Improved Lattice Enumeration Algorithms by Primal and Dual Reordering Methods

Kazuki Yamamura[1] and Yuntao Wang[2⋆] Eiichiro Fujisaki[2]

[1] NTT Social Informatics Laboratories kazuki.yamamura.by@iecl.ntt.co.jp
[2] School of Information Science, Japan Advanced Institute of Science and Technology
y-wang@jaist.ac.jp fujisaki@jaist.ac.jp

Abstract. The security of lattice-based cryptosystems is generally based on the hardness of the Shortest Vector Problem (SVP). There are two common categories of lattice algorithms to solve SVP: search algorithms and reduction algorithms. The original enumeration algorithm (ENUM) is one of the former algorithms which run in exponential time due to the exhaustive search. Further, ENUM is used as a subroutine for the BKZ algorithm, which is one of the most practical reduction algorithms. It is a critical issue to reduce the computational complexity of ENUM. In this paper, first, we improve the mechanism in the so-called reordering method proposed by Wang in ACISP 2018. We call this improvement Primal Projective Reordering (PPR) method which permutates the projected vectors by decreasing norms; therefore it performs better to reduce the number of search nodes in ENUM. Then, we propose a Dual Projective Reordering (DPR) method permutating the projected vectors in its dual lattice. In addition, we propose a condition to decide whether the reordering method should be adopted or not. Preliminary experimental results show that our proposed reordering methods can successfully reduce the number of ENUM search nodes comparing to the predecessor, e.g., PPR reduces around 9.6% on average in 30-dimensional random lattices, and DPR reduces around 32.8% on average in 45-dimensional random lattices. Moreover, our simulation shows that the higher the lattice dimension, the more the proposed reordering method can reduce ENUM search nodes.

Keywords: Lattice cryptography, Enumeration algorithm, Reordering method, Dual lattice

1 Intruduction

1.1 Background

Cryptosystems such as RSA [13] and ECC [9,11] are currently used to protect private information, relying on hard mathematical problems like integer factoring problem (IFP) and discrete logarithm problem (DLP). However, if a quantum computer is developed in the near future, it can be compromised by quantum

⋆ corresponding author, ORCID: 0000-0002-2872-4508.

algorithms such as Shor's algorithm [16], which can solve IFP and DLP in polynomial time. Therefore, we need to move forward with post-quantum cryptography (PQC) as soon as possible. Hence NIST officially started PQC standardization project in 2016 and announced third round PQC candidates in 2020, including four public key cryptography (CRYSTALS-KYBER, NTRU, SABER, and Classic McEliece) and three digital signature schemes (CRYSTALS-DILITHIUM, FALCON, Rainbow). Among these candidates, around 70% are lattice-based cryptosystems [1]. At the evaluating stage, cryptanalysis is essential to work. Namely, it is necessary to decide on secure and practical parameters. The security of lattice-based cryptography such as NTRU [8] and LWE-based schemes [12] depend on the hardness of some lattice problems, such as SVP, CVP, and their variants. The analysis of the concrete hardness of these problems is essential to decide the proper parameter settings. In particular, TU Darmstadt published open problems called the lattice challenge [5] to analyze the practical hardness of lattice problems. In order to evaluate our proposed methods, We utilize a random lattice provided by the lattice challenge.

Various algorithms for SVP have been proposed, and they can be classified into two main categories. First, there are lattice basis reduction algorithms, which convert a bad basis of a lattice into a good one, such as LLL reduction (Lenstra-Lenstra-Lovász) reduction and BKZ (block Korkin-Zolotarev) reduction [15,4,3]. LLL reduction is a remarkable lattice reduction that runs in polynomial time; therefore, we often use the LLL reduction to get a good basis before using other algorithms to solve hard lattice problems. In our experiments, we use LLL reduction, which is implemented in the open-source library NTL [17].

BKZ reduction, proposed by Schnorr and Euchner, was a hybrid LLL reduction and ENUM algorithms that finds the shortest vector of a lattice. Given a lattice basis and a block size parameter β, BKZ reduction reduced the block size lattice basis by the LLL reduction before inputting them into ENUM iteratively. Since the complexity of ENUM is much larger than LLL reduction, the complexity of BKZ reduction depends on ENUM.

Second, lattice point search algorithms such as ENUM [15] and Sieve [2]. ENUM proposed in the same paper of BKZ reduction is an exhaustive search method that finds the shortest lattice vector by the depth-first search in a tree constructed with nodes labeled by coefficients. The complexity of ENUM is $2^{O(n^2)}$ for a given n-dimensional lattice basis. As mentioned above, the BKZ reduction depends on the complexity of ENUM; therefore, we can see that it is important to work to reduce the complexity of ENUM for BKZ reduction.

The sieve algorithm proposed by Ajtai is well-known lattice point search algorithms, which requires a runtime of $2^{0.292n+O(n)}$ and exponential memory of $2^{0.2n+(n)}$ in lattice dimension n.

Quick reordering technique (QRT) [18,19] is an initial reordering method applied in lattice reduction. QRT reorders the output reduced basis vectors by their decreasing norm, which is applied in BKZ reduction to reduce the number of search nodes in ENUM by a certain probability.

1.2 Our contribution

The BKZ, one of the lattice basis reductions, is one of the most powerful algorithms known today. The runtime of the BKZ depends on the complexity of a subroutine called ENUM, an enumeration search algorithm. In other words, reducing the runtime of ENUM leads to reducing that of the BKZ, which is expected to bring us much closer to the SVP solution. In this paper, we examine how much the runtime of ENUM can be reduced by changing the order of the input basis into ENUM, using the properties of projective and dual lattices. Our contributions are as follows:

1. improving the previous reordering method using the property of projected lattice;
2. proposing a reordering method using the property of dual lattice;
3. proposing condition to decide whether the reordering method should be adopted or not.

