因する計算量の増加は無視できない．そこで，掃き出し計算を回避するために，SGR アルゴリズムの代わりに，Mulders-Storjohann(2003) による簡約アルゴリズムを採用することを提案した．特に，Λ_v の生成集合の簡約では，Wang-Zhu-Pei(2004) による並べ替えの手法を用いることにより，係数ベクトルを一定の条件 (三角条件) に保ったまま，簡約を行うことが可能となる．これらの方法を使いやすい形に修正し，ピボット格子簡約アルゴリズムとして与え，インダクティブ・プロジェクションおよび状態表現と組み合わせ，PIS 法と名付けた．

PIS 法の計算量として，すべての $k(v)$ $(v = w, \ldots, 1)$ の計算に要するビット演算量の上限

$$2w \dim(S)^2 + \frac{1}{2}w^2 \dim(S) + \frac{1}{2}w^2(w+1)$$

を導出し，SIS 法における $\dim(S)^1$ の係数並びに $\dim(S)$ を含まない項における w のオーダーが共に 1 次減少することを示した．これは，w が増加した場合に PIS 法の方が有効であることを示唆している．実際，SIS 法と PIS 法による計算機実験を行い，MT 法および WELL 法に対して，すべての $k(v)$ を求めるのにかかった CPU 時間を計測した．$\dim(S) = 19937$ の場合，$w = 32$ の例では SIS 法の約 1.5 倍，$w = 64$ の場合には約 3 倍高速化された．

一方，異なった方向性の高速化技法として，初期状態 $s_0 \in S$ の取り方について考察した．Panneton-L'Ecuyer-松本 (2006) は，MT 法の初期状態に，極端に 0 の多い零超過状態を与えると，長時間に渡って，出力の 0 と 1 のバランスに偏りが見られる MT 法の欠点を述べている．本論文では，この現象を逆手にとり，零超過状態 s_0 を用いて Λ_v の生成集合の一元 $\chi_v(s_0)$ を与えると，MT 法の格子簡約に現れる状態表現に 1 が疎な状態が続くことから，非常に高速に簡約基底が求まることを発見した．PIS 法と組み合わせ，実験を行ったところ，MT 法においては，通常の初期化を用いた SIS 法に比べて 10 倍以上の高速化を得た．また，零超過状態による初期化の有無にかかわらず，いずれの場合も，PIS 法は SIS 法よりも高速となった．

このアルゴリズムは MT 型発生法のパラメータ探索ソフトウェア Dynamic Creator 及び MTGPDC に組み込まれ，ホームページから配布されている．