IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 17, 2018, accepted October 6, 2018, date of publication October 16, 2018, date of current version November 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2876401

An Efficient Algorithm for the Shortest Vector Problem

YU-LUN CHUANG[1], CHUN-I FAN[1,2,3], AND YI-FAN TSENG[1]

[1]Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan
[2]Information Security Research Center, National Sun Yat-sen University, Kaohsiung 80424, Taiwan
[3]Intelligent Electronic Commerce Research Center, National Sun Yat-sen University, Kaohsiung 80424, Taiwan

Corresponding author: Chun-I Fan (cifan@mail.cse.nsysu.edu.tw)

This work was supported in part by the Taiwan Information Security Center, in part by the Ministry of Science and Technology of Taiwan under Grant MOST 105-2923-E-110-001-MY3 and Grant MOST 107-2218-E-110-014, in part by the Information Security Research Center, National Sun Yat-sen University, Taiwan, and in part by the Intelligent Electronic Commerce Research Center through the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education in Taiwan.

ABSTRACT Lattice is widely used in cryptography since it has potential for defending quantum attacks. One of the significant problems in such cryptography is the shortest vector problem (SVP). This problem is to find the non-zero shortest vector in lattice. The SVP is an NP-hard problem under randomized reductions proven by Ajtai, and many cryptosystems are secure under the assumption that SVP is hard, such as NTRU. On the other hand, some primitives of lattice-based cryptography require relatively short vectors. In this paper, we propose a new SVP algorithm that can be performed in time complexity $O(n^3)$. We also prove that the Hermite factor of the proposed algorithm is polynomial-bounded.

INDEX TERMS Shortest vector problem, algorithm analysis, optimization theory, lattice, lattice-based cryptography.

I. INTRODUCTION

Lattice is a discrete set consisting of some linearly independent vectors, $L(B) = \{\vec{v} \mid \vec{v} = \sum_{i=1}^{d} c_i\vec{b}_i$ where $\vec{b}_i \in B$ and $c_i \in \mathbb{Z}, \forall i\}$. It has been widely studied in cryptography [1], [2] since it is believed that lattice-based cryptography has potential to resist the attacks from quantum computers. One of the core hard problems in lattice-based cryptography is the shortest vector problem (SVP). That is, given a linearly independent basis $B = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n\} \in \mathbb{Z}^{m\times n}$, find a non-zero vector $\vec{v}$ such that $\|\vec{v}\| = \min_{\vec{z}\in B} \|\vec{z}\|$. SVP is proved to be an NP-Hard problem under randomized reductions by Ajtai [3] in 1998. In 2001 Micciancio [4] proved that the SVP problem is NP-Hard within any factor less than $\sqrt{2}$. The researches on solving the SVP play an important role in cryptography. In some lattice-based cryptosystems, the user needs to find a short vector, such as [5]. On the other hand, when we are constructing a lattice-based cryptosystem, we can derive the most appropriate security parameters according to the time/space complexity of the best algorithm in solving the SVP. Given an algorithm in solving the SVP, one can evaluate the algorithm by its time complexity, space complexity, and approximation factor α. An algorithm with approximation factor α means that it is able to compute a short vector whose length is not greater than $\alpha\lambda_1(L)$, where $\lambda_1(L)$ is the length of the shortest vector. If $\alpha = 1$, then the algorithm is able to find the shortest vector. The existing algorithms can be divided into two types, where one can be run in polynomial time while the approximation factor is exponentially large; the other is able to find a vector with approximation factor exponentially close to 1, i.e. $\alpha = 1$, however its time and space complexity are exponential.