Our experimental results show that the proposed improvement method for the previous method, named **PPR**, can reduce around 9.6% on average in 30-dimensional random lattices and that the proposed reordering method using the property of dual lattice, named **DPR**, can successfully reduce the large number of ENUM search nodes, e.g., reducing around 32.8% on average in 45-dimensional random lattices. Moreover, we experimentally show that DPR overcomes the QRT weakness that the reduction of the search node number decreases as the dimension increase and can increase the reduction as the dimension increase.

1.3 Organization

We introduce mathematical backgrounds in section 2 and the details of some classic lattice algorithms and the quick reordering technique (QRT), which is invited to improve ENUM in section 3. In section 4, we propose our new reordering methods, which are called PPR and DPR, and condition t_R to decide whether the reordering method should be adopted or not. We then present the experimental results on our proposed method in section 5. Finally, the conclusion is given in section 6.

2 Preliminaries

Given n linearly independent vectors $\mathbf{B} := (\mathbf{b}_1, \cdots, \mathbf{b}_n) \in \mathbb{R}^m$, the lattice generated by them is defined as $L(\mathbf{b}_1, \ldots, \mathbf{b}_n) := \{\sum_{i=1}^{n} v_i \mathbf{b}_i \mid v_i \in \mathbb{Z}\}$. Here, n is the rank of L, and m is the dimension of L. If n=m, L is called a full-rank lattice. The fundamental domain of L corresponding to this basis $\mathbf{B}$ is the set $P(L) := \{\sum_{i=1}^{n} x_i \mathbf{b_i} \mid 0 \leq x_i < 1\}$ called the fundamental parallelepiped of L. $vol(P(L))$ is called the volume of a lattice L which depends on the basis $\mathbf{B}$.

Shortest Vector Problem (SVP). A lattice L has at least nonzero shortest vectors. The Shortest Vector Problem (SVP) is to find one shortest nonzero

vector of L given the basis B of lattice L, which is expected to be very difficult to solve in polynomial time.

Gram-Schmidt Orthogonalization (GSO). Gram-Schmidt Orthogonalization (GSO) is a classic procedure in linear algebra that creates a set of orthogonal vectors given a set of linearly independent vectors. It works by projecting each vector on the space orthogonal to the span of the previous vectors in order from front to front of linearly independent vectors. Note that on changing the order of the inputs of linearly independent vectors, Gram-Schmidt Orthogonalization outputs different orthogonal vectors. We denote by $\mathbf{B}^* = (\mathbf{b}_1^*, \cdots, \mathbf{b}_n^*)$ Gram-Schmidt orthogonal vector (GSO vectors) of the given lattice basis $\mathbf{B} = (\mathbf{b}_1, \cdots \mathbf{b}_n)$. The GSO vectors of a linearly independent vectors $\mathbf{B}$ in order from front to front can be computed as follows:

$$\begin{cases} \mathbf{b}_1^* := \mathbf{b}_1 \\ \\ \mathbf{b}_i^* := \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{i,j} \mathbf{b}_j^* \ (2 \leq i \leq n) \end{cases}$$

where

$$\mu_{i,j} := \frac{\langle \mathbf{b}_i, \mathbf{b}_j^* \rangle}{\|\mathbf{b}_j^*\|^2} \ (1 \leq j < i \leq n)$$

and let $\langle \cdot, \cdot \rangle : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ to be inner product.

The GSO vector of a linearly independent vectors $\mathbf{B}$ in order from back to front can be computed as follows

$$\begin{cases} \mathbf{b}_n^\dagger := \mathbf{b}_n \\ \\ \mathbf{b}_i^\dagger := \mathbf{b}_i - \sum_{j=n}^{i+1} \nu_{i,j} \mathbf{b}_j^\dagger \ (1 \leq i \leq n-1) \end{cases}$$

where

$$\nu_{i,j} := \frac{\langle \mathbf{b}_i, \mathbf{b}_j^\dagger \rangle}{\|\mathbf{b}_j^\dagger\|^2} \ (1 \leq i < j \leq n)$$

Note that the volume of $L(\mathbf{B})$ can also be computed by $vol(L(\mathbf{B})) = \prod_{i=1}^n \|\mathbf{b}_i^*\|$. For any $1 \leq i \leq n$, let $\pi_i : \mathbb{R}^n \to span_{\mathbb{R}}(\mathbf{b}_1, \cdots, \mathbf{b}_{i-1})^\perp$ be an orthogonal projection of a vector onto $span_{\mathbb{R}}(\mathbf{b}_1, \cdots, \mathbf{b}_{i-1})^\perp$. We also denote by $\pi_i(L)$ a projective sublattice with basis vectors of $(\pi_i(\mathbf{b}_i), \ldots, \pi_i(\mathbf{b}_n)$.

Dual lattice. We define the notation of the dual of a lattice and see its applications. We denote the dual lattice of L by $\hat{L} := \{\mathbf{x} \in span_{\mathbb{R}}(L) \mid \langle \mathbf{x}, \mathbf{y} \rangle \in \mathbb{Z} \ (\forall \mathbf{y} \in L)\}$, where let $\langle \cdot, \cdot \rangle : \mathbb{R}^m \times \mathbb{R}^m \to \mathbb{R}$ to be inner product. If $\mathbf{D} := (\mathbf{d}_1, \cdots, \mathbf{d}_n)$ is the basis of a dual lattice $\hat{L}(\mathbf{B})$ then $\mathbf{D} = (\mathbf{B}\mathbf{B}^{\mathrm{T}})^{-1}\mathbf{B}$. A dual lattice has good properties: let $\{\mathbf{b}_1^*, \cdots, \mathbf{b}_n^*\}$ be GSO vectors of lattice basis $\mathbf{B}$ and let

$\{\mathbf{d}_1^\dagger, \cdots, \mathbf{d}_n^\dagger\}$ be GSO vectors of dual lattice basis $\mathbf{D}$ in reverse order, then for all i,

$$\|\mathbf{b}_i^*\| \cdot \|\mathbf{d}_i^\dagger\| = 1.$$

The properties of the dual lattice have been applied to attacks to lattice cryptography [6,10]. Our proposed algorithm also uses the properties of a dual lattice.

