An efficient lattice reduction method for F_2-linear pseudorandom number generators using Mulders and Storjohann algorithm

Shin Harase (The University of Tokyo)

MCQMC2010 (Warsaw), August 16th, 2010

This work was partially supported by Grant-in-Aid for JSPS Fellows 21·4427.

Purpose

- The dimensions of equidistribution are important measure for the quality of pseudorandom number generators (PRNGs).

- In their computation, certain lattice reduction is used (Couture, L'Ecuyer, and Tezuka (1993) and Tezuka (1994)).

- We use Mulders and Storjohann lattice reduction algorithm (2003) and lessen the order of complexity for our previous work (Harase, Matsumoto and Saito, to appear in Math. Comp).

$\mathbf{F}_2$-linear generators

Let $\mathbf{F}_2 := \{0, 1\}$ be the two element field.
We only treat $\mathbf{F}_2$-linear generators as follows.

$S = \mathbf{F}_2^p$: a state space ($p = \dim(S)$)
(the possible states of the memory assigned for the PRNG).
$f : S \to S$: an $\mathbf{F}_2$-linear transition function.
$O = \mathbf{F}_2^w$: the set of outputs
(w is intended for the word size of the machine).
$o : S \to O$: an $\mathbf{F}_2$-linear output function.

For a given initial state $s_0 \in S$, the generator computes the next state by the following recursion

$$s_0, s_1 := f(s_0), s_2 := f(s_1), \ldots \in S$$

and the output sequence is given by

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

We identify the output set $O = \mathbf{F}_2^w$ with the set of unsigned w-bit binary integers.

The dimension of equidistribution (k-distribution property)

We recall the dimension of equidistribution as a criterion of uniformity.

Let v be $1 \leq v \leq w$.
Look only the most significant v bits (v MSBs) in the outputs (the outputs with v-bit accuracy). The corresponding output function is 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

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

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

Definition *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 equidistribution with* v*-bit accuracy, denoted by* $k(v)$*.*

If the generator has the maximal period $2^p - 1$ ($p = \dim(S)$),
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.

As a criterion of uniformity,
the larger $k(v)$ for each of $1 \le v \le w$ is desirable.
(An upper bound $k(v) \le \lfloor p/v \rfloor$.)

$\rightsquigarrow$ Problem: compute $k(v)$ for all $1 \le v \le w$ efficiently.

Lattice structure

To compute $k(v)$, the lattice methods are known.

Let K denote the (negative) formal power series field:

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

Example.

$$t^2 + 1 + t^{-3} + t^{-4} + t^{-6} + \cdots \in K.$$

For $\alpha = \sum_{j=j_0}^{\infty} a_j t^{-j} \in K = \mathbf{F}_2((t^{-1}))$,
we define the length $|\alpha|$ by the degree of the highest degree term

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

For a vector $\gamma = (\alpha_1, \alpha_2, \ldots, \alpha_v) \in K^v$,
we define the length $||\gamma|| := \max_{1 \leq i \leq v} |\alpha_i|$.

Example. We consider the case for $v = 3$.
Let

$$\begin{aligned} \alpha_1 &= t^{-2} + t^{-4} \cdots, \\ \alpha_2 &= t^{-1} + t^{-3} + t^{-4} \cdots, \\ \alpha_3 &= t^{-3} + t^{-4} \cdots, \end{aligned}$$

and $\gamma = (\alpha_1, \alpha_2, \alpha_3)$.

We have the lengths $|\alpha_1| = -2, |\alpha_2| = -1$, and $|\alpha_3| = -3$.
Also the length $||\gamma|| = \max_{1 \leq i \leq 3} |\alpha_i| = -1$.
We say that α_2 is longer than α_1 and α_3.

For a nonzero $\gamma = (\alpha_1, \alpha_2, \ldots, \alpha_v) \in K^v$,
we define its coefficient vector at the leading term $\pi(\gamma) \in \mathbf{F}_2^v$ by

$$\gamma = \pi(\gamma) t^{||\gamma||} + \text{lower degree terms in } t.$$

Example (Continued).

$$\begin{aligned}
\alpha_1 &= 0 \cdot t^{-1} + 1 \cdot t^{-2} + 0 \cdot t^{-3} + 1 \cdot t^{-4} \cdots, \\
\alpha_2 &= 1 \cdot t^{-1} + 0 \cdot t^{-2} + 1 \cdot t^{-3} + 1 \cdot t^{-4} \cdots, \\
\alpha_3 &= 0 \cdot t^{-1} + 0 \cdot t^{-2} + 1 \cdot t^{-3} + 1 \cdot t^{-4} \cdots,
\end{aligned}$$

and $\gamma = (0,1,0) \cdot t^{-1} + (1,0,0) \cdot t^{-2} + (0,1,1) \cdot t^{-3} + (1,1,1) \cdot t^{-4} \cdots$.
$||\gamma|| = -1$, so that $\pi(\gamma) = (0,1,0)$.

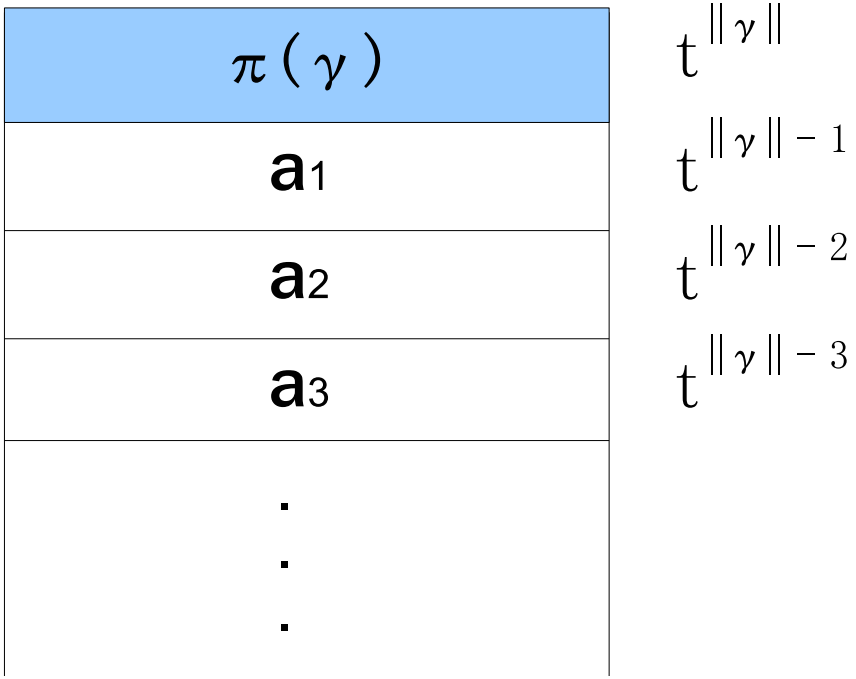
γ is denoted as follows:

$$\begin{array}{ll}
(0,1,0) & t^{-1} \\
(1,0,0) & t^{-2} \\
(0,1,1) & t^{-3} \\
(1,1,1) & t^{-4} \\
\vdots &
\end{array}$$

In general, a (v-dimensional) nonzero vector

$$\gamma = \pi(\gamma)t^{||\gamma||} + \mathbf{a}_1 t^{||\gamma||-1} + \mathbf{a}_2 t^{||\gamma||-2} + \mathbf{a}_3 t^{||\gamma||-3} + \cdots \in K^v,$$

with $\pi(\gamma) \in \mathbf{F}_2^v$ and $\mathbf{a}_i \in \mathbf{F}_2^v \quad (i = 1, 2, 3, \ldots)$ is picturized as follows:



Later, we use this picture to explain our new lattice reduction algorithm.

Let $L \subset K^v$ be an $\mathbf{F}_2[t]$-lattice,
$\overset{\text{def}}{\iff}$ there exist K-linearly independent vectors $\omega_1, \ldots, \omega_v \in K^v$
such that the $\mathbf{F}_2[t]$-linear span $\langle \omega_1, \ldots, \omega_v \rangle_{\mathbf{F}_2[t]}$ is L.

Such a set of vectors is called a basis of L.

A basis $\{\omega_1, \ldots, \omega_v\}$ is said to be a reduced basis of L
if $\pi(\omega_1), \ldots, \pi(\omega_v)$ are linearly independent over $\mathbf{F}_2$.

Sort a reduced basis $\{\omega_1, \omega_2, \ldots, \omega_v\}$ so that

$$||\omega_1|| \leq ||\omega_2|| \leq \ldots \leq ||\omega_v||.$$

Then, $\nu_1 := ||\omega_1||, \ldots, \nu_v := ||\omega_v||$ are called successive minima of L(Mahler, 1941).

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:

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

We define a (resolution-wise) lattice Λ_v of K^v as

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

where $e_1, \ldots, e_v$ are the unit vectors. Note that $\{e_1, e_2, \ldots, e_v, \chi_v(s_0)\}$ is not a basis but a generating set. Still we can prove that Λ_v is an $\mathbf{F}_2[t]$-lattice.

Theorem ((Couture et al, 1993, Tezuka, 1994)) *Assume that the characteristic polynomial of f is irreducible. Take nonzero $s_0 \in S$. Then, $k(v) = -\nu_v$ holds, where ν_v is the v-th successive minimum of Λ_v (i.e., the length of the longest vector in a reduced basis).*

Fast lattice reduction for F_2-linear PRNGs (H-M-S, to appear in Math. Comp.)

To compute a reduced basis of Λ_v for all $1 \leq v \leq w$, Harase et al. proposed the following SIS method:

- Uses a modification of Wolfgang M. Schmidt's algorithm (1991) for computing a reduced basis.
- Computes $k(v)$ for $v = w, w-1, w-2, \ldots$, inductively (Inductive projection).
- Uses the state space S to represent vectors with components in the formal power series (State representation).

These are motivated by the dual lattice method (Couture and L'Ecuyer, 2000).

Our proposal

In this talk, we propose to replace Schmidt algorithm in (H-M-S, Math. Comp.) with Pivot index reduction algorithm by (Mulders and Storjohann, 2003).

For computing a reduced basis of Λ_v,
upper bounds for the number of bit operations are:

$$\begin{aligned} \text{Schmidt:} \quad & (v+1)\dim(S)^2 \ + (v^3+v^2)\dim(S), \\ & \downarrow \\ \text{Pivot index reduction:} \quad & (v+1)\dim(S)^2 \ + \frac{1}{2}v^2(v+1), \end{aligned}$$

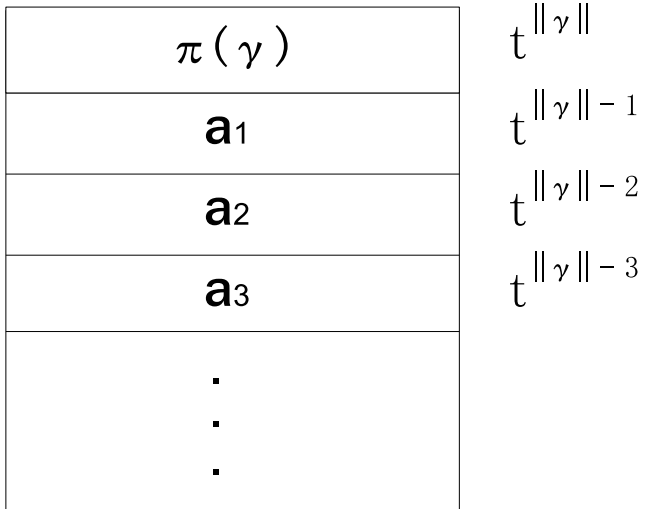
under a mild assumption.

Mulders and Storjohann algorithm

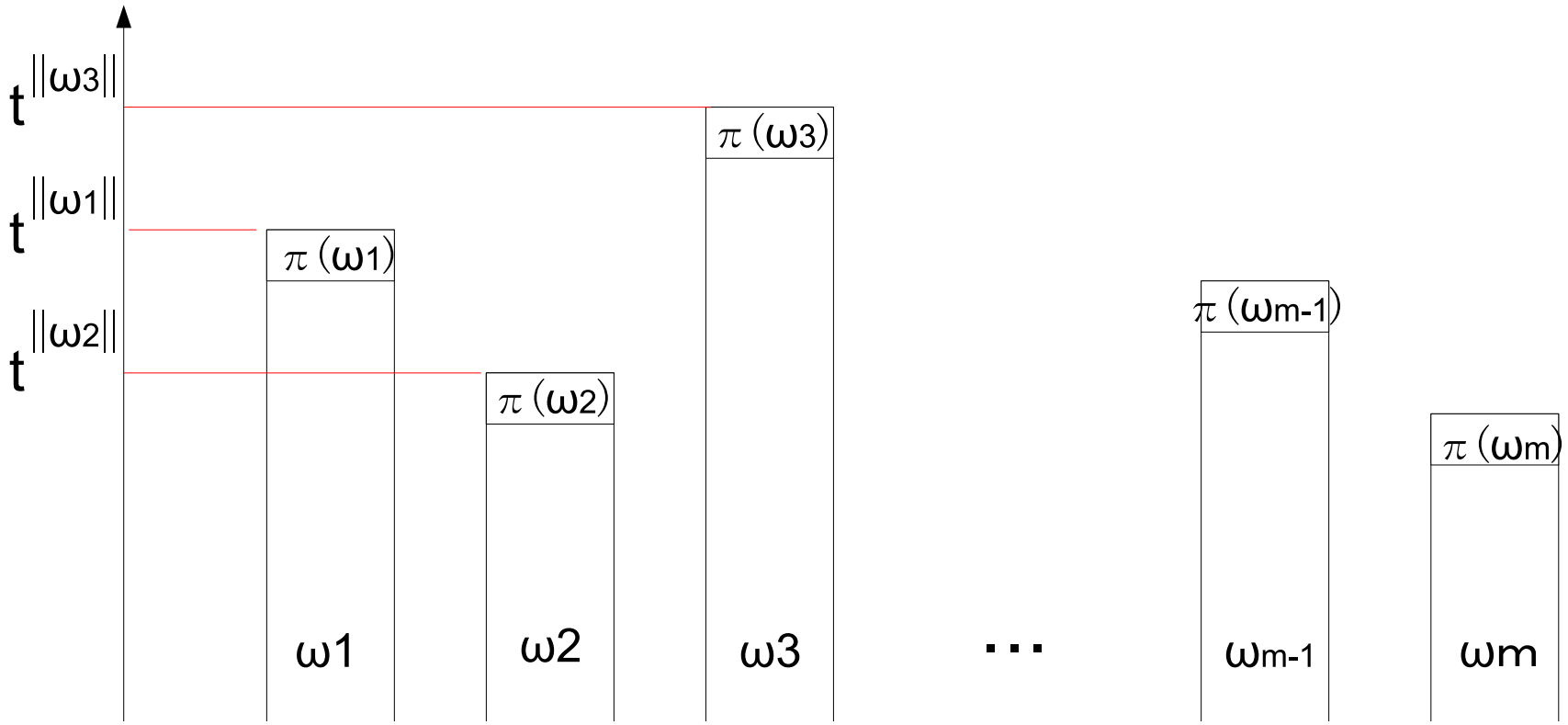
We consider an $\mathbf{F}_2[t]$-lattice $L \subset K^v$.
Input: a generating set of L, namely, $\omega_1, \omega_2, \ldots, \omega_m \in L$.
Output: a reduced basis of L.

Recall the following picture for a nonzero vector

$$\gamma = \pi(\gamma)t^{||\gamma||} + \mathbf{a}_1 t^{||\gamma||-1} + \mathbf{a}_2 t^{||\gamma||-2} + \mathbf{a}_3 t^{||\gamma||-3} + \cdots \in K^v :$$

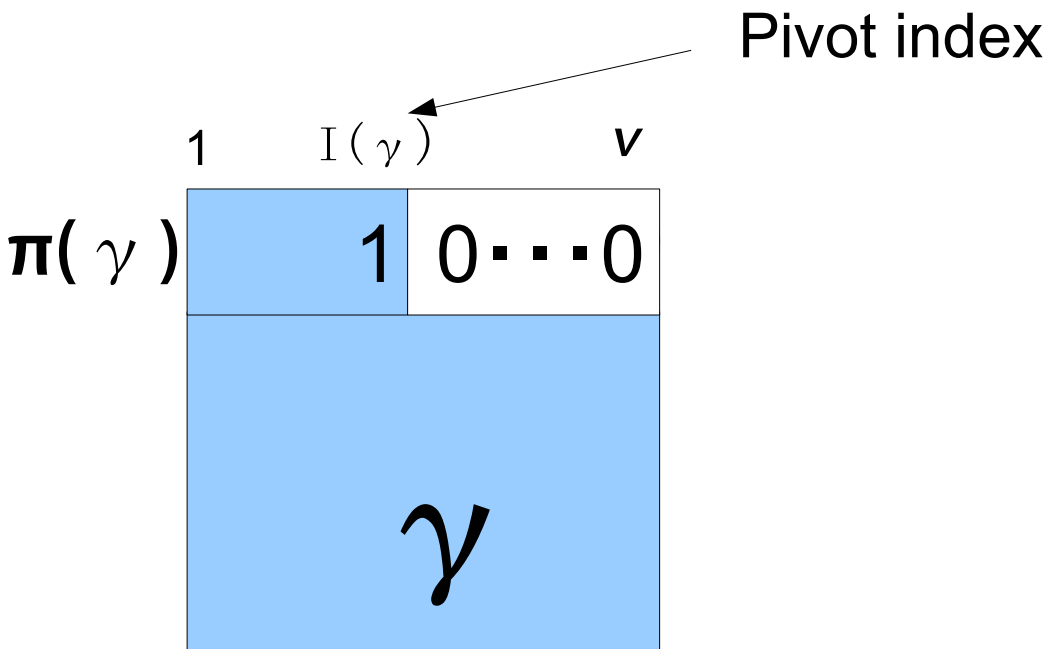


The following boxes represent a generating set $\omega_1, \omega_2, \ldots, \omega_m$.
The heights represent the lengths $||\omega_1||, ||\omega_2||, \ldots, ||\omega_m||$.
The top rows are the leading terms $\pi(\omega_1), \pi(\omega_2), \ldots, \pi(\omega_m)$.



The definition of the pivot index

For a nonzero vector $\gamma = (\alpha_1, \ldots, \alpha_v) \in K^v$,
we define the pivot index of γ (denoted by $I(\gamma)$) by the coordinate index of the rightmost nonzero component in the leading term $\pi(\gamma)$.

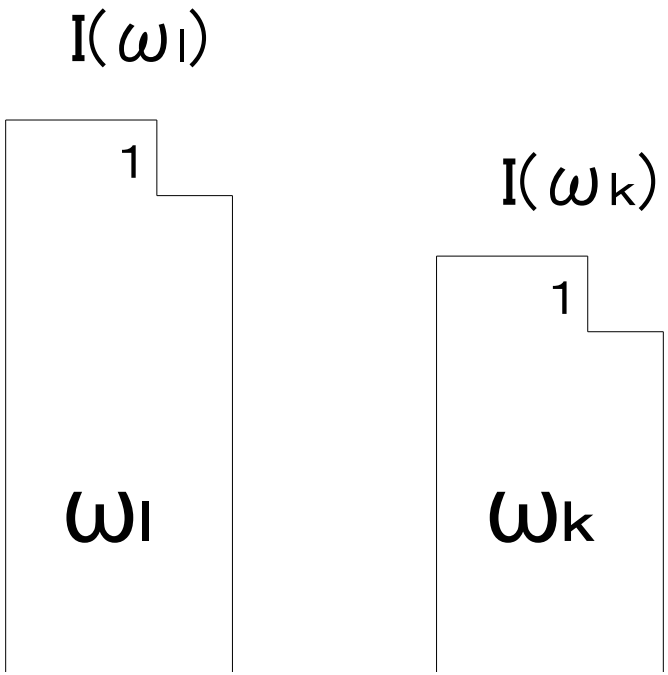


For $\gamma = (0, \ldots, 0)$, let $I(\gamma) := 0$.

Definition *For nonzero* ω_k *and* ω_l $(k \neq l)$*, assume*

$$I(\omega_l) = I(\omega_k) \text{ and } ||\omega_l|| \geq ||\omega_k||.$$

Pivot index reduction of ω_l *by* $\omega_k \overset{\text{def}}{\Longleftrightarrow}$ *replace* ω_l *with* $\omega_l' := \omega_l - \omega_k \cdot t^{||\omega_l|| - ||\omega_k||}$.

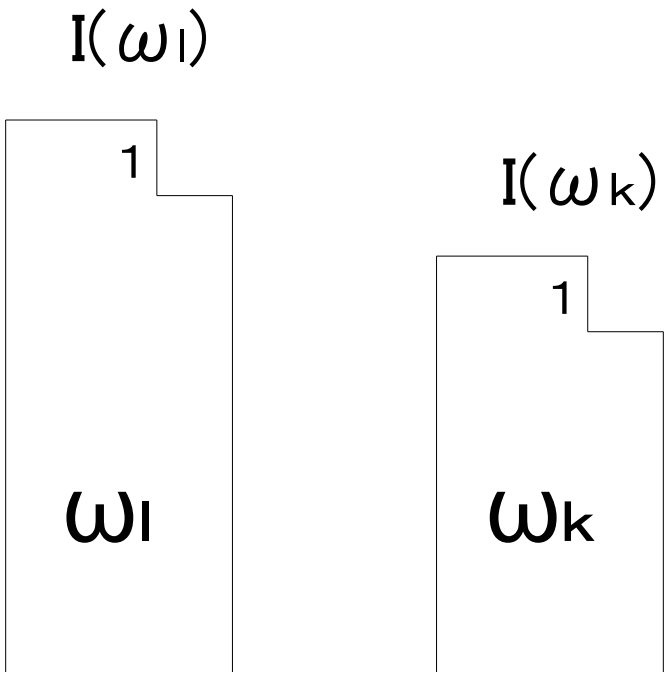


A pair of $(||\omega_l'||, I(\omega_l')) < (||\omega_l||, I(\omega_l))$ decreases in the lexicographic order.

Definition *For nonzero* ω_k *and* ω_l $(k \neq l)$*, assume*

$$I(\omega_l) = I(\omega_k) \text{ and } ||\omega_l|| \geq ||\omega_k||.$$

Pivot index reduction of ω_l *by* $\omega_k \overset{\text{def}}{\Longleftrightarrow}$ *replace* ω_l *with* $\omega_l' := \omega_l - \omega_k \cdot t^{||\omega_l|| - ||\omega_k||}$.

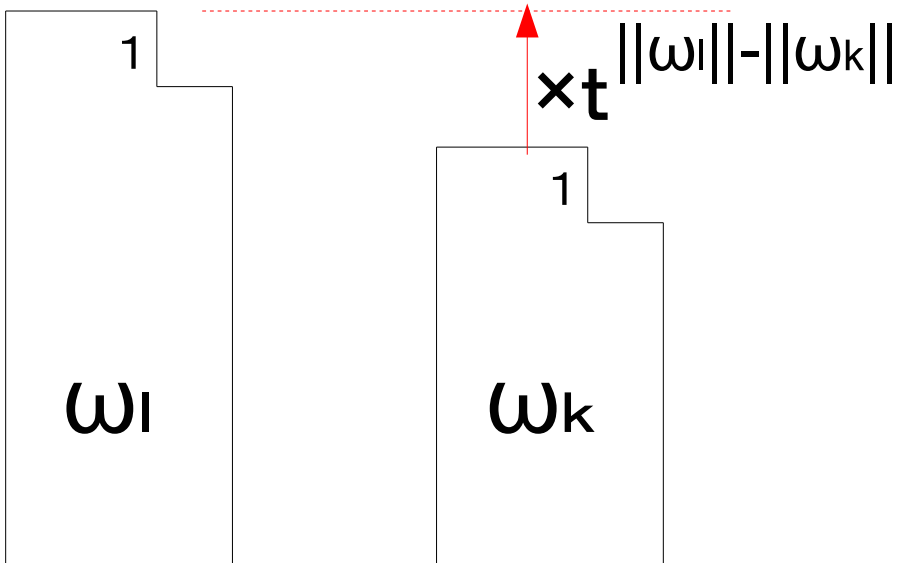


A pair of $(||\omega_l'||, I(\omega_l')) < (||\omega_l||, I(\omega_l))$ decreases in the lexicographic order.

Definition *For nonzero* ω_k *and* ω_l $(k \neq l)$*, assume*

$$I(\omega_l) = I(\omega_k) \text{ and } ||\omega_l|| \geq ||\omega_k||.$$

Pivot index reduction of ω_l *by* $\omega_k \overset{\text{def}}{\Longleftrightarrow}$ *replace* ω_l *with* $\omega_l' := \omega_l - \omega_k \cdot t^{||\omega_l|| - ||\omega_k||}$.

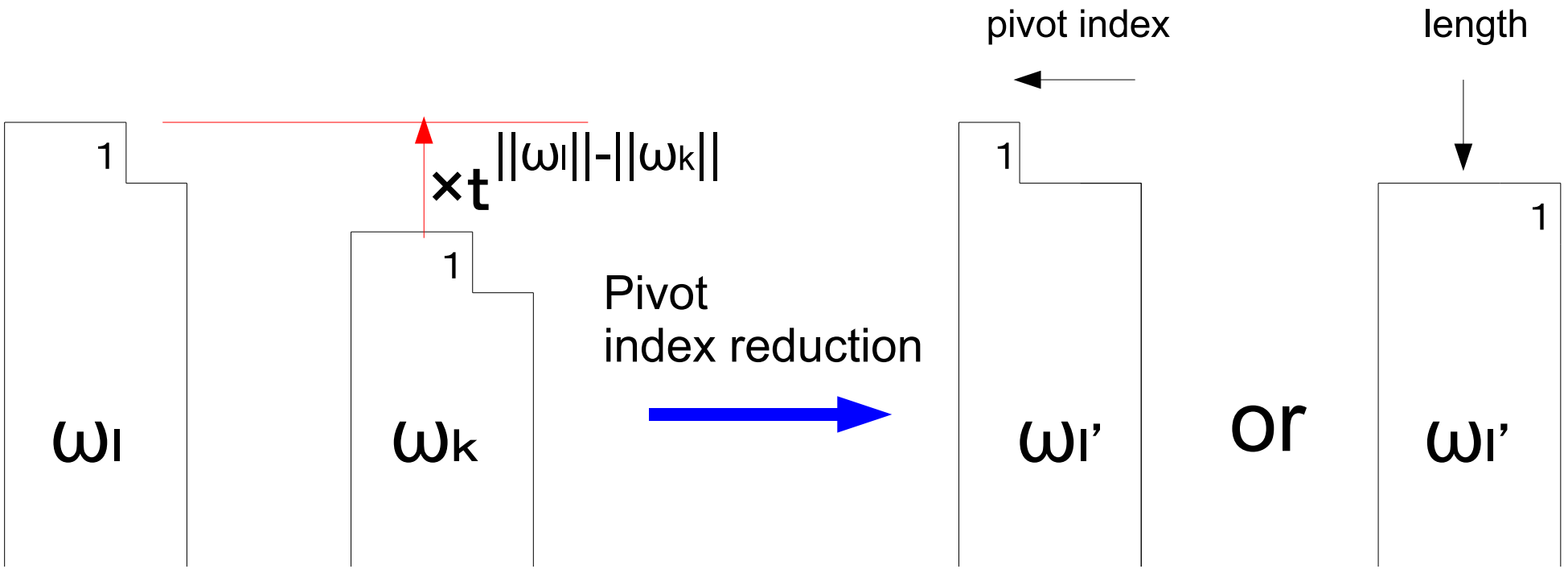


A pair of $(||\omega_l'||, I(\omega_l')) < (||\omega_l||, I(\omega_l))$ decreases in the lexicographic order.

Definition *For nonzero* ω_k *and* ω_l $(k \neq l)$*, assume*

$$I(\omega_l) = I(\omega_k) \text{ and } ||\omega_l|| \geq ||\omega_k||.$$

Pivot index reduction of ω_l *by* $\omega_k \overset{\text{def}}{\Longleftrightarrow}$ *replace* ω_l *with* $\omega_l' := \omega_l - \omega_k \cdot t^{||\omega_l|| - ||\omega_k||}$.



A pair of $(||\omega_l'||, I(\omega_l')) < (||\omega_l||, I(\omega_l))$ decreases in the lexicographic order.

Lemma *Assume that the nonzero pivot indices of* $\omega_1, \ldots, \omega_m$ *are all different. Then*

$$\{\omega_i \mid \omega_i \neq 0, i = 1, \ldots, m\}$$

is a reduced basis of L*.*

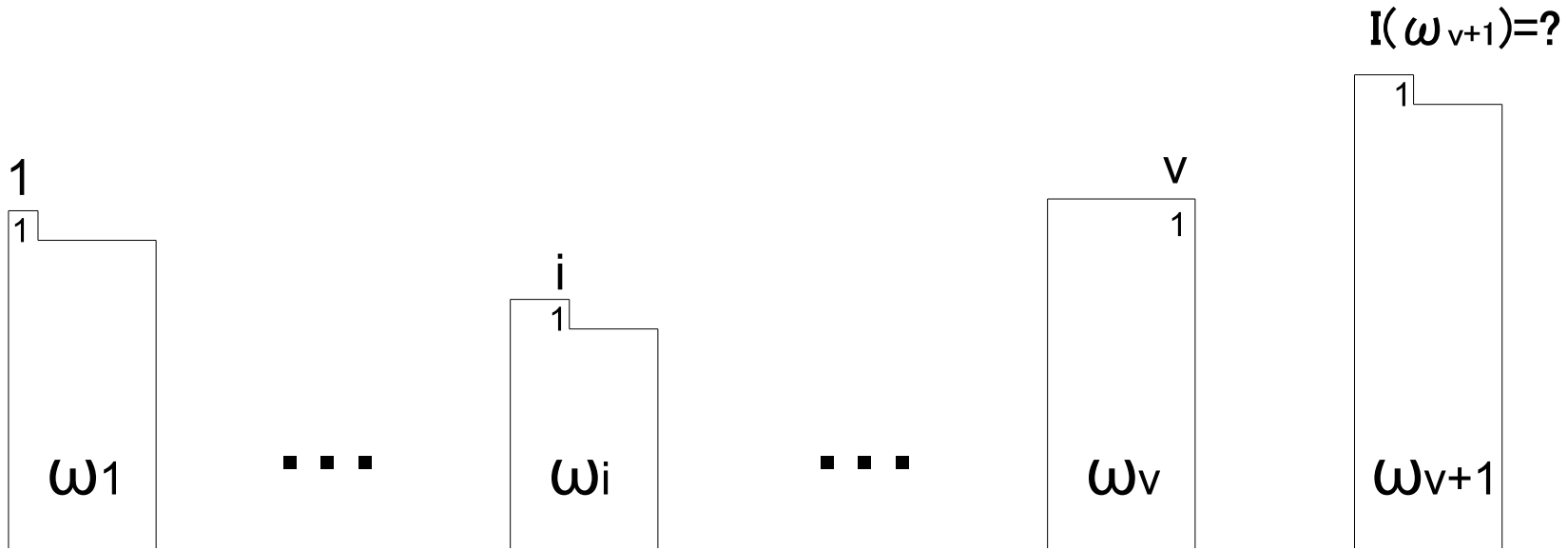
Mulders and Storjohann algorithm (2003) for a reduced basis is to iterate the pivot index reductions until the above condition holds.

(The algorithm terminates since $(||\omega_l'||, I(\omega_l')) \in \mathbf{Z}^2$ are bounded below.)

A variant of M-S algorithm by Wang, Zhu, and Pei (2004)

Let $L \subset K^v$ be an $\mathbf{F}_2[t]$-lattice.
Assume that a generating set $\{\omega_1, \ldots . \omega_v, \omega_{v+1}\}$ is given, satisfying the triangular condition:

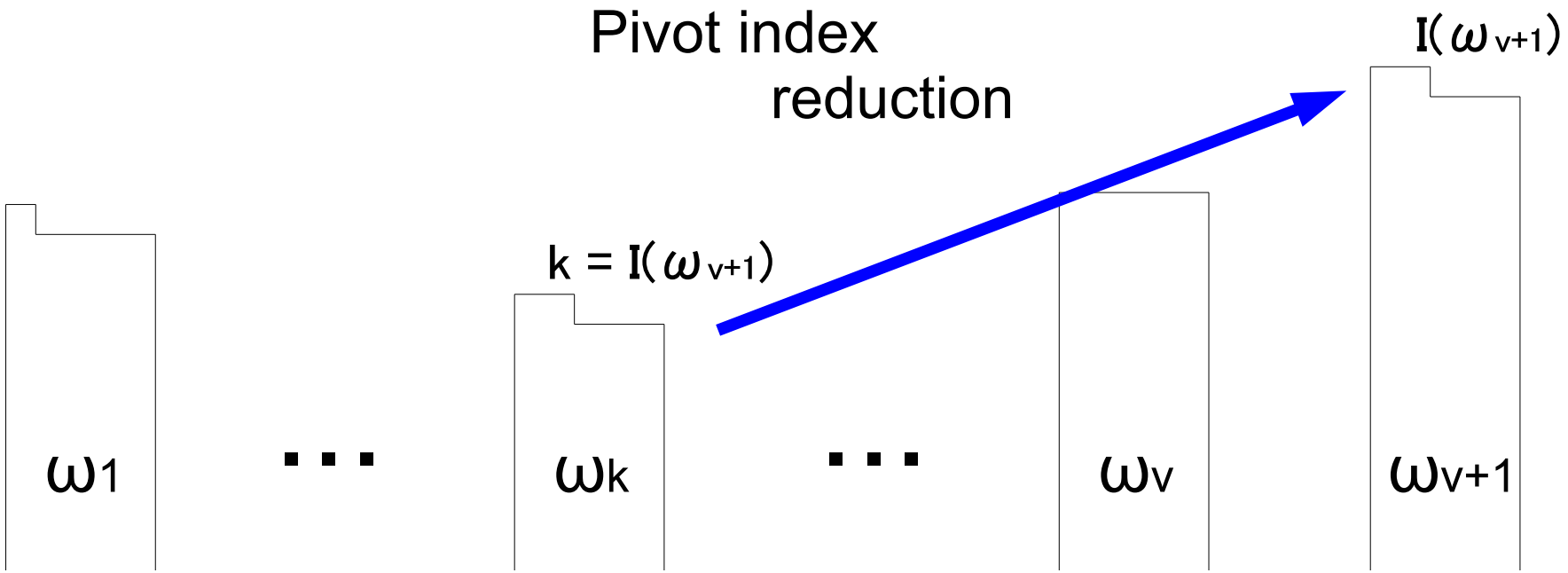
$$I(\omega_i) = i \text{ for } i = 1, \ldots, v.$$



Note that a generating set $\{e_1, \ldots, e_v, \chi_v(s_0)\}$ of Λ_v satisfies the triangular condition. There is a pivot index reduction that preserves this condition as follows.

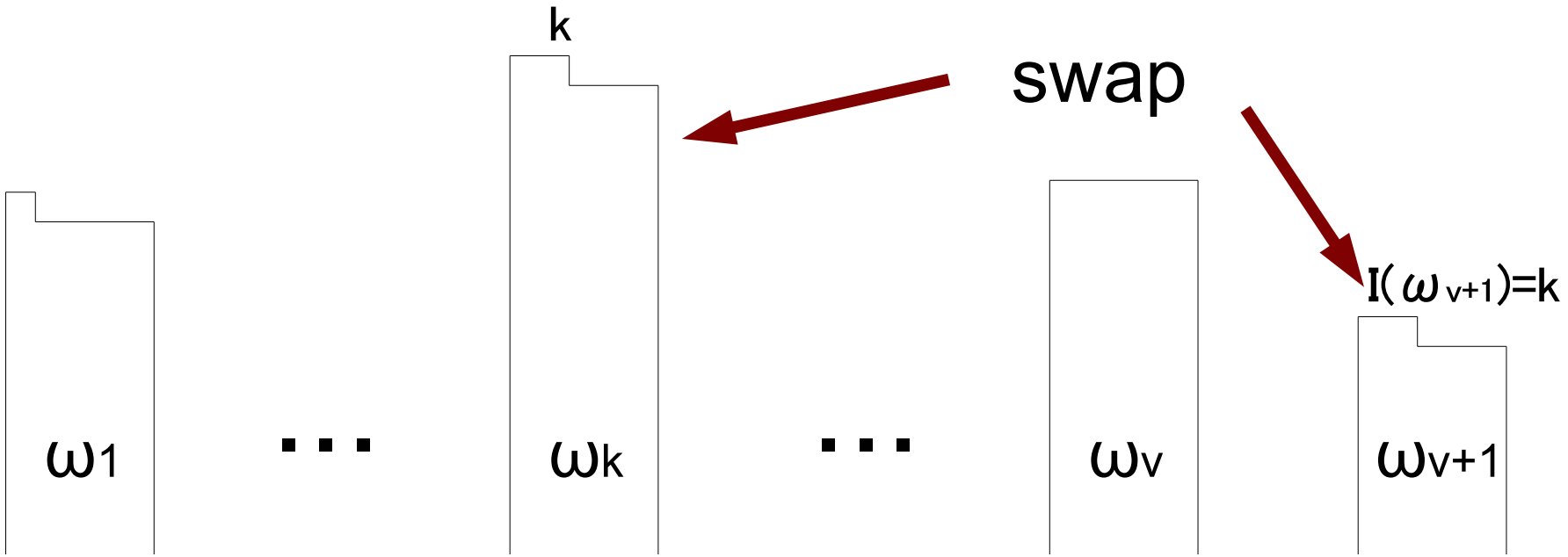
Find the pivot index $I(\omega_{v+1})$.
Set $k \leftarrow I(\omega_{v+1})$.

Case 1: if $||\omega_{v+1}|| \geq ||\omega_k||$ then
Apply the pivot index reduction of ω_{v+1} by ω_k.



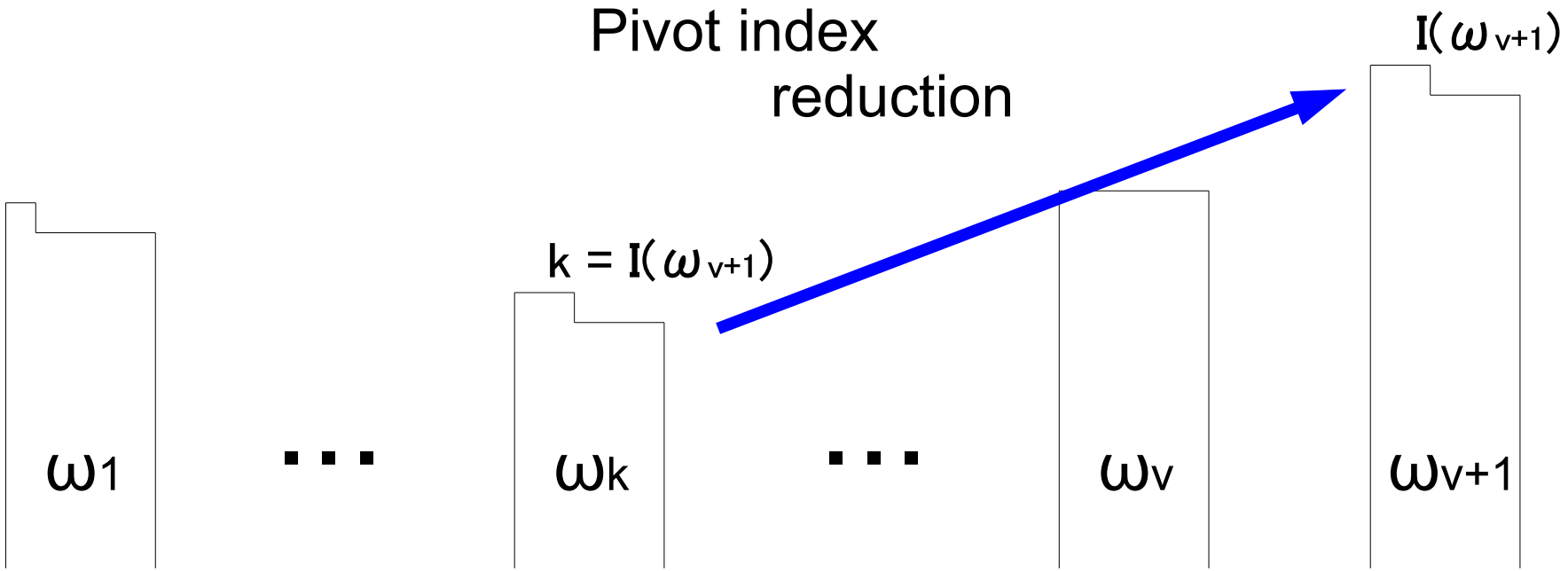
Find the pivot index $I(\omega_{v+1})$.
Set $k \leftarrow I(\omega_{v+1})$.

Case 2: if $||\omega_{v+1}|| < ||\omega_k||$ then
Swap ω_k and ω_{v+1}.
Apply the pivot index reduction of ω_{v+1} by ω_k.



Find the pivot index $I(\omega_{v+1})$.
Set $k \leftarrow I(\omega_{v+1})$.

Case 2: if $||\omega_{v+1}|| < ||\omega_k||$ then
Swap ω_k and ω_{v+1}.
Apply the pivot index reduction of ω_{v+1} by ω_k.



Procedure: Triangular pivot index reduction

input: a generating set $\omega_1, \omega_2, \ldots, \omega_{v+1}$ of L with the triangular condition.
output: a reduced basis $\omega_1, \omega_2, \ldots, \omega_v \in L$.
begin
While $\omega_{v+1} \neq (0, \ldots, 0)$ do
(reduction step)
Set $k \leftarrow I(\omega_{v+1})$ (i.e., the pivot index of ω_{v+1}).
If $||\omega_{v+1}|| \geq ||\omega_k||$
Set $\omega_{v+1} \leftarrow \omega_{v+1} - \omega_k \cdot t^{||\omega_{v+1}|| - ||\omega_I||}$.
else
Swap ω_k and ω_{v+1}.
Set $\omega_{v+1} \leftarrow \omega_{v+1} - \omega_k \cdot t^{||\omega_{v+1}|| - ||\omega_k||}$.
end if
end while
end

We iterate the reduction step until $\omega_{v+1} = (0, \ldots, 0)$.

Computational complexities

We propose PIS (Pivot index reduction with Inductive projection and State representation) for computing all $k(v)$ $(w \geq v \geq 1)$.

Upper bounds for the number of bit operations for all $k(v)$:

$$\text{SIS method} \quad 2w\,\dim(S)^2 + \frac{4}{3}w^3\,\dim(S) + \frac{1}{4}w^4,$$

$$\downarrow$$

$$\text{PIS method} \quad 2w\,\dim(S)^2 + \frac{1}{2}w^2\,\dim(S) + \frac{1}{2}w^2(w+1),$$

under a mild assumption.

This is because SIS requires Gaussian elimination at each reduction step (i.e., the cost $O(v^3)$).

Speed comparison

We choose the following w-bit $\mathbf{F}_2$-linear generators ($\dim(S) = 19937$):

- 32-bit WELL generator WELL19937a'
  (Pannton-L'Ecuyer-Matsumoto, 2006).
- 64-bit Mersenne Twister MT19937-64 (Nishimura, 2000).

We compared the following two methods:

1. PIS method (our proposal)
2. SIS method (H-M-S, to appear in Math. Comp.)

We measured the CPU time (in seconds) for computing each $k(v)$ for $1 \leq v \leq w$.

The experiments were coded in C language (`-O2` optimization flag) on Linux operating system.

WELL19937a'			MT19937-64					
$k(v)$	PIS	SIS	$k(v)$	PIS	SIS	$k(v)$	PIS	SIS
$k(32)$	0.275	0.451	$k(64)$	0.289	0.933	$k(32)$	0.004	0.009
$k(31)$	0.009	0.014	$k(63)$	0.000	0.000	$k(31)$	0.000	0.003
$k(30)$	0.009	0.014	$k(62)$	0.000	0.001	$k(30)$	0.001	0.002
$k(29)$	0.009	0.014	$k(61)$	0.000	0.001	$k(29)$	0.001	0.004
$k(28)$	0.008	0.014	$k(60)$	0.000	0.000	$k(28)$	0.002	0.004
$k(27)$	0.008	0.013	$k(59)$	0.000	0.001	$k(27)$	0.002	0.005
$k(26)$	0.008	0.013	$k(58)$	0.000	0.001	$k(26)$	0.002	0.007
$k(25)$	0.008	0.012	$k(57)$	0.000	0.000	$k(25)$	0.003	0.007
$k(24)$	0.010	0.012	$k(56)$	0.000	0.001	$k(24)$	0.004	0.007
$k(23)$	0.009	0.012	$k(55)$	0.000	0.001	$k(23)$	0.004	0.008
$k(22)$	0.009	0.012	$k(54)$	0.000	0.001	$k(22)$	0.004	0.008
$k(21)$	0.009	0.012	$k(53)$	0.000	0.001	$k(21)$	0.003	0.007
$k(20)$	0.009	0.012	$k(52)$	0.000	0.000	$k(20)$	0.001	0.003
$k(19)$	0.009	0.012	$k(51)$	0.000	0.001	$k(19)$	0.002	0.005
$k(18)$	0.009	0.012	$k(50)$	0.001	0.001	$k(18)$	0.003	0.006
$k(17)$	0.009	0.011	$k(49)$	0.000	0.001	$k(17)$	0.003	0.006
$k(16)$	0.008	0.012	$k(48)$	0.001	0.001	$k(16)$	0.004	0.007
$k(15)$	0.008	0.012	$k(47)$	0.000	0.001	$k(15)$	0.001	0.003
$k(14)$	0.008	0.011	$k(46)$	0.000	0.001	$k(14)$	0.003	0.005
$k(13)$	0.008	0.010	$k(45)$	0.000	0.001	$k(13)$	0.002	0.003
$k(12)$	0.009	0.010	$k(44)$	0.001	0.001	$k(12)$	0.003	0.005
$k(11)$	0.008	0.011	$k(43)$	0.000	0.001	$k(11)$	0.003	0.003
$k(10)$	0.009	0.010	$k(42)$	0.001	0.001	$k(10)$	0.003	0.004
$k(9)$	0.009	0.011	$k(41)$	0.001	0.002	$k(9)$	0.003	0.002
$k(8)$	0.008	0.010	$k(40)$	0.000	0.001	$k(8)$	0.003	0.003
$k(7)$	0.008	0.011	$k(39)$	0.001	0.002	$k(7)$	0.002	0.004
$k(6)$	0.008	0.010	$k(38)$	0.001	0.002	$k(6)$	0.003	0.003
$k(5)$	0.009	0.009	$k(37)$	0.001	0.002	$k(5)$	0.003	0.004
$k(4)$	0.009	0.010	$k(36)$	0.000	0.003	$k(4)$	0.003	0.004
$k(3)$	0.009	0.010	$k(35)$	0.001	0.004	$k(3)$	0.004	0.005
$k(2)$	0.009	0.009	$k(34)$	0.002	0.005	$k(2)$	0.005	0.005
total	0.534	0.798	$k(33)$	0.002	0.005	total	0.386	1.128

We also experimented with other 32-bit $\mathbf{F}_2$-linear generators:

- 32-bit Mersenne Twister MT19937
  $\dim(S) = 19937$ (Matsumoto-Nishimura, 1998)
- 32-bit WELL generator WELL44497a'
  $\dim(S) = 44497$ (Panneton et al., 2006)

The following table is a summary of the cumulative CPU time (in seconds) for computing all $k(v)$ $(w \geq v \geq 1)$.

	PIS	SIS	Δ
MT19937-64	0.386	1.128	7820
MT19937	0.294	0.481	6750
WELL19937a'	0.534	0.798	0
WELL44497a'	2.591	3.193	0

(The total defect $\Delta := \sum_{v=1}^{w}(\lfloor \dim(S)/v \rfloor - k(v))$.)

PIS is faster than SIS.

Appendix: a simple idea for speed-up of lattice reduction (effective for sparse transition generators)

We propose to choose $s_0 \in S$ whose bits are all zero except one bit (i.e., 0-excess states) for computing a reduced basis of

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

The 0-excess state was proposed as a bad initialization for Mersenne Twisters by (Panneton et al., 2006), because the sparse output is kept for a long term.

On the other hand, the 0-excess states can accelerate the reduction speeds.

The following table is a summary of the cumulative CPU time (in seconds) for computing all $k(v)$ $(w \geq v \geq 1)$.

	PIS(0-ex.)	SIS(0-ex.)	PIS	SIS	Δ
MT19937-64	0.105	0.197	0.386	1.128	7820
MT19937	0.029	0.036	0.294	0.481	6750
WELL19937a'	0.525	0.794	0.534	0.798	0
WELL44497a'	2.587	3.137	2.591	3.193	0

(The total defect $\Delta := \sum_{v=1}^{w}(\lfloor \dim(S)/v \rfloor - k(v))$.)

- WELL generators recover from 0-excess states very quickly.
- The lattice reductions for Mersenne Twisters are significantly accelerated.

In every case, our PIS is faster.