A. RELATED WORKS

In 1982, Lenstra, Lenstra and Lovász proposed a lattice reduction algorithm LLL [6], [7] which can be performed in $O(n^5)$ with $\alpha \leq (4/3)^{(n-1)/2}$, where n is dimension. In 1983, Kannan [8], [9] proposed an exact algorithm HKZ which can be performed in $n^{\frac{n}{2e}+O(n)}$. In 1994, Schnorr and Euchner proposed a blockwise algorithm BKZ-β [10] where β is the block size, and it was implemented in NTL [11]. The block size β is an important parameter for the time complexity and α in BKZ. However, there is no good upper bound of time complexity about β and n. The experiment [12]–[14] showed that the performing time is sub-exponential in n as $\beta < 25$ and exponential in n as $\beta \geq 25$. In 2001, Ajtai *et al.* [15] proposed a sieve algorithm AKS which required exponentially large time and space complexity, and showed that α

2169-3536 © 2018 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

is exponentially close to 1. Nguyen and Vidick [16] showed that the time complexity of AKS is $O(2^{5.9n+O(n)})$ with the space complexity $O(2^{2.95n+O(n)})$. Moreover, Nguyen also proposed another sieve algorithm, called Listsieve, which is performed in time $O(2^{3.199n+O(n)})$ and space $O(2^{1.325n+O(n)})$. In 2008, Nguyen *et al.* used Mordell's inequality to propose slide-reduction algorithm and proved $\alpha = (\gamma_k(1+\varepsilon))^{\frac{n-k}{k-1}}$.

We can classify these algorithms into four types:

- Approximation algorithm (non-blockwise) [6]:
  It can be performed in polynomial time, but α is exponentially large in n.
- Approximation algorithm (blockwise) [10], [17]–[19]:
  It can be performed in sub-exponential time with appropriate k and α is exponentially large in $\frac{n}{k}$.
- Exact algorithm (polynomial space complexity) [8]:
  It can be performed in time complexity $2^{O(n\log n)}$.
- Exact algorithm (exponenial space complexity) [15], [16], [20]–[24]
  It can be performed in time complexity $2^{O(n)}$.

B. CONTRIBUTION

In this paper, we propose a new approximation algorithm for solving the SVP. Our algorithm is motivated from some techniques in the optimization theory. That is, add some noise and find the critical point of a distance function. The time complexity of the proposed algorithm is $O(n^3)$ and only polynomially large space is needed. The most special feature of the proposed algorithm is that the Hermite factor of our algorithm is a polynomial-bounded function in the number of the dimensions. With the best of our knowledge, it is the first algorithm with polynomial-bounded Hermite factor and polynomial time complexity.

II. PRELIMINARY

In this section, we introduce the shortest vector problem and some theorems on lattice.

A. LATTICE

Lattice is a set containing all integer linear combinations of a basis. Given a basis $B = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n\} \in \mathbb{Z}^{m\times n}$, define the lattice of B as follows.

$$L(B) = \{\vec{v} \,|\, \vec{v} = \sum_{i=1}^{n} c_i\vec{b}_i, \forall \vec{b}_i \in B, \forall c_i \in \mathbb{Z}, i = 1, 2, \ldots, n\}.$$

B. MATRIX FORM

Let $B = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n\} \in \mathbb{Z}^{m\times n}$, then we denote the matrix form of B as $M(B)$ where

$$M(B) = \left[\, \vec{b}_1 \middle| \vec{b}_2 \middle| \vec{b}_3 \middle| \cdots \middle| \vec{b}_n \,\right].$$

C. THE SHORTEST VECTOR PROBLEM

Given a basis $B = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n\} \in \mathbb{Z}^{m\times n}$, the shortest vector problem is to find a vector $\vec{v}$ satisfying

$$\|\vec{v}\| = \min_{\vec{u}\in L(B)/0} \|\vec{u}\| = \lambda_1(L(B)).$$

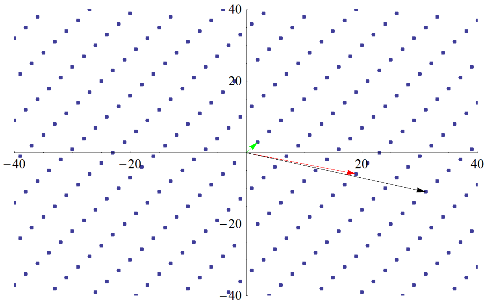

FIGURE 1. Example of the shortest vector problem. Let $B = \{[19, -6], [31, -11]\}$, then the nonzero shortest vector is $5 \times [19, -6] - 3 \times [31, -11] = [2, 3]$.