Gaussian heuristic. Given a lattice L and a continuous subset C of $\mathbb{R}$, we can estimate the number of points in $C \cap L$ approximately $vol(C)/vol(L)$, which is called the Gaussian heuristic. Using Gaussian heuristic, We can estimate the shortest lattice vector norm approximately, denoted by $GH(L) := \sqrt{\frac{n}{2\pi e}} vol(L)^{\frac{1}{n}}$.

Geometric series assumption (GSA) The geometric assumption (GSA)[14] says that the norms of GSO vectors $\|\mathbf{b}_i^*\|$ in the LLL-reduced basis decline geometrically with quotient q such as $\|\mathbf{b}_i^*\|^2/\|\mathbf{b}_{i-1}^*\|^2 = q$ for $i = 1, \cdots, n$ and $q \in [3/4, 1)$ Here, q is called the GSA constant, whose size depends on the reduction algorithm and the corresponding parameter setting.

3 Lattice Algorithms

This section introduces some classic lattice algorithms, such as the LLL basis reduction algorithm, the enumeration search algorithm (ENUM), and the BKZ algorithm [15]. Furthermore, we recall the quick reordering technique (QRT), which is invited to improve the previous algorithms.

3.1 LLL reduction

The LLL reduction, an approximation algorithm to SVP, was developed in 1982 by A.K.Lenstra, J.W.Lenstra, Jr., and L.Lovasz. Given a lattice basis and a parameter $3/4 < \delta_{LLL} < 1$, LLL reduction repeats size reduction and the swap of basis neighbors until the basis is a good one which means nearly orthogonal. Note that the closer δ_{LLL} is to one, the better the LLL-reduced basis is. Since LLL reduction terminates in the polynomial time, it is applied to many attacks on cryptosystems.

3.2 ENUM

We describe Schnorr-Euchner's enumeration algorithm (**ENUM**) [15] associated with our proposal algorithms. Given a lattice basis $\mathbf{B} = (\mathbf{b}_1, \cdots, \mathbf{b}_n)$, the inputs of the ENUM are GSO coefficients $(\mu_{i,j})_{1 \le j \le i \le n}$, the square norms $\{\|\mathbf{b}_1^*\|, \cdots, \|\mathbf{b}_n^*\|\}$ of $\mathbf{B}^*$ and search bound R which is usually $GH(L) \times 1.05$. The output is one shortest vector $\mathbf{v} = \sum_{i=1}^n u_i \mathbf{b}_i$, where $\{u_i\}_{i=1}^n$ is the set of integer coefficients that ENUM searches.

ENUM performs the depth-first search of the ENUM tree formed by half vectors in the projected lattice $\pi_n(L), \pi_{n-1}(L), \cdots, \pi_1(L)$ within the norm bound R. The depth of the ENUM tree is equal to the lattice dimension n. For $0 \le k \le n$,

Algorithm 1 Quick Reordering Technique (QRT)

Input: A basis $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$, index q'_n
Output: A basis $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$
if $n \geq 10$ **then**
  Compute the slope q_{curr} *of current GSO vector lengths by LSF.*
  if $q_{curr} < q'_n$ **then**
    $\{\mathbf{b}'_1, \cdots, \mathbf{b}'_n\} \leftarrow$ *Normal Reordering*$(\{\mathbf{b}_1, \cdots, \mathbf{b}_n\})$
    Compute AveGSO and AveGSO′
    if *AveGSO* $\leq$ *AveGSO′* **then**
      $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\} \leftarrow \{\mathbf{b}'_1, \cdots, \mathbf{b}'_n\}$
      Compute GSO information.
    end if
  end if
end if

the nodes at depth k are half of the number of the vectors in the rank-k projected lattice $\pi_{n+1-k}(L)$ with norm $\leq R$. Therefore The Gaussian heuristic estimates of the number of nodes at depth k as:

$$H_k(R) := \frac{1}{2} \cdot \frac{vol(B_k(R))}{vol(\pi_{n+1-k}(L))} = \frac{1}{2} \cdot \frac{vol(B_k(R))}{\prod_{i=n+1-k}^{n} \|\mathbf{b}_i^*\|}$$

where $B_k(R)$ is the k-dimensional Euclidean ball of radius R centered around 0. Then the total number of search nodes in ENUM is approximately $N = \sum_{k=1}^{n} H_k(R) = \frac{1}{2} \cdot \sum_{k=1}^{n} \cdot \frac{vol(B_k(R))}{\prod_{i=n+1-k}^{n} \|\mathbf{b}_i^*\|}$. From [7], $H_k(R)$ is maximal around the middle depth $k \simeq n/2$(see an example of 30-dimension in Fig2)

3.3 BKZ reduction

The BKZ reduction is a powerful lattice reduction algorithm [3,4,15]. Given a lattice basis, one sets a proper blocksize $\beta \geq 2$ on which both the runtime and the output quality depend. Assuming that j is the first index of each local block $B_{j,min(j+\beta-1,n)}$, BKZ reduction iteratively performs the LLL reduction and the ENUM on each local block for j from 1 to $n-1$. Note that the ENUM subroutine is the most expensive part of the BKZ reduction. Therefore it is important to decrease the total number of search nodes in ENUM in order to reduce the runtime of BKZ reduction.

