An Improvement of Babai's Rounding Procedure for CVP

Shuying Yang
Department of Data and Computer Science, Shandong Women's University
No. 2399, University Road, Jinan 250022, Shandong, P. R. China
Email: ysystudy2005@163.com

(Received Aug. 15, 2022; Revised and Accepted Dec. 1, 2022; First Online Feb. 18, 2023)

Abstract

CVP, together with SVP, are two of the central problems in lattice-based cryptography. Their hardness paved the way for the proposals of many different lattice-based cryptographic schemes. Meanwhile, efficient algorithms for solving or approximately solving these problems have become essential tools in public key cryptanalysis and have successfully attacked many cryptosystems. Among these algorithms, Babai's rounding procedure is one of the most classic. Recently, the rounding procedure has been implemented in RNS and improved using optimal Hermite Normal Form lattices. In this paper, we propose four improved rounding procedures to approximately solve the CVP problem based on the famous Babai's rounding procedure and Gram-Schmidt orthogonalization technique. The first two procedures are general for any lattice basis, while the latter two algorithms are unique versions of the first two for which the input basis is in HNF form. We also show that all four algorithms perform better than Babai's procedure concerning errors by examples and experiments, although some efficiency loss is taken as the cost.

Keywords: Closest Vector Problem (CVP); Hermite Normal Form (HNF); Rounding Procedure

1 Introduction

Recently in the literature of cryptology, lattice-based cryptographic constructions hold a great promise for post-quantum cryptography [3,4,11–13,15,21], since they enjoy relatively strong security proofs, remarkable efficient implementations, as well as very simplicity. Furthermore, lattice-based cryptography is believed to be quantum resistant. At the bottom of the constructions of lattice-based cryptography, however, it is the hardness of the computational problem in lattices that lays a secure foundation [9, 20].

The most two basics of computational problems in lattices are the *Shortest Vector Problem*(SVP) and the *Closest Vector Problem*(CVP) [19]. The *Shortest Vector Problem* asks to find the shortest nonzero vector in a lattice L, however, the *Closest Vector Problem* is the inhomogeneous version of SVP, and asks to find the lattice point closest to a given target. The CVP has been proved to be NP-complete by reduction from subset sum [10], and therefore no algorithm can solve CVP in deterministic polynomial time, unless $P = NP$ [10]. Reducing CVP to SVP is an interesting problem [7], as it is widely believed that SVP is not harder than CVP, and many even believe that SVP is strictly easier. Empirical evidence to these beliefs is provided by the gap between known hardness results for both problems. Whearas it is relatively easy to establish the NP-hardness of CVP, the question of whether SVP is NP-hard was open for almost two decades, originally conjectured in [10] and resolved in the affirmative in [5], and only for randomized reductions.

The hardness of solving SVP and CVP has led reseachers to consider approximation versions of these problems [19]. Approximation algorithms return solutions that are only guaranteed to be within some specified factor γ from the optimal. Approximation CVP in n-dimensional lattices is known to be NP-hard for any constant approximation factor or even some slowly increasing function of the dimension [14].

However, NP-hardness itself does not exclude the possiblity of sub-exponential time algorithms. To rule out such algorithms, several hypothesis have been introduced, such as the Strong Exponential Time Hypothesis (SETH), the Exponential Time Hypothesis(ETH), or the Gap-Exponential Time Hypothesis (Gap-ETH), and are by now quite standard. Based on these, a few recent results have shown hardness for approximation CVP and SVP are closely related [1, 2].

On the algorithmic side, known polynomial-time algorithms like the one of LLL algorithm [16] and its descendants such as [6] obtain slightly subexponential approximation factors $\gamma = 2^{\Theta(n \log\log n/\log n)}$ for SVP. Based on these algorithms, Babai's nearest plane algorithm can obtain a similar approximation factor for CVP. In [7], Babai also propose a rounding procedure for CVP which is more efficient, although it loses a little approximation factor. Recently, the rounding procedure has been implemented in RNS [8] and improved with the use of lattices of optimal Hermite Normal Form [17, 18]. In this paper, we impove the Babai's rounding procedure.

Our Contributions. We provide four improved rounding procedures and show that the proposed rounding procedures are stronger than Babai's rounding procedure with respect to the appoximation parameters.

1) The first two improved rounding procedures are general for arbitrary lattice basis. The underlying main ideas are inspired by the following three observations: The first one is that every lattice basis can be easily decomposed as multiplication of a orthogonal basis (a basis of $\mathbb{R}^n$ for some n, not necessarily a lattice basis) and an upper triangular matrix by Gram-Schmidt orthogonalization procedure. Second, the orthogonal basis makes it easy to quantify the distance between vector in lattice and the input target vector. The third observation is that optimizing this distance can be done by appropriately choosing round up or round down of corresponding parameters.

2) The second two improved rounding procedures are special for lattice basis in its Hermite Nornal Form (HNF). The underlying main ideas are based on the observation: The special form of the HNF basis makes the Gram-Schmidt orthogonal decomposition of this basis very special. The obtained orthogonal basis is a digonal matrix, and the diagonal elements are the corresponding diagonal elements of the original HNF basis. Furthermore, the resulting upper triangular matrix has diagonal entries of 1, and all nonzero entries are positive. These characteristics make our algorithm easy to do some checking operations, thus improving the efficiency. Since every lattice has a unique HNF basis which can be efficiently computed from any basis of the lattice, the second two rounding algorithms can actually be made general easily, just by converting the basis to an HNF basis.

2 Preliminaries

Notation. We use $\mathbb{Z}$ for the set of integers, and $\mathbb{R}$ for the set of real numbers. Elements of these sets are denoted by lowercase letters. We use uppercase letters to denote matrices M, and usually arrange a set of vectors in columns into a matrix. We denote the i-th coordinate of vector v by v_i.

2.1 Lattices

Definition 1. *A lattice is defined as the set of all integer linear combinations*

$$L(b_1, \cdots, b_n) = \left\{ \sum_{i=1}^{n} x_i b_i : x_i \in \mathbb{Z} \text{ for } 1 \le i \le n \right\}$$

of n linearly independent vectors $b_1, \cdots, b_n$ in $\mathbb{R}^d$, where d and n are called dimension and rank of the lattice. If $n = d$, the lattice is called full rank. The set of vectors $b_1, \cdots, b_n$ is called a basis for the lattice. A basis is usually represented by the matrix $B = [b_1, \cdots, b_n] \in \mathbb{R}^{d\times n}$ having the basis vectors as columns.

When studying lattices from an algorithm point of view, it is customary to assume that the basis vectors (and therefore any lattice vector) have all rational coordinates. Moreover, by appropriately scaling the lattice, all rational lattices can be easily converted to integer lattices. So, without loss of generality, we concentrate on integer lattices.

2.2 Minimum Distance

Definition 2. *For any lattice L, the minimum distance of L is the smallest distance between any two lattice points:*

$$\lambda(L) = \inf\{\|x - y\| : x, y \in L, x \neq y\}$$

Equivalently, the minimum distance can also be defined as the length of the shortest nonzero lattice vector:

$$\lambda(L) = \inf\{\|v\| : v \in L \backslash 0\}$$

Minkowski's first theorem states the following with regards to the minimun distance:

Theorem 1. *For any rank n lattice L, the length of the shortest nonzero vector satisfies $\lambda(L) < \sqrt{n}\det(L)^{1\backslash n}$.*

2.3 Computational Problems

Minkowski's first theorem implies that any lattice of rank n contains a nonzero vector of length at most $\sqrt{n}\det(L)^{1\backslash n}$. Its proof, however, is non-constructive: it does not give us an algorithm to find such a lattice vector. To discuss such computational issues, let us define the most two basic computational problems involving lattices.

Definition 3 (Approximate SVP)**.** *Given a lattice basis $B \in \mathbb{Z}^{d\times n}$, find a nonzero lattice vector $Bx(x \in \mathbb{Z}^n \backslash 0)$ such that $\|Bx\| \le \gamma\lambda(L(B))$. In particular, the problem is called SVP if $\gamma = 1$.*

Definition 4 (Approximate CVP)**.** *Given a lattice basis $B \in \mathbb{Z}^{d\times n}$ and a target vector $t \in \mathbb{Z}^m$, find a lattice vector $Bx(x \in \mathbb{Z}^n)$ such that $\|Bx - t\| \le \gamma\|By - t\|$ for any other $y \in \mathbb{Z}^n$. In particular, the problem is called CVP if $\gamma = 1$.*

3 Babai's Rounding Procedure

In this section, we provide a brief overview of Babai's rounding procedure (Algorithm 1). Given an arbitrary lattice basis B and a target t, in order to find a lattice point close to a target t we may

- first apply the inverse transformation B^{-1} to get $B^{-1}t$,
- round $B^{-1}t$ to the closest integer vector $\lfloor B^{-1}t \rceil \in \mathbb{Z}^n$,
- map the resulting integer vector to the lattice point $v = B\lfloor B^{-1}t \rceil$.

Algorithm 1 Babai's Rounding Procedure

```
Input:
    Lattice basis $B$, and vector $t \in \mathbb{R}^n$
Output:
1: Compute $B^{-1}$ and get $B^{-1}t$;
2: Round $B^{-1}t$ to the closest integer vector $\lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
3: return the resulting integer vector to the lattice point $v = B\lfloor B^{-1}t \rceil$.
```

In order to analyse Babai's algorithm easily, we introduce the two quantities

$$s_{\min} = \min_{x\in\mathbb{R}^n} \|Bx\| \,/\, \|x\|$$

$$s_{\max} = \max_{x\in\mathbb{R}^n} \|Bx\| \,/\, \|x\|$$

which express by how much the transformation B can shrink or expand the length of a vector.

Theorem 2. *Babai's rounding procedure always outputs a lattice point within distance $\sqrt{n} \cdot S_{\max}/2$ from t.*

Proof.

$$\begin{aligned}
&\left\|B\lfloor B^{-1}t\rceil - t\right\| \\
&= \left\|B(\lfloor B^{-1}t\rceil - B^{-1}t)\right\| \\
&= \left\|B\frac{\lfloor B^{-1}t\rceil - B^{-1}t}{\|B(\lfloor B^{-1}t\rceil - B^{-1}t)\|} \cdot \left\|B(\lfloor B^{-1}t\rceil - B^{-1}t)\right\|\right\| \\
&= \left\|(\lfloor B^{-1}t\rceil - B^{-1}t)\right\| \cdot \left\|B\frac{\lfloor B^{-1}t\rceil - B^{-1}t}{\|B(\lfloor B^{-1}t\rceil - B^{-1}t)\|}\right\| \\
&\le \left\|(\lfloor B^{-1}t\rceil - B^{-1}t)\right\| \cdot S_{\max}(B) \\
&\le \frac{1}{2}\left\|\begin{bmatrix}1\\ \vdots \\ 1\end{bmatrix}\right\| \cdot S_{\max}(B) \\
&= \frac{\sqrt{n}}{2} \cdot S_{\max}(B)
\end{aligned}$$

where the first inequality is due to the definition of $S_{\max}(B)$, and the second inequality is since the rule of notation $\lfloor\cdot\rceil$. □

Theorem 3. *Let t is within distance $S_{\min}/2$ from the lattice, then Babai's rounding procedure returns the (necessarily unique) lattice point within distance $S_{\min}/2$ from t.*