For example, given $B = \{[19, -6], [31, -11]\}$, then we can generate the lattic set (Figure 1), and the shortest vector is [2, 3]. SVP is an NP-hard problem under randomized reductions proved by Ajtai [3]. Currently, there is no polynomial time algorithm to verify whether a vector is the solution of SVP or not. Therefore, we will use Minkowski's theorems or Hermite factor to test the solution. Note that the shortest vector might not be unique in lattice.

D. MINKOWSKI'S THEOREM

Theorem 1 (Minkowski's First Theorem): Let B be a basis in $\mathbb{R}^n$ and $\lambda_1(L(B))$ be the first Minkowski's minimum in ∞-norm of $L(B)$, then $\lambda_1(L(B)) \le det(L(B))^{1/n}$.

Theorem 2 (Minkowski's Second Theorem): Let B be a basis in $\mathbb{R}^n$ and $\lambda_i(L(B))$ be the i-th Minkowski's minimum in ∞-norm of $L(B)$ for $i = 1, 2, \ldots, n$, then $\prod_{i=1}^{n} \lambda_i(L(B)) \le 2^n * det(L(B))$.

Let $\vec{w} = \sum_{i=1}^{n} f_i\vec{b}_i$ be the shortest vector of $L(B)$ for some vector $\vec{f} = [f_i]_n$. We can estimate the upper bound of $\|\vec{f}\|_\infty$ by Minkowski's theorems. Then we have

$$\begin{aligned}\|\vec{f}\|_\infty &= \|M(B)^{-1}\vec{w}\|_\infty \\ &\le \|M(B)^{-1}\|_\infty\|\vec{w}\|_\infty \le \|M(B)^{-1}\|_\infty det(L(B))^{1/n}.\end{aligned}$$

Remark: $M(B)^{-1}$ is the pseudoinverse of $M(B)$.

E. NORM SPACE

Definition 1: Let $\vec{x}$ be a vector in $\mathbb{R}^n$ and $q \in \mathbb{R}$, then we define its q-norm and ∞-norm as follow:

$$\|\vec{x}\|_q = \sqrt[q]{x_1^q + x_2^q + \ldots + x_n^q}.$$
$$\|\vec{x}\|_\infty = \max_i |x_i|.$$

Definition 2: Let $M(B)$ be a matrix in $\mathbb{R}^{m\times n}$ and $q \in \mathbb{R}$, then we define its q-norm as follow:

$$\begin{aligned}\|M(B)\|_q &= \max_{\|\vec{x}\|_q\neq 0} \frac{\|M(B)\vec{x}\|_q}{\|\vec{x}\|_q} \\ &= \max_{\|\vec{x}\|_q\neq 0} \|M(B)\frac{\vec{x}}{\|\vec{x}\|_q}\|_q = \max_{\|\vec{x}\|_q=1} \|M(B)x\|_q\end{aligned}$$

Algorithm 1

```
Input: A basis $B = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n\}$, where $\vec{b}_i \in \mathbb{Z}^m$.
Step 1: Set $e_i = 1$ for $i = 1, 2, \ldots, n$.
Step 2: Construct a distance function
    $S = \sum_{i=1}^{m}(\sum_{j=1}^{n}(\vec{b}_j)_i x_j)^2 + (-1 + \sum_{j=1}^{n} e_j x_j)^2.$
Step 3: Compute $\frac{\partial S}{\partial x_i}$ for $i = 1, 2, \ldots, n$.
Step 4: Get $x_i = c_i$ by solving the linear system below.
    $\begin{cases} \frac{\partial S}{\partial x_1} = 0 \\ \frac{\partial S}{\partial x_2} = 0 \\ \vdots \\ \frac{\partial S}{\partial x_n} = 0 \end{cases}$
Step 5: Choose $t = \max_{1 \le i \le n} |c_i|$, compute $u_i = \frac{c_i}{t}$.
Step 6: For $i = 1, 2, \ldots, \|M(B)^{-1}\|_\infty det(L(B))^{\frac{1}{n}}$, compute $\vec{v}_i = \sum_{j=1}^{n} r_{ij}\vec{b}_j$, where $r_{ij} = \lceil u_j * i \rfloor$.
Step 7: output $\vec{v}'$, where $\|\vec{v}'\|_2 = \min_i \|\vec{v}_i\|_2$.
Note: $\lceil \cdot \rfloor$ is a round function.
```

