MATHEMATICS OF COMPUTATION
Volume 80, Number 273, January 2011, Pages 395–407
S 0025-5718(2010)02391-9
Article electronically published on June 18, 2010

FAST LATTICE REDUCTION FOR $\mathbf{F}_2$-LINEAR PSEUDORANDOM NUMBER GENERATORS

SHIN HARASE, MAKOTO MATSUMOTO, AND MUTSUO SAITO

Abstract. Sequences generated by an $\mathbf{F}_2$-linear recursion have wide applications, in particular, pseudorandom number generation. The dimension of equidistribution with v-bit accuracy is a most important criterion for the uniformity of the generated sequence. The fastest known method for computing these dimensions is proposed by Couture and L'Ecuyer, based on Lenstra's lattice basis reduction and the dual lattice to the lattice of vector-valued generating functions (with components in the formal power series $\mathbf{F}_2[[t^{-1}]]$) associated to the output $\mathbf{F}_2$-vector sequence. In this paper we propose a similar but faster algorithm, where (1) the state space is used to represent vectors with components in the formal power series, (2) the dual lattice is not necessary, and (3) Lenstra reduction is replaced with a simpler basis reduction. The computational complexity of our method is smaller than for the Couture-L'Ecuyer method. Experiments show that our method improves the speed by a factor of 10 for Mersenne Twister MT19937 and for WELL generators with state sizes of 19937 bits and 44497 bits.

1. Introduction

Let $\mathbf{F}_2 := \{0, 1\}$ denote the two element field. Sequence generators based on $\mathbf{F}_2$-linear recursion are widely used in practical applications, in particular, as pseudorandom number generators. Among the quality criteria of the generators, the notion of *the dimension of equidistribution with v-bit accuracy* is widely used as a most informative criterion for the higher dimensional uniformness of the distribution of the sequence (see [6]). When the state space is large, the computation of these dimensions is time consuming, and at the designing stage of the generator, it becomes a bottleneck in finding good parameters. Couture, L'Ecuyer and Tezuka [2] introduced a lattice basis reduction method to compute these dimensions, over the formal power series field $\mathbf{F}_2((t^{-1}))$. Couture and L'Ecuyer [1] improved Tezuka's resolutionwise lattice method [15] by using the dual lattice. The aim of this paper is to propose a simpler and more efficient method. In §2, we briefly recall the notion of $\mathbf{F}_2$-linear generators and computation of the dimensions of equidistribution using lattices. In §3, we introduce a method to compute a reduced basis from a generating set, a method to compute the dimension of equidistribution with v-bit

Received by the editor July 15, 2009 and, in revised form, October 18, 2009.

2010 *Mathematics Subject Classification.* Primary 11K45, Secondary 65C10.

The first author was partially supported by Grant-in-Aid for JSPS Fellows 21·4427.

The second author was partially supported by JSPS Grant-in-Aid for Challenging Exploratory Research 21654017, Scientific Research (A) 19204002 and JSPS Core-to-Core Program 18005.

The third author was partially supported by JSPS Grant-in-Aid for Challenging Exploratory Research 21654004.

accuracy for $v = w, w-1, w-2, \ldots$ inductively in this order, and an efficient representation of the lattice elements (and operations on them) in terms of the state space. We give the computational complexity of the proposed method in §4. We compare the speeds of the state representation method and the dual basis method [1] in §5 using a C++ implementation.

2. Linear generator and lattice method

2.1. **Dimension of equidistribution.** We recall basic materials; see [6] and its references for the original definitions. An $\mathbf{F}_2$-linear sequence generator consists of a state space $S = \mathbf{F}_2^p$, an $\mathbf{F}_2$-linear state transition function $f : S \to S$, an $\mathbf{F}_2$-linear output function $o : S \to O$ where $O = \mathbf{F}_2^w$ is the set of outputs (w intended for the word size of the machine). Once an initial state $s_0 \in S$ is given, the generator computes the next state by the recursion $s_{i+1} = f(s_i)$ $(i = 0, 1, 2, \ldots)$ every time unit, and the output sequence is given by $o(s_0), o(s_1), o(s_2), \ldots \in O$. Throughout this paper, $P(t)$ denotes the characteristic polynomial of f.

The dimension of equidistribution $k(v)$ of such a generator is defined as follows. We identify the output set $\mathbf{F}_2^w$ with the set of unsigned w-bit binary integers. Let us consider the most significant v bits (v MSBs) in the outputs. We regard this *to consider the output with v-bit accuracy.* This amounts to consider the composition $o_v : S \xrightarrow{o} \mathbf{F}_2^w \to \mathbf{F}_2^v$, where the latter map denotes taking the v MSBs. Define the k-tuple output function for any $k \geq 0$ by

$$o_v^{(k)} : S \to (\mathbf{F}_2^v)^k, \quad s_0 \mapsto (o_v(s_0), o_v(f(s_0)), \ldots, o_v(f^{k-1}(s_0))),$$

namely, $o_v^{(k)}(s_0)$ is the consecutive k-tuple of the outputs from the state s_0.

Definition 2.1. If $o_v^{(k)} : S \to (\mathbf{F}_2^v)^k$ is surjective, then the generator is said to be k-dimensionally equidistributed with v-bit accuracy. The largest value of k with this property is called the dimension of the equidistribution with v-bit accuracy, denoted by $k(v)$.

Since $o_v^{(k)}$ is linear, k-dimensional equidistribution means that every element in $(\mathbf{F}_2^v)^k$ occurs with the same probability, when the initial state s_0 is uniformly distributed over the state space. If the generator has the maximal period $2^p - 1$, then this amounts to saying that every kv-bit pattern occurs as consecutive overlapping k-tuples of v-bit integers equally often for the whole period, except the all-zero pattern which occurs once less often.

