I. J. Mathematical Sciences and Computing, **2022, 3, 1-20**
Published Online August 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijmsc.2022.03.01

Revised Method for Sampling Coefficient Vector of GNR-enumeration Solution

Gholam Reza Moghissi
Department of ICT, Malek-Ashtar University of Technology, Tehran, Iran
E-mail: fumoghissi@chmail.ir

Ali Payandeh
Department of ICT, Malek-Ashtar University of Technology, Tehran, Iran

Received: 02 October 2021; Revised: 18 November 2021; Accepted: 25 December 2021; Published: 08 August 2022

Abstract: For selecting security parameters in lattice-based cryptographic primitives, the exact manner of BKZ algorithm (as total cost and specification of output basis) should be estimated in high block sizes. The simulations of BKZ are used to predict (estimate) the exact manner of BKZ algorithm which cannot be studied by practical running of BKZ algorithm for higher block sizes. Sampling method of GNR (Gamma-Nguyen-Regev) enumeration solution vector $\boldsymbol{v}$ is one of the main components of designing BKZ-simulation and it includes two phases: sampling the norm of solution vector $\boldsymbol{v}$ and sampling corresponding coefficient vectors. Our work, by Moghissi and Payandeh in 2021, entitled as "*Better Sampling Method of Enumeration Solution for BKZ-Simulation*", introduces a simple and efficient idea for sampling the norm and coefficient vectors of GNR enumeration solution $\boldsymbol{v}$. This paper proposes much better analysis for approximating the expected value and variance of the entries of these coefficient vectors. By this analysis, our previous idea for sampling the coefficient vectors is revised, which means that the expected value and variance of every entry in these coefficient vectors sampled by our new sampling method, are more close to the expected value and variance of corresponding entries in original sampling method, while these new sampled coefficient vectors include no violation from main condition of GNR bounding function (i.e., our new sampling method is not a rejection sampling).

Index Terms: BKZ Simulation, GNR Enumeration, Coefficient Vector, Sampling Method, Expected Value, Variance.

1. Introduction

Lattice reduction is the main phase of most lattice attacks. BKZ reduction is the current main practical lattice reduction algorithm [1,2]. For determining the security parameters in lattice cryptographic primitives (such as the work in [3]), the total cost and output quality of BKZ algorithm should be estimated in high block sizes. For predicting the manner of BKZ in higher block sizes, practical running is not the way, therefore BKZ-simulations are introduced (e.g., the simulation by Shi Bai et al. [4] is one of the last versions of BKZ-simulation). Designing a BKZ-simulation with GNR-pruned enumeration includes some necessary components such as enumeration radius, generation of bounding function, estimation of success probability, LLL simulation, estimation of GNR enumeration cost, sampling method for enumeration solution and simulation of updating GSO [5]. However other SVP-solvers (such as the works in [6-8]), instead of GNR-enumeration, can be considered in designing BKZ-simulations, (to the best of our knowledge) previous researches in design of BKZ-simulation just focus on BKZ with enumeration function. Also, design of pure version of BKZ-simulation (with GNR-enumeration) can be extended simply into the simulation of different versions of BKZ (such as BKZ with progressive block size [9] or progressive success probability [10] or pressed BKZ [4]).

One of the main components in design of BKZ-simulation with GNR enumeration is sampling the enumeration solution v. Enumeration solution v in former BKZ-simulations is often considered in the form of the expected value of the norm of solution $\|v\|$. GNR enumeration function in BKZ algorithm over a lattice block $\mathcal{L}_{[b_j,\ldots,b_k]}$ returns the coefficient vector y, which can be used to compute the solution vector v by linear combination of vectors of this lattice block as $v = \sum_{l=j}^{k} y_l b_l$ [5]. Our work in [5] tries to introduce an approximate sampling method for enumeration solution v which samples both norm and coefficient vectors of enumeration solution. Our idea in Lemma 3 from [5] introduces a suitable method for sampling the norm of enumeration solution v. In other side, to the best of our knowledge, first precise and explicit analysis of sampling the coefficient vector y is introduced in [5] and following sampling methods of coefficient vectors are proposed in [5]:

- Our sampling method 1 (which is introduced in Claim 1 and Lemma 11 from [5]; see the pseudo-code in Algorithm 5 from [5]) is defined just for piecewise-linear bounding functions with condition of small enumeration radius or small (static) success probability. In other words, the condition that the expected number of solutions in GNR enumeration (i.e., dynamic success frequency) belongs to $\approx O(1)$; However our results in [5] verify our sampling method 1, this version of sampling is not too accurate for much small success probability;
- Our sampling method 2 (see the pseudo-code in Algorithm 6 from [5]) is defined just for piecewise-linear bounding functions with any enumeration radius and any success probability; However our results in [5] verify our sampling method 2, this version of sampling is not too accurate for enumeration radius of $R \approx \|v\|$ and much small success probability;
- Our sampling method 3 (see the pseudo-code in Algorithm 6 from [5]) is adapted for any bounding function with any enumeration radius and any success probability (i.e., with no constraint); Our test results in [5] show that the probability distribution (and expected value) of $\mathrm{X}_i = \sum_{l=1}^{i} z_l^2/R^2$ for different types of bounding functions with same success probability P is not sufficiently similar, unless for big dynamic success frequency (i.e., $f_{succ} \gg 1$). Accordingly, this version of sampling is not much accurate.

As mentioned, our idea in Lemma 3 from [5] is accepted in this paper, but our sampling methods of 1, 2 and 3 cannot be accepted as exact sampling methods of coefficient vectors and would be revised in this paper.

The accuracy of our sampling methods of 1, 2 and 3 in [5] is just determined based on the expected value of each entry in coefficient vector of enumeration solution! As shown in test results of this paper and [5], when the success probability of bounding function is much small or enumeration radius tends to the solution norm (i.e., $\boldsymbol{R} \approx \|\boldsymbol{v}\|$), the proposed sampling method 1, 2 and 3 would be less accurate. In this paper, a novel idea for approximating this expected value is introduced which is more close to the expected value of corresponding entries in basic rejection sampling in Algorithm 6 from [2]. In other side, there is no analysis on variance of each entry in coefficient vector of enumeration solution in [5], while our test results in this paper show disappointing variance for these three sampling methods! Also a novel idea for approximating this variance is introduced which is more close to corresponding entries in basic rejection sampling in Algorithm 6 from [2]. Moreover there is no analysis on the number of violations from main condition of bounding function at each entry in coefficient vector of enumeration solution in [5], while our test results in this paper show surprising much number of violations for these three sampling methods! Finally in this paper, our sampling method 1, 2 and 3 (for sampling coefficient vectors) are revised, in the way that, our new method is not a rejection sampling (i.e., it includes no violation of main condition of GNR bounding function at each entry of coefficient vectors), also the expected value and variance of each entry of coefficient vectors tend to our new exact approximation of expected value and variance of corresponding entries in Algorithm 6 from [2]. This is worthy of noting that for designing a BKZ-simulation, our novel idea for approximating the expected value of corresponding entries in Algorithm 6 from [2] is sufficiently exact (instead of "Our Sampling Method 5"), while its variance and number of violations at each entry are absolutely zero. In fact, using our new sampling method just tries to inject more reasonable randomness to the BKZ-simulation.

The remainder of this paper is organized as follows. Section 2 is dedicated to essential background for understanding our contributions in this paper. Our contributions in design of sampling method of coefficient vectors of GNR enumeration solution would be introduced in section 3. Our test results for verifying our claims and value of our contributions in this paper are introduced in section 4. Finally, in section 5, the conclusions and further studies for this work are expressed.

2. Background

In this section, a sufficient background is introduced to make this work easy to be understood. At first of this section, some basic preliminaries for this paper are introduced. Next, some necessary information on GNR-enumeration and coefficient vector of GNR-enumeration solution is introduced.

2.1. Basic Preliminaries

A lattice in this paper is defined as follows:

Lattices. A lattice is a set of points in the n-dimensional space [11], in the way that for given n-linearly independent vectors $b_1, \dots, b_n \in \mathbb{R}^m$, the lattice can be generated by them as the following set of vectors:

$$\mathcal{L}(b_1, \dots, b_n) = \{\textstyle\sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z}\}. \tag{1}$$

The set of vectors of $[b_1, \dots, b_n]$ is known as the lattice basis shown by column matrix of B. Also a lattice block of $[b_j, b_{j+1}, \dots, b_k]$ is shown by $B_{[j,k]}$ or $\mathcal{L}_{[b_j,\dots,b_k]}$ or $\mathcal{L}_{[j,k]}$. For cryptographic applications, this is assumed to $b_i \in \mathbb{Z}^m$. The rank and dimension of lattice $\mathcal{L}(B)$ are shown respectively by n and m.

Euclidean norm. The length or norm of a lattice vector $v = (v_1, \dots, v_m)$ is measured by $\|v\| = \sqrt[2]{{v_1}^2 + \cdots + {v_m}^2}$.

Fundamental Domain. For a lattice $\mathcal{L}(B)$, the fundamental domain is defined as following set:

$$\mathcal{F}(\mathcal{L}) = \{t_1 b_1 + t_2 b_2 + \cdots + t_n b_n : 0 \le t_i < 1\}. \tag{2}$$

Volume of Lattice. The volume of a lattice $\mathcal{L}(B)$ is defined by the volume of the parallelepiped of fundamental domain $\mathcal{F}(\mathcal{L})$ which is computed as follows:

$$Vol\big(\mathcal{L}(B)\big) = Vol\Big(\mathcal{F}\big(\mathcal{L}(B)\big)\Big) = |\det B|. \tag{3}$$

One of the main heuristic in lattice theory is Gaussian Heuristic, which estimates the number of points in a set S.

Heuristic 1 (Gaussian Heuristic). *"Given a lattice $\mathcal{L}$ and a set S, the number of points in $S \cap \mathcal{L}$ is approximated by $Vol(S)/Vol(\mathcal{L})$"* [1];

By using Gaussian Heuristic, if lattice $\mathcal{L}$ is limited in a central ball with radius of $R = \lambda_1(\mathcal{L})$, then it is expected that there is at least one lattice vector in $Ball_n(R)$ with radius R, which is the shortest vector. Therefore, the value of $\lambda_1(\mathcal{L})$ can be estimated by Gaussian Heuristic of lattice $GH(\mathcal{L})$ as follows (by using sterling approximation):

$$GH(\mathcal{L}) = \left(\frac{Vol\big(\mathcal{L}(B)\big)}{Vol\big(Ball_n(1)\big)}\right)^{\frac{1}{n}} \approx \sqrt{\frac{n}{2\pi e}}(\det B)^{\frac{1}{n}}. \tag{4}$$

Also, Gram-Schmidt Orthogonalization and projected sub-lattices are other fundamental definitions in BKZ reduction.

Orthogonal projection ($\pi_i(\dots)$). For a given lattice basis $B = (b_1, b_2, \dots, b_n)$, the orthogonal projection $\pi_i(\dots)$ is defined as follows:

$$\pi_i : \mathbb{R}^m \mapsto \mathrm{span}(b_1, \dots, b_{i-1})^{\perp}, \qquad \text{where } 1 \le i \le n.$$

Gram-Schmidt Orthogonal basis (GSO basis). For given lattice basis $B = (b_1, b_2, \dots, b_n)$, the Gram-Schmidt Orthogonal basis $B^* = (b_1^*, b_2^*, \dots, b_n^*)$ is defined as follows:

$$\pi_i(b_i) = b_i^* = b_i - \textstyle\sum_{j=1}^{i-1} \mu_{i,j} . b_j^*, \tag{5}$$
$$\text{where } \mu_{i,j} = \frac{b_i . b_j^*}{\|b_j^*\|^2} \text{ and } 1 \le j < i \le n.$$

Note: The parameter of b_i^* refers to i-th vector of GSO basis of B^*.

Also in our statistical analysis, Gamma distribution is massively used:

Gamma distribution. The Gamma distribution is a two-parameter and continuous probability distribution which is defined as follows (for input shape-parameter of k and scale-parameter of θ):

$$\mathrm{Gamma}(x; k, \theta) = \frac{x^{k-1} . e^{-x/\theta}}{\Gamma(k)\, \theta^k}, \quad \text{where } x > 0. \tag{6}$$

The mean and variance in Gamma distribution respectively are determined by $k\theta$ and $k\theta^2$.

Note: In this paper, the expected value and variance of random variable X respectively are shown by E[X] and $\mathcal{V}$[X] (i.e., for random variable of x, the notation of $\mathcal{V}[x]$ and the notation of $\sigma^2[x]$ are equivalent). Also, the standard deviation is shown by notation of STD[X] $\approx \sqrt{\mathcal{V}[\mathrm{X}]}$.

2.2. GNR-Enumeration function

There are many hard problems in lattice theory, where the shortest vector problem (SVP) is the basic one. For given lattice basis B, solvers of SVP try to find the shortest nonzero vector in this lattice. One of the main SVP-solvers is enumeration function which searches the solution vector in a tree. Different pruning method is introduced for enumeration function.

Full-enumeration. For initial radius R, the tree of full-enumeration enumerates all lattice points in n-dimensional ball of radius R (i.e., no pruning is used in this type of enumeration function).

The sound pruning technique (also named as GNR-pruning), which is introduced by Gama, Nguyen and Regev, uses the concept of cylinder-intersection. The cylinder-intersection, bounding function and GNR-pruning are defined as follows:

Cylinder-intersection. The l-dimensional cylinder-intersection with radius $(R_1, \dots, R_l)$ is defined as following set

[1]:

$$C_{R_1,\dots,R_l} = \{(x_1,\dots,x_l) \in \mathbb{R}^l, \forall\, 1 \le i \le l, \sum_{t=1}^{i} x_t{}^2 \le R_i{}^2\}. \tag{7}$$

Bounding function. The vector of $\mathcal{R}$: $0 \le \mathcal{R}_1 \le \mathcal{R}_2 \le \cdots \le \mathcal{R}_\beta = 1$, when multiplied by initial radius R, defines a bounded cylinder-intersections with radius $(R_1,\dots,R_l) = (R.\mathcal{R}_1,\dots,R.\mathcal{R}_l)$ for $1 \le l \le \beta$, and consequently is used to prune the enumeration trees [1].

Note: In this paper, the entry i in bounding function $\mathcal{R}$ is shown by $\mathcal{R}[i]$ and $\mathcal{R}_i$.