3.4 Quick Reordering Technique

The Quick Reordering Technique (QRT) [18,19] is a reordering method to reduce the runtime of the BKZ using ENUM as subroutines. Using a quick sort to reorder the input basis vectors by their decreasing norms (in this paper, we call this operation **Normal Reordering (NR)**. QRT can decrease the number of search nodes in ENUM with high probability both when the GSA assumption [14]

does not hold and when the average of $\|\mathbf{b}^*_{\lfloor n/2 \rceil -1}\|$, $\|\mathbf{b}^*_{\lfloor n/2 \rceil}\|$, $\|\mathbf{b}^*_{\lfloor n/2 \rceil +1}\|$ is bent larger after reordering the basis.

We show the QRT algorithm in **Algorithm 1**. The first step is to decide whether the basis is a good basis, i.e., nearly orthogonal. To do this, we calculate the constant q under the GSA assumption in the input basis using the least-squares method (LSF). LSF is a method frequently used in regression analysis. Given n points $\{(x_i, y_i) \,|\, 1 \leq i \leq n\}$, find the line $y = ax + b$ that minimizes the distance between these points. Specifically, it can be obtained by $a = \frac{\sum_{i=1}^{n} x_i\, y_i - n \cdot \bar{x}\, \bar{y}}{\sum_{i=1} n x_i^2 - n \cdot (\bar{x})^2}$ and $b = \bar{y} - a$. If $q_{curr} = a$ and $q_{curr} < q'_n$, i.e., the basis is judged to be bad, It calculates the GSO of $\{\mathbf{b}'_1, \cdots, \mathbf{b}'_n\}$ that is Normal Reordered and obtains vectors $\{\mathbf{b}'^*_1, \cdots, \mathbf{b}'^*_n\}$ before getting $AveGSO$ such that

$$AveGSO := \frac{\|\mathbf{b}^*_{\lfloor n/2 \rceil -1}\| + \|\mathbf{b}^*_{\lfloor n/2 \rceil}\| + \|\mathbf{b}^*_{\lfloor n/2 \rceil +1}\|}{3}$$

If $AveGSO < AveGSO'$ holds, we decide that the number of search nodes in ENUM is less for $\{\mathbf{b}'_1, \cdots, \mathbf{b}'_n\}$ than for $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$ and thus we set $\{\mathbf{b}'_1, \cdots, \mathbf{b}'_n\}$ to be ENUM input. The reason why we consider the ENUM to be less computationally intensive when $AveGSO < AveGSO'$ is that the number of search nodes in ENUM is highest around $\lfloor \frac{n}{2} \rceil$ and the surrounding GSO vector norm is large, the number of search nodes can decrease.

It has been pointed out that QRT can efficiently reduce the total ENUM runtime up to about 30 dimensions of the lattice, but the reduction decreases as the dimensionality increases.

4 Our Proposals

In this section, we propose two methods to decrease the number of search nodes N in ENUM. Recall that $N \approx \sum_{k=1}^{n} \frac{vol(B_k(R))}{\prod_{i=n-k+1}^{n} \|\mathbf{b}^*_i\|}$, which depends on the input basis. By enlarging the back half of the GSO vectors' norm, it is possible to decrease the number of search nodes in the ENUM process. Note that because the lattice volume is invariant, the shorter the front half of the GSO vectors' norm, the longer the back half of the GSO vectors' norm. In this paper, we propose two lattice basis reordering methods with the following strategies.

1. Shorten the former half of the GSO vectors' norm, which correspondingly lengthens the latter half of the GSO vectors' norm;
2. Directly lengthen the back half of the GSO vectors' norm.

Primal Projective Reordering (PPR) introduced in section 4.1 is based on the former strategy, while **Dual Projective Reordering (DPR)** introduced in section 4.2 is based on the latter strategy.

4.1 Primal Projective Reordring

First, we describe the Primal Projective Reordering (PPR) method to decrease the number of search nodes in ENUM by shortening the former half of the input

Algorithm 2 Primal Projective Reordering (PPR)

Input: A basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \in \mathbb{R}^m$
Output: A basis $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\} \in \mathbb{R}^m$ such that $\forall i \in \{1, \cdots, n\}$, $\|\mathbf{b}_i^*\| = \min_{i \le j \le n} \|\pi_i(\mathbf{b}_j)\|$
$(\mathbf{b}_0^*, \mathbf{b}_1^*, \cdots, \mathbf{b}_n^*) \leftarrow (\mathbf{0}, \mathbf{b}_1, \cdots, \mathbf{b}_n)$;
$(B_{-1}, B_0, B_1, \cdots, B_n) \leftarrow (\infty, 1, \|\mathbf{b}_1\|^2, \cdots, \|\mathbf{b}_n\|^2)$;
for $i = 1$ to $n - 1$ **do**
  $k \leftarrow -1$
  for $j = i$ to n **do**
    $t \leftarrow \langle \mathbf{b}_j, \mathbf{b}_{i-1}^* \rangle$;
    $\mathbf{b}_j^* \leftarrow \mathbf{b}_j^* - \frac{t}{B_{i-1}} \mathbf{b}_{i-1}^*$;
    $B_j \leftarrow B_j - \frac{t^2}{B_{i-1}}$;
    if $B_k > B_j$ **then**
      $k \leftarrow j$;
    end if
  end for
  Swap$(\mathbf{b}_i, \mathbf{b}_k)$;
end for

GSO vectors' norm. As a result, the latter half of the GSO vectors' norm become larger. This strategy is similar to the previous study [18,19], Normal Reordering (NR) method, which directly reorders the input basis vectors by decreasing norm. Specifically, in PPR method, we move the shortest basis vector to ahead, and move the vector whose projective norm onto $\mathbf{b_1}$ is the shortest in the projected sublattice $\pi_1(L)$. Then we repeat this procedure and get a reordered basis. PPR can be regarded as an improved version of NR method.