Proof. Since t is within distance $S_{\min}/2$ from the lattice, there exists a integer vector $Y \in \mathbb{Z}$ such that

$$\|BY - t\| \le \frac{1}{2}S_{\min}(B)$$

Hence,

$$\begin{aligned}
&\frac{1}{2}S_{\min}(B) \\
&\ge \|BY - t\| \\
&= \left\|B(Y - B^{-1}t)\right\| \\
&= \left\|B\frac{Y - B^{-1}t}{\|Y - B^{-1}t\|} \cdot \left\|Y - B^{-1}t\right\|\right\| \\
&= \left\|Y - B^{-1}t\right\| \cdot \left\|B\frac{Y - B^{-1}t}{\|Y - B^{-1}t\|}\right\| \\
&\ge \left\|Y - B^{-1}t\right\| \cdot S_{\min}(B),
\end{aligned}$$

Therefore, $\left\|Y - B^{-1}t\right\| \le \frac{1}{2}$. We claim that the absolution of each coordinate of vector $Y - B^{-1}t$ is less than $\frac{1}{2}$. Otherwise, the norm of vector $Y - B^{-1}t$ will be strictly greater than $\frac{1}{2}$ which contracts to $\left\|Y - B^{-1}t\right\| \le \frac{1}{2}$. Since $Y \in \mathbb{Z}$ is an integer vector, $Y = \lfloor B^{-1}t\rceil$. This also shows that t which is within distance $S_{\min}/2$ from the lattice is necessarily unique. □

4 Improved Rounding Procedure

In this section,we provide Four improved rounding procedures. Two are general for arbitrary lattice basis, the other two special for lattice basis in its Hermite normal form. And we also show that the proposed rouding procedures are stronger than Babai's rounding procedure with respect to the appoximation parametres.

4.1 General Version

Given an arbitrary lattice basis B and a target t, in order to find a lattice point close to a target t we conduct the following steps (Algorithm 2).

1) First compute the Gram-Schmidt orthogonalization B^* of basis B and the corresponding upper-triangular matrix Λ such that $B = B^* \cdot \Lambda$;

2) Apply the inverse transformation B^{-1} to get $B^{-1}t$;

3) Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t\rceil \in \mathbb{Z}^n$;

4) Compute the vector $w = u - B^{-1}t$;

5) If $\lambda_{ij} \cdot w_j \ge 0$ for all $j > i$, goto step 6); Otherwise, goto step 7);

6) For j from 1 to $n-1$, if $|w_j| = 0.5$, set $w_j = -w_j$. Then, goto step 7) ;

7) Map the resulting integer vector to the lattice point $v = B^{-1}t + w$.

Theorem 4. *Given an arbitrary lattice B and a target vector t. Let v and $v^\dagger$ be the outputs of Algorithm 1 and Algorithm 2, respectively. Then the distance of t from*

Algorithm 2 General Edition I

```
Input:
    Lattice basis $B$, and vector $t \in \mathbb{R}^n$
Output:
1:  Compute the GS-basis $B^*$ of $B$ and $\Lambda$ such that $B = B^*\Lambda$;
2:  Compute $B^{-1}$ and get $B^{-1}t$;
3:  Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
4:  Compute the vector $w = u - B^{-1}t$;
5:  $k = 0$;
6:  for $i = 1$ to $n$ do
7:    for $j = i$ to $n$ do
8:      if $\lambda_{ij} \cdot w_j \geq 0$ then
9:        k++;
10:     end if
11:   end for
12: end for
13: if $k = \frac{n(n+1)}{2}$ then
14:   for $j = 1$ to $n-1$ do
15:     if $|w_j| = 0.5$ then
16:       $w_j = -w_j$;
17:     end if
18:   end for
19: end if
20: return the resulting integer vector to the lattice point $v = B(B^{-1}t + w)$
```

$v^\dagger$ is not greater than that of t from v. In particular, if there is at least one coordinate of w appeared in Algorithm 2 whose absolute value is exactly the value 0.5, then the distance of t from $v^\dagger$ will be strictly smaller than that of t from v.

Proof. Due to the step 1 in Algorithm 2, B^* is the Gram-Schmidt orthogonalization of basis B, and Λ is the corresponding upper-triangular matrix with 1 on it's principal diagonal such that $B = B^* \cdot \Lambda$. Denote

$$B^* = [b_1^*, b_2^*, \cdots, b_n^*],$$

and

$$\Lambda = \begin{bmatrix} 1 & \lambda_{12} & \cdots & \lambda_{1n} \\ 0 & 1 & \cdots & \lambda_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Since v is the output of Algorithm 1, the square of the distance between the input vector t and v can be presented as

$$\begin{aligned} \|t - v\|^2 &= \left\|B(B^{-1}t - \lfloor B^{-1}t \rceil)\right\|^2 \\ &= \left\|B^*\Lambda(B^{-1}t - \lfloor B^{-1}t \rceil)\right\|^2 \\ &= \|B^*\Lambda y\|^2 \end{aligned}$$

where $B^{-1}t - \lfloor B^{-1}t \rceil$ is denoted by y, which is also equal to the vector w in the step 4 of Algorithm 2. After finishing excuting all steps of Algorithm 2, the vector w would be exchanged. We denote w at this point as $y^\dagger$. Since $v^\dagger$ is the output of Algorithm 2, the square of the distance between the input vector t and $v^\dagger$ can be presented as

$$\begin{aligned} \left\|t - v^\dagger\right\|^2 &= \left\|t - B(B^{-1}t - w)\right\|^2 \\ &= \left\|t - B(B^{-1}t - y^\dagger)\right\|^2 \\ &= \left\|By^\dagger\right\|^2 \\ &= \left\|B^*\Lambda y^\dagger\right\|^2 \end{aligned}$$

Now we denote Λy and $\Lambda y^\dagger$ by x and $x^\dagger$ respectively, and analyse their relation due to the process between Step 5 and Step 19 in Algorithm 2. The functionality from step 5 to step 12 is to check whether $\lambda_{ij} \cdot w_j \geq 0$ for all $j > i$. If so, set $w_j = -w_j$ when $|w_j| = 0.5$ for all j; Otherwise, follow Babai's rounding procedure without any change. Thus

$$\begin{aligned} x_i^\dagger &= \sum_{j=1}^{j=n} \lambda_{ij} \cdot y_j^\dagger = \sum_{j=i}^{j=n} \lambda_{ij} \cdot y_j^\dagger \leq |\sum_{j=i}^{j=n} \lambda_{ij} \cdot y_j^\dagger| \\ &\leq \sum_{j=i}^{j=n} |\lambda_{ij} \cdot y_j^\dagger| = \sum_{j=i}^{j=n} |\lambda_{ij} \cdot y_j| = \sum_{j=1}^{j=n} |\lambda_{ij} \cdot y_j| = x_i. \end{aligned}$$

Therefore,

$$\begin{aligned} \left\|t - v^\dagger\right\|^2 &= \left\|B^*\Lambda y^\dagger\right\|^2 = \left\|B^* x^\dagger\right\|^2 \\ &= |x_1^\dagger|\,\|b_1^*\|^2 + |x_2^\dagger|\,\|b_2^*\|^2 + \cdots + |x_n^\dagger|\,\|b_n^*\|^2 \\ &\leq |x_1|\,\|b_1^*\|^2 + |x_2|\,\|b_2^*\|^2 + \cdots + |x_n|\,\|b_n^*\|^2 \\ &= \|B^*x\|^2 = \|B^*\Lambda y\|^2 = \|t - v\|^2 \end{aligned}$$

where the third and fourth equalities follow from the fact that B^* is the Gram-Schmidt orthogonalization of basis B.

In particular, if there is at least one coordinate of w appeared in Algorithm 2 whose absolute value is exactly the value 0.5, Step 16 in Algorithm 2 will be performed. Thus, there is at least one coordinate $x_i^\dagger$ which would be strictly smaller than it's corresponding element x_i, i.e. $x_i^\dagger < x_i$. Then

$$\begin{aligned} \left\|t - v^\dagger\right\|^2 &= \left\|B^*\Lambda y^\dagger\right\|^2 = \left\|B^* x^\dagger\right\|^2 \\ &= |x_1^\dagger|\,\|b_1^*\|^2 + |x_2^\dagger|\,\|b_2^*\|^2 + \cdots + |x_n^\dagger|\,\|b_n^*\|^2 \\ &< |x_1|\,\|b_1^*\|^2 + |x_2|\,\|b_2^*\|^2 + \cdots + |x_n|\,\|b_n^*\|^2 \\ &= \|B^*x\|^2 = \|B^*\Lambda y\|^2 = \|t - v\|^2 . \end{aligned}$$

□

Theorem 5. *Given an arbitrary lattice B and a target vector t. Let v and $v^\ddagger$ be the outputs of Algorithm 1 and Algorithm 3, respectively. Then the distance of t from $v^\ddagger$ is not greater than that of t from v. In particular, if there is at least one coordinate of w appeared in Algorithm 2 whose absolute value is exactly the value 0.5, then the distance of t from $v^\ddagger$ will be strictly smaller than that of t from v.*

Algorithm 3 General Edition II

```
Input:
    Lattice basis $B$, and vector $t \in \mathbb{R}^n$
Output:
 1: Compute the GS-basis $B^*$ of $B$ and $\Lambda$ such that $B = B^*\Lambda$;
 2: Compute $B^{-1}$ and get $B^{-1}t$;
 3: Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
 4: Compute the vector $w = u - B^{-1}t$;
 5: $k = 0$;$l = 0$;$r = 0$
 6: for $i = 1$ to $n$ do
 7:   for $j = i$ to $n$ do
 8:     if $\lambda_{ij} \cdot w_j \geq 0$ then
 9:       k++;
10:     end if
11:   end for
12: end for
13: if $k = \frac{n(n+1)}{2}$ then
14:   for $j = n - 1$ to 1 do
15:     if $|w_j| = 0.5$ then
16:       $l+ = w_j$;
17:       $r+ = -w_j$;
18:       for $i = j + 1$ to $n$ do
19:         $l+ = \lambda_{ji} \cdot w_i$;
20:         $r+ = \lambda_{ji} \cdot w_i$;
21:       end for
22:       if $| l |\geq| r |$ then
23:         $w_j = -w_j$;
24:       end if
25:     end if
26:   end for
27: end if
28: return the resulting integer vector to the lattice point $v = B(B^{-1}t + w)$
```

Proof. For simplicity, we continue to use the notations in the proof of Theorem 3. Same as Algorithm 2, the vector w in Step 4 of Algorithm 3 is also equal to y. After finishing excuting all steps of Algorithm 3, the vector w would be exchanged. We denote w at this point as $y^{\ddagger}$. Since $v^{\ddagger}$ is the output of Algorithm 3, the square of the distance between the input vector t and $v^{\ddagger}$ can be presented as

$$\begin{aligned}\left\|t - v^{\ddagger}\right\|^2 &= \left\|t - B(B^{-1}t - w)\right\|^2\\ &= \left\|t - B(B^{-1}t - y^{\ddagger})\right\|^2\\ &= \left\|By^{\ddagger}\right\|^2\\ &= \left\|B^*\Lambda y^{\ddagger}\right\|^2\end{aligned}$$

Now we denote $\Lambda y^{\ddagger}$ by $x^{\ddagger}$ and ananlyse it's relation to x. The functionality from Step 5 to Step 12 in Algorithm 3 is to check whether $\lambda_{ij} \cdot w_j \geq 0$ for all $j > i$. If so, set $w_j = -w_j$ only when $|l| \geq |r|$ in Algorithm 3 rather than seting $w_j = -w_j$ when $|w_j| = 0.5$ for all j in Algorithm 2; Otherwise, same to Algorithm 2, follow Babai's rounding procedure without any change. Hence,

$$\begin{aligned}x_i^{\ddagger} &= \sum_{j=1}^{j=n} \lambda_{ij} \cdot y_j^{\ddagger} = \sum_{j=i}^{j=n} \lambda_{ij} \cdot y_j^{\ddagger} \leq |\sum_{j=i}^{j=n} \lambda_{ij} \cdot y_j^{\ddagger}|\\ &\leq \sum_{j=i}^{j=n} |\lambda_{ij} \cdot y_j^{\ddagger}| = \sum_{j=i}^{j=n} |\lambda_{ij} \cdot y_j| = \sum_{j=1}^{j=n} |\lambda_{ij} \cdot y_j| = x_i.\end{aligned}$$

Therefore,

$$\begin{aligned}\left\|t - v^{\ddagger}\right\|^2 &= \left\|B^*\Lambda y^{\ddagger}\right\|^2 = \left\|B^*x^{\ddagger}\right\|^2\\ &= |x_1^{\ddagger}|\left\|b_1^*\right\|^2 + |x_2^{\ddagger}|\left\|b_2^*\right\|^2 + \cdots + |x_n^{\ddagger}|\left\|b_n^*\right\|^2\\ &\leq |x_1|\left\|b_1^*\right\|^2 + |x_2|\left\|b_2^*\right\|^2 + \cdots + |x_n|\left\|b_n^*\right\|^2\\ &= \left\|B^*x\right\|^2 = \left\|B^*\Lambda y\right\|^2 = \left\|t - v\right\|^2\end{aligned}$$

where the third and fourth equalities follow from the fact that B^* is the Gram-Schmidt orthogonalization of basis B. In particular, if there is at least one coordinate of w appeared in Algorithm 2 whose absolute value is exactly the value 0.5, the step 16 in algrithm 2 will be performed. Thus, there is at least one coordinate $x_i^{\ddagger}$ which would be strictly smaller than it's corresponding element x_i, i.e. $x_i^{\ddagger} < x_i$. Then

$$\begin{aligned}\left\|t - v^{\ddagger}\right\|^2 &= \left\|B^*\Lambda y^{\ddagger}\right\|^2 = \left\|B^*x^{\ddagger}\right\|^2\\ &= |x_1^{\ddagger}|\left\|b_1^*\right\|^2 + |x_2^{\ddagger}|\left\|b_2^*\right\|^2 + \cdots + |x_n^{\ddagger}|\left\|b_n^*\right\|^2\\ &< |x_1|\left\|b_1^*\right\|^2 + |x_2|\left\|b_2^*\right\|^2 + \cdots + |x_n|\left\|b_n^*\right\|^2\\ &= \left\|B^*x\right\|^2 = \left\|B^*\Lambda y\right\|^2 = \left\|t - v\right\|^2.\end{aligned}$$

□

4.2 Special Version for HNF

An important fact in lattice theory is that every integer lattice can be represented in its unique Hermite Normal Form basis, and the HNF basis can be efficiently computed from any lattice basis. In this section, we will modify Algorithm 2 and Algorithm 3 according to the characteristics of HNF basis and present their special editions for HNF basis.

The Hermite Normal Form basis of an integer lattice is defined as either row-style or column-style. In this paper, we present the HNF basis as the column-style one which is defined as follows.

Definition 5. *Let L be a m dimentional integer lattice with rank n. A basis $H \in \mathbb{Z}^{m\times n}$ is in Hermite Normal Form if*

- *H is upper triangular, that is, there exists a sequence of integers $j_1 < \cdots < j_l$ such that for all $1 \leq i \leq l$ we have $h_{i,j} = 0$ for all $j < j_i$.*
- *For $1 \leq k < i \leq l$, we have $0 \leq h_{j_i,k} < h_{j_i,i}$, that is, the pivot element is the greatest along its row and the coefficients right are non-negative.*
- *For $i > l$, we have $h_{i,j} = 0$ for all $j = 1, \cdots, m$.*

Lemma 1. *For every integer lattice L, there exists a unique basis H that is in Hermite Normal Form.*

Lemma 2. *For any m dimentional integer lattice with rank n, there exists an polymomial time algorithm to compute it's HNF basis with $O(n^2 logM)$ space complexity and $O(mn^4 log^2 M)$ running time, where M is a bound on its entries.*

Given HNF basis B of an arbitrary integer lattice and a target t, in order to find a lattice point close to a target t we conduct the following steps (Algorithm 4).

1) First compute the Gram-Schmidt orthogonalization B^* of basis B and the corresponding upper-triangular matrix Λ such that $B = B^* \cdot \Lambda$;
2) Apply the inverse transformation B^{-1} to get $B^{-1}t$;
3) Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
4) Compute the vector $w = u - B^{-1}t$;
5) If $w_j \geq 0$ for all $j > i$, goto step 6); Otherwise, goto step 7);
6) For j from 1 to $n-1$, if $|w_j| = 0.5$, set $w_j = -w_j$. Then, goto step 7) ;
7) Map the resulting integer vector to the lattice point $v = B^{-1}t - w$.

Algorithm 4 Special Edition for HNF I

```
Input:
    Lattice HNF basis $B$, and vector $t \in \mathbb{R}^n$
Output:
 1: Compute the GS-basis $B^*$ of $B$ and $\Lambda$ such that $B = B^*\Lambda$;
 2: Compute $B^{-1}$ and get $B^{-1}t$;
 3: Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
 4: Compute the vector $w = u - B^{-1}t$;
 5: $k = 0$;
 6: for $i = 1$ to $n$ do
 7:     if $w_i \geq 0$ then
 8:         k++;
 9:     end if
10: end for
11: if $k = n$ then
12:     for $j = 1$ to $n-1$ do
13:         if $w_j = 0.5$ then
14:             $w_j = -w_j$;
15:         end if
16:     end for
17: end if
18: return the resulting integer vector to the lattice point $v = B(B^{-1}t + w)$
```

Corollary 1. *Given an arbitrary lattice presented in its Hermite normal form(HNF) basis B and a target vector t. Let v and v' be the outputs of Algorithm 1 and Algorithm 4, respectively. Then the distance of t from v' is not greater than that of t from v. In particular, if there is at least one coordinate of w appeared in Algorithm 2 which is exactly the value 0.5, then the distance of t from v' will be strictly smaller than that of t from v.*

Proof. The main differences between Algorithm 2 and Algorithm 4 are several steps which are used to finish some checks. The function of Steps 6 through 12 in Algorithm 2 is to check whether $\lambda_{ij} \cdot w_j \geq 0$ for all $j > i$, while the function of steps 6 through 10 in Algorithm 4 is to check whether $w_i \geq 0$ for all i from 1 to n.

Since the input of Algorithm 4 is a HNF basis H, H is upper triangular and each element is non-negative. Therefore, the Gram-Schmidt orthogonalization can be represented in the following form

$$\begin{aligned} H &= \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ 0 & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{nn} \end{bmatrix} \\ &= \begin{bmatrix} h_{11} & 0 & \cdots & 0 \\ 0 & h_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{nn} \end{bmatrix} \cdot \begin{bmatrix} 1 & \lambda_{12} & \cdots & \lambda_{1n} \\ 0 & 1 & \cdots & \lambda_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \\ &= H^*\Lambda \end{aligned}$$

where $\lambda_{ij} = \frac{h_{ij}}{h_{ii}}$. Hence, $\lambda_{ij} \geq 0$. Once the input HNF basis passes the check steps 6 through 10 in Algorithm 4, it means that the HNF basis can also pass the check Steps 6 through 12 in Algorithm 2 since λ_{ij} and w_j are both non-negative for all i and j. Therefore, the outputs $v^\dagger$ of Algorithm 2 and v' of Algorithm 4 are actually the same vector if the input of these two algorithms is the same. According to Theorem 3, the conclusion of this corollary holds. □

Corollary 2. *Given an arbitrary lattice presented on its Hermite normal form(HNF) basis B and a target vector t. Let v and v'' be the outputs of Algorithm 1 and Algorithm 5, respectively. Then the distance of t from v'' is not greater than that of t from v. In particular, if there is at least one coordinate of w appeared in Algorithm 2 whose absolute value is exactly the value 0.5, then the distance of t from $v^\ddagger$ will be strictly smaller than that of t from v.*

Proof. This proof is similar to the one above except that it is based on Theorem 4 instead of Theorem 3, omitted here. □

5 Examples and Experiments

In this section, we analyse our proposed algorithms through examples and experiments, and show that these

Algorithm 5 Special Edition for HNF II

```
Input:
    Lattice HNF basis $B$, and vector $t \in \mathbb{R}^n$
Output:
1:  Compute the GS-basis $B^*$ of $B$ and $\Lambda$ such that $B = B^*\Lambda$;
2:  Compute $B^{-1}$ and get $B^{-1}t$;
3:  Round $B^{-1}t$ to the closest integer vector $u = \lfloor B^{-1}t \rceil \in \mathbb{Z}^n$;
4:  Compute the vector $w = u - B^{-1}t$;
5:  $l = 0$;$r = 0$
6:  $k = 0$;
7:  for $i = 1$ to $n$ do
8:      if $w_i \geq 0$ then
9:          k++;
10:     end if
11: end for
12: if k=n then
13:     for $j = n - 1$ to 1 do
14:         if $|w_j| = 0.5$ then
15:             $l+ = w_j$;
16:             $r+ = -w_j$;
17:             for $i = j + 1$ to $n$ do
18:                 $l+ = \lambda_{ji} \cdot w_i$;
19:                 $r+ = \lambda_{ji} \cdot w_i$;
20:             end for
21:             if $| l | \geq | r |$ then
22:                 $w_j = -w_j$;
23:             end if
24:         end if
25:     end for
26: end if
27: return the resulting integer vector to the lattice point $v = B(B^{-1}t - w)$
```

algorithms outperform the classic Babai's algorithm with respect to their corresponding errors.

5.1 Examples

Example 1. Given a 6-rank lattice basis

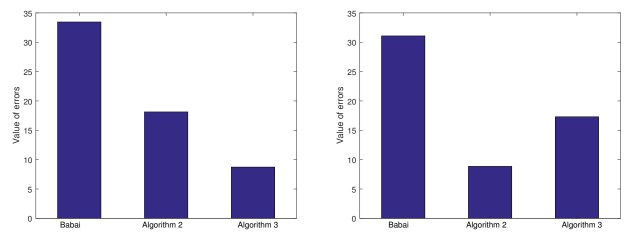
$$B = \begin{bmatrix} 10 & 10 & 2 & 3 & 5 & 6 \\ 9 & 2 & 6 & 5 & 1 & 7 \\ 5 & 0 & 5 & 8 & 10 & 6 \\ 1 & 6 & 6 & 6 & 9 & 3 \\ 5 & 3 & 8 & 2 & 1 & 4 \\ 4 & 9 & 6 & 9 & 1 & 4 \end{bmatrix}$$

and a target vector

$$t = [189.1, 157.6, 133.6, 129, 122.9, 175.6]$$

as input of Algorithms 1, 2, and 3, it is not difficult to obtain the outputs of these three algorithms as

$$v_1 = [205, 170, 148, 143, 133, 190],$$
$$v_2 = [178, 152, 128, 121, 116, 170],$$
$$v_3 = [188, 154, 128, 127, 119, 179],$$



(a) The case of Example 1 (b) The case of Example 2

Figure 1. The error comparison of three algorithms

respectively. The errors between these outputs and the target vector are as follows,

$$error_1 = \|v_1 - t\| = 33.4559,$$
$$error_2 = \|v_2 - t\| = 18.1356,$$
$$error_3 = \|v_3 - t\| = \ 8.7350,$$

Figure 1(a) shows a comparison of these errors. Both the proposed Algorithm 2 and Algorithm 3 output vectors that are closer to the target vector t than Algorithm 1 (i.e. Babai's algorithm) . In addition, $error_3$ is smaller than $error_2$, which means that Algorithm 3 outperforms Algorithm 2 for this input. In fact, the following experiments show that this phenomenon is common in the vast majority of inputs. However, there are exceptions, such as the input given in example 2.

Example 2. Given a 6-rank lattice basis

$$B = \begin{bmatrix} 3 & 4 & 7 & 4 & 3 & 6 \\ 1 & 4 & 2 & 9 & 1 & 9 \\ 1 & 7 & 7 & 5 & 1 & 8 \\ 6 & 6 & 3 & 6 & 1 & 8 \\ 3 & 2 & 4 & 8 & 7 & 6 \\ 10 & 4 & 4 & 9 & 0 & 2 \end{bmatrix}$$

and a target vector

$$t = [125.7, 110.7, 137.4, 150.8, 120.2, 170.2]$$

as input of Algorithms 1, 2 and 3. It is not difficult to obtain the outputs of these three algorithms as

$$v_1 = [138, 122, 150, 164, 134, 183],$$
$$v_2 = [125, 118, 141, 154, 120, 169],$$
$$v_3 = [132, 120, 148, 157, 124, 173],$$

respectively. The errors between these outputs and the target vector are as follows,

$$error_1 = \|v_1 - t\| = 31.0847,$$
$$error_2 = \|v_2 - t\| = \ 8.8578,$$
$$error_3 = \|v_3 - t\| = 17.2991,$$

Figure 1(b) shows a comparison of these errors. Both the proposed Algorithm 2 and Algorithm 3 also output vectors that are closer to the target vector t than Algorithm 1 (i.e. Babai's algorithm) . In this case, $error_2$ is smaller than $error_3$, which means that Algorithm 2 outperforms Algorithm 3 for this input.

5.2 Experiments

Our experiments are done on a Windows desktop PC with an Intel Core i5-7200U CPU running at 2.50 GHz. The algorithms are implemented in Matlab R2016b.

For comparison, we randomly generate 500 lattice bases with full rank 8 that can trigger the adjustment strategies in Algorithm 2 and Algorithm 3. Figure 2 shows a comparison of the errors generated by Algorithms 1, 2 and 3 with these randomly chosen lattice bases as inputs. It can be seen easily that our proposed Algorithm 2 and Algorithm 3 both outperform Babai's algorithm with respect to the corresponding errors. Figure 2 also shows Algorithm 3 outperforms Algorithm 2 for the vast majority of inputs. This is in line with our algorithm design expectations. After all, Algorithm 3 has a more accurate adjustment strategy than Algorithm 2.

Figure 3 shows the time overhead of these three algorithms. Our algorithms are slower than Babai's due to the time-consuming Gram-Schimdt procedure. However, this is not a big problem for cryptographic applications, since the implementation of a cryptographic algorithm involves only one selected lattice basis, and attacking the cryptographic algorithm only needs to solve the specific CVP problem of the lattice basis.

In addition to the above general case, we have also done experiments for the special case of HNF. For comparison, we randomly generate 100 lattice bases in HNF with full rank 10. Figure 4 shows a comparison of errors generated by all the five algorithms appeared in the paper. A strange thing is that there are only three curves in Figure 4. This is because Algorithms 4 and 5 are special cases of Algorithm 2 and Algorithm 3 respectively, so the errors they generate are exactly the same.

Although the errors generated are the same, it can be easily seen from Figure 5 that the computational cost of Algorithm 4 and Algorithm 5 is smaller than that of Algorithm 2 and Algorithm 3. In particular, Algorithm 4, in which there is no Gram-Schimidt process, has almost the same computational cost as Babai's algorithm.

6 Conclusions

In this paper, we propose four improved rounding procedures for CVP problem based on Babai's rouding procedure. The first two procedures are general for any type of lattice basis, while the latter two algorithms are special versions of the first two for which the input basis is in HNF form. We also show that all four algorithms perform better than Babai's procedure with respect to approximation factor, although they lose some efficiency as the cost.

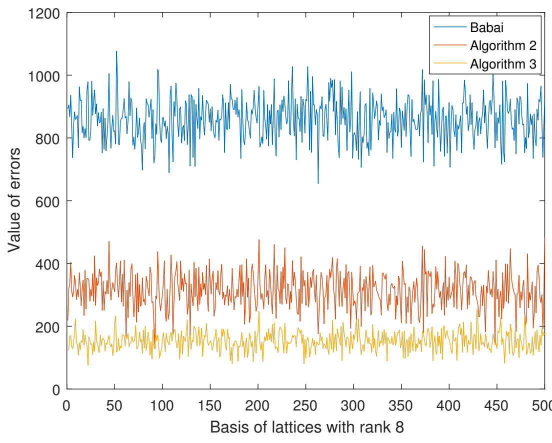


Figure 2. The error comparison of three algorithms

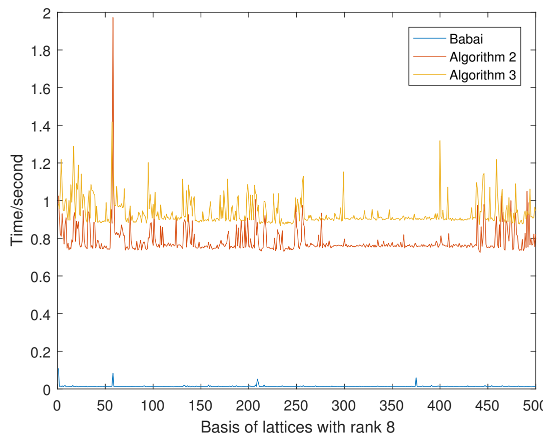


Figure 3. The efficience comparison of three algorithms

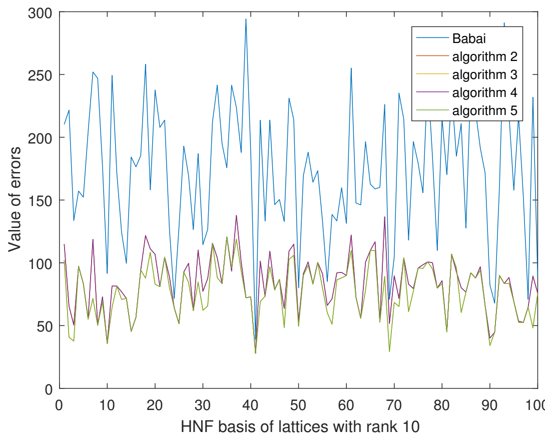


Figure 4. The efficience comparison of five algorithms

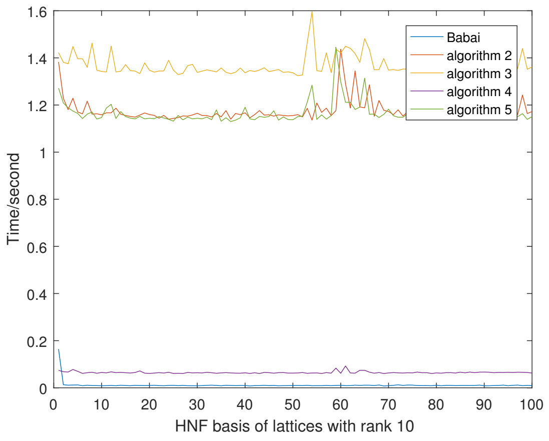


Figure 5. The efficience comparison of five algorithms

Acknowledgments

This work was supported by the Doctoral Fund of University of Jinan (Granted No. XBS1455), and National Science Foundation of Shandong Province (No. ZR2018LF006).

References

[1] D. Aggarwal, H. Bennett, A. Golovnev, and N. Stephens-Davidowitz, "Fine-grained hardness of cvp(p): everything that we can prove (and nothing else)," in *Symposium on Discrete Algorithms*, 2021.

[2] D. Aggarwal and Eldon Chung, "A note on the concrete hardness of the shortest independent vector in lattices," *Information Processing Letters*, vol. 167, no. 106065, pp. 1–5, 2021.

[3] S. Agrawal, E. Kirshanova, D. Stehle, and A. Yadav, "Practical, round-optimal lattice-based blind signatures," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2022.

[4] S. Agrawal, A. Yadav, and S. Yamada, "Multi-input attribute based encryption and predicate encryption," in *Proceedings of Crypto*, 2022.

[5] M. Ajtai, "Generating hard instances of lattice problems (extended abstract)," in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing*, pp. 99–108, ACM, 1996.

[6] M. Ajtai, R. Kumar, and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem," in *STOC*, pp. 601–610, 2001.

[7] L. Babai, "On lovász lattice reduction and the nearest lattice point problem," *Combinatorica*, vol. 6, no. 1, pp. 1–13, 1986.

[8] J. Bajard, J. Eynard, N. Merkiche, and T. Plantard, "Babai round-off cvp method in rns: Application to lattice based cryptographic protocols," *International Symposium on Integrated Circuits (ISIC)*, pp. 440–443, 2014.

[9] H. Bennett, C. Peikert, and Y. Tang, "Improved hardness of bdd and svp under gap-(s)eth," in *Proceedings of Innovations in Theoretical Computer Science (ITCS)*, 2022.

[10] V. E. Boas. "Another np-complete problem and the complexity of computing short vectors in a lattice,". Tech. Rep. Technical Report 81-04, University of Amsterdam, 1981.

[11] Z. Brakerski, R. Tsabary, V. Vaikuntanathan, and H. Wee, "Private constrained prfs (and more) from lwe," in *TCC 2017*, pp. 264–302, Baltimore, USA, November 2017.

[12] Z. Brakerski and V. Vaikuntanathan, "Lattice-inspired broadcast encryption and succinct ciphertext policy abe," in *ITCS*, 2022.

[13] R. Challa and G. V. Kumari, "Additively lwe based homomorphic encryption for compact devices with enhanced security," *International Journal of Network Security*, vol. 21, no. 3, pp. 378–383, 2019.

[14] O. Goldreich and S. Goldwasser, "On the limits of nonapproximability of lattice problems," *Journal of Computer and System Sciences*, vol. 60, no. 3, pp. 540–563, 2000.

[15] M. Jiang, Q. Chen, Y. Guo, and D. Zhang, "Multi-bit functional encryption for inner product predicate over lattice," *International Journal of Network Security*, vol. 24, no. 6, pp. 1106–1113, 2022.

[16] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients," *Mathematische Annalen*, vol. 261, no. 4, pp. 515–534, 1982.

[17] A. Mandangan, H. Kamarulhaili, and M. A. Asbullah, "Good basis vs bad basis: On the ability of babai's round-off method for solving the closest vector problem," *Journal of Physics: Conference Series*, vol. 1366, no. 1, p. 012016, 2019.

[18] P. Martins, J. Eynard, J. Bajard, and L. Sousa, "Arithmetical improvement of the round-off for cryptosystems in high-dimensional lattices," *IEEE Trans. on Computers*, vol. 66, no. 12, pp. 2005–2018, 2017.

[19] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective*. Netherlands: Kluwer Academic Publisher, 2002.

[20] D. Micciancio and C. Peikert, "Trapdoors for lattices: Simpler, tighter, faster, smaller," in *EUROCRYPT 2012*, pp. 700–718, Cambridge, United Kingdom, April 2012.

[21] S. Zhang, "A lwe-based oblivious transfer protocol from indistinguishability obfuscation," *International Journal of Network Security*, vol. 22, no. 5, pp. 801–808, 2020.

Biography

Shuying Yang received the B.S. and M.S. degree in Mathematics from Shandong Normal University, China, in 2004 and 2007, respectively. Currently, She is an associate professor in department of data and computer

science at Shandong Women's University. Her research interests include information security and cryptology. Prof. Yang may be reached at ysystudy2005@163.com.