Algorithm 2

```
Input: A basis $B = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n\}$, where $\vec{b}_i \in \mathbb{Z}^m$.
Step 1: Set $e_i = 1$ for $i = 1, 2, \ldots, n$.
Step 2: Construct a distance function
    $S = \sum_{i=1}^{m}(\sum_{j=1}^{n}(\vec{b}_j)_i x_j)^2 + (-1 + \sum_{j=1}^{n} e_j x_j)^2.$
Step 3: Compute $\frac{\partial S}{\partial x_i}$ for $i = 1, 2, \ldots, n$.
Step 4: Get $x_i = c_i$ by solving the linear system below.
    $\begin{cases} \frac{\partial S}{\partial x_1} = 0 \\ \frac{\partial S}{\partial x_2} = 0 \\ \vdots \\ \frac{\partial S}{\partial x_n} = 0 \end{cases}$
Step 5: If $\exists c_i e_i < 0$, then $e_i \leftarrow -e_i$ and go to Step 2.
Step 6: Choose $t = \max_{1 \le i \le n} |c_i|$, compute $u_i = \frac{c_i}{t}$.
Step 7: For $i = 1, 2, \ldots, \|M(B)^{-1}\|_\infty det(L(B))^{\frac{1}{n}}$, compute $\vec{v}_i = \sum_{j=1}^{n} r_{ij}\vec{b}_j$, where $r_{ij} = \lceil u_j * i \rfloor$.
Step 8: output $\vec{v}'$, where $\|\vec{v}'\|_2 = \min_i \|\vec{v}_i\|_2$.
```

Let B be a basis in $\mathbb{R}^{m\times n}$ and $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$ be the singular values of $M(B)$. Since $M(B)^T M(B)$ is Hermitian matrix, it can be written as $M(B)^T M(B) = V\Lambda V^T$, where $VV^T = I$. Then we have

$$\begin{aligned} \|M(B)\|_2^2 &= \max_{\|\vec{x}\|_2=1} \|M(B)\vec{x}\|_2^2 \\ &= \max_{\|\vec{x}\|_2=1} \vec{x}^T(M(B)^T M(B))\vec{x} \\ &= \max_{\|\vec{x}\|_2=1} \vec{x}^T V\Lambda V^T \vec{x} \\ &= \max_{\|\vec{y}\|_2=1} \vec{y}^T \Lambda \vec{y} \\ &= \max_{\|\vec{y}\|_2=1} y_1^2\sigma_1^2 + y_2^2\sigma_2^2 + \ldots + y_n^2\sigma_n^2 \\ &\leq \max_{\|\vec{y}\|_2=1} \sigma_1^2(y_1^2 + y_2^2 + \ldots + y_n^2) \\ &= \sigma_1^2 \end{aligned}$$

Remark:

- $\|\vec{y}\|_2^2 = \|V^T\vec{x}\|_2^2 = \vec{x}^T VV^T\vec{x} = \vec{x}^T\vec{x} = \|\vec{x}\|_2^2 = 1$.
- The equality is hold as $\vec{y} = [1, 0, 0, \ldots, 0]$. Thus $\|M(B)\|_2 = \sigma_1$.
- It is obvious that $\|M(B)^{-1}\|_2 = \frac{1}{\sigma_n}$ since $\Lambda^{-1} = diag(\frac{1}{\sigma_1^2}, \frac{1}{\sigma_2^2}, \ldots, \frac{1}{\sigma_n^2})$.