We denote by $\mathbf{B} := \{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$ the lattice basis and by $\{\mathbf{b}_1^*, \cdots, \mathbf{b}_n^*\}$ the GSO vectors of $\mathbf{B}$. Then the PPR changes the lattice basis order such that

$$\begin{aligned}
\|\mathbf{b}_1^*\| &= \min\{\|\mathbf{b}_1\|, \cdots, \|\mathbf{b}_n\|\} \\
\|\mathbf{b}_2^*\| &= \min\{\|\pi_2(\mathbf{b}_2)\|, \cdots, \|\pi_2(\mathbf{b}_n)\|\} \\
&\vdots \\
\|\mathbf{b}_{n-1}^*\| &= \min\{\|\pi_{n-1}(\mathbf{b}_{n-1})\|, \|\pi_{n-1}(\mathbf{b}_n)\|\} \\
\|\mathbf{b}_n^*\| &= \|\pi_n(\mathbf{b}_n)\|
\end{aligned}$$

which is equivalent to

$$\forall i \in \{1, \cdots, n\},\ \|\mathbf{b}_i^*\| = \min_{i \le j \le n} \|\pi_i(\mathbf{b}_j)\|$$

We show the PPR in **Algorithm 2**. Note that for all $1 \le i \le n$

$$\pi_i(\mathbf{b}_j) = \mathbf{b}_j - \sum_{k=1}^{i-1} \frac{\langle \mathbf{b}_j, \mathbf{b}_k^* \rangle}{\|\mathbf{b}_k^*\|^2} \mathbf{b}_k^*.$$

Algorithm 3 Dual Projective Reordering (DPR)

Input: A basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \in \mathbb{R}^m$,
Output: $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\} \in \mathbb{R}^m$ such that $\forall i \in \{1, \cdots, n\},\ \|\mathbf{d}_i^\dagger\| = \min_{1 \le j \le i} \|\tau_i(\mathbf{d}_j)\|$
1: $\mathbf{D} \leftarrow (\mathbf{B}\mathbf{B}^{\mathrm{T}})^{-1}\mathbf{B}$;
2: $(\mathbf{d}_1, \cdots, \mathbf{d}_n) \leftarrow (\mathbf{d}_n, \cdots, \mathbf{d}_1),$;
3: PPR$((\mathbf{d}_1, \cdots, \mathbf{d}_n))$;
4: $(\mathbf{d}_1, \cdots, \mathbf{d}_n) \leftarrow (\mathbf{d}_n, \cdots, \mathbf{d}_1)$;
5: $\mathbf{B} \leftarrow (\mathbf{D}\mathbf{D}^{\mathrm{T}})^{-1}\mathbf{D}$;

Therefore,

$$\pi_i(\mathbf{b}_j) = \pi_{i-1}(\mathbf{b}_j) - \frac{\langle \mathbf{b}_j, \mathbf{b}_{i-1}^* \rangle}{\|\mathbf{b}_{i-1}^*\|^2} \mathbf{b}_{i-1}^*$$

and

$$\|\pi_i(\mathbf{b}_j)\|^2 = \|\pi_{i-1}(\mathbf{b}_j)\|^2 - \frac{\langle \mathbf{b}_j, \mathbf{b}_{i-1}^* \rangle^2}{\|\mathbf{b}_{i-1}^*\|^2}$$

Note that the complexity of PPR is at most $O(n^3)$ for a given n-dimensional lattice, which is negligible compared to the complexity of ENUM.

4.2 Dual Projective Reordering

We further introduce another reordering method to decrease the number of search nodes in ENUM by directly lengthening the back half of the GSO vectors' norm. Here we use the property of a dual lattice. Let $\{\mathbf{d}_n^\dagger, \cdots, \mathbf{d}_1^\dagger\}$ be the dual lattice basis$\{\mathbf{d}_1, \cdots, \mathbf{d}_n\}$ in reverse order. Since $\|\mathbf{b}_i^*\| \cdot \|\mathbf{d}_i^\dagger\| = 1$ holds, reducing the norm of each GSO vector $\mathbf{d}_i^\dagger$ incurs enlarging the norm of $\mathbf{b}_i^*$ correspondingly. Based on this idea, we propose the Dual Projective Reordering (DPR) method which applies the PPR method to the reversed dual basis $\{\mathbf{d}_n^\dagger, \cdots, \mathbf{d}_1^\dagger\}$. Then DPR changes the lattice basis order such that

$$\begin{aligned}
\|\mathbf{d}_n^\dagger\| &= \min\{\|\mathbf{d}_1\|, \cdots, \|\mathbf{d}_n\|\} \\
\|\mathbf{d}_{n-1}^\dagger\| &= \min\{\|\tau_{n-1}(\mathbf{d}_1)\|, \cdots, \|\tau_{n-1}(\mathbf{d}_{n-1})\|\} \\
&\vdots \\
\|\mathbf{d}_2^\dagger\| &= \min\{\|\tau_2(\mathbf{d}_1)\|, \|\tau_2(\mathbf{d}_2)\|\} \\
\|\mathbf{d}_1^\dagger\| &= \|\tau_1(\mathbf{d}_1)\|
\end{aligned}$$

which is equivalent to

$$\forall i \in \{1, \cdots, n\},\ \|\mathbf{d}_i^\dagger\| = \min_{1 \le j \le i} \|\tau_i(\mathbf{d}_j)\|.$$

We show the DPR in **Algorithm 3**. Note that the DPR method also cost at most $O(n^3)$ for a given n-dimensional lattice, which is negligible compared to the ENUM complexity.

Algorithm 4 Reordering with t_R

Input: A basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \in \mathbb{R}^m$, its GSO vectors $\{\mathbf{b}_1^*, \cdots, \mathbf{b}_n^*\}$ and a parameter $t_{succ} \geq 1$.
Output: A reordered basis $\{\mathbf{b}'_1, \cdot, \mathbf{b}'_n\}$ or a input basis $\{\mathbf{b}_1, \ldots, \mathbf{b}_n\}$.
$\{\mathbf{b}_1^*, \cdots, \mathbf{b}_n^*\} \leftarrow \text{computeGSO}(\{\mathbf{b}_1, \cdots, \mathbf{b}_n\})$;
$\{\mathbf{b}'_1, \cdots, \mathbf{b}'_n\} \leftarrow \text{Reordering}(\{\mathbf{b}_1, \cdots, \mathbf{b}_n\})$;
$\{\mathbf{b}'^*_1, \cdots, \mathbf{b}'^*_n\} \leftarrow \text{computeGSO}(\{\mathbf{b}'_1, \cdots, \mathbf{b}'_n\})$;
$t_R \leftarrow 1$;
for $i = \lfloor n/2 \rceil$ to n **do**
    $t_R \leftarrow t_R \cdot \frac{\|\mathbf{b}'^*_i\|}{\|\mathbf{b}^*_i\|}$;
end for
if $t_{succ} < t_R$ **then**
    output $\{\mathbf{b}'_1, \cdots, \mathbf{b}'_n\}$;
end if
output $\{\mathbf{b}_1, \cdots, \mathbf{b}_n\}$;

4.3 Observation

Because of the possibility that reordering basis vectors may incur an increase of search nodes in ENUM, it is necessary to observe and set an threshold to enhance the effect of the reordering methods. Let the GSO vectors of the lattice basis be $\{\mathbf{b}_1^*, \cdots, \mathbf{b}_n^*\}$, and let the GSO vector of the DPR-reordered basis be $\{\mathbf{b}'^*_1, \cdots, \mathbf{b}'^*_n\}$. Here we define a parameter t_R as

$$t_R := \prod_{i=\lfloor n/2 \rceil}^{n} \frac{\|\mathbf{b}'^*_i\|}{\|\mathbf{b}^*_i\|}.$$

A larger t_R indicates a larger GSO vectors of the latter reordered basis vectors, and a potentially better performance of the reordering method. We can apply this idea to PPR, DPR, and NR, too.

We show the algorithm of reordering with t_R in **Algorithm 4**. An input parameter $t_{succ} \geq 1$ is a threshold, where the larger t_R is than t_{succ}, the more search nodes will be reduced by the reordering method.

5 Experimental Results

In this section, we show the experimental results of the PPR method and the DPR method. The implementation was done in C++ language using the number theory library NTL [17]. For the random lattice, we generate the random lattice bases from the SVP Challenge [5]. The upper bound R input into ENUM is fixed at $1.05 \times GH(L)$.

5.1 Experimental result for 30-dimensional random lattice

We performed the following experiment. We have prepared 1000 cases of 30-dimensional random lattice bases LLL-reduced with $\delta_{LLL} = 0.8$ and applied NR,

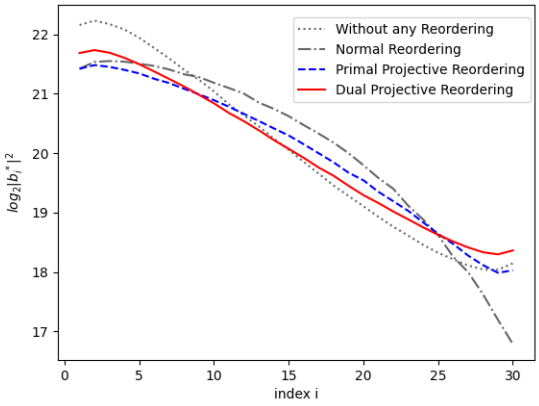

Fig. 1. The distribution of $\|\mathbf{b}_i^*\|$. (average value of 1000 cases of 30-dimensional random lattice LLL-reduced with $\delta_{LLL} = 0.8$)

PPR and DPR to them before inputting that basis into ENUM, respectively. Fig. 1 shows the distribution of the GSO vectors' norm respectively, which is the average value of 1000 cases of 30-dimensional random lattice LLL-reduced with $\delta_{LLL} = 0.8$.

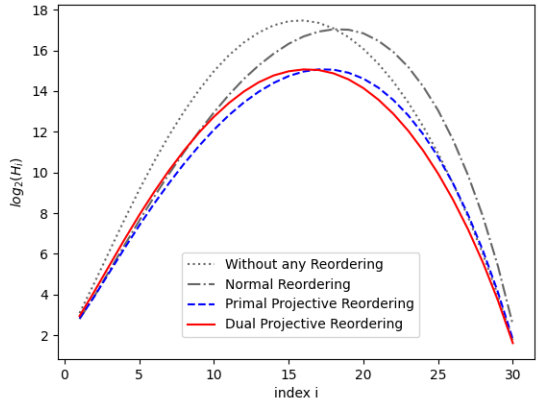

Fig. 2. Total number of search nodes at each level in ENUM (average value of 1000 cases of 30-dimensional random lattice LLL-reduced with $\delta = 0.8$)

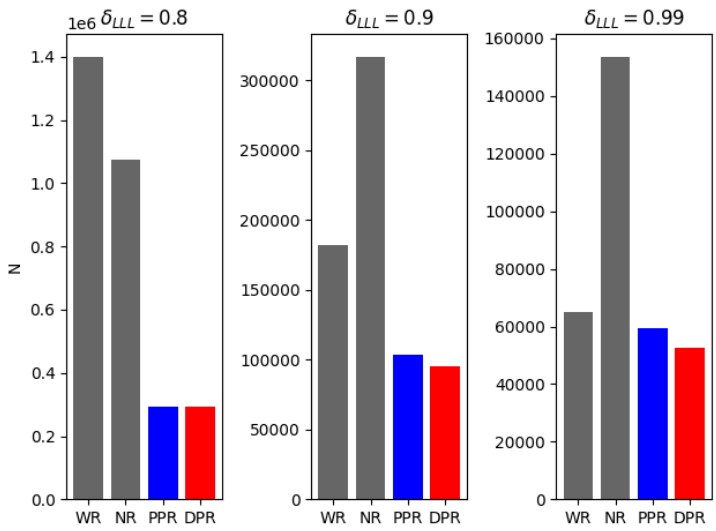

Fig. 3. The number of search nodes in ENUM (average value of 1000 cases of 30-dimensional random lattice LLL-reduced with $\delta = 0.8, 0.9, 0.99$). WR: Without any Reordering, NR: Normal Reordering, PPR: Primal Projective Reordering, DPR: Dual Projective Reordering.

From Fig. 1, we can see that PPR and DPR satisfy the condition to lengthen the back half of the GSO vectors' norm, while Normal Reordering (NR), unfortunately, shortens the back half of the GSO vectors' compared to PPR, DPR. Therefore we can see that PPR and DPR are an improvement on N. Next, Fig. 2 shows the number of search nodes at each level in the ENUM tree (the average value of 1000 cases of 30-dimensional random lattices). From Fig. 2, we can see that the number of search node in ENUM is the largest around $\frac{n}{2}$ as pointed out in [7]. Moreover, the numbers of search nodes in ENUM using PPR and DPR are respectively much smaller than that of the case with NR or without Reordering. The large-small relationship of the experimental results on the total search node number ($= \sum_{i=1}^{n} H_i$) for each method was DPR $<$ PPR $<$ NR $<$ Without any Reordering. (i.e., The number of ENUM search node with DPR was the lowest of four.) When we usually use LLL reduction, we set δ_{LLL} as close to 1 as possible, e.g., $\delta_{LLL} = 0.99$. Fig. 3 shows the search node number in ENUM when lattice basis LLL-reduced with $\delta = 0.8$, 0.9, 0.99 in the four cases: without reordering, using NR, PPR and DPR.

From Fig. 3, we can see that every three reordering methods reduce the number of search nodes compared to the case without reordering for $\delta_{LLL} = 0.8$. However, when $\delta_{LLL} = 0.9$ and 0.99, the total number of search nodes increases. The reason is that the norm in the middle of the GSO vectors NR-reordered becomes large, while the back half of the GSO vectors' norm becomes extremely small. Although previous studies claim that the larger the norm in the middle of

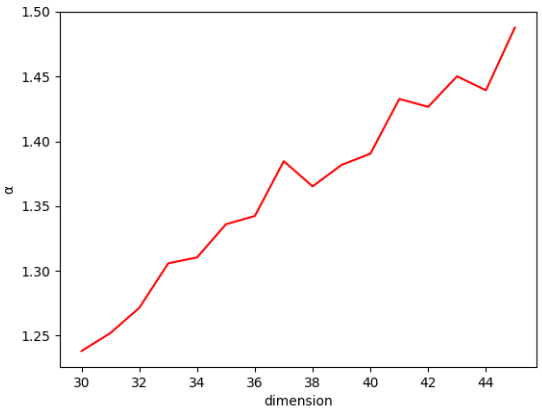

Fig. 4. $\alpha := \bar{N}_{WR}/\bar{N}_{DPR}$ at each dimensional lattices. (average value of 1000 cases of 30 to 45 dimensional random lattices LLL-reduced with $\delta_{LLL} = 0.99$)

the GSO vector is, the more the total number of search nodes can be reduced, we claim that it is necessary to lengthen the back half of the GSO vectors' norm in order to decrease the number of search nodes. We can see that both PPR and DPR can reduce the number of search nodes compared to the case without reordering with $\delta_{LLL} = 0.9$ or 0.99.

Moreover, the experimental result shows that DPR decreases the number of search nodes more than PPR. i.e., Directly lengthening the back half of the GSO vectors' norm decreases the number of search nodes more than shortening the front half. Therefore in the following subsection, we will focus on our discussion on the case with DPR.

5.2 Experimental results on a high-dimensional lattice

Next, to see the performance of DPR in more than 30-dimensional lattice, we performed the same experiment about DPR, not only 30 dimensions but also 30-45 dimensions. (The same Experiments in more than 46-dimensional lattice were impossible due to the exponential computation time of ENUM.). Fig. 4 shows that experimental result. $\bar{N}_{DPR}$ is the number of search nodes when the basis without reordering is input to ENUM, while $\bar{N}_{DPR}$ is the number of search nodes when DPR.

Fig. 4 shows that the more the dimension increases, the more DPR decreases the average number of search nodes, although it has been pointed out that NR decreases as the dimension increases. Therefore we overcome the QRT weakness that the reduction of the number of search nodes decreases as the dimension increase, and obtained the result that the reduction can increase as the dimension increase. Furthermore, from experimental results, DPR decreased the number of search nodes by 32.8% on average on 45-dimensional lattices.

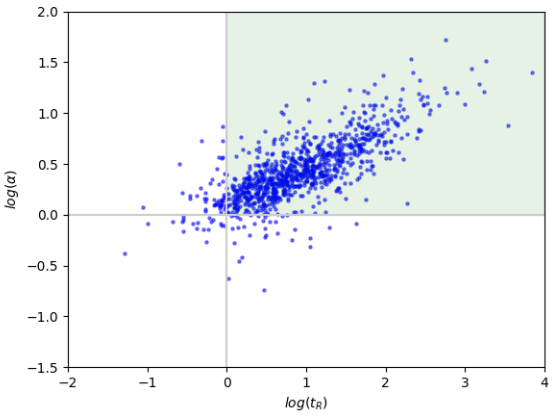

Fig. 5. The scatter plot of t_R and $\alpha := N_{WR}/N_{DPR}$. (1000 cases of 40-dimensional random lattices LLL-reduced with $\delta_{LLL} = 0.99$)

5.3 Experimental results on t_R

To observe the lattice basis condition on which DPR can decrease the number of search nodes in ENUM, we have confirmed the usefulness of DPR's t_R through experiments by 40-dimensional random lattices. Fig. 5 shows the experimental results of the relation between t_R and $\alpha := N_{WR}/N_{DPR}$. N_{WR} is the number of search nodes in ENUM without reordering, while N_{DPR} is the number of search nodes when using DPR.