GNR-pruning (Sound pruning). For a lattice block of $B_{[j,k]} = [b_j, b_{j+1},\dots,b_k]$ and the coefficient vector $x \in \mathbb{Z}^\beta$, this pruning replaces the inequalities of $\|\pi_{k+1-i}(x.B_{[j,k]})\| \le R$ for $1 \le i \le k-j+1$ as a bounded ball (in full-enumeration) by $\|\pi_{k+1-i}(x.B_{[j,k]})\| \le \mathcal{R}_i R$, where $0 \le \mathcal{R}_1 \le \cdots \le \mathcal{R}_{k-j+1}$=1 as a cylinder-intersection [2].

The pseudo-code of the sound pruned enumeration (GNR-enumeration) can be studied in Appendix B from paper [1]. Success probability is one of the main features of bounding function which is defined as follows [1]:

Success probability of bounding function. For any lattice block of $[b_j, b_{j+1},\dots,b_k]$, initial enumeration radius R and bounding function $\mathcal{R}$, if there is just one lattice vector v in n-dimensional ball with radius of R (i.e., $\|v\| \le R$), the probability of finding solution vector v after GNR-pruning by $\mathcal{R}$ in enumeration tree is defined as the success probability of $\mathcal{R}$, and is shown by $p_{succ}(\mathcal{R})$.

For analysis of the success probability of GNR bounding function, Gama et al. use following heuristic assumptions (in addition to Gaussian Heuristic) [1]:

Heuristic 2. *"The distribution of the coordinates of the target vector v, when written in the normalized Gram-Schmidt basis $(b_1^*/\|b_1^*\|,\dots,b_n^*/\|b_n^*\|)$ of the input basis, look like those of a uniformly distributed vector of norm $\|v\|$".*

The success probability of the bounding function $\mathcal{R}$ can be estimated by Monte Carlo simulation (see Algorithm 8 in paper [2]), however Algorithm 7 from paper [1] can be used as more efficient estimation of success probability.

At the end of this subsection, two families of bounding functions including "Step bounding function" and "Piecewise-linear bounding function" are defined as follows [1]:

Step bounding function. The Step bounding function is defined as $\mathcal{R}_i^2 = \mathrm{a}$, for $i = 1,\dots,\beta/2$ and $\mathcal{R}_i^2 = 1$ in otherwise, where $0 < \mathrm{a} \le 1$ [1].

Piecewise linear pruning. The piecewise linear bounding function is defined as $\mathcal{R}_i^2 = \frac{2i\mathrm{a}}{\beta}$, for $i = 1,\dots,\beta/2$ and $\mathcal{R}_i^2 = \left(2\mathrm{a} - 1 + \frac{2i(1-\mathrm{a})}{\beta}\right)$ in otherwise, where $0 < \mathrm{a} \le 1$ [1].

We show in [12], if success probability of optimal bounding function is closed to $\%100$, then the cost of GNR-enumeration pruned by this bounding function can be approximated as $\frac{1}{2^{\beta/4}}$ times the cost of full-enumeration (in consistent with the analysis in [1]). Also, we show in [12], for practical block size of $100 \le \beta \le 250$, the upper-bound for speedup of GNR-enumeration pruned by a piecewise-linear bounding function with much small success probability (as extreme pruning) when the lattice block is preprocessed by BKZ with enumeration would be estimated from $2^{\frac{\beta}{6.6}}$ to $2^{\frac{\beta}{4.4}}$ (and tries to reject the claimed speedup of $2^{\frac{\beta}{2}}$ by extreme-pruning over full-enumeration in [1]). Furthermore, a global search strategy, a practical search algorithm and an efficient approximate technique to generate the optimal bounding function are introduced in [14].

2.3. Coefficient Vector of GNR-Enumeration Solution

GNR enumeration function over a lattice block $\mathcal{L}_{[b_j,\dots,b_k]}$ returns the coefficient vector y, which can be used to compute the solution vector v by linear combination of vectors of this lattice block as $v = \sum_{l=j}^{k} y_l b_l$. The coefficient of orthonormal basis vector $z = (z_1, z_2,\dots,z_{k-j+1=d})$ in Heuristic 2 which corresponds with the target lattice vector of v can be formulated as follows (note that, $b_i^*/\|b_i^*\|$ is the ith vector of the orthonormal basis of $b_1^*/\|b_1^*\|,\dots,b_n^*/\|b_n^*\|$) [1]:

$$v = (z_1,\cdots,z_d).\begin{bmatrix} b_k^*/\|b_k^*\| \\ \vdots \\ b_j^*/\|b_j^*\| \end{bmatrix} = (v_1,\cdots,v_m). \tag{8}$$

The coordinates of the coefficient vector z are reversed (i.e., z_i corresponds with $b_{k-i+1}^*/\|b_{k-i+1}^*\|$), and it is clear that $\|z\| = \|v\|$ [1]. Also, the vector $u = (u_1, u_2,\dots,u_{k-j+1=d}) = (z_1/R, z_2/R,\dots,z_d/R)$ is chosen to be uniformly distributed from d-dimensional ball of the radius 1 (by the notation of $u \backsim Ball_d$). By using these formulations, success probability of a GNR bounding function $\mathcal{R}$ can be formally defined as follows [1]:

$$p_{succ}(\mathcal{R}) = \Pr_{u\backsim Ball_d}\left(\forall\, i \in [1,d], \sum_{l=1}^{i} u_l{}^2 \le \frac{R_i{}^2}{R_d{}^2}\right) = \Pr_{u\backsim Ball_d}\left(\forall\, i \in [1,d], \sum_{l=1}^{i} u_l{}^2 \le \mathcal{R}_i{}^2\right). \tag{9}$$

Note: Since in last block of BKZ, the size of blocks becomes less than initial block size of β, so the variable of $d = k - j + 1$ is used to emphasize this fact.

The main condition (or constraint) which defines the success probability of finding true coefficient vector of enumeration solution for an input bounding function is defined as follows:

$$\left(\forall\, i \in [1,d], \sum_{l=1}^{i} z_l^2 \le \mathcal{R}_i{}^2 . R_d{}^2\right) \equiv \left(\forall\, i \in [1,d], \sum_{l=1}^{i} u_l{}^2 \le \frac{R_i{}^2}{R_d{}^2} = \mathcal{R}_i{}^2\right) \equiv \left(\forall\, i \in [1,d], \sum_{l=1}^{i} \frac{\omega_{d-l+1}}{\sum_{t=1}^{d}\omega_t} \le \mathcal{R}_i{}^2\right), \quad \text{where } \omega_i \leftarrow \text{Gamma}(1/2,2). \tag{10}$$

The solution vector v can be written by the coefficient vector $w = \left(\frac{z_d}{\|b_1^*\|}, \dots, \frac{z_2}{\|b_{d-1}^*\|}, \frac{z_1}{\|b_d^*\|}\right)$ on GSO block as follows [5]:

$$v = (v_1, \cdots, v_m) = (w_1, \cdots, w_d). \begin{bmatrix} b_1^* \\ \vdots \\ b_d^* \end{bmatrix}. \tag{11}$$

Note: The solution vector v from enumeration over lattice block $\mathcal{L}_{[j,k]}$ is a GSO projected vector which is orthogonal over the previous basis vectors in $\mathcal{L}_{[1,j-1]}$ (remember that, here the notation of $\mathcal{L}_{[1,d]}$ represents $\mathcal{L}_{[j,k]}$).

By inserting the solution vector v at first of the lattice block $\mathcal{L}_{[1,d]}$ (which results in the block of $(v, b_1^*, \dots, b_d^*)$ with $d+1$ vectors), one of the vectors from the GSO block $[v, b_1^*, \dots, b_d^*]$ should be eliminated after updating GSO coefficients of these $d+1$ vectors. Lattice enumeration uses the integer coefficients y_i for enumerating over the projected lattice block $\mathcal{L}_{[1,d]}$, therefore the coefficients w_i in vector w depend on integer entries in the vector of y, as follows [5]:

Note: Here, the projection notation of $\pi_1(\dots)$ represents $\pi_j(\dots)$, also the dimension of b_i^* and v is m which differs from the rank of lattice block (i.e., block size of d).

$$v = (y_1, y_2, \cdots, y_d). \begin{bmatrix} \pi_1(b_1) \\ \vdots \\ \pi_1(b_d) \end{bmatrix} = (y_1, y_2, \cdots, y_d). \begin{bmatrix} b_1^* \\ \vdots \\ b_d^* + \sum_{i=1}^{d-1} \mu_{d,i} . b_i^* \end{bmatrix}. \tag{12}$$

$$\|v\|^2 = \underbrace{\left(y_1 + \sum_{i=2}^{d} y_i \mu_{i,1}\right)^2 \|b_1^*\|^2}_{z_d^2} + \cdots + \underbrace{\left(y_g + \sum_{i=g+1}^{d} y_i . \mu_{i,g}\right)^2 \left\|b_g^*\right\|^2}_{z_{d-g+1}^2} + \cdots + \underbrace{y_d^2 \|b_d^*\|^2}_{z_1^2} \Rightarrow \tag{13}$$

$$w = \left[\underbrace{y_1 + \sum_{i=2}^{d} y_i . \mu_{i,1}}_{w_1}, \dots, \underbrace{y_g + \sum_{i=g+1}^{d} y_i . \mu_{i,g}}_{w_g}, \dots, \underbrace{y_d}_{w_d}\right]. \tag{14}$$

Based on (14), a zero coefficient of y_i does not always result in $w_i = 0$, except for indices after the last non-zero coefficient of y_g [5]. Also by using Theorem 2 in [5], following relations can be expanded:

$$w_i = \frac{z_{d-i+1}}{\|b_i^*\|} \Rightarrow$$
$$w_g = \frac{z_{d-g+1}}{\|b_g^*\|} = 1 \Rightarrow z_{d-g+1} = \left\|b_g^*\right\|. \tag{15}$$

$$w_l = \frac{z_{d-l+1}}{\|b_l^*\|} = 0 \Rightarrow z_{d-l+1} = 0, \text{ for } g < l \le d. \tag{16}$$

The entries of coefficient vectors of w and z can be sampled for full-enumeration as follows [5]:

$$w_i \leftarrow (-1)^{\lfloor \text{rand}_{[0\dots2)} \rfloor} . \sqrt{\frac{\omega_i . \left(\|v\|^2 - \|b_g^*\|^2\right)}{\|b_i^*\|^2 . \left(\sum_{t=1}^{g-1} \omega_t\right)}}, \quad \text{where } \omega_i \leftarrow \text{Gamma}(1/2,2). \tag{17}$$

$$z_{d-i+1} \leftarrow (-1)^{\lfloor \text{rand}_{[0\dots2)} \rfloor} . \sqrt{\frac{\omega_i . \left(\|v\|^2 - \|b_g^*\|^2\right)}{\sum_{t=1}^{g-1} \omega_t}}, \quad \text{where } \omega_i \leftarrow \text{Gamma}(1/2,2). \tag{18}$$

3. Our Contribution

As mentioned, GNR enumeration in BKZ algorithm over a lattice block $\mathcal{L}_{[b_j,\ldots,b_k]}$ returns the coefficient vector y, which can be used to compute the solution vector v by linear combination of this lattice block vectors as $v = \sum_{l=j}^{k} y_l b_l$.

To the best of our knowledge, our work in [5] is first precise analysis of sampling the coefficient vector y. As mentioned, our sampling methods for coefficient vectors is developed in three versions in [5], as follows:

- Our sampling method 1 (which is introduced in Claim 1 and Lemma 11 from [5]; see the pseudo-code in Algorithm 5 from [5]);
- Our sampling method 2 (see the pseudo-code in Algorithm 6 from [5]);
- Our sampling method 3 (see the pseudo-code in Algorithm 6 from [5]);

This paper proposes Algorithm 1 as the skeleton of our sampling method of coefficient vectors.

Note: For simplicity, Algorithm 1 assumes that cutting point is $\mathrm{Cut} = d$, while in practical purposes, if $\mathrm{Cut} < d$ then this is only needed to change the block of $\mathcal{L}_{[1,d]}$ with block size of d into the block of $\mathcal{L}_{[1,\mathrm{Cut}]}$ with block size of Cut.

Algorithm 1 Sample_of_w

Input: GSO norms of block given as $B^*_{[1,d]} = [\|b_1^*\| \ldots \|b_d^*\|]$ at stage j, any type of bounding function $\mathcal{R}$, enumeration radii R,
Sampled norm of enumeration solution as ℓ_1, last nonzero index of $\mathscr{g}$,/* boolean parameter of RANDBLOCK,*/GSO Coefficient Matrix of μ.

$[\ell_1^2, \ell_2^2, \ldots, \ell_{\mathscr{g}}^2] \leftarrow$ Sample_of_$\ell_t^2(B^*_{[1,d]}, \mathcal{R}, \mathrm{R}, \ell_1, \mathscr{g})$;
for$(l = \mathscr{g} + 1; l \leq d; l++)$ $y_l = 0$;
$y_{\mathscr{g}} = 1$;
for$(l = \mathscr{g} - 1; l \geq 1; l--)\{$
    $y_l = \frac{\sqrt{\ell_l^2 - \ell_{l+1}^2}}{\|b_l^*\|} - \sum_{i=l+1}^{\mathscr{g}} y_i \mu_{i,l}$;//$\ell_x^2 = \|\pi_x(v)\|^2$
}//end for
for$(l = 1; l \leq d; l++)\{$
    $w_l = y_l + \sum_{i=l+1}^{d} y_i \mu_{i,l}$;
    $z_{d-l+1} = w_l \|b_l^*\|$;
}//end for

Output: Sampled cofficient vectors of w, z and y.

Lemma 1. For a GNR-enumeration over GSO block $[b_1^*, \ldots, b_d^*]$ by any type of bounding function $\mathcal{R}$, enumeration radii R and current GSO Coefficient Matrix of μ, if there is a solution which is returned with sampled norm of ℓ_1 and last nonzero index of $\mathscr{g}$, Algorithm 1 gives the samples of coefficient vectors of w, z and y for this sampled solution based on Heuristic 2, just if the function "Sample_of_ℓ_t^2" truly samples $\|\pi_t(v)\|$ for $1 \leq t \leq \mathscr{g}$ based on Heuristic 2.