We give some properties of norms. Let $x, y \in S$, and then $p : S \to \mathbb{R}$ is a norm if and only if

- $p(x) \geq 0$ and $p(x) = 0$ if and only if $x = 0$.
- For all $a \in F$, $p(ax) = |a|p(x)$.
- $p(x + y) \leq p(x) + p(y)$.

III. OUR ALGORITHM

This section presents a new algorithm (**Algorithm 1**) for the shortest vector problem. The details of the algorithm will be shown in Section III-A. Our main concept is based on finding the critical point of a distance function and then we find the ratio between each component of the shortest vector. However, we will just get the trivial solution if we solve it directly. Hence in the initial step (Step 1), we add some noises $e_i = 1$ for each vector such that $\sum_{i=1}^{n} c_i e_i \approx 1$. In Step 2 and Step 3, in order to simplify the procedure, we construct the linear system by computing $\frac{\partial S}{\partial x_i}$, but it can be computed by performing the inner product in the implementation. That is, for each component of the matrix $M = [a_{ij}]_{n\times n}$, a_{ij} is equal to $< \vec{b}_i, \vec{b}_j > +e_i e_j$. In Step 4, we can find the critical point by Gaussian elimination or any effective algorithm. In Step 5 to Step 7, we will recover the ratio from a rational number to an integer. Moreover, we improve **Algorithm 1** and give **Algorithm 2**. For the simplicity of the analysis, we analyse the algorithm under some assumptions, as shown in Section IV-C.

A. DISCUSSION

1) COST OF STEP 3

Consider a basis $B = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n\} \in \mathbb{Z}^{m\times n}$ with its corresponding distance function

$$S = \sum_{i=1}^{m}(\sum_{j=1}^{n}(\vec{b}_j)_i x_j)^2 + (-1 + \sum_{j=1}^{n} e_j x_j)^2.$$

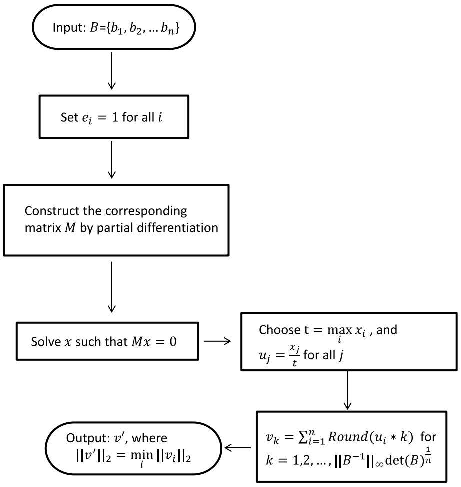
Input: B={$b_1, b_2, \ldots b_n$}
Set e_i = 1 for all i
Construct the corresponding matrix M by partial differentiation
Solve x such that $Mx = 0$
Choose t = $\max_i x_i$, and $u_j = \frac{x_j}{t}$ for all j
$v_k = \sum_{i=1}^n Round(u_i * k)$ for $k = 1,2,\ldots, ||B^{-1}||_\infty \det(B)^{\frac{1}{n}}$
Output: v', where $||v'||_2 = \min_i ||v_i||_2$