From Fig. 5, we can see that the larger α, the larger t_R. In other words, the larger t_R, the larger the number of search nodes. Moreover, we can see that the events often happen that t_R exceeds one and α exceeds one. (i.e., when the back half of the GSO vectors' norm becomes large, the number of search nodes successfully decreases). The experimental result shows that when the t_R is greater than one, the DPR-reordered basis can efficiently decrease the number of search nodes in ENUM and that events often happen. (See the green area in Fig.5) Therefore we can see that if t_{succ} in **Algorithm 4** is larger than one, the number of search nodes decreases with high probabilities.

6 Conclusion and Future Work

In this paper, we observed whether the number of search nodes in ENUM decreases by changing the order of the inputted into ENUM, using the properties of projective and dual lattices. Our contributions are as follows.

1. A proposal of a basis reordering method, **PPR**, that extends the methods of previous studies;

2. A proposal of a basis reordering method **DPR** using the properties of a dual lattice;
3. A proposal of t_R to decide how much the reordered basis reduces the computational complexity of ENUM compared to the original basis.

The experimental results show that PPR can decrease the number of search nodes in ENUM by decreasing the projection length orders, which improves the QRT in the previous study. Moreover, they show that DPR has better performance than PPR and decreases the number of search nodes by 32.8% on average on 45-dimensional lattices. Furthermore, we experimentally show that DPR can increase the quantities to decrease the number of search nodes as dimensions increase: i.e., DPR overcomes the QRT weakness that the quantities to decrease the number of search nodes decreases as the dimension increase.

The reordered basis cannot always decrease the number of search nodes in ENUM compared to the original basis. Therefore, based on the idea that the larger the norm of the latter GSO vector of DPR-reordered basis, the more the reordered basis can decrease the number of search nodes in ENUM, we proposed t_R to decide how much the reordered basis reduces the computational complexity of ENUM compared to the original basis. We experimentally showed that the larger the t_R, the more the number of search node in ENUM decreases.

Future work includes the application of DPR to the BKZ and Extreme Pruning [7]. If we can efficiently decide whether the number of search nodes can decrease by using t_R, we can reduce the complexity of these algorithms by using DPR.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number JP20K23322, JP21K11751 and JP19K11960, Japan.

References

1. PQC Standardization Process: Third Round Candidate Announcement, 2020. `https://csrc.nist.gov/News/2020/pqc-third-round-candidate-announcement`.
2. M. Ajtai, R. Kumar, and D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, pages 601–610, 2001.
3. Y. Aono, Y. Wang, T. Hayashi, and T. Takagi. Improved progressive BKZ algorithms and their precise cost estimation by sharp simulator. In *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings, Part I*, pages 789–819, 2016.
4. Y. Chen and P. Q. Nguyen. BKZ 2.0: Better lattice security estimates. In D. H. Lee and X. Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings*, volume 7073 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2011.
5. T. Darmstadt. Svp challenge. Available at `https://www.latticechallenge.org/svp-challenge`, 2019.

6. N. Gama and P. Q. Nguyen. Finding short lattice vectors within mordell's inequality. In C. Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 207–216. ACM, 2008.
7. N. Gama, P. Q. Nguyen, and O. Regev. Lattice enumeration using extreme pruning. In *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Proceedings*, pages 257–278, 2010.
8. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: A ring-based public key cryptosystem. In J. Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.
9. N. Koblitz. Constructing elliptic curve cryptosystems in characteristic 2. In A. Menezes and S. A. Vanstone, editors, *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990, Proceedings*, volume 537 of *Lecture Notes in Computer Science*, pages 156–167. Springer, 1990.
10. D. Micciancio and M. Walter. Practical, predictable lattice basis reduction. In M. Fischlin and J. Coron, editors, *Advances in Cryptology - EUROCRYPT 2016 - 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, May 8-12, 2016, Proceedings, Part I*, volume 9665 of *Lecture Notes in Computer Science*, pages 820–849. Springer, 2016.
11. V. S. Miller. Use of elliptic curves in cryptography. In H. C. Williams, editor, *Advances in Cryptology - CRYPTO '85, Santa Barbara, California, USA, August 18-22, 1985, Proceedings*, volume 218 of *Lecture Notes in Computer Science*, pages 417–426. Springer, 1985.
12. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6):34:1–34:40, 2009.
13. R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.
14. C. Schnorr. Lattice reduction by random sampling and birthday methods. In H. Alt and M. Habib, editors, *STACS 2003, 20th Annual Symposium on Theoretical Aspects of Computer Science, Berlin, Germany, February 27 - March 1, 2003, Proceedings*, volume 2607 of *Lecture Notes in Computer Science*, pages 145–156. Springer, 2003.
15. C. Schnorr and M. Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Math. Program.*, 66:181–199, 1994.
16. P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 124–134. IEEE Computer Society, 1994.
17. V. Shoup. NTL, a library for doing number theory. Available at `http://www.shoup.net/ntl/`, 2017.
18. Y. Wang and T. Takagi. Improving the BKZ reduction algorithm by quick reordering technique. In *Information Security and Privacy - 23rd Australasian Conference, ACISP 2018, Proceedings*, pages 787–795, 2018.
19. Y. Wang and T. Takagi. Studying lattice reduction algorithms improved by quick reordering technique. *Int. J. Inf. Sec.*, 20(2):257–268, 2021.