See proof in Appendix C.

As mentioned in Lemma 1, Algorithm 1 gives the samples of coefficient vectors of w, z and y for this sampled solution based on Heuristic 2, just if the function of "Sample_of_ℓ_t^2" truly samples the norm of $\pi_t(v)$ for $1 \leq t \leq \mathscr{g}$ based on Heuristic 2. Algorithm 3 in Appendix A uses Lemma 3 in [5] for implementing the function of "Sample_of_ℓ_t^2". Lemma 2 declares that this idea in Algorithm 3 (which is named as "Our Sampling method 4") is not consistent with Heuristic 2 (i.e., it is impossible to use Lemma 3 in [5] for implementing the function of Sample_of_ℓ_t^2).

Lemma 2. Under Heuristic 2, if the sampled solution of $v = \pi_1(v)$ would be projected on $\mathrm{span}(b_1, \ldots, b_t)^{\perp}$, then $\pi_t(v)$ cannot be considered as an independent uniformly distributed sampled vector on the projected lattice block of $[\pi_2(b_t), \ldots, \pi_2(b_d)]$, and consequently the norm of $\pi_t(v)$ cannot be sampled by using Lemma 3 in [5] in the way which is introduced in Algorithm 3 (with corresponding parameters which are updated for this projected block in lines 3-10 from Algorithm 3 in Appendix A).

Proof. This proof is done by introducing a Counter-example. Lets assume a GSO basis of $B^*_{[1,2]} = [b_1^* = (1,0), b_2^* = (0,1)]$. For enumeration radius R and full-enumeration bounding function (i.e., vector of $\mathcal{R} = [1,1]$), if a solution vector v would be sampled uniformly from these lattice vectors in the circle of radius R based on Heuristic 2, then is the distribution of projected vector of $\pi_2(v)$ equal to uniform distribution over vertical vertex with basis of $[b_2^* = (0,1)]$? The answer is clearly "NO". In fact, the distribution of projected vector of $\pi_2(v)$ has most frequency in point of $(0,0)$ from vertical vertex, while the least frequency can be observed in points of $(0, \pm\lfloor R \rfloor)$ from vertical vertex, this means that the distribution of the projected vector of $\pi_2(v)$ in this example is not uniform. In other side, Lemma 3 in [5] is mainly defined based on uniformly distribution of lattice vectors in corresponding cylinder-intersection of the bounding function, so Lemma 3 in [5] cannot be used as "Sample_of_ℓ_t^2" for the projected vector $\pi_i(v)$ where $i > 1$.

Our results Test 1 from Section 4 verifies Lemma 2 (see Fig.2).

For best accuracy in sampling coefficient vectors, this paper introduces Algorithm 2 as "Our sampling method 5":

Algorithm 2 Sample_of_ℓ_t^2_Ver5 (Our Sampling Method 5)

Input: GSO norms of block given as $B^*_{[1,d]} = [b^*_1, \dots, b^*_d]$, bounding function $\mathcal{R}$, enumeration radii R, Sampled norm of enumeration solution as ℓ_1, last nonzero index of g.

1: $\mathcal{R}_{\text{UP1}} \leftarrow$ Get_UpperBound_of_$\mathcal{R}$_1$(d, \mathcal{R})$;
2: $\mathcal{R}_{\text{UP2}} \leftarrow$ Get_UpperBound_of_$\mathcal{R}$_2$(d, \mathcal{R}, \mathcal{R}_{\text{UP1}})$;
3: $\mathcal{R}_{\text{LOW}} \leftarrow$ Get_LowerBound_of_$\mathcal{R}(d, \mathcal{R}, g)$;
4: $[\text{E}[\ell_1^2], \dots, \text{E}[\ell_d^2]] \leftarrow$ Get_ExpectedValues_of_$\ell_t^2(B^*_{[1,d]}, \mathcal{R}, \text{R}, \ell_1, g, \mathcal{R}_{\text{LOW}}, \mathcal{R}_{\text{UP1}}, \mathcal{R}_{\text{UP2}})$;
5: $[\mathcal{V}[\ell_1^2], \dots, \mathcal{V}[\ell_d^2]] \leftarrow$ Get_Variances_of_$\ell_t^2(B^*_{[1,d]}, \mathcal{R}, \text{R}, \ell_1, [\text{E}[\ell_1^2], \dots, \text{E}[\ell_d^2]])$;
6: $[\ell_1^2, \dots, \ell_d^2] \leftarrow$ Get_Sample_of_$\ell_t^2(B^*_{[1,d]}, \ell_1, g, [\text{E}[\ell_1^2], \dots, \text{E}[\ell_d^2]], [\mathcal{V}[\ell_1^2], \dots, \mathcal{V}[\ell_d^2]])$;

Output: Sampled norms of $\ell_1^2, \ell_2^2, \dots, \ell_g^2$.

Algorithm 2 includes 5 main steps which are defined as follows.

Step 1. Computing Upper-bound bounding functions:

This phase (lines 1-2 in Algorithm 2) defines a Step bounding function of $\mathcal{R}_{\text{UP1}}$ and a modification of $\mathcal{R}_{\text{UP1}}$ as $\mathcal{R}_{\text{UP2}}$ including the entries which are closely lower-bounded by input bounding function $\mathcal{R}$. See the pseudo-codes of Get_UpperBound_of_$\mathcal{R}$_1 and Get_UpperBound_of_$\mathcal{R}$_2 respectively in Algorithm 4 and Algorithm 5 from Appendix B. Fig.3 shows the upper-bound bounding functions of $\mathcal{R}_{\text{UP1}}$ and $\mathcal{R}_{\text{UP2}}$ for input bounding function of "BF2 [Psucc=0.0002]".

Step 2. Computing Lower-bound bounding functions:

This phase (lines 3 in Algorithm 2) defines a piecewise-linear bounding function of $\mathcal{R}_{\text{LOW}}$ including the entries which are closely upper-bounded by input bounding function $\mathcal{R}$. See the pseudo-codes of Get_LowerBound_of_$\mathcal{R}$ in Algorithm 6 from Appendix B. Fig.3 shows the lower-bound bounding function of $\mathcal{R}_{\text{LOW}}$ for an input bounding function of "BF2 [Psucc=0.0002]".

Step 3. Computing the expected value of ℓ_t^2:

In this step, the expected value of $\ell_{d-i+1}^2 = \|\pi_{d-i+1}(v)\|^2$ can be computed as follows.

$\text{X}_i = \sum_{t=1}^{i} z_t^2 / R^2$, for $1 \le i \le d \Rightarrow$

$R^2\text{E}[\text{X}_i] = R^2\text{E}\left[\sum_{t=1}^{i} z_t^2 / R^2\right] = \text{E}\left[\sum_{t=1}^{i} z_t^2\right] = \text{E}\left[z_1^2 + \cdots + z_{d-g+1}^2 + z_{d-g+2}^2 + \cdots + z_i^2\right] = \text{E}\left[z_{d-g+1}^2 + z_{d-g+2}^2 + \cdots + z_i^2\right] \Rightarrow$

By using (13), Theorem 2 in [5] and (32):

$$R^2\text{E}[\text{X}_i] = \text{E}[\|\pi_{d-i+1}(v)\|^2] = \text{E}[\ell_{d-i+1}^2]. \tag{19}$$

By using Claim 1, the expected value of $R^2\text{X}_i = \sum_{t=1}^{i} z_t^2$ for Step bounding function $\mathcal{R}_{\text{Step}}$ can be approximated.

Claim 1. Under Heuristic 2, Algorithm 8 closely approximates the expected value of $\ell_{d-i+1}^2 = \|\pi_{d-i+1}(v)\|^2 = R^2\text{X}_i = \sum_{t=1}^{i} z_t^2$ for Step bounding function $\mathcal{R}_{\text{Step}}$.

To have better sense, see the shape of $\text{E}[\ell_{d-i+1}^2]$ for Step bounding function of "BF6 [Psucc=0.000147]" by our results in Fig.4. Also, by using Claim 2, the expected value of $R^2\text{X}_i = \sum_{t=1}^{i} z_t^2$ for Piecewise-linear bounding function $\mathcal{R}_{\text{Piece}}$ can be approximated.

Claim 2. Under Heuristic 2, Algorithm 9 closely approximates the expected value of $\ell_{d-i+1}^2 = \|\pi_{d-i+1}(v)\|^2 = R^2\text{X}_i = \sum_{t=1}^{i} z_t^2$ for Piecewise-linear bounding function $\mathcal{R}_{\text{Piece}}$.

To have better sense, see the shape of $\text{E}[\ell_{d-i+1}^2]$ for Piecewise-linear bounding function of "BF5 [Psucc=0.00000011]" by our results in Fig.5. Finally, by using Claim 3, the expected value of $R^2\text{X}_i = \sum_{t=1}^{i} z_t^2$ for any shape of bounding function $\mathcal{R}$ can be approximated.

Claim 3. Under Heuristic 2, for any shape of bounding function $\mathcal{R}$, if $\frac{\partial(\mathcal{R}[i])}{\partial i} < \frac{2}{d}$ then Algorithm 7 closely approximates the expected value of $\ell_{d-i+1}^2 = \|\pi_{d-i+1}(v)\|^2 = R^2\text{X}_i = \sum_{t=1}^{i} z_t^2$.

To have better sense, see the shape of $\text{E}[\ell_t^2]$ for bounding function of "BF2 [Psucc=0.0002]" by our results in Fig.6.

Step 4. Computing the variance of ℓ_t^2:

In this step, the variance of $\ell_{d-i+1}^2 = \|\pi_{d-i+1}(v)\|^2$ can be computed as follows:

$$R^4\mathcal{V}[\text{X}_i] = \mathcal{V}[\|\pi_{d-i+1}(v)\|^2] = \mathcal{V}[\ell_{d-i+1}^2]. \tag{20}$$

To approximate the variance of ℓ_{d-i+1}^2 for an input bounding function $\mathcal{R}$, at first, this paper estimates the variance of ℓ_{d-i+1}^2 for full-enumeration bounding function, then update this estimation for any arbitrary bounding function $\mathcal{R}$. The variance of $\mathrm{X}_i = \sum_{t=1}^{i} z_t^2/R^2$ for full-enumeration can be approximated by following lemma:

Lemma 3. Under Heuristic 2, if random variable $w_l^2 = z_i^2/\|b_l^*\|^2$ (for $l = d - i + 1$) would be sampled by (17) under full-enumeration with solution norm of $\|v\|$, the variance of $\mathrm{X}_i = \sum_{t=1}^{i} z_t^2/R^2$ can be approximated by (21):

$$\mathcal{V}[\mathrm{X}_i]_{\mathrm{full}} = \left(\frac{\|v\|^2-\|b_d^*\|^2}{R^2}\right)^2 \frac{2id-2i^2-2d+2i}{d^3-d^2-d+1}. \tag{21}$$

See proof in Appendix D.

Corollary 1. By Lemma 3 and (20), the variance of ℓ_{d-i+1}^2 for full-enumeration can be computed as follows:

$$R^4\mathcal{V}[\mathrm{X}_i]_{\mathrm{full}} = \mathcal{V}[\|\pi_{d-i+1}(v)\|^2]_{\mathrm{full}} = \mathcal{V}[\ell_{d-i+1}^2]_{\mathrm{full}} = (\|v\|^2 - \|b_d^*\|^2)^2\,\frac{2id-2i^2-2d+2i}{d^3-d^2-d+1}. \tag{22}$$

The variance of $\mathrm{X}_i = \sum_{t=1}^{i} z_t^2/R^2$ for an arbitrary bounding function $\mathcal{R}$ can be estimated by Approximation 1:

Approximation 1. Under Heuristic 2, for an arbitrary bounding function $\mathcal{R}$, if the variance of $\mathrm{X}_i = \sum_{t=1}^{i} z_t^2/R^2$ for full-enumeration is approximated as $\mathcal{V}[\mathrm{X}_i]_{\mathrm{full}}$ and the expected value of X_i is approximated as $\mathrm{E}[\mathrm{X}_i]$, then the variance of X_i for $\mathcal{R}$ can be approximated by (23):

$$\mathcal{V}[\mathrm{X}_i] \leftarrow \mathcal{V}[\mathrm{X}_i]_{\mathrm{full}} \times \min\left(\left(\frac{\mathcal{R}^2[i]-\mathrm{E}[\mathrm{X}_i]}{\mathcal{R}_{\mathrm{UP3}}^2[i]-\mathrm{E}[\mathrm{X}_i]_{\mathrm{full}}}\right)^2, 1\right), \tag{23}$$

where $\mathcal{R}_{\mathrm{UP3}}^2[i] = \min\left(0.7 \times \left(\mathrm{E}[\mathrm{X}_i]_{\mathrm{full}} + 3 \times \sqrt{\mathcal{V}[\mathrm{X}_i]_{\mathrm{full}}}\right) + 0.3, 1\right)$ and $\mathrm{E}[\mathrm{X}_i]_{\mathrm{full}}$ can be computed by using Algorithm 8 in Claim 1 (since full-enumeration bounding function is a step bounding function) or it can be computed by Lemma 9 in [5]. By Approximation 1 and (20), the variance of ℓ_{d-i+1}^2 for an input bounding function $\mathcal{R}$ can be estimated by (24):

$$\mathcal{V}[\ell_{d-i+1}^2] \leftarrow R^4\mathcal{V}[\mathrm{X}_i]. \tag{24}$$

The pseudo-code of computing the variance of ℓ_t^2 is introduced in Algorithm 10.

Step 5. Generating the samples of $\boldsymbol{\ell_t^2}$:

Under Heuristic 2, Approximation 2 generates the samples of $\ell_{t=d-i+1}^2$ for an arbitrary bounding function $\mathcal{R}$:

Approximation 2. Under Heuristic 2, the values of $\ell_{t=d-i+1}^2 = \|\pi_{d-i+1}(v)\|^2 = R^2\mathrm{X}_i = \sum_{t=1}^{i} z_t^2$ for an arbitrary bounding function $\mathcal{R}$ can be sampled by following formula:

$$\ell_t^2 \leftarrow \begin{cases} 1{:}\ 0, & \text{for } t = d \text{ downto } g + 1, \\ 2{:}\ \|b_g^*\|, & \text{for } t = g, \\ 3{:}\ \min(\min(\mathcal{R}^2[d-t+1] \times \mathrm{R}^2, \|v\|^2), \ell_{t+1}^2 + \max(\mathrm{Y}_t + \min(\mathrm{E}[\ell_t^2] - \ell_{t+1}^2 - \varepsilon, 0), 0)), & \text{for } t = g - 1 \text{ downto } 2, \\ 4{:}\ \|v\|^2, & \text{for } t = 1, \end{cases} \tag{25}$$

where $\varepsilon = 1/d$, $\Delta\mathrm{E}_t = \max(\mathrm{E}[\ell_t^2] - \ell_{t+1}^2, \varepsilon)$ and $\mathrm{Y}_t \leftarrow \mathrm{Gamma}\left(k = \frac{(\Delta\mathrm{E}_t)^2}{\mathcal{V}[\ell_t^2]}, \theta = \frac{\mathcal{V}[\ell_t^2]}{\Delta\mathrm{E}_t}\right)$.

The pseudo-code of sampling by Approximation 2 is introduced in Algorithm 11. Also, our results in Test 6 from Section 4, verify the statistical exactness of "Our Sampling Method 5" including Approximation 2.

Table 1. The configuration of our tests in this paper

Specification	**Value**
Dimension (β or d)	200
Last non-zero index (g)	200
Determinant of $\mathcal{L}_{[1,\beta]}$	0.2058911321e45
$\mathrm{GH}^2(\mathcal{L}_{[1,\beta]})$	33.5495
Radius Factor (r_{FAC}^2)	1.1
Radius R^2	36.9045
$\|b_1^*\|^2$	45657810
$\|b_g^*\|^2$	0.12004e-11
Count of solutions in Full-Enum $r_{\mathrm{FAC}}^{\beta}/2$	6890.31

4. Test Results

This section introduces sufficient test results to show the correctness, exactness and value of our claims and contributions in sampling the coefficient vector of GNR-enumeration solution. The configuration of our tests in this section uses the parameters in Table 1. Also the success probability of bounding functions in this paper is computed by using Algorithm 7 from paper [2]. Following notes are introduced for better study of our results in this section:

- Based on the definition of $\ell_{t=d-i+1}^2 = R^2 \mathrm{X}_i$ and $\mathrm{X}_i = \sum_{t=1}^{i} z_t^2 / R^2$, the expressions of "Sample_X_i by R" in Figure 3 in [5], Figure 4 in [5] and Figure 5 in [5] should be re-written as "Sample_$\ell_{t=d-i+1}^2$ by R".
- The entries of all bounding functions in the test results of this section are squared.
- The block size d in all results of this section is even.
- The expression of "Our Sampling Method 4" refers to Algorithm 1 by using Algorithm 3 as Sample_of_ℓ_t^2.
- The expression of "Our Sampling Method 5" refers to Algorithm 1 by using Algorithm 2 as Sample_of_ℓ_t^2.

Also, the shape of all bounding functions which are used in our tests are shown in Fig.1.

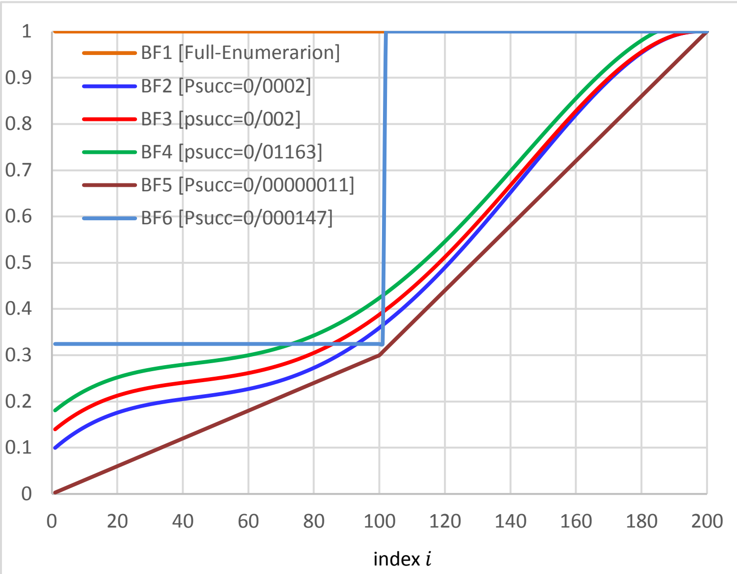


Fig.1. All bounding functions used in our tests results of this paper

- BF1 [Full-Enumerarion]: It is a vector including the entries with values of 1 for full-enumeration.
- BF2 [Psucc=0.0002]: It is an arbitrary bounding function with success probability of p_{succ} =0.0002.
- BF3 [Psucc=0.002]: It is an arbitrary bounding function with success probability of p_{succ} =0.002.
- BF4 [Psucc=0. 01163]: It is an arbitrary bounding function with success probability of p_{succ} =0. 01163.
- BF5 [Psucc=0.00000011]: It is a piecewise-linear bounding function with parameter of a = 0.3 and success probability of p_{succ} =0.00000011.
- BF6 [Psucc=0.000147]: It is a Step bounding function with parameter of a = 0.3244 and success probability of p_{succ} =0.000147.

***Test 1. Rejecting the correctness of Algorithm 3 under Heuristic 2*:** In this test, by using the configuration in Table 1, our test results for 169 successful samples verify Lemma 2 which declares Algorithm 3 as "our sampling method 4" cannot be defined under Heuristic 2, since as shown in Fig.2, the expected value of samples by original sampling method are not closing to the expected value of samples by our sampling method 4. In fact, in this paper, this is tried to introduce some sampling method which generates some samples of coefficient vectors in the way that the statistical measurements of entries of these samples (e.g., expected value) are close to the corresponding measurements of entries of samples by original sampling method by (18), while Algorithm 3 as "our sampling method 4" dose not satisfy this goal at all!

Note: All figures in this paper which shows the shape of $\ell_{t=d-i+1}^2$ or $\mathrm{E}[\ell_{t=d-i+1}^2]$ for some input bounding function $\mathcal{R}$, also shows the scaled entries of these bounding functions as $\mathcal{R}^2[i] \times \mathrm{R}^2$ (such as: Fig.2, Fig4, Fig.5, Fig.6, Fig.7 and Column 1 in Fig.8), while Fig.1 and Fig.3 just shows $\mathcal{R}^2[i]$.

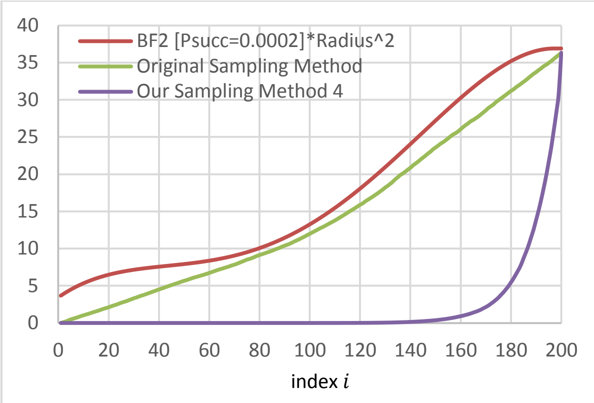


Fig.2. Comparison of expected value of entries of samples by original sampling method by (18) as $\mathrm{E}[\ell^2_{t=d-i+1}]$ and our sampling method 4 by Algorithm 3 for bounding function of BF2 [Psucc=0.0002]

Test 2. Some Observations for Step 1, Step 2 and Step 3 in Our Sampling Method 5: In this test, by using the configuration in Table 1, Fig.3 shows the upper-bound bounding functions of $\mathcal{R}_{\mathrm{UP1}}$ (as explained in Step 1 in Section 3) and $\mathcal{R}_{\mathrm{UP2}}$ (as explained in Step 1 in Section 3) and lower-bound bounding function of $\mathcal{R}_{\mathrm{LOW}}$ (as explained in Step 2 in Section 3) for an input bounding function "BF2 [Psucc=0.0002]".

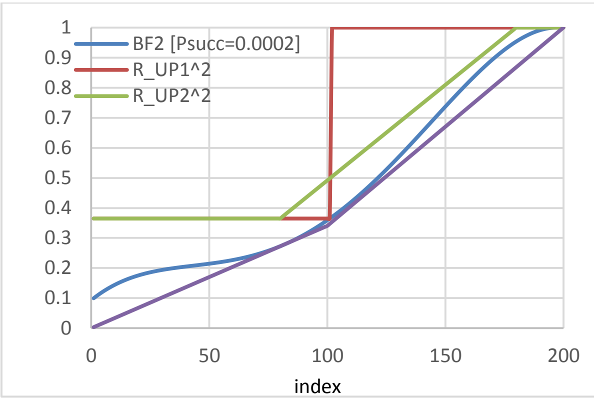
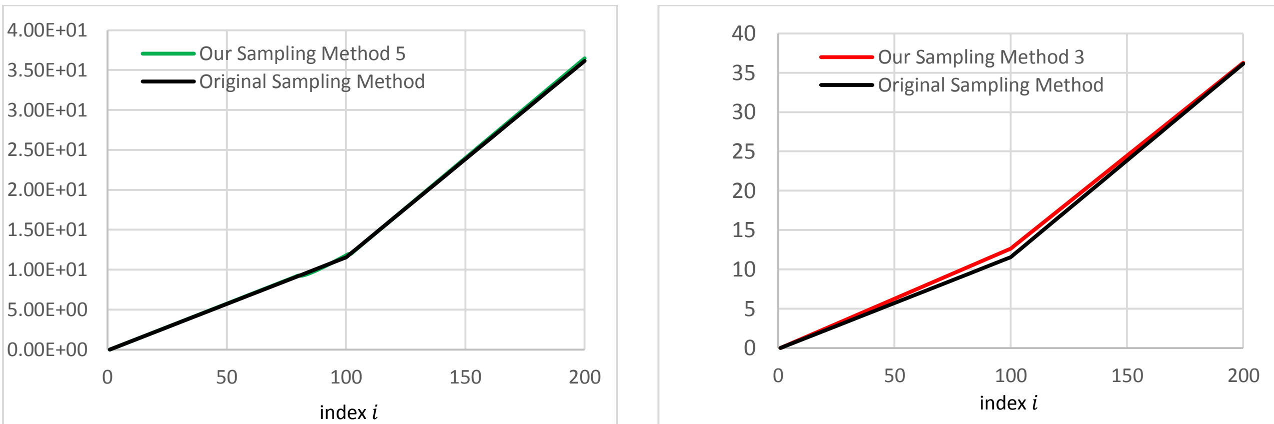


Fig.3. Upper-bound and Lower-bound bounding functions for input bounding function of "BF2 [Psucc=0.0002]"

Note: As mentioned, the entries of all bounding functions in the test results of this section (including "BF2 [Psucc=0.0002]" in Fig.3) are squared.



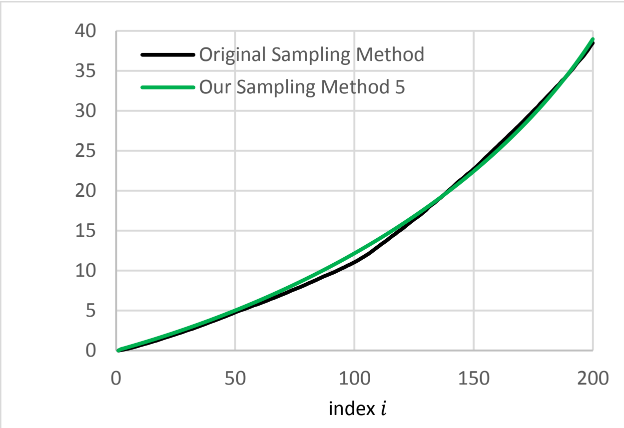
(1) Comparison of samples of "Original Sampling Method" and "Our Sampling Method 5"

(2) Comparison of samples of "Original Sampling Method" and "Our Sampling Method 3"

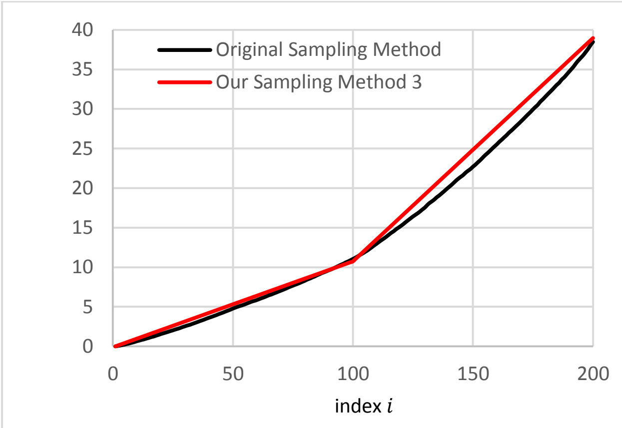
Fig.4. Comparison of $\mathrm{E}[\ell^2_{t=d-i+1}]$ as the expected value of entries of samples for Original Sampling Method by (18) and Our Sampling Method 5 by Algorithm 2 and Our Sampling Method 3 for Step bounding function of "BF6 [Psucc=0.000147]"

***Test 3. Verification of Claim 1*:** In this test, by using the configuration in Table 1, our test results for nearly 20000 successful samples verify the exactness of Claim 1, since as shown in Fig.4, the expected value of samples by original sampling method are more close to the expected value of samples by our sampling method 5 than the one by our sampling method 3. In fact, as shown in Fig.4, Claim 1 can be assumed as a strong assumption.

***Test 4. Verification of Claim 2*:** In this test, by using the configuration in Table 1, our test results for nearly 20000 successful samples verify the exactness of Claim 2, since as shown in Fig.5, the expected value of samples by original sampling method are more close to the expected value of samples by our sampling method 5 than the one by our sampling method 3. However, as shown in Fig.5, Claim 2 is not as exact as Claim 1 (see Fig.4), this claim (Claim 2) can be assumed as a better assumption than using "Our sampling method 3"!



(1) Comparison of samples of "Original Sampling Method" and "Our Sampling Method 5"



(2) Comparison of samples of "Original Sampling Method" and "Our Sampling Method 3"

Fig.5. Comparison of $\mathrm{E}[\ell_{t=d-i+1}^2]$ as the expected value of entries of samples for Original Sampling Method by (18) and Our Sampling Method 5 by Algorithm 3 and Our Sampling Method 3 for Piecewise-linear bounding function of "BF5 [Psucc=0.00000011]"

***Test 5. Some Observations for Claim 3*:** Fig.6 shows that how the estimation of $\mathrm{E}[\ell_{t=d-i+1}^2]$ in "Our Sampling Method 5" can be drawn between the estimations of $\mathrm{E}[\ell_{t=d-i+1}^2]$ for $\mathcal{R}_{\mathrm{UP1}}$ and $\mathcal{R}_{\mathrm{LOW}}$. The sufficient comparison of the estimation of $\mathrm{E}[\ell_{t=d-i+1}^2]$ in "Our Sampling Method 5" with "Our Sampling Method 3" is introduced in Test 6.

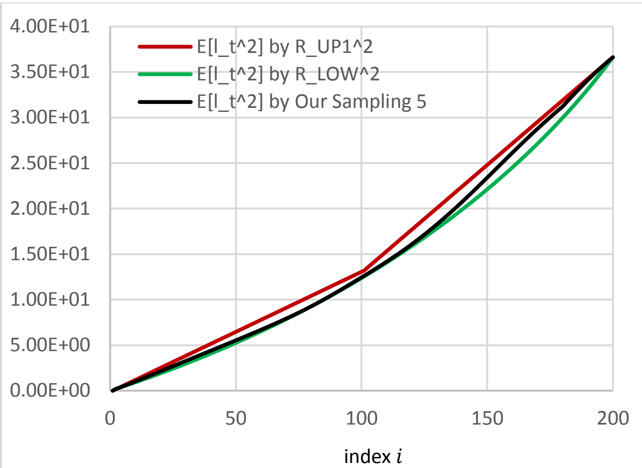
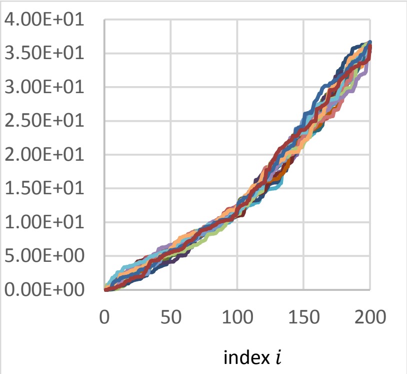
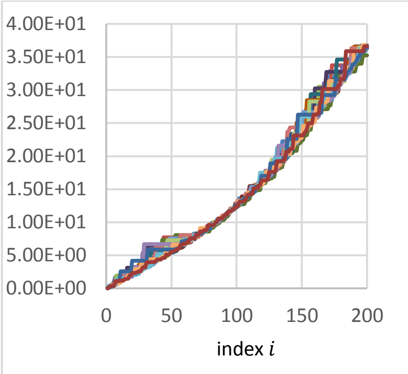


Fig.6. Estimation of $\mathrm{E}[\ell_{t=d-i+1}^2]$ as the expected value of entries of samples for in "Our Sampling Method 5" between the estimations of $\mathrm{E}[\ell_{t=d-i+1}^2]$ for $\mathcal{R}_{\mathrm{UP1}}$ and $\mathcal{R}_{\mathrm{LOW}}$ for input bounding function of "BF2 [Psucc=0.0002]"

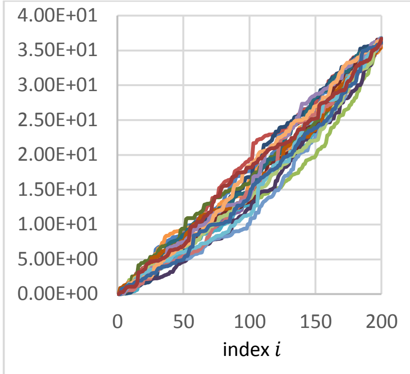
***Test 6. Final Comparison of "Our Sampling Method 5" and "Our Sampling Method 3"*:** In this test, by using the configuration in Table 1, we introduce our main (and final) comparison of "Our Sampling Method 5" and "Our Sampling Method 3" based on statistical measurements. Fig.7 shows the shapes of 20 samples of $\ell_{t=d-i+1}^2$ for input bounding function of "BF2 [Psucc=0.0002]" which are generated by "Original Sampling Method", "Our Sampling Method 5" and "Our Sampling Method 3". Based on Fig.7, this is shown that the samples of "Original Sampling Method" are more close to the samples of "Our Sampling Method 5" than the one by "Our Sampling Method 3". Also Fig.8 introduces our best comparison between the samples generated by "Original Sampling Method", "Our Sampling Method 5" and "Our Sampling Method 3". This test uses nearly 100 successful samples for each methods in Fig.8.



(1) By "Original Sampling Method"

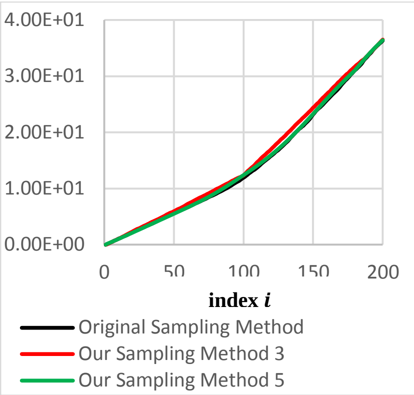


(2) By "Our Sampling Method 5"

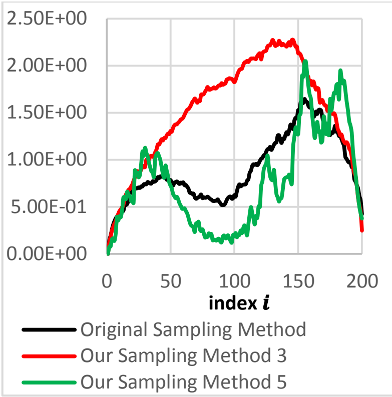


(3) By "Our Sampling Method 3"

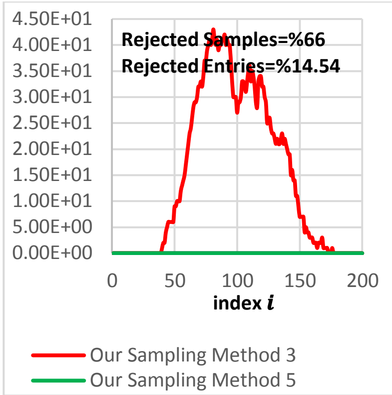
Fig.7. The shape of 20 samples of $\ell^2_{t=d-i+1}$ for input bounding function of "BF2 [Psucc=0.0002]"



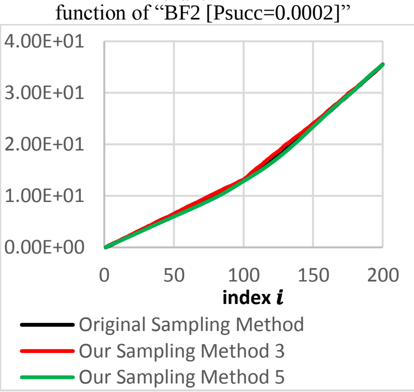
(1-a) Shape of E[$\ell^2_{t=d-i+1}$] for input bounding function of "BF2 [Psucc=0.0002]"



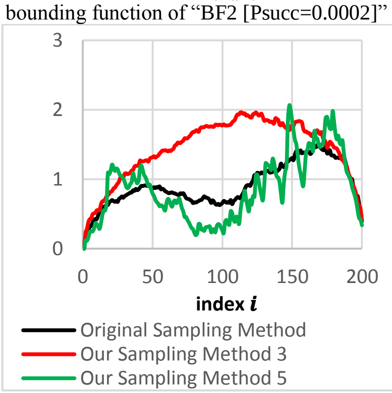
(1-b) Shape of STD[$\ell^2_{t=d-i+1}$] for input bounding function of "BF2 [Psucc=0.0002]"



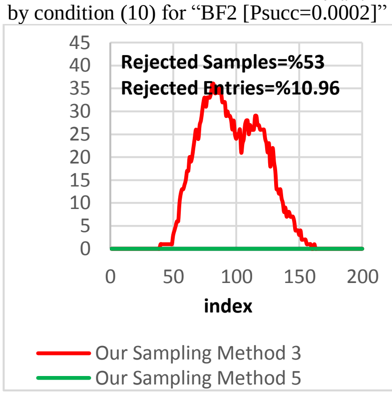
(1-c) Number of rejected entries of $\ell^2_{t=d-i+1}$ by condition (10) for "BF2 [Psucc=0.0002]"



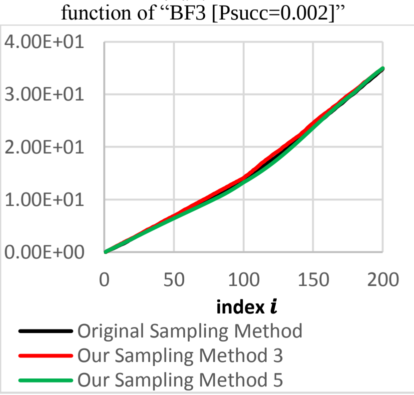
(2-a) Shape of E[$\ell^2_{t=d-i+1}$] for input bounding function of "BF3 [Psucc=0.002]"



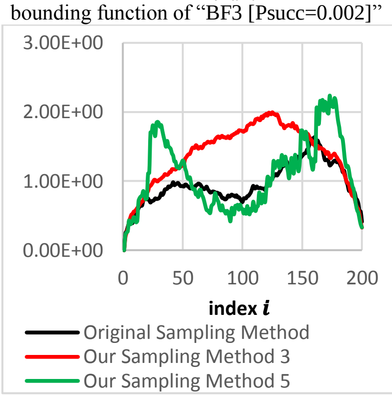
(2-b) Shape of STD[$\ell^2_{t=d-i+1}$] for input bounding function of "BF3 [Psucc=0.002]"



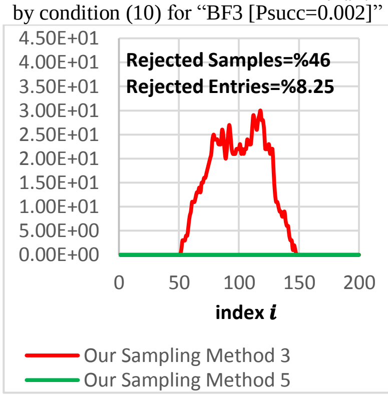
(2-c) Number of rejected entries of $\ell^2_{t=d-i+1}$ by condition (10) for "BF3 [Psucc=0.002]"



(3-a) Shape of E[$\ell^2_{t=d-i+1}$] for input bounding function of "BF4 [Psucc=0. 01163]"



(3-b) Shape of STD[$\ell^2_{t=d-i+1}$] for bounding function of "BF4 [Psucc=0. 01163]"



(3-c) Number of rejected entries of $\ell^2_{t=d-i+1}$ by condition (10) for "BF4 [Psucc=0. 01163]"

Fig.8. Statistical specifications of nearly 100 assumed successful samples of coefficient vector ℓ or z for three input bounding functions by methods of "Original Sampling Method", "Our Sampling Method 3" and "Our Sampling Method 5"

Fig.8 shows the superiorities of “Our Sampling Method 5” over “Our Sampling Method 3” based on statistical measurements:

- The column 1 in Fig.8 shows that the expected value of the entries of the samples by “Original Sampling Method” are more close to the ones by “Our Sampling Method 5” than the ones by “Our Sampling Method 3”, however Fig.4 and Fig.5 introduce better comparisons for showing this superiority of “Our Sampling Method 5” over “Our Sampling Method 3”.
- The column 2 in Fig.8 shows that the standard deviation of the entries of the samples by “Original Sampling Method” are more close to the ones by “Our Sampling Method 5” than the ones by “Our Sampling Method 3”.
- The column 3 in Fig.8 shows the surprising number of violations of main condition of (10) by each entry of the coefficient vector samples of $\ell^2_{t=d-i+1}$ or z_i^2 in “Our Sampling Method 3”, while the number of violations of main condition of (10) by each entry of the samples by “Our Sampling Method 5” are strictly zero! Also, the total number of rejected samples and total number of violations of main condition of (10) by each entry of the samples in “Our Sampling Method 3” are written on the corresponding figures in column 3 of Fig.8.

5. Conclusion

For predicting the manner of BKZ in higher block sizes, practical running is not the way, therefore BKZ-simulations are introduced. One of the main components in design of BKZ-simulation with GNR enumeration is sampling the enumeration solution v. Our idea in Lemma 3 from [5] introduces a suitable method for sampling the norm of enumeration solution v, also first precise and explicit analysis of sampling the coefficient vector y is introduced in [5]. Our idea in Lemma 3 from [5] is accepted in this paper, but our sampling methods of 1, 2 and 3 cannot be accepted as exact sampling methods of coefficient vectors and would be revised in this paper.

Based on the test results of this paper and [5], if the success probability of bounding function would be much small or enumeration radius tends to the solution norm (i.e., $\boldsymbol{R} \approx \|\boldsymbol{v}\|$), the proposed sampling method 1, 2 and 3 would be less accurate. A new idea for approximating the expected value of coefficient vector entries of enumeration solution is introduced which is more close to the expected value of corresponding entries in basic rejection sampling in Algorithm 6 from [2]. There is no analysis on variance of each entry in coefficient vector of enumeration solution in [5], while our test results in this paper show disappointing variance for these three sampling methods (our sampling method 1, 2, 3)! Also a novel idea for approximating this variance is introduced which is more close to the variance of corresponding entries in basic rejection sampling in Algorithm 6 from [2]. Moreover there is no analysis on the number of violations from main condition of bounding function at each entry in coefficient vector of enumeration solution in [5] (see this main condition in relation (10)), while our test results in this paper show surprising much number of violations for these three sampling methods (our sampling method 1, 2, 3)! For example, for some arbitrary bounding functions with success probability of $\boldsymbol{p_{succ} = 0.011}$ up to $\boldsymbol{p_{succ} = 0.0002}$, the number of rejected samples would be 46% up to 66%. By using these new achievements and analysis, our sampling method 1, 2 and 3 (for sampling coefficient vectors) are revised. Our new revised method is not a rejection sampling, this means that, it includes no violation from main condition of GNR bounding function at each entry of coefficient vectors (see this main condition in relation (10)). Also the expected value and variance of each entry of coefficient vectors by our new sampling method are nearly more close to our new exact approximation of expected value and variance of corresponding entries in Algorithm 6 from [2] than the entries of coefficient vectors by our sampling method 1, 2, 3.

In fact, for designing a BKZ-simulation, even our novel idea for approximating the expected value of corresponding entries in Algorithm 6 from [2] is sufficiently exact (instead of sampling by “Our Sampling Method 5” in this paper), while its variance and number of violations at each entry are absolutely zero. In other words, using our new sampling method just tries to inject more reasonable randomness to the BKZ-simulation. However this work introduces some acceptable sampling method for coefficient vectors of GNR-enumeration, more efficient and more exact samplers are expected to be studied and introduced in future researches.

Appendix A. "Our Sampling Method 4"

Algorithm 3 shows the pseudo-code of "Our Sampling Method 4".

Algorithm 3 Sample_of_ℓ_i^2_Ver4 (Our Sampling Method 4)

Input: GSO norms of block given as $B_{[1,d]}^* = [b_1^*, \ldots, b_d^*]$ at stage j, any type of bounding function $\mathcal{R}$, enumeration radii R, Sampled norm of enumeration solution as ℓ_{new}, last nonzero index of $\mathcal{g}$.

$\ell_1^2 = \ell_{new}^2$;
for$(i = 2; i \leq \mathcal{g} - 1; i++)\{$
  gh $= \mathrm{GH}\left(\pi_i\left(\mathcal{L}_{[i,d]}\right)\right)$;
  $\mathrm{R}_{new} = \min(\mathrm{R} \times \mathcal{R}_{d-i+1}, \ell_{i-1})$;
  $r_{\mathrm{FAC}} = \frac{\mathrm{R}_{new}}{\mathrm{gh}}$;
  for$(l = i; l \leq d; l++)\left\{\mathcal{R}_{d-l+1} \leftarrow \frac{\min(\mathrm{R} \times \mathcal{R}_{d-l+1}, \ell_{i-1})}{\mathrm{R}_{new}};\right\}$
  $\mathcal{R}_{new} = [\mathcal{R}_1, \ldots, \mathcal{R}_{d-i+1}]$; //$\mathcal{R}_{new}$ is a new bounding function with length of $d - i + 1$
  $\mathrm{R} \leftarrow \mathrm{R}_{new}$;
  $\mathrm{p} = p_{succ}(\mathcal{R}_{new})$;
  $\mathrm{V} = \mathrm{Determinant}([\pi_i(b_i), \pi_i(b_{i+1}), \ldots, \pi_i(b_d)])$;
  $\ell_i^2 = \mathrm{Sample}_{\|v\|}(d - i + 1, \mathrm{p}, \mathrm{gh}, r_{\mathrm{FAC}}, \mathrm{V}, v_{d-i+1})$; //$v_{d-i+1}$ as volume of $(d - i + 1)$_dimensional ball with unit radius
}//end for
$\ell_{\mathcal{g}}^2 = \left\|b_{\mathcal{g}}^*\right\|^2$;

Output: Sampled norms of $\ell_1^2, \ell_2^2, \ldots, \ell_{\mathcal{g}}^2$.

Appendix B. Sub-functions of "Our Sampling Method 5"

The pseudo-codes of all sub-functions of "Our Sampling Method 5" are introduced in this appendix. As mentioned, the block size d in all algorithms of this appendix is assumed even. Algorithm 4 generates a Step bounding function including the entries which are closely lower-bounded by input bounding function $\mathcal{R}$.

Algorithm 4 Get_UpperBound_of_$\mathcal{R}$_1

Input: Block size d, bounding function $\mathcal{R}$.

$\mathrm{a} = \mathcal{R}^2\left[\frac{d}{2}\right] + 0.000001$;
for$\left(i = 1; i \leq \frac{d}{2}; i++\right)$ $\mathcal{R}_{\mathrm{UP1}}^2[i] = \mathrm{a}$;
for$\left(i = \frac{d}{2} + 1; i \leq d; i++\right)$ $\mathcal{R}_{\mathrm{UP1}}^2[i] = 1$;

Output: Upper_bound bounding function $\mathcal{R}_{\mathrm{UP1}}$.

Algorithm 5 generates a modification of upper-bound Step bounding function of $\mathcal{R}_{\mathrm{UP1}}$, as $\mathcal{R}_{\mathrm{UP2}}$, including the entries which are closely lower-bounded by input bounding function $\mathcal{R}$.

Algorithm 5 Get_UpperBound_of_$\mathcal{R}$_2

Input: Block size d, bounding function $\mathcal{R}$, Upper_bound bounding function $\mathcal{R}_{\mathrm{UP1}}$.

$\mathrm{a} = \mathcal{R}_{\mathrm{UP1}}^2[1]$;
$\mathrm{A} = \frac{1-\mathrm{a}}{0.5d}$;
for$(i = 0.9d; i \geq 0.4d; i--)\{$//begin for1
  $y = \mathrm{A}(i - 0.4d) + \mathrm{a}$;
  if$(y < \mathcal{R}_i^2)$ $\mathrm{A} \leftarrow \frac{\mathcal{R}_i^2 - \mathrm{a}}{i - 0.4d}$;}//end for1
$i = 1$;
for$(; i < 0.4d; i++)$ $\mathcal{R}_{\mathrm{UP2}}^2[i] = \mathrm{a}$;
for(;;){//begin for2
  $y = \mathrm{A}(i - 0.4d) + \mathrm{a}$;
  if$(y \leq 1)$ $\{\mathcal{R}_{\mathrm{UP2}}^2[i] = y;\ i++;\}$
  else break;} //end for2
for$(; i \leq d; i++)$ $\mathcal{R}_{\mathrm{UP2}}^2[i] = 1$;

Output: Upper_bound bounding function $\mathcal{R}_{\mathrm{UP2}}$.

Algorithm 6 defines a piecewise-linear bounding function including the entries which are closely upper-bounded by input bounding function $\mathcal{R}$.

Algorithm 6 Get_LowerBound_of_$\mathcal{R}$

Input: Block size d, bounding function $\mathcal{R}$, Last non_zero index of g.

$\mathrm{a} = 1;$
$i = d - g + 1;$
$\mathrm{for}(; i \le d/2; i++)\{$
$\quad \mathrm{if}\left(\frac{2i\mathrm{a}}{d} > \mathcal{R}^2[i]\right)\ \mathrm{a} = \frac{\mathcal{R}^2[i] \times d}{2i};\}$
$\mathrm{for}(i = d/2 + 1; i \le d; i++)\{$
$\quad \mathrm{if}\left(2a - 1 + \frac{2i(1-a)}{d} > \mathcal{R}^2[i]\right)\{$
$\quad\quad \mathrm{a} = \frac{\mathcal{R}_i^2 d + d - 2i}{2d - 2i};\}\}$
$i = 1;$
$\mathrm{for}(; i \le d/2; i++)\{\mathcal{R}^2_{\mathrm{LOW}}[i] = 2i\mathrm{a}/d;\}$
$\mathrm{for}(; i \le d/2; i++)\{\mathcal{R}^2_{\mathrm{LOW}}[i] = 2\mathrm{a} - 1 + 2i(1 - \mathrm{a})/d;\}$

Output: Lower_bound bounding function $\mathcal{R}_{\mathrm{LOW}}$.

Algorithm 7 returns the expected value of ℓ_t^2 for an input bounding function $\mathcal{R}$ and lattice block $\mathcal{L}_{[1,d]}$.

Note: The notation of $\mathrm{E}[\ell_t^2]$ shows the expected value of ℓ_t^2, also $\mathrm{e}\ell_i^2$ shows the value of $\mathrm{E}[\ell_{d-i+1}^2]/\|v\|^2$ (i.e., the index of i in $\mathrm{e}\ell_i^2$ and bounding functions $\mathcal{R}[i]$ is consistent, while $i = d - t + 1$ in $\mathrm{E}[\ell_t^2]$).

Algorithm 7 Get_ExpectedValues_of_ℓ_t^2

Input: GSO norms of block given as $B^*_{[1,d]} = [b_1^*, \dots, b_d^*]$, bounding function $\mathcal{R}$, enumeration radii R,
Sampled norm of enumeration solution as ℓ_1, last nonzero index of g, bounding functions of $\mathcal{R}_{\mathrm{LOW}}$, $\mathcal{R}_{\mathrm{UP1}}$ and $\mathcal{R}_{\mathrm{UP2}}$.

$\left[\mathrm{E}[\ell_1^2]_{\mathrm{up}}, \dots, \mathrm{E}[\ell_d^2]_{\mathrm{up}}\right] \leftarrow \mathrm{Get\_E}[\ell_t^2]\mathrm{\_of\_}\mathcal{R}_{\mathrm{Step}}\left(B^*_{[1,d]}, \mathcal{R}_{\mathrm{UP1}}, \mathrm{R}, \ell_1, g\right);$
$[\mathrm{E}[\ell_1^2]_{\mathrm{low}}, \dots, \mathrm{E}[\ell_d^2]_{\mathrm{low}}] \leftarrow \mathrm{Get\_E}[\ell_t^2]\mathrm{\_of\_}\mathcal{R}_{\mathrm{Piece}}\left(B^*_{[1,d]}, \mathcal{R}_{\mathrm{LOW}}, \mathrm{R}, \ell_1, g\right);$
$\mathrm{for}(i = 1; i \le d; i++)\{$
$\quad \mathrm{E}[\ell_{d-i+1}^2] \leftarrow \frac{(\mathcal{R}^2[i] - \mathcal{R}^2_{\mathrm{LOW}}[i]) \times \mathrm{E}[\ell_{d-i+1}^2]_{\mathrm{up}} + (\mathcal{R}^2_{\mathrm{UP2}}[i] - \mathcal{R}^2[i]) \times \mathrm{E}[\ell_{d-i+1}^2]_{\mathrm{low}}}{\mathcal{R}^2_{\mathrm{UP2}}[i] - \mathcal{R}^2_{\mathrm{LOW}}[i]};\}$

Output: $\mathrm{E}[\ell_1^2], \dots, \mathrm{E}[\ell_d^2]$ for bounding function $\mathcal{R}$.

Algorithm 8 introduces the pseudo-code of one of the main sub-function of Algorithm 7.

Algorithm 8 Get_E[ℓ_t^2]_of_$\mathcal{R}_{\mathrm{Step}}$

Input: GSO norms of block given as $B^*_{[1,d]} = [b_1^*, \dots, b_d^*]$, Step bounding function $\mathcal{R}$, enumeration radii R,
Sampled norm of enumeration solution as ℓ_1, last nonzero index of g.

$\mathrm{gh} = \mathrm{GH}\left(\pi_{d/2+1}\left(\mathcal{L}_{[1,d]}\right)\right);$
$r_{\mathrm{FAC}} = \mathrm{R} \times \mathcal{R}[d/2]/\mathrm{gh};$
$\mathrm{E}\left[\ell_{d/2+1}^2\right] = \min\left(\frac{\mathrm{R}^2}{2}, \mathrm{R}^2\mathcal{R}^2[d/2]\left(1 - \frac{4}{d}\right)\right);$
$\mathrm{A} = \frac{\mathrm{E}\left[\ell_{d/2+1}^2\right] - \|b_g^*\|^2}{g - d/2 - 1};$
$\mathrm{for}(i = 1; i \le d/2; i++)\{$
$\quad \mathrm{E}[\ell_{d-i+1}^2] \leftarrow \mathrm{A} \times (i - d + g - 1) + \|b_g^*\|^2;\}$
$\mathrm{A} = \frac{\ell_1^2 - \mathrm{E}[\ell_{d/2+1}^2]}{d/2};$
$\mathrm{for}(i = d/2 + 1; i \le d; i++)\{$
$\quad \mathrm{E}[\ell_{d-i+1}^2] \leftarrow \mathrm{A} \times (i - d/2) + \mathrm{E}[\ell_{d/2+1}^2];\}$

Output: $\mathrm{E}[\ell_1^2], \dots, \mathrm{E}[\ell_d^2]$ for Step bounding function $\mathcal{R}$.

Algorithm 9 introduces the pseudo-code of other main sub-function of Algorithm 7.

Algorithm 9 Get_E[ℓ_t^2]_of_$\mathcal{R}_{\mathrm{Piece}}$

Input: GSO norms of block given as $B^*_{[1,d]} = [b_1^*, \dots, b_d^*]$, Piecewise_linear bounding function $\mathcal{R}$, enumeration radii R,
Sampled norm of enumeration solution as ℓ_1, last nonzero index of g.

$\mathrm{p}_0 = \frac{p_{succ}(\mathcal{R})}{d};$
$\mathrm{step} = 1/32; \varepsilon = \mathrm{p}_0/100;$
$\mathrm{for}(\mathrm{p} = 0; \mathrm{p} < \mathrm{p}_0 - \varepsilon; \mathrm{step} = \mathrm{step} \times 2)\{$//begin for1
$\quad \mathrm{B} = \mathrm{B} + \mathrm{step};$
$\quad \mathrm{A} = \mathrm{B} + \sqrt{2\mathrm{B}^2 + 2\mathrm{B}};$
$\quad \mathrm{for}(i = 1; i \le d; i++)\{$
$\quad\quad \mathrm{e}\ell_i^2 = 1 + \mathrm{A} - \sqrt{(1 + \mathrm{A} + \mathrm{B})^2 - (i/d + \mathrm{A})^2};\}$
$\quad \mathrm{for}(i = 1; i \le d; i++)\{$
$\quad\quad \mathrm{e}\ell_i^2 \leftarrow \left((\mathrm{e}\ell_i^2 - \mathrm{e}\ell_1^2) \times \frac{1 - \|b_g^*\|^2/\|v\|^2}{1 - \mathrm{e}\ell_1^2} + \|b_g^*\|^2/\|v\|^2\right);\}$
$\quad \mathrm{p} = p_{succ}(e\ell^2);\}$//end for1

$\quad$ $\text{step} = \frac{\text{step}}{2}$; $\text{B} = \text{B} - \text{step}$; $\text{A} = \text{B} + \sqrt{2\text{B}^2 + 2\text{B}}$;
$\quad$ while(true){//begin while1
$\quad\quad$ if($\text{p}_0 - \varepsilon \le \text{p}$ &&$\text{p} \le \text{p}_0 + \varepsilon$) break;
$\quad\quad$ for($i = 1; i \le d; i++$){
$\quad\quad\quad$ $\text{e}\ell_i^2 = 1 + \text{A} - \sqrt{(1 + \text{A} + \text{B})^2 - (i/d + \text{A})^2}$;}
$\quad\quad$ for($i = 1; i \le d; i++$){
$\quad\quad\quad$ $\text{e}\ell_i^2 \leftarrow \left((\text{e}\ell_i^2 - \text{e}\ell_1^2) \times \frac{1 - \|b_g^*\|^2/\|v\|^2}{1 - \text{e}\ell_1^2} + \|b_g^*\|^2/\|v\|^2 \right)$;}
$\quad\quad$ $\text{p} = p_{succ}(e\ell^2)$;
$\quad\quad$ if($\text{p} < \text{p}_0 - \varepsilon$){
$\quad\quad\quad$ $\text{step} = \frac{\text{step}}{2}$; $\text{B} = \text{B} + \text{step}$; $\text{A} = \text{B} + \sqrt{2\text{B}^2 + 2\text{B}}$;}
$\quad\quad$ if($\text{p}_0 + \varepsilon < \text{p}$){
$\quad\quad\quad$ $\text{step} = \frac{\text{step}}{2}$; $\text{B} = \text{B} - \text{step}$; $\text{A} = \text{B} + \sqrt{2\text{B}^2 + 2\text{B}}$;}
$\quad$ }//end while1
$\quad$ for($i = 1; i \le d; i++$) {$\text{E}[\ell_{d-i+1}^2] \leftarrow \text{e}\ell_i^2 \times \|v\|^2$;}
Output: $\text{E}[\ell_1^2], \ldots, \text{E}[\ell_d^2]$ for Piecewise_linear bounding function $\mathcal{R}$.

Algorithm 10 returns the variance of ℓ_t^2 for an arbitrary bounding function $\mathcal{R}$ and lattice block $\mathcal{L}_{[1,d]}$.

Algorithm 10 Get_Variances_of_ℓ_t^2
Input: GSO norms of block given as $B_{[1,d]}^* = [b_1^*, \ldots, b_d^*]$, Step bounding function $\mathcal{R}$, enumeration radii R, Sampled norm of enumeration solution as ℓ_1, $\left[\text{E}[\ell_1^2], \ldots, \text{E}[\ell_d^2]\right]$ for bounding function $\mathcal{R}$.
$\quad$ $\mathcal{R}_{\text{full}} \leftarrow [1,1,\ldots,1]$;
$\quad$ $[\text{E}[\ell_1^2]_{\text{full}}, \ldots, \text{E}[\ell_d^2]_{\text{full}}] \leftarrow$ Get_E[ℓ_t^2]_of_$\mathcal{R}_{\text{Step}}\left(B_{[1,d]}^*, \mathcal{R}_{\text{full}}, \text{R}, \ell_1, g_0 = d\right)$;
$\quad$ for($i = 1; i \le d; i++$){
$\quad\quad$ $\mathcal{V}[\ell_{d-i+1}^2]_{\text{full}} = (\|v\|^2 - \|b_d^*\|^2)^2 \frac{2id - 2i^2 - 2d + 2i}{d^3 - d^2 - d + 1}$;
$\quad\quad$ $\mathcal{R}_{\text{UP3}}^2[i] = \min\left(\frac{0.7}{R^2}\left(\text{E}[\ell_{d-i+1}^2]_{\text{full}} + 3 \times \sqrt{\mathcal{V}[\ell_{d-i+1}^2]_{\text{full}}}\right) + 0.3, 1\right)$;
$\quad\quad$ $\mathcal{V}[\ell_{d-i+1}^2] \leftarrow \mathcal{V}[\ell_{d-i+1}^2]_{\text{full}} \times \min\left(\left(\frac{R^2\mathcal{R}^2[i] - \text{E}[\ell_{d-i+1}^2]}{R^2\mathcal{R}_{\text{UP3}}^2[i] - \text{E}[\ell_{d-i+1}^2]_{\text{full}}}\right)^2, 1\right)$;}
Output: $[\mathcal{V}[\ell_1^2], \ldots, \mathcal{V}[\ell_d^2]]$ for bounding function $\mathcal{R}$.

Algorithm 11 generates the samples of ℓ_t^2 in "Our Sampling Method 5" efficiently, for an arbitrary bounding function $\mathcal{R}$ and lattice block $\mathcal{L}_{[1,d]}$.

Algorithm 11 Get_Sample_of_ℓ_t^2
Input: GSO norms of block given as $B_{[1,d]}^* = [b_1^*, \ldots, b_d^*]$, Sampled norm of enumeration solution as ℓ_1, last nonzero index of g, enumeration radii R, $\left[\text{E}[\ell_1^2], \ldots, \text{E}[\ell_d^2]\right]$ for bounding function $\mathcal{R}$, $[\mathcal{V}[\ell_1^2], \ldots, \mathcal{V}[\ell_d^2]]$ for bounding function $\mathcal{R}$.
$\quad$ for($t = d; t \ge g + 1; t--$) $\ell_t^2 \leftarrow 0$;
$\quad$ $\ell_g^2 \leftarrow \|b_g^*\|^2$;
$\quad$ $\varepsilon = \ell_1^2/d$;
$\quad$ for($t = g - 1; t \ge 2; t--$){
$\quad\quad$ $\Delta\text{E}_t = \max(\text{E}[\ell_t^2] - \ell_{t+1}^2, \varepsilon)$;
$\quad\quad$ $\text{Y}_t \leftarrow \text{Gamma}\left(k = \frac{(\Delta\text{E}_t)^2}{\mathcal{V}[\ell_t^2]}, \theta = \frac{\mathcal{V}[\ell_t^2]}{\Delta\text{E}_t}\right)$;
$\quad\quad$ $\ell_t^2 \leftarrow \min(\min(\mathcal{R}^2[d - t + 1] \times \text{R}^2, \|v\|^2), \ell_{t+1}^2 + \max(\text{Y}_t + \min(\text{E}[\ell_t^2] - \ell_{t+1}^2 - \varepsilon, 0), 0))$;
$\quad$ }
$\quad$ $//\ell_1^2 \leftarrow \|v\|^2$;
Output: samples of $\ell_1^2, \ldots, \ell_d^2$ for bounding function $\mathcal{R}$.

Appendix C. Proof of Lemma 1

If sampled solution of $v = \pi_1(v)$ would be projected on $\text{span}(b_1)^\perp$ as $\pi_2(v)$, the under-laying lattice block of $[\pi_1(b_1), \pi_1(b_2), \ldots, \pi_1(b_d)]$ is modified into $[\pi_2(b_2), \pi_2(b_3), \ldots, \pi_2(b_d)]$, then this projected solution of $v = \pi_1(v)$ can be re-written with coefficient vector y by using (12) as follows:

$$\pi_1(v) = (y_1, y_2, \cdots, y_d) \begin{bmatrix} \pi_1(b_1) \\ \vdots \\ \pi_1(b_d) \end{bmatrix} = (y_1, y_2, \cdots, y_d) \begin{bmatrix} b_1^* \\ \vdots \\ b_d^* + \sum_{t=1}^{d-1} \mu_{d,t} b_t^* \end{bmatrix}, \quad (26)$$

$$\pi_2(v) = (y_2, y_3, \cdots, y_d)\begin{bmatrix}\pi_2(b_2)\\ \vdots \\ \pi_2(b_d)\end{bmatrix} = (y_2, y_3, \cdots, y_d)\begin{bmatrix} b_2^* \\ \vdots \\ b_d^* + \sum_{t=2}^{d-1}\mu_{d,t}b_t^*\end{bmatrix}. \tag{27}$$

For more projection over $\text{span}(b_1, b_2, \ldots)^{\perp}$, this projected solution can be re-written as follows:

$$\vdots$$

$$\pi_l(v) = (y_l, y_{l+1}, \cdots, y_d)\begin{bmatrix}\pi_l(b_l)\\ \vdots \\ \pi_l(b_d)\end{bmatrix} = (y_l, y_{l+1}, \cdots, y_d)\begin{bmatrix} b_l^* \\ \vdots \\ b_d^* + \sum_{t=l}^{d-1}\mu_{d,t}b_t^*\end{bmatrix}, \tag{28}$$

$$\vdots$$

$$\pi_g(v) = (y_g, y_{g+1}, \cdots, y_d)\begin{bmatrix}\pi_g(b_g)\\ \vdots \\ \pi_g(b_d)\end{bmatrix} = (y_g, y_{g+1}, \cdots, y_d)\begin{bmatrix} b_g^* \\ \vdots \\ b_d^* + \sum_{t=g}^{d-1}\mu_{d,t}b_t^*\end{bmatrix}. \tag{29}$$

where $y_g = 1$, $y_{g+1} = 0$, …, $y_d = 0$.

The norm of enumeration solution v can be expanded on the original lattice block of $[\pi_1(b_1),, \ldots, \pi_1(b_d)]$ as follows:

$$\|\pi_1(v)\|^2 = \left(y_1 + \sum_{t=2}^{d} y_t\mu_{t,1}\right)^2\|b_1^*\|^2 + \cdots + \left(y_g + \sum_{t=g+1}^{d} y_t\mu_{t,g}\right)^2\left\|b_g^*\right\|^2 + \cdots + y_d^2\|b_d^*\|^2 =$$
$$\left(y_1 + \sum_{t=2}^{g} y_t\mu_{t,1}\right)^2\|b_1^*\|^2 + \cdots + \left(y_{g-1} + \mu_{g,g-1}\right)^2\left\|b_{g-1}^*\right\|^2 + \left\|b_g^*\right\|^2 = \left(y_1 + \sum_{t=2}^{g} y_t\mu_{t,1}\right)^2\|b_1^*\|^2 + \|\pi_2(v)\|^2, \tag{30}$$

where $y_g = 1$ and $y_{g+1} = 0 \ldots y_d = 0$.

Note: The sampled norm of $\|\pi_l(v)\|^2$ is shown by ℓ_l^2 in this paper (also $\|\pi_1(v)\|^2$ is shown by $\ell_{new}^2 = \ell_1^2$).

The sampled norm of $\|\pi_2(v)\|^2$ shown by ℓ_2^2 in Algorithm 1, can be expanded as follows:

$$\|\pi_2(v)\|^2 = \left(y_2 + \sum_{t=3}^{g} y_t\mu_{t,2}\right)^2\|b_2^*\|^2 + \cdots + \left(y_{g-1} + \mu_{g,g-1}\right)^2\left\|b_{g-1}^*\right\|^2 + \left\|b_g^*\right\|^2 =$$
$$\left(y_2 + \sum_{t=3}^{g} y_t\mu_{t,2}\right)^2\|b_2^*\|^2 + \|\pi_3(v)\|^2. \tag{31}$$

For more projection of $\pi_l(v)$ up to $\pi_g(v)$, the norm of projected solution can be expanded as follows (note: the sampled norm of $\|\pi_l(v)\|^2$ is shown by ℓ_l^2 in this paper):

$$\vdots$$

$$\|\pi_l(v)\|^2 = \left(y_l + \sum_{t=l+1}^{g} y_t\mu_{t,l}\right)^2\|b_l^*\|^2 + \cdots + \left(y_{g-1} + \mu_{g,g-1}\right)^2\left\|b_{g-1}^*\right\|^2 + \left\|b_g^*\right\|^2 =$$
$$\left(y_l + \sum_{t=l+1}^{g} y_t\mu_{t,l}\right)^2\|b_l^*\|^2 + \|\pi_{l+1}(v)\|^2. \tag{32}$$

$$\vdots$$

$$\left\|\pi_{g-1}(v)\right\|^2 = \left(y_{g-1} + \mu_{g,g-1}\right)^2\left\|b_{g-1}^*\right\|^2 + \left\|b_g^*\right\|^2 = \left(y_{g-1} + \mu_{g,g-1}\right)^2\left\|b_{g-1}^*\right\|^2 + \left\|\pi_g(v)\right\|^2. \tag{33}$$

$$\left\|\pi_g(v)\right\|^2 = \left\|b_g^*\right\|^2. \tag{34}$$

At this point, the coefficient vector of y can be sampled from y_d down to y_1 (in inverse direction) as follows:

By using Theorem 2 in [5]: $y_g = 1$ and $y_{g+1} = y_{g+2} = \cdots = y_d = 0$.

By using (33) and (34): $y_{g-1} = \frac{\sqrt{\left\|\pi_{g-1}(v)\right\|^2 - \left\|b_g^*\right\|^2}}{\left\|b_{g-1}^*\right\|} - \mu_{g,g-1}$, where $\left\|\pi_{g-1}(v)\right\|^2$ is sampled as ℓ_{g-1}^2.

By using (32) for each y_l where $1 \le l \le g$: $y_l = \frac{\sqrt{\|\pi_l(v)\|^2 - \|\pi_{l+1}(v)\|^2}}{\|b_l^*\|} - \sum_{t=l+1}^{g} y_t\mu_{t,l}$, where $\|\pi_l(v)\|^2$ is sampled in Algorithm 1 as ℓ_l^2 and also $y_{l+1}, \ldots, y_g$ are sampled (found) in previous steps.

Finally, by using (14) and (13), lines 7-10 in Algorithm 1 compute the corresponding coefficients of w_t and z_{d-t+1} for $1 \le t \le d$.

Note: This lemma can be assumed as a well start point for entering the theorem proving methods in this context. In fact, this is possible to use formal methods by some theorem prover such as Isabelle/HOL in proving Lemma 1 by using the basic ideas, structures and propositions which are proposed in [15].

Appendix D. Proof of Lemma 3

The cutting point in full-enumeration with bounding function $\mathcal{R} = (1,1,1,\dots,1)$ is always $\mathrm{Cut} = d$. Also, by using Lemma 7 in [5] for full-enumeration, this is expected to $g \approx d$. The variance of X_i is defined as follows:

$$\mathcal{V}[\mathrm{X}_i] = \mathrm{E}[\mathrm{X}_i^2] - \mathrm{E}[\mathrm{X}_i]^2. \tag{35}$$

At first, the parameter of A_i is defined as follows: $\sum_{t=1}^{d} z_t^2 = \|v\|^2 \Rightarrow$

By using Theorem 2 in [5]: $\sum_{t=2}^{d} z_t^2 = \|v\|^2 - \|b_d^*\|^2 \Rightarrow \sum_{t=2}^{d} \frac{z_t^2}{\|v\|^2-\|b_d^*\|^2} = 1 \Rightarrow$

By sampling method of (18) in this lemma (for full-enumeration): $z_{d-l+1} = (-1)^{\lfloor \mathrm{rand}_{[0\dots2)} \rfloor} \cdot \sqrt{\frac{\omega_l \cdot \left(\|v\|^2-\|b_d^*\|^2\right)}{\sum_{t=1}^{d-1} \omega_t}} \Rightarrow$

$$\mathrm{A}_i = \sum_{t=2}^{i} \frac{z_t^2}{\|v\|^2-\|b_d^*\|^2} = \frac{\sum_{t=d-i+1}^{d-1} \omega_t}{\sum_{t=1}^{d-1} \omega_t}, \tag{36}$$

where $\omega_t \leftarrow \mathrm{Gamma}(1/2,2)$. The value of A_i in (36) is defined as $\mathrm{X}_i = \sum_{t=1}^{i} \frac{z_t^2}{R^2}$ in [1], since enumeration radius R in [1] is assumed to be $R = \|v\|$, also paper [1] dose not notice the fact by Theorem 2 in [5]. In Section 4 from [5], to show our results, we use the definition of $\mathrm{X}_i = \sum_{t=1}^{i} \frac{z_t^2}{R^2}$. Therefore, the definition of $\mathrm{X}_i = \sum_{t=1}^{i} \frac{z_t^2}{R^2}$ is not equal to $\mathrm{A}_i = \sum_{t=2}^{i} \frac{z_t^2}{\|v\|^2-\|b_d^*\|^2}$, especially when $\|v\| < R$.

$$\mathrm{X}_i = \sum_{t=1}^{i} \frac{z_t^2}{R^2} = \frac{\|b_d^*\|^2}{R^2} + \sum_{t=2}^{i} \frac{z_t^2}{R^2} = \frac{\|b_d^*\|^2}{R^2} + \frac{\|v\|^2-\|b_d^*\|^2}{R^2} \sum_{t=2}^{i} \frac{z_t^2}{\|v\|^2-\|b_d^*\|^2} \Rightarrow$$

$$\mathrm{X}_i = \frac{\|b_d^*\|^2}{R^2} + \frac{\|v\|^2-\|b_d^*\|^2}{R^2} \mathrm{A}_i. \tag{37}$$

For invariant factors of $\frac{\|b_d^*\|^2}{R^2}$ and $\frac{\|v\|^2-\|b_d^*\|^2}{R^2}$, by using the definition of variance and relation (37):

$$\mathcal{V}[\mathrm{X}_i] = \mathcal{V}\left[\frac{\|b_d^*\|^2}{R^2} + \frac{\|v\|^2-\|b_d^*\|^2}{R^2} \mathrm{A}_i\right] = \left(\frac{\|v\|^2-\|b_d^*\|^2}{R^2}\right)^2 \mathcal{V}[\mathrm{A}_i]. \tag{38}$$

Also, by using the definition of variance:

$$\mathcal{V}[\mathrm{A}_i] = \mathrm{E}[\mathrm{A}_i^2] - \mathrm{E}[\mathrm{A}_i]^2. \tag{39}$$

Step 1: In this step the value of $\mathrm{E}[\mathrm{A}_i^2]$ is estimated. The parameter of A_i^2 can be expanded as follows:
$\mathrm{A}_i^2 = \left(\sum_{t=2}^{i} \frac{z_t^2}{\|v\|^2-\|b_d^*\|^2}\right)^2 = \left(\frac{\sum_{t=d-i+1}^{d-1} \omega_t}{\sum_{t=1}^{d-1} \omega_t}\right)^2$, where $\omega_t \leftarrow \mathrm{Gamma}(1/2,2)$.

For index of $x = d - i + 1$: $\mathrm{A}_i^2 = \left(\frac{\sum_{t=x}^{d-1} \omega_t}{\sum_{t=1}^{d-1} \omega_t}\right)^2 = \frac{\left(\sum_{t=x}^{d-1} \omega_t\right)\left(\sum_{t=x}^{d-1} \omega_t\right)}{\left(\sum_{t=1}^{d-1} \omega_t\right)\left(\sum_{t=1}^{d-1} \omega_t\right)} \Rightarrow \mathrm{A}_i^2 = \frac{\sum_{t=x}^{d-1} \omega_t^2 + \sum_{x \le t \le d-1, x \le r \le d-1, t \ne r} (\omega_t \omega_r)}{\sum_{t=1}^{d-1} \omega_t^2 + \sum_{1 \le t \le d-1, 1 \le r \le d-1, t \ne r} (\omega_t \omega_r)}$.

In other side, for random variable A_i^2, since $\mathrm{A}_i^2 > 0$, the function $\varphi(\mathrm{A}_i^2) = \frac{1}{\mathrm{A}_i^2}$ is convex, therefore by using Jensen's inequality [13]: $\mathrm{A}_i^2 > 0 \Rightarrow \frac{1}{\mathrm{E}[\mathrm{A}_i^2]} \le \mathrm{E}\left[\frac{1}{\mathrm{A}_i^2}\right]$.

$\frac{1}{\mathrm{A}_i^2} = \frac{\mathrm{Y}_1+\mathrm{Y}_2}{\mathrm{Y}_2}$, where $\mathrm{Y}_1 = \left(\sum_{t=1,r=1}^{t=x-1,r=x-1} \omega_t \omega_r\right) + \left(\sum_{t=1,r=x}^{t=x-1,r=d-1} \omega_t \omega_r\right) + \left(\sum_{t=x,r=1}^{t=d-1,r=x-1} \omega_t \omega_r\right)$ and $\mathrm{Y}_2 = \left(\sum_{t=x,r=x}^{t=d-1,r=d-1} \omega_t \omega_r\right)$.

The random variables of Y_1 and Y_2 are independent. By using Fact 2 in [5] and Approximation 1 in [5]:

$$\frac{1}{\mathrm{E}[\mathrm{A}_i^2]} \approx \frac{\mathrm{E}[\mathrm{Y}_1]}{\mathrm{E}[\mathrm{Y}_2]} + 1 \Rightarrow \frac{1}{\mathrm{E}[\mathrm{A}_i^2]} \approx \frac{\mathrm{E}\left[\sum_{t=1,r=1}^{t=x-1,r=x-1} \omega_t \omega_r\right] + \mathrm{E}\left[\sum_{t=1,r=x}^{t=x-1,r=d-1} \omega_t \omega_r\right] + \mathrm{E}\left[\sum_{t=x,r=1}^{t=d-1,r=x-1} \omega_t \omega_r\right]}{\mathrm{E}\left[\sum_{t=x,r=x}^{t=d-1,r=d-1} \omega_t \omega_r\right]} + 1 \Rightarrow$$

Remark 1. In approximation of variance in this context, following facts is used:

- $E[\omega_i\omega_j] = E[\omega_j] \times E[\omega_i] = 1$ for $i \neq j$ where $\omega_t \in \text{Gamma}(1/2,2)$,
- $E[\omega_i\omega_i] = 3$ where $\omega_t \in \text{Gamma}(1/2,2)$.

By using Remark 1: $\frac{1}{\mathrm{E}[\mathrm{A}_i^2]} \approx \frac{d^2-1}{i^2-1} \Rightarrow$

$$\mathrm{E}[\mathrm{A}_i^2] \approx \frac{i^2-1}{d^2-1}. \tag{40}$$

Step 2: In this step the value of $\mathrm{E}[\mathrm{A}_i]^2$ is estimated. For random variable $\mathrm{A}_i = \frac{\sum_{t=x}^{d-1}\omega_t}{\sum_{t=1}^{d-1}\omega_t}$ where $x = d - i + 1$, since $\mathrm{A}_i > 0$, the function $\varphi(\mathrm{A}_i) = \frac{1}{\mathrm{A}_i}$ is convex, therefore by using Jensen's inequality [13]: $\mathrm{A}_i > 0 \Rightarrow \frac{1}{\mathrm{E}[\mathrm{A}_i]} \leq \mathrm{E}\left[\frac{1}{\mathrm{A}_i}\right]$.

$$\frac{1}{\mathrm{A}_i} = \frac{\sum_{t=1}^{d-1}\omega_t}{\sum_{t=x}^{d-1}\omega_t}, \text{ where } \mathrm{Y}_1 = \sum_{t=1}^{x-1}\omega_t \text{ and } \mathrm{Y}_2 = \sum_{t=x}^{d-1}\omega_t.$$

The random variables of Y_1 and Y_2 are independent. By using Fact 2 in [5] and Approximation 1 in [5]:

$$\frac{1}{\mathrm{E}[\mathrm{A}_i]} \approx \frac{\mathrm{E}[\mathrm{Y}_1]}{\mathrm{E}[\mathrm{Y}_2]} + 1 \Rightarrow \frac{1}{\mathrm{E}[\mathrm{A}_i]} \approx \frac{\mathrm{E}[\sum_{t=1}^{x-1}\omega_t]}{\mathrm{E}[\sum_{t=x}^{d-1}\omega_t]} + 1 \Rightarrow$$

By using Remark 1: $\frac{1}{\mathrm{E}[\mathrm{A}_i]} \approx \frac{d-1}{i-1} \Rightarrow \mathrm{E}[\mathrm{A}_i] \approx \frac{i-1}{d-1} \Rightarrow$

$$\mathrm{E}[\mathrm{A}_i]^2 \approx \frac{(i-1)^2}{(d-1)^2} = \frac{i^2+1-2i}{d^2+1-2d}. \tag{41}$$

By using (39), (40) and (41): $\mathcal{V}[\mathrm{A}_i] = \mathrm{E}[\mathrm{A}_i^2] - \mathrm{E}[\mathrm{A}_i]^2 = \frac{i^2-1}{d^2-1} - \frac{(i-1)^2}{(d-1)^2} = \frac{2id-2i^2-2d+2i}{d^3-d^2-d+1}$.

By using (38): $\mathcal{V}[\mathrm{X}_i] = \left(\frac{\|v\|^2-\|b_d^*\|^2}{R^2}\right)^2 \frac{2id-2i^2-2d+2i}{d^3-d^2-d+1}$.

References

[1] N. Gama, P. Q. Nguyen, and O. Regev, "Lattice enumeration using extreme pruning", In Proc. EUROCRYPT '10, volume 6110 of LNCS. Springer, 2010.

[2] Y. Chen, and P. Q. Nguyen, "BKZ 2.0: Better lattice security estimates", In International Conference on the Theory and Application of Cryptology and Information Security, pp. 1-20. Springer Berlin Heidelberg, 2011.

[3] M. R. Albrecht, B. R. Curtis, A. Deo, A. Davidson, R. Player, E. W. Postlethwaite, F. Virdia, and T. Wunderer. "Estimate all the {LWE, NTRU} schemes!", In SCN, pages 351–357, 2018.

[4] Shi Bai, Damien Stehlé, Weiqiang Wen, "Measuring, Simulating and Exploiting the Head Concavity Phenomenon in BKZ", ASIACRYPT 2018.

[5] Moghissi, G., Payandeh, A. (2021). "Better Sampling Method of Enumeration Solution for BKZ-Simulation", *The ISC International Journal of Information Security*, 13(2), pp. 177-208. doi: 10.22042/isecure.2021.225886.531.

[6] L. Ducas, "Shortest vector from lattice sieving: A few dimensions for free", In EUROCRYPT, pages 125-145, 2018.

[7] Zhongxiang Zheng, Xiaoyun Wang, Yang Yu, "Orthogonalized lattice enumeration for solving SVP", Science China Information Sciences 2018.

[8] Moghissi, Gholam Reza, and Ali Payandeh. *"A Parallel Evolutionary Search for Shortest Vector Problem."* International Journal of Information Technology and Computer Science, (2019).

[9] Y. Aono, Y. Wang, T. Hayashi, and T. Takagi, "Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator", In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pp. 789-819. Springer, Berlin, Heidelberg, 2016.

[10] Moghissi, Gholam Reza, and Ali Payandeh. *"Using Progressive Success Probabilities for Sound-pruned Enumerations in BKZ Algorithm."* International Journal of Computer Network and Information Security 9.9 (2018): 10.

[11] D. Micciancio and O. Regev, "Lattice-based cryptography", In Post-quantum cryptography, pp. 147-191, Springer Berlin Heidelberg, 2009.

[12] Moghissi, Gholam Reza, and Ali Payandeh. "Rejecting claimed speedup of $2^{\beta/2}$ in extreme pruning and revising BKZ 2.0 for better speedup", *Journal of Computing and Security*, 2021; 8(1): 65-91. doi: 10.22108/jcs.2021.121191.1044.

[13] Jensen, Johan Ludwig William Valdemar, *"Sur les fonctions convexes et les inégalités entre les valeurs moyennes"*, *Acta mathematica* 30 (1906): 175-193.

[14] Gholam Reza Moghissi, Ali Payandeh, *"Optimal bounding function for GNR-enumeration"*, International Journal of Mathematical Sciences and Computing (IJMSC), Vol.8, No.1, pp. 1-17, 2022. DOI: 10.5815/ijmsc.2022.01.01.

[15] Moghissi, Gholam Reza, and Ali Payandeh. "*Formal Verification of NTRUEncrypt Scheme*." International Journal of Computer Network and Information Security 8.4 (2016): 44.

Authors' Profiles

Gholam Reza Moghissi received the M.S. degree in the Department of ICT at Malek-e-Ashtar University of Technology, Tehran, Iran, in 2016. His researches focus on Information Security.

Ali Payandeh received the M.S. degree in Electrical Engineering from Tarbiat Modares University in 1994, and the Ph.D. degree in Electrical Engineering from K.N. Toosi University of Technology (Tehran, Iran) in 2006. He is now an assistant professor in the Department of Information and Communications Technology at the Malek-e-Ashtar University of Technology, Iran. He has published many papers in international journals and conferences. His research interests include information theory, coding theory, cryptography, security protocols, secure communications, and satellite communications.

How to cite this paper: Gholam Reza Moghissi, Ali Payandeh, "Revised Method for Sampling Coefficient Vector of GNR-enumeration Solution", International Journal of Mathematical Sciences and Computing(IJMSC), Vol.8, No.3, pp. 1-20, 2022. DOI:10.5815/ijmsc.2022.03.01