FIGURE 2. **The flowchart of Algorithm 1.**

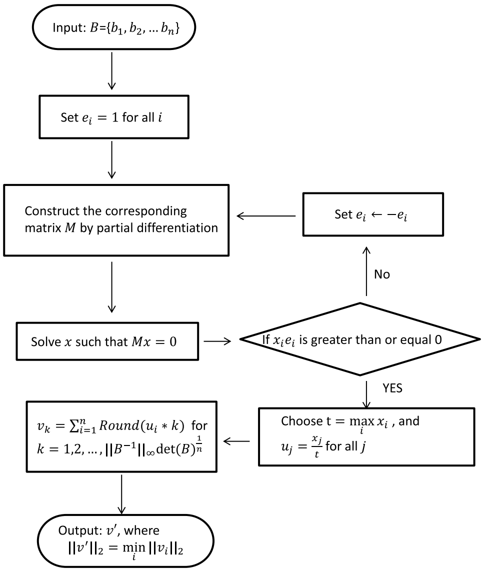
Input: B={$b_1, b_2, \ldots b_n$}
Set e_i = 1 for all i
Construct the corresponding matrix M by partial differentiation
Set $e_i \leftarrow -e_i$
No
Solve x such that $Mx = 0$
If x_ie_i is greater than or equal 0
YES
Choose t = $\max_i x_i$, and $u_j = \frac{x_j}{t}$ for all j
$v_k = \sum_{i=1}^n Round(u_i * k)$ for $k = 1,2,\ldots, ||B^{-1}||_\infty \det(B)^{\frac{1}{n}}$
Output: v', where $||v'||_2 = \min_i ||v_i||_2$

FIGURE 3. **The flowchart of Algorithm 2.**

Let $\vec{b}'_i = [\vec{b}_i, e_i]$ for $i = 1, 2, \ldots, n$. Then we compute

$$
\begin{aligned}
&\frac{\partial S}{\partial x_q}\\
&= \sum\nolimits_{i=1}^{m} 2(\sum\nolimits_{j=1}^{n} (\vec{b}_j)_i x_j) \times (\vec{b}_q)_i + 2(-1 + \sum\nolimits_{j=1}^{n} e_j x_j) \times e_q\\
&= 2\sum\nolimits_{j=1}^{n} (\sum\nolimits_{i=1}^{m} (\vec{b}_j)_i (\vec{b}_q)_i x_j) + 2(\sum\nolimits_{j=1}^{n} e_j e_q x_j) - 2e_q\\
&= 2[\sum\nolimits_{j=1}^{n} (\sum\nolimits_{i=1}^{m} (\vec{b}_j)_i (\vec{b}_q)_i + e_j e_q) x_j - e_q]\\
&= 2[\sum\nolimits_{j=1}^{n} < \vec{b}'_j, \vec{b}'_q > x_j - e_q].
\end{aligned}
$$

It means that we can perform the inner product to compute each component of M.

2) THE UPPER BOUND OF COEFFICIENT

Let $\vec{w} = \sum_{i=1}^{n} f_i \vec{b}_i$ be the shortest vector in $L(B)$, then $\vec{f} = [f_1, f_2, \ldots, f_n]$ can be written as follow.

$$
\vec{f} = \left[\vec{b}_1 \middle| \vec{b}_2 \middle| \cdots \middle| \vec{b}_n\right]^{-1} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}
$$

By Minkowski's first theorem, we have $\|\vec{w}\|_\infty \leq det(L(B))^{\frac{1}{n}}$. Thus, the inequality will hold.

$$
\begin{aligned}
\|\vec{f}\|_\infty &= \|M(B)^{-1}\vec{w}\|_\infty\\
&\leq \|M(B)^{-1}\|_\infty \|\vec{w}\|_\infty\\
&\leq \|M(B)^{-1}\|_\infty det(L(B))^{\frac{1}{n}}
\end{aligned}
$$

3) CHANGE THE SIGN IN STEP 5

In algorithm 2, we want to minimize the distance function S in our main concept. It can be shown as $S = D_1 + D_2$, where

$$
\begin{cases}
D_1 = \sum_{i=1}^{m} (\sum_{j=1}^{n} (\vec{b}_j)_i x_j)^2\\
D_2 = (-1 + \sum_{j=1}^{n} e_j x_j)^2.
\end{cases}
$$

In D_2, there exist two kinds of situations by Step 4, (a) $\exists c_i e_i < 0$ or (b) $\forall c'_i e'_i \geq 0$. Since the goal is $\sum_{i=1}^{n} c_i e_i \approx 1$, we predict that the variability of $(c_1, c_2, \ldots, c_n)$ is relatively large in (a). That is, $\sum_{i=1}^{n} (c_i)^2 > \sum_{i=1}^{n} (c'_i)^2$.