The larger $k(v)$ for each of $1 \leq v \leq w$ is desirable. By comparing the dimensions of S and $(\mathbf{F}_2^v)^k$, we have a trivial bound $p = \dim(S) \geq kv$, and hence $k(v) \leq \lfloor p/v \rfloor$. If the equality holds, then the generator is said to be *maximally equidistributed.*

We may compute $k(v)$ by checking the surjectivity by linear algebra [3]. For a large p, Couture et al. [2] and Tezuka [15] proposed much faster algorithms based on lattice structures over power series.

2.2. **Lattice structure.** We briefly recall the above-mentioned lattice method. Let K denote the formal power series field

$$K := \mathbf{F}_2((t^{-1})) = \left\{ \sum_{j=j_0}^{\infty} a_j t^{-j} \mid a_j \in \mathbf{F}_2, j_0 \in \mathbf{Z} \right\}.$$

For $\alpha = \sum_{j=j_0}^{\infty} a_j t^{-j} \in K$, we define a standard norm by

$$|\alpha| := \begin{cases} \max\{-j \in \mathbf{Z} \mid a_j \neq 0\} & \text{if } \alpha \neq 0, \\ -\infty & \text{if } \alpha = 0. \end{cases}$$

For a vector $\gamma = (\alpha_1, \alpha_2, \ldots, \alpha_v) \in K^v$, we define $||\gamma|| := \max_{1 \le i \le v} |\alpha_i|$. Note that $|\alpha|$ and $||\gamma||$ are often negative integers. For $\gamma \neq 0$, we define its *coefficient vector at the leading term* $\pi(\gamma) \in \mathbf{F}_2^v$ by

$$\pi(\gamma) := (a_{1,||\gamma||}, a_{2,||\gamma||}, \ldots, a_{v,||\gamma||}), \text{ so that } \gamma = \pi(\gamma) t^{||\gamma||} + \text{lower degree terms in } t.$$

A subset $L \subset K^v$ is called an $\mathbf{F}_2[t]$-*lattice* if it is the set of a linear combination of $\omega_1, \omega_2, \ldots, \omega_v$ with coefficients in $\mathbf{F}_2[t]$, where $\omega_1, \ldots, \omega_v$ are linearly independent over K. Such a set of vectors is called a basis of L.

Theorem 2.2 (Lemma 1 of [9]). *Let $\omega_1, \ldots, \omega_v$ be the points in an $\mathbf{F}_2[t]$-lattice $L \subset K^v$ with the following properties:*

(1) *ω_1 is a shortest nonzero vector in L;*
(2) *for $i = 2, \ldots, v$, ω_i is a shortest vector among the set of vectors ω in L such that $\omega_1, \ldots, \omega_{i-1}, \omega$ are linearly independent over K.*

Then $\omega_1, \ldots, \omega_v$ form a basis of L.

Such a basis is called a *reduced basis* for L. It is not unique, but the numbers $\nu_i := ||\omega_i||$ are invariants of the lattice, called *successive minima* (see [8]). Let us consider an $\mathbf{F}_2$-linear generator. For a v-bit output sequence from an initial state $s_0 \in S$, let χ_v denote a (v-dimensional vector-valued) generating function in K^v:

$$\chi_v(s_0) := \sum_{j=0}^{\infty} o_v(f^j(s_0)) t^{-1-j} = o_v(s_0) t^{-1} + o_v(s_1) t^{-2} + \cdots \in K^v. \tag{2.1}$$

This gives an $\mathbf{F}_2$-linear map $\chi_v : S \to K^v$. We define a lattice Λ_v in K^v as the set of $\mathbf{F}_2[t]$-linear combinations of $\chi_v(s_0)$ and the unit vectors, namely, the $\mathbf{F}_2[t]$-linear span

$$\Lambda_v := \langle \chi_v(s_0), e_1, e_2, \ldots, e_v \rangle_{\mathbf{F}_2[t]}, \tag{2.2}$$

where e_i is the vector whose i-th component is 1 and 0 for the other components ($i = 1, \ldots, v$). The vectors $\chi_v(s_0), e_1, e_2, \ldots, e_v$ form a generating set of Λ_v over $\mathbf{F}_2[t]$, but they are K-linearly dependent because they form a set of cardinality $v + 1$. Still, these $v + 1$ vectors generate a lattice: after multiplying each vector by the characteristic polynomial $P(t)$, every coordinate becomes a polynomial, and thus there is a basis consisting of v vectors.

Theorem 2.3 ([2, 15]). *Assume that $P(t)$ is irreducible. Take nonzero $s_0 \in S$. Then, $k(v) = -\nu_v$ holds, where ν_v is the v-th successive minimum of the lattice Λ_v in K^v.*

In [1], to obtain a reduced basis, the authors used the Lenstra reduction algorithm [7], which requires a basis of Λ_v as an initial input. Since $P(t)\chi_v(s_0) \in \mathbf{F}_2[t]^v$, one can define polynomials $(g_1(t), g_2(t), \ldots, g_v(t)) := P(t)\chi_v(s_0)$. Let $g_1(t)^{-1}$ denote a polynomial which is a multiplicative inverse to $g_1(t)$ modulo $P(t)$ (which exists if the MSB of the sequence is not constantly 0), and define a vector with polynomial components

$$\Psi := (g_1(t)^{-1} \cdot P(t)\chi_v(s_0)) \mod P(t).$$

The first component of Ψ is 1, and the vectors $\Psi/P(t), e_2, \ldots, e_v$ form a basis of Λ_v. In applying the Lenstra basis reduction, to avoid the infinite formal power series $\Psi/P(t)$, we multiply this basis by $P(t)$ to reduce to the polynomial computation. We apply Lenstra's reduction to the polynomial vectors

$$(1, \bar{g}_2(t), \ldots, \bar{g}_v(t)), (0, P(t), \ldots, 0), \ldots, (0, \ldots, 0, P(t)) \in \mathbf{F}_2[t]^v, \tag{2.3}$$

where $\bar{g}_j(t) := g_1(t)^{-1} g_j(t) \mod P(t)$ $(2 \leq j \leq v)$.

Later we use the following.

Lemma 2.4. *Let $\nu_1, \ldots, \nu_v$ be the successive minima of Λ_v. We have*

$$-\dim(S) = \sum_{i=1}^{v} \nu_i.$$

Proof. From (25) in [8] (or (3) in [1]), the lemma follows. □

2.3. **Dual lattice.** For a lattice $L \subset K^v$, its *dual lattice* L' is defined by

$$L' := \{h' \in K^v \mid h \cdot h' \in \mathbf{F}_2[t], \text{ for all } h \in L\},$$

where $h \cdot h' = \sum_{j=1}^{l} h_j(t) \cdot h'_j(t)$ (the scalar product) for $h = (h_1(t), \ldots, h_v(t))$ and $h' = (h'_1(t), \ldots, h'_v(t))$. The vectors

$$(P(t), 0, \ldots, 0), (-\bar{g}_2(t), 1, \ldots, 0, 0), \ldots, (-\bar{g}_v(t), 0, \cdots, 0, 1) \tag{2.4}$$

form (the so-called dual) basis of the dual lattice Λ'_v. The next theorem reduces the computation of successive minima of Λ_v to those of the dual.

Theorem 2.5 ([1]). *Let $\nu_1, \nu_2, \ldots, \nu_v$ be the successive minima of Λ_v, and let ν'_1, $\nu'_2, \ldots, \nu'_v$ be the successive minima of Λ'_v. We have, for $i = 1, 2, \ldots, v$,*

$$\nu_i + \nu'_{v-i+1} = 0.$$

A big advantage of using the dual in [1] is that we can use the reduced basis of Λ'_{v-1} to compute that of Λ'_v.

Assume $1 < v \leq w$. Let $\iota : K^{v-1} \to K^v$ be an inclusion by supplementing 0 at the vth coordinate, and $\rho : K^v \to K^{v-1}$ the projection by deletion of the vth coordinate.

Theorem 2.6 ([1]). *For $1 < v \leq w$, we have*

(1) $\rho(\Lambda_v) = \Lambda_{v-1}$.
(2) $\Lambda'_v = \iota(\Lambda'_{v-1}) \oplus \langle(-\bar{g}_v(t), 0, \cdots, 0, 1)\rangle_{\mathbf{F}_2[t]}$,

where $\langle(-\bar{g}_v(t), 0, \cdots, 0, 1)\rangle_{\mathbf{F}_2[t]}$ denotes the space spanned by a single vector.

The first claim follows from the definition of Λ_v. The second follows from (2.4). Thus, for the dual lattice Λ'_v, we may choose its lattice basis as the union of a reduced basis of Λ'_{v-1} and the vector $(-\bar{g}_v(t), 0, \cdots, 0, 1)$. Consequently, if we compute reduced bases of $\Lambda'_2, \Lambda'_3, \ldots$ in this order, then we can use a reduced basis of Λ'_{v-1} in computing that of Λ'_v. Computational complexity given in [1] shows a significant advantage of this method, which we call the *dual lattice method*.

3. Main result

3.1. **Schmidt's generating set reduction.** The Lenstra basis reduction requires a basis of the lattice. Although it is not difficult to obtain a basis from a set of generating vectors (cf. [5]), there is an even simpler reduction algorithm by Schmidt [14, p. 200], which can be easily generalized to an algorithm to obtain a reduced basis from a generating set. We describe this generalized version, which we call *Schmidt's generating set reduction* (SGR).

```
procedure : Schmidt's generating set reduction
input : a generating set $\omega_1, \omega_2, \ldots, \omega_m$ which spans a lattice $L$ over $\mathbf{F}_2[t]$.
output : a reduced basis $\omega_1, \omega_2, \ldots, \omega_v \in L$.
begin
while $\pi(\omega_1), \pi(\omega_2), \ldots, \pi(\omega_m)$ are linearly dependent over $\mathbf{F}_2$ do
   (reduction step)
   Find a vector $(\alpha_1, \alpha_2, \ldots, \alpha_m) \in \mathbf{F}_2^v$ such that $\sum_{i=1}^m \alpha_i \pi(\omega_i) = (0, \ldots, 0)$.
   Find an integer $i_{\max}$ such that $||\omega_{i_{\max}}|| = \max\{||\omega_i|| \mid 1 \le i \le m, \alpha_i \ne 0\}$.
   Set $\omega_{i_{\max}} \leftarrow \sum_{i=1}^m \alpha_i t^{-||\omega_i|| + ||\omega_{i_{\max}}||} \omega_i$.
   If $\omega_{i_{\max}} = 0$ then swap $\omega_{i_{\max}}$ and $\omega_m$, and set $m \leftarrow m - 1$.
end while
Renumber $\omega_1, \omega_2, \ldots, \omega_m$ in such a way that $||\omega_1|| \le ||\omega_2|| \le \ldots \le ||\omega_m||$.
end
```

Each reduction step decreases $\sum_i ||\omega_i||$, and since there is a shortest vector in a lattice, the algorithm terminates. Then, the number m of vectors is reduced to v, and $\pi(\omega_1), \ldots, \pi(\omega_v)$ are linearly independent. Such a basis is called a *reduced basis* in [14, p. 199], and its equivalence to that in Theorem 2.2 follows from the uniqueness of their length ([14, p. 201]).

Using SGR, we can use ρ and (1) in Theorem 2.6 to obtain a reduced basis of Λ_v from that of Λ_{v+1}: if $\omega_1, \ldots, \omega_{v+1}$ is a reduced basis of Λ_{v+1}, we obtain a generating set $\rho(\omega_1), \ldots, \rho(\omega_{v+1})$ of Λ_v, hence we may apply SGR. Since $\rho(\omega_1), \ldots, \rho(\omega_{v+1})$ are short, this lowers computational complexity significantly when one computes all $k(v)$ for $1 \le v \le w$ (see §3). We call this method *inductive projection.* Note that this method computes $k(w), k(w-1), \ldots, k(1)$ in this order, which is converse to the standard techniques (e.g., [1], [5]).

SGR in the inductive projection is proved to terminate when one vector is eliminated, as follows.

Theorem 3.1. *Let $\omega_1, \ldots, \omega_{v+1}$ be a generating set of an $\mathbf{F}_2[t]$-lattice $L \subset K^v$. Suppose that $\pi(\omega_1), \ldots, \pi(\omega_{v+1})$ has rank v. When we apply the SGR algorithm to this generating set, then it terminates when the number of the vectors becomes v, namely, when a vector is reduced to zero.*

Let $\omega'_1, \ldots, \omega'_v$ be the obtained reduced basis. The number of reduction steps in SGR required before the termination is bounded from above by

$$\sum_{i=1}^{v+1} ||\omega_i|| - \sum_{i=1}^{v} ||\omega'_i|| - ||\omega^{\text{last}}|| + 1, \tag{3.1}$$

where ω^{last} denotes the last vector reduced to zero at the final step in SGR.

In particular, let L be Λ_v in (2.2). Then, the value of (3.1) is bounded from above by $-||\omega^{\text{last}}|| + 1$, if $\omega_1, \ldots, \omega_{v+1}$ are the image by ρ of a reduced basis of the lattice Λ_{v+1} and the characteristic polynomial $P(t)$ is irreducible.

Proof. In SGR, in one reduction step, one vector is reduced by subtracting an $\mathbf{F}_2[t]$-linear combination of the other v vectors. Looking at the coefficient vectors at the leading term, this amounts to eliminating one $\mathbf{F}_2$ vector by subtracting an $\mathbf{F}_2$-linear combination of the other v vectors. The coefficients of the leading term of the reduced vector changes, but the other v vectors do not change. The reducibility implies that even if we throw away the reduced vector, still the rank of the coefficient vectors at the leading terms does not decrease. Thus, the rank of vectors $\pi(\omega_1), \dots, \pi(\omega_{v+1})$ is always v. Consequently, if the reduced vector becomes zero, then the other v vectors have rank v at the leading terms, which means the termination.

In each reduction step, the sum $\sum_{i=1}^{v+1} ||\omega_i||$ is decreased at least by one. When one vector is reduced to zero, then this value becomes $-\infty$. We look at the last step of reduction. There is a vector ω^{last} that is reduced to zero, while the other v vectors are unchanged and become the reduced basis. At this stage, the above sum is $||\omega^{\mathrm{last}}|| + \sum_{i=1}^{v} ||\omega'_i||$. Hence, the number of steps is bounded by their difference + 1, namely (3.1).

If the lattices are from an $\mathbf{F}_2$-linear generator, then $\sum_{i=1}^{v} ||\omega'_i|| = -\dim(S)$ holds by Lemma 2.4. If $\tilde{\omega}_i$ $(i = 1, \dots, v+1)$ is a reduced basis of Λ_{v+1}, then

$$\sum_{i=1}^{v+1} ||\rho(\tilde{\omega}_i)|| \leq \sum_{i=1}^{v+1} ||\tilde{\omega}_i|| = -\dim(S),$$

hence the result. □

We check that in inductive projection, SGR satisfies the condition in Theorem 3.1. For $v = w$, we apply SGR to a generating set (2.2), whose cardinality is $v + 1$. In the induction step, we have $v + 1$ generators projected from a reduced basis, and SGR reduces them to v generators. In both cases, the coefficient vectors of the leading terms of the generating set have rank v as $\mathbf{F}_2$ vectors. Namely, $\pi(\chi_w(s_0)), \pi(e_1), \pi(e_2), \dots, \pi(e_w)$ have rank w. In the induction step, let $\omega_1, \dots, \omega_{v+1}$ be a reduced basis of Λ_{v+1}. Then $\pi(\rho(\omega_1)), \dots, \pi(\rho(\omega_{v+1}))$ have rank v, since $\pi(\omega_1), \dots, \pi(\omega_{v+1})$ have rank $v + 1$ and the rank of all $\pi(\rho(\omega_i))$ is at least the rank of $\rho(\pi(\omega_1)), \dots, \rho(\pi(\omega_{v+1}))$, which is v, and consequently the rank must be v. Thus, both cases satisfy the condition of Theorem 3.1.

Corollary 3.2. *Under a heuristic assumption that on average* $||\omega^{\mathrm{last}}|| \geq -\dim(S)/v$ *holds, the average number of the reduction steps in SGR to obtain a reduced basis of* Λ_v *from that of* Λ_{v+1} *is bounded from above by* $\dim(S)/v + 1$.

Proof. The assumption is that, $||\omega^{\mathrm{last}}||$ in the proof of the theorem is on average larger than or equal to the average of $||\omega'_1||, \dots, ||\omega'_v||$. This is justified by the fact that ω^{last} is reduced by ω'_i, hence has on average a larger norm than the average norm of $||\omega'_i||$, which is $-\dim(S)/v$ by Lemma 2.4. □

Note that ω^{last} is reduced often by using the longest vector or the second longest vector among ω'_i, hence the above bound $\dim(S)/v + 1$ seems overestimated: SGR tends to stop in a smaller number of steps, which agrees with our experiments.

Remark 3.3. There is a modified Lenstra reduction algorithm [13] applicable to a generating set of a lattice, but its efficiency seems comparable to SGR. Wang and Zhu [17] and Wang, Zhu and Pei [18] applied SGR to compute the linear complexity of a multisequence. A more informative complexity, based on the successive minima obtained using SGR, is given by Wang and Niederreiter [16].

3.2. **State representation.** Another merit of the dual lattice method in [1] is that the space complexity is reduced. If we apply SGR to the generating set (2.2) polynomialized by multiplying by $P(t)$, each vector has components being polynomials of degree smaller or equal to $\deg(P(t)) = \dim(S)$. Thus, one vector requires $\dim(S) \times v$ bits of memory, and the generating set consumes $v(v+1)\dim(S)$ bits. We need to start from $v = w$, which costs a lot if $\dim(S)$ is large. On the contrary, in the dual lattice method, for $v = 1$ we have no reduction step (and $k(1) = \dim(S)$), and for $v = 2$ we need $2\dim(S)$ bits of memory for each of two vectors, and after a basis reduction, the components of the vectors in a reduced basis have degree $\dim(S)/2$ on average, thus $\dim(S)$ bits for one vector and $2\dim(S)$ bits for a reduced basis. In the same way, the reduced basis for Λ'_v consumes $v\dim(S)$ bits, which improves on $v(v+1)\dim(S)$ for the original lattice.

Instead of using the dual lattice, we propose a method to represent a vector in the lattice Λ_v by a state, which we call the *state representation*. Since one vector consumes only $\dim(S)$ bits of memory, memory efficiency is comparable to the dual lattice method (or better, since we need no assumption on the reducedness). Recall the map $\chi_v : S \to K^v$ defined in (2.1). Note that $K = \mathbf{F}_2[t] \oplus (\mathbf{F}_2[[t^{-1}]] \cdot t^{-1})$ as an $\mathbf{F}_2$-vector space, since any element of K is a sum of its polynomial part (namely, a linear combination of t^j with $j \geq 0$) and its fractional part (namely, an infinite linear combination of t^j with $j < 0$) in a unique way. Hence we have

$$K^v = \mathbf{F}_2[t]^v \oplus (\mathbf{F}_2[[t^{-1}]] \cdot t^{-1})^v.$$

The first direct summand is called the *polynomial part*, and the second is the *fractional part* which we denote by F^v. Let $F(\Lambda_v)$ be the fractional part $F^v \cap \Lambda_v$. Since Λ_v contains $\mathbf{F}_2[t]^v$, the polynomial part of any element in the lattice is in Λ_v, and so is the fractional part, namely:

$$\Lambda_v = \mathbf{F}_2[t]^v \oplus F(\Lambda_v)$$

as an $\mathbf{F}_2$-vector space. Note that the image of χ_v lies in $F(\Lambda_v)$. Note also that $\Lambda_v/(\mathbf{F}_2[t]^v)$ is an $\mathbf{F}_2[t]$-module.

Lemma 3.4.

$$\chi_v : S \to \Lambda_v/(\mathbf{F}_2[t]^v)$$

is a homomorphism as $\mathbf{F}_2[t]$*-modules. If the characteristic polynomial* $P(t)$ *is irreducible and* χ_v *is nonzero, then* χ_v *is an isomorphism.*

If $P(t)$ is irreducible, then this lemma implies that the fractional part of a lattice element has a unique representation by a state in S, and the sum and multiplication by t for lattice elements can be computed by those for the corresponding states. Thus, we can implement lattice reduction algorithms using operations on S. This is a key to reduce the space and time complexities by the state representation.

Proof. The action of t on $s \in S$ is defined by $t \cdot s := f(s)$. Since χ_v is linear, to show homomorphy, it suffices to show that

$$\chi_v(f(s_0)) \equiv t \cdot \chi_v(s_0) \mod \mathbf{F}_2[t]^v.$$

But by the definition (2.1), $\chi_v(f(s_0)) = \sum_{j=0}^{\infty} o_v(f^{j+1}(s_0))t^{-1-j}$ and $t \cdot \chi_v(s_0) = \sum_{j=0}^{\infty} o_v(f^j(s_0))t^{-j}$, hence their difference is an $\mathbf{F}_2$ vector $o_v(s_0) \in \mathbf{F}_2[t]^v$.

Suppose that $P(t)$ is irreducible. Then, since $P(t)$ trivially acts on S, S is a $k := \mathbf{F}_2[t]/(P(t))$-module with k being a field. Since $\dim(S) = \deg(P(t))$, S is a one-dimensional k-vector space. On the other hand, $\Lambda_v/(\mathbf{F}_2[t]^v)$ is also a k-vector space, which is generated by a single element $\chi_v(s_0)$. Thus, χ_v is a K-linear map between two one-dimensional spaces, so χ_v is an isomorphism if nonzero. □

From now on, we assume irreducibility of $P(t)$. By the above lemma, we can represent the fractional part of an element of Λ_v as $\chi_v(s)$ in a unique way. Thus, any element of Λ_v has a unique representation as $\mathit{poly} + \chi_v(s)$ with polynomial part *poly* and the fractional part $\chi_v(s)$.

Definition 3.5. A pair of a polynomial vector *poly* and a state $s \in S$ is called the state representation of $\mathit{poly} + \chi_v(s) \in \Lambda_v$.

The addition of two representations is given by adding their polynomial parts, and by adding the states in the state space. Multiplication by t is given by

$$t(\mathit{poly} + \chi_v(s)) = (t \cdot \mathit{poly} + o_v(s)) + \chi_v(f(s)).$$

In applying SGR to (2.2), note that $e_1, \ldots, e_v$ are not in the image of χ_v, but once such a vector is reduced, then the result has only the fractional part, having a representation $\chi_v(s)$. Thus, most computation can be done inside the state space.

There is a slightly improved version. In a lattice-reduction procedure, we need to compute the norm and the leading term of $\chi_v(s)$. A direct method is to compute $o_v(s), o_v(f(s)), o_v(f^2(s)), \ldots$ in this order, until one gets a nonzero vector. If $o_v(f^j(s))$ is the first nonzero vector, then this vector is the leading coefficient $\pi(\chi_v(s))$ and $||\chi_v(s)|| = -j-1$ holds. This method is time-consuming if the norm is small, which is the case for the last steps of the reduction.

To avoid this, we adopted the following representation. Let s be a nonzero state that represents a lattice element $\chi_v(s)$. If $||\chi_v(s)|| = -n$, then we keep the pair $(n-1, f^{n-1}(s))$ as a representation of $\chi_v(s)$, instead of s. More precisely, consider the set $\tilde{S} := \{(m, s') \in \mathbf{Z} \times S \mid m \geq 0, ||\chi_v(f^{-m}(s'))|| = -m-1\}$. The above mapping $s \mapsto (n-1, f^{n-1}(s)) \in \tilde{S}$ gives the inverse to the mapping $\phi : \tilde{S} - \{0\} \to S - \{0\}; (m, s') \mapsto f^{-m}(s')$. Through this bijection, we use elements in $\tilde{S}$ as representations of lattice elements. It is easy to check that $||\chi_v(\phi(m, s'))|| = -m-1$ and $\pi(\chi_v(\phi(m, s'))) = o_v(s')$, so there is no need to search for the first nonzero term. One can check that in the reduction steps in SGR, we need only the norm and the leading term, hence this representation works. We leave it as an exercise to detail how to compute the sum and the multiplication by t in $\tilde{S}$.

We propose a combination of SGR, inductive projection, and state representation for computing all $k(v)$'s, which we call *SIS* for short. Thus, first SIS computes a reduced basis of Λ_w using SGR with state representation. By taking the projection, SIS computes a generating set of Λ_{w-1}, then reduces it to a reduced basis by SGR with state representation. SIS inductively computes reduced bases of Λ_w, Λ_{w-1}, Λ_{w-2}, …, Λ_2, in this order (inductive projection). Theorem 2.3 gives $k(v)$ for $v = w, w-1, \ldots, 2$.

4. Computational complexities

In a practical $\mathbf{F}_2$-linear generator, f and o_v can be computed by a few operations, often independently of the size of the state space, which we assume is negligible from the total cost of the computation.

Theorem 4.1. *The average number of bit operations to obtain the reduced basis by the SGR from the generating set in* (2.2) *is bounded by* $(v+1)\dim(S)^2 + (v^3 + v^2)\dim(S)$*, when using the state representation.*

Proof. One step of the reduction in SGR consists of v^3 bit operations for Gaussian elimination to find a linear relation among $v+1$ $\mathbf{F}_2$ vectors, and v additions to reduce a vector. Each addition requires $\dim(S)$ bit operations in the state representation. Thus, one reduction step has $v\dim(S) + v^3$ bit operations. By Theorem 3.1 and Corollary 3.2, on average, the number of reduction steps does not exceed

$$||\chi_v(s_0)|| + ||e_1|| + \cdots + ||e_v|| - \sum_{i=1}^{v} ||\omega_i'|| + \dim(S)/v + 1.$$

From Lemma 2.4, $||\chi_v(s_0)|| \leq -1$ and $||e_i|| = 0$, it follows that this bound is equal to $\dim(S)(1 + 1/v)$. By multiplying, we have a complexity upper bound $\dim(S)(1+1/v)(v\dim(S) + v^3) = (v+1)\dim(S)^2 + (v^3+v^2)\dim(S)$. □

Theorem 4.2. *The average number of bit operations for an SGR algorithm to obtain a reduced basis of* Λ_v *from that of* Λ_{v+1} *has an upper bound* $\dim(S)^2 + (v^2 + v)\dim(S) + v^3$*, when we use the state representation.*

Proof. By Corollary 3.2, SGR needs at most $\dim(S)/v + 1$ reduction steps on average. As in the proof of Theorem 4.1, each reduction step has $v\dim(S) + v^3$ bit operations, hence we have $(\dim(S)/v + 1)(v\dim(S) + v^3) = \dim(S)^2 + (v^2 + v)\dim(S) + v^3$. □

These theorems give an upper bound of computational complexity of SIS. Theorem 4.1 implies that the first step of SIS computing requires at most $w\dim(S)^2 + w^3\dim(S)$ bit operations. At the step of the inductive projection from Λ_v to Λ_{v-1} in SIS, Theorem 4.2 gives an upper bound of the complexity $\dim(S)^2 + v^2\dim(S) + v^3$. By summing for $v = w-1, w-2, \ldots, 1$, $w\dim(S)^2 + \frac{1}{3}w^3\dim(S) + \frac{1}{4}w^4$ bit operations will suffice to compute the other $w-1$ values $k(w-1), k(w-2), \ldots, k(1)$. By summing, we have:

Theorem 4.3. *SIS requires at most* $2w\dim(S)^2 + \frac{4}{3}w^3\dim(S) + \frac{1}{4}w^4$ *bit operations to compute all* $k(v)$*,* $w \geq v \geq 1$*.*

We compare this complexity to the following result for the dual lattice method described in §2.3. A lattice Λ_v is said to be *regular* if the minimum and the maximum of its successive minima have a difference of at most 1.

Theorem 4.4 ([1, Theorem 2 and §4])**.** *Suppose that* Λ_{v-1}' *is regular. The number of bit operations for computing a reduced basis of* Λ_v' *from that of* Λ_{v-1}' *does not exceed*

$$Cv(\dim(S) + v - 1)^2, \quad v \geq 2,$$

where C *is an absolute constant.*

The number of bit operations for computing all $k(1),\ldots,k(w)$ *does not exceed* $C'\frac{w^2}{2}(\dim(S)+w-1)^2$ *for an absolute constant* C'*, under the regularity assumption for each lattice* Λ'_v.

The comparison of the orders show that our SIS method is expected to be more efficient than this by a factor of w. (As pointed out by a referee, strictly speaking, these are only upper bounds and do not compare the efficiency). Note that there are differences in the estimation: our estimation does not assume the regularity on the lattices, but does depend on a heuristic argument on the average. Actually, the regularity implies that our estimation in Corollary 3.2 plus 1 gives an upper bound as a worst case analysis, since $||\omega^{\mathrm{last}}|| \geq \nu_1 \geq -\dim(S)/v - 1$ is regular.

Remark 4.5. We are also interested in whether SGR is more efficient than Lenstra's algorithm or not, when used for dual lattice. According to our experiments, the answer is yes, but not that much, see the next section. We implemented a version of the dual lattice method, replacing the Lenstra algorithm with SGR.

There is one caution when SGR is used with the dual lattice method: to keep the efficiency, we need a triangulation process, as stated below. In the dual lattice method, let $\omega_1,\ldots,\omega_{v-1}$ be the computed reduced basis of Λ'_{v-1}. Let B be the square matrix of size $v-1$ whose j-th column is ω_j. As explained after Theorem 2.6, the vector ${}^t(-\bar{g}_v(t),0,\cdots,0,1)$ is reduced by using $\iota(\omega_1),\ldots,\iota(\omega_{v-1})$ (called the first phase, see [1, Proof of Theorem 2]), then the Lenstra algorithm is applied to obtain a reduced basis of Λ'_v (the second phase). Let $\pi(B)$ be the square matrix whose j-th column is $\pi(\omega_j)$. In the first phase, in one reduction step, a linear equation $\pi(B)x = y$ with coefficients in $\mathbf{F}_2$ is solved, until the vector becomes non-reducible by $\iota(\omega_1),\ldots,\iota(\omega_{v-1})$ (where y may change at every step). If $\pi(B)$ happens to be triangular, then solving these linear equations is efficient.

If B is obtained by the Lenstra algorithm, then $\pi(B)$ is triangular. If B is obtained by SGR, $\pi(B)$ may be not triangular, but it is easy to transform B to another reduced basis B' with $\pi(B')$ being triangular. We use SGR with this triangulation procedure, then the dual lattice method with SGR is often faster than that using the Lenstra algorithm.

5. Speed comparison

The following computer experiments compare our SIS method and the dual lattice method. We assume $w = 32$, and choose three $\mathbf{F}_2$-linear generators. We measured the CPU time for computing each $k(v)$ for $2 \leq v \leq 32$, by using the following three methods:

(1) SIS method (our proposal in §4).
(2) the dual lattice method with
 (a) SGR algorithm applied as in Remark 4.5.
 (b) Lenstra reduction algorithm (the method proposed in [1]).

We implemented these three methods in C++. In the SIS method, we compute $k(32)$ from (2.2), and then $k(31),\ldots,k(2)$ inductively. In the dual lattice method, we need to compute the characteristic polynomial $P(t)$ by the Berlekamp-Massey algorithm [10], and then transform (2.3) into (2.4) by arithmetic operations modulo $P(t)$. We refer to this phase as the *precomputation* for the dual lattice method. For polynomial arithmetic in the precomputation, we used the library NTL (`http://www.shoup.net/ntl`). We compute $k(2)$ by applying SGR algorithm (a), or Lenstra's algorithm (b), to the basis (2.4), and then inductively compute $k(3), \ldots, k(32)$. The tests were performed on a computer with 64-bit AMD-Athlon 64 3200+ CPU and 2.0 GB of memory, on a Linux operating system. The programs were compiled using gcc compiler version 4 with the -O2 optimization flag.

We applied these methods to WELL44497a', which is a maximally equidistributed version of WELL44497a [12] by improving its tempering (see [4]). The generator has $\dim(S) = 44497$. The lattice Λ_v turns out to be regular for every $2 \leq v \leq 32$, except for $v = 7$. Table 1 gives the CPU time (in seconds) for computing a reduced basis. As predicted from the computational complexities (Theorems 4.1 and 4.2), in the SIS method, the CPU time for computing $k(32)$ is comparable to the sum of all the rest of the computations. Note that the consumed time for $k(31)$ to $k(2)$ is almost the same, as predicted from Theorem 4.2. In the dual lattice methods, computation of $k(2)$ is fast, and computation time of $k(v)$ increases, roughly proportional to v, as predicted from Theorem 4.4. In these experiments, SGR is a little faster than Lenstra's algorithm.

The comparison of the CPU time is in accordance with the ratio v between the complexity of our method and that of the dual lattice method, in inductive computation of $k(v)$. In total, our method is much faster.

We also compared the timings for two other $\mathbf{F}_2$-linear generators with $\dim(S) = 19937$, namely WELL19937a' (a maximally equidistributed version of WELL19937a [12] introduced in [4], Λ_v being regular except for $v = 6$) and Mersenne Twister MT19937 [11] whose lattices are far from being regular (whose total dimension defect [12] Δ is 6750). Table 2 lists the total CPU time (in seconds) to compute all $k(v)$ $(2 \leq v \leq 32)$ by the three methods for the three generators. The first line lists the total time for each method applied for WELL44497a', copied from the last line of Table 1. The experiments on WELL19937a' show the same tendency. In MT19937, Lenstra's algorithm is faster than SGR.

Table 1. The CPU time for computing $k(v)$ ($2 \leq v \leq 32$) of WELL44497a' (in seconds). For the SIS methods, they are listed in descending order with respect to v, according to the order of computation.

SIS		dual lattice		
	SGR		SGR	Lenstra
		precom.	1.829	1.845
$k(32)$	1.996	$k(2)$	0.064	0.064
$k(31)$	0.059	$k(3)$	0.138	0.140
$k(30)$	0.058	$k(4)$	0.212	0.217
$k(29)$	0.058	$k(5)$	0.286	0.298
$k(28)$	0.059	$k(6)$	0.364	0.379
$k(27)$	0.057	$k(7)$	0.445	0.468
$k(26)$	0.057	$k(8)$	0.527	0.553
$k(25)$	0.057	$k(9)$	0.612	0.646
$k(24)$	0.058	$k(10)$	0.700	0.739
$k(23)$	0.056	$k(11)$	0.791	0.836
$k(22)$	0.057	$k(12)$	0.879	0.940
$k(21)$	0.057	$k(13)$	0.977	1.047
$k(20)$	0.057	$k(14)$	1.071	1.157
$k(19)$	0.056	$k(15)$	1.183	1.266
$k(18)$	0.057	$k(16)$	1.284	1.383
$k(17)$	0.058	$k(17)$	1.386	1.491
$k(16)$	0.058	$k(18)$	1.505	1.619
$k(15)$	0.051	$k(19)$	1.623	1.761
$k(14)$	0.052	$k(20)$	1.742	1.895
$k(13)$	0.053	$k(21)$	1.857	2.008
$k(12)$	0.053	$k(22)$	1.985	2.148
$k(11)$	0.054	$k(23)$	2.114	2.288
$k(10)$	0.050	$k(24)$	2.247	2.468
$k(9)$	0.051	$k(25)$	2.365	2.595
$k(8)$	0.052	$k(26)$	2.516	2.739
$k(7)$	0.048	$k(27)$	2.682	2.929
$k(6)$	0.050	$k(28)$	2.824	3.098
$k(5)$	0.049	$k(29)$	2.969	3.248
$k(4)$	0.049	$k(30)$	3.117	3.443
$k(3)$	0.048	$k(31)$	3.308	3.630
$k(2)$	0.048	$k(32)$	3.456	3.806
total	3.622	total	49.053	53.139

Table 2. The cumulative CPU-time (in seconds) for computation of all $k(v)$ ($2 \leq v \leq 32$) of three $\mathbf{F}_2$-linear generators, by the SIS method and the dual lattice methods. The number in () shows the pre-computation time. The column Δ shows the total dimension defect.

generators	SIS	dual lattice		Δ
		SGR	Lenstra	
WELL44497a'	3.622	49.053(1.829)	53.139(1.845)	0
WELL19937a'	0.939	12.360(0.398)	13.476(0.392)	0
MT19937	0.529	9.399(0.403)	5.654(0.408)	6750

Acknowledgement

The authors are thankful to the anonymous referees for many useful comments.

References

1. R. Couture and P. L'Ecuyer, *Lattice computations for random numbers*, Math. Comput. **69** (2000), no. 230, 757–765. MR1651748 (2000i:11125)
2. R. Couture, P. L'Ecuyer, and S. Tezuka, *On the distribution of k-dimensional vectors for simple and combined tausworthe sequences*, Math. Comput. **60** (1993), 749–761. MR1176708 (93h:11085)
3. M. Fushimi and S. Tezuka, *The k-distribution of generalized feedback shift register pseudorandom numbers*, Commun. ACM **26** (1983), no. 7, 516–523.
4. S. Harase, *Maximally equidistributed pseudorandom number generators via linear output transformations*, Math. Comput. Simul. **79** (2009), no. 5, 1512–1519. MR2488100 (2010a:65012)
5. P. L'Ecuyer and R. Couture, *An implementation of the lattice and spectral tests for multiple recursive linear random number generators*, INFORMS Journal on Computing **9** (1997), no. 2, 206–217. MR1477315
6. P. L'Ecuyer and F. Panneton, *$\mathbf{F}_2$-linear random number generators*, Advancing the Frontiers of Simulation: A Festschrift in Honor of George Samuel Fishman (C. Alexopoulos, D. Goldsman, and J. R. Wilson, eds.), Springer-Verlag, 2009, pp. 169–193.
7. A. K. Lenstra, *Factoring multivariate polynomials over finite fields*, Journal of Computer and System Sciences **30** (1985), no. 2, 235 – 248. MR801825 (87a:11124)
8. K. Mahler, *An analogue to Minkowski's geometry of numbers in a field of series*, The Annals of Mathematics **42** (1941), no. 2, 488–522. MR0004272 (2:350c)
9. ———, *On a theorem in the geometry of numbers in a space of Laurent series*, Journal of Number Theory **17** (1983), no. 3, 403–416. MR724538 (85e:11043)
10. J. L. Massey, *Shift-register synthesis and BCH decoding*, IEEE Trans. Inform. Theory **IT-15** (1969), 122–127. MR0242556 (39:3887)
11. M. Matsumoto and T. Nishimura, *Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator*, ACM Trans. Model. Comput. Simul. **8** (1998), no. 1, 3–30.
12. F. Panneton, P. L'Ecuyer, and M. Matsumoto, *Improved long-period generators based on linear recurrences modulo* 2, ACM Trans. Math. Softw. **32** (2006), no. 1, 1–16. MR2272349 (2007h:94033)
13. S. Paulus, *Lattice basis reduction in function fields.*, Algorithmic Number Theory. Lecture Notes in Computer Science (Berlin), vol. 1423, Springer-Verlag, 1998. MR1726102 (2000i:11193)
14. W. M. Schmidt, *Construction and estimation of bases in function fields*, J. Number Theory **39** (1991), no. 2, 181 – 224. MR1129568 (93b:11079)
15. S. Tezuka, *The k-dimensional distribution of combined GFSR sequences*, Math. Comput. **62** (1994), no. 206, 809–817. MR1223233 (94i:65014)
16. L. Wang and H. Niederreiter, *Successive minima profile, lattice profile, and joint linear complexity profile of pseudorandom multisequences*, J. Complex. **24** (2008), no. 2, 144–153. MR2400313 (2009d:94063)
17. L. Wang and Y. Zhu, *$F[x]$-lattice basis reduction algorithm and multisequence synthesis*, Sci. in China Ser. F **44** (2001), 321–328. MR1895107 (2003g:94031)
18. L. Wang, Y. Zhu, and D.-Y. Pei, *On the lattice basis reduction multisequence synthesis algorithm*, IEEE Trans. Inform. Theory **50** (2004), no. 11, 2905–2910. MR2097012

Graduate School of Mathematical Sciences, The University of Tokyo, Tokyo, Japan
E-mail address: `sharase@orange.ocn.ne.jp`

Graduate School of Mathematical Sciences, The University of Tokyo, Tokyo, Japan
E-mail address: `matumoto@ms.u-tokyo.ac.jp`

Department of Mathematics, Hiroshima University, Hiroshima, Japan
E-mail address: `saito@math.sci.hiroshima-u.ac.jp`