Now, we consider D_1, the upper bound of D_1 is

$$
\begin{aligned}
D_1^2 &= \| \sum\nolimits_{i=1}^{n} x_i \vec{b}_i \|^2\\
&= \sum\nolimits_{i=1}^{n} x_i^2 \|\vec{b}_i\|^2\\
&\leq (\sum\nolimits_{i=1}^{n} x_i^2)(\sum\nolimits_{i=1}^{n} \|\vec{b}_i\|^2).
\end{aligned}
$$

It can be found that the upper bound in (b) is better than (a). Therefore, we require the situation of (b).

4) THE SUM OF c_ie_i IS IN [0, 1]

In geometric meaning, we generate a plane Ξ which containing all vector $\{\vec{v} \mid \vec{v} = \sum_{i=1}^{n} t_i [\vec{b}_i, e_i]$ for all $t_i \in \mathbb{R}$ and $\vec{b}_i \in B\}$. Since $0 \in \Xi$, the plane of Ξ can be written as

$$
\tau_1 \, x_1 + \tau_2 \, x_2 + \ldots + \tau_n x_n + y = 0.
$$

Through optimization theory, we find a point $\vec{c}$ on Ξ such that $\|\vec{c} - \vec{\varphi}\|_2$ has the minimum, where $\vec{\varphi} = [0, 0, \ldots, 0, 1]$. That is, the vector $\vec{c} - \vec{\varphi}$ is perpendicular to Ξ. Thus, $\vec{c}$ can be written as

$$
\begin{cases}
c_1 = 0 + t\tau_1\\
c_2 = 0 + t\tau_2\\
\vdots\\
c_n = 0 + t\tau_n\\
c_{n+1} = 1 + t
\end{cases}
$$
$$
\Rightarrow t\tau_1^2 + t\tau_2^2 + \ldots + t\tau_n^2 + 1 + t = 0.
$$

Finally, we have $t = \frac{-1}{1+\sum\limits_{i=1}^{n}\tau_i^2}$.

$$\sum\nolimits_{i=1}^{n} c_i e_i = c_{n+1} = \frac{\sum\limits_{i=1}^{n}\tau_i^2}{1+\sum\limits_{i=1}^{n}\tau_i^2}.$$

IV. ANALYSIS

A. CORRECTNESS

In the proposed algorithm, we can find a ratio $(u_1, u_2, \ldots, u_n)$ between each component of the shortest vector. Let $\vec{v}' = \sum\limits_{i=1}^{n} u_i k \vec{b}_i$ and $\vec{w} = \sum\limits_{i=1}^{n} f_i \vec{b}_i$ be the shortest vector. Now, we prove that $\exists k$ such that $\forall i$, $|u_i * k - f_i| \leq \eta = \frac{\sigma_1}{\sigma_n}\sqrt{n}$ where σ_1 and σ_n are the maximum and minimum singular value of $M(B)$.

First, let $k = \frac{\sum_{i=1}^{n} f_i}{\sum_{i=1}^{n} u_i}$ and $r_i = u_i * k$. Thus $\sum\limits_{i=1}^{n} r_i - \sum\limits_{i=1}^{n} f_i = 0$. Let $\sum\limits_{i=1}^{n} r_i \vec{b}_i - \sum\limits_{i=1}^{n} f_i \vec{b}_i = \sum\limits_{i=1}^{n} z_i \vec{b}_i$ where $z_i = r_i - f_i$. It is obvious that

$$\begin{aligned}
\sum_{i=1}^{n} z_i &= \sum_{i=1}^{n} r_i - \sum_{i=1}^{n} f_i \\
&= \sum_{i=1}^{n} u_i * k - \sum_{i=1}^{n} f_i \\
&= \frac{\sum_{i=1}^{n} f_i}{\sum_{i=1}^{n} u_i} \sum_{i=1}^{n} u_i - \sum_{i=1}^{n} f_i \\
&= 0.
\end{aligned}$$

Definition 3: Let E_p be the plane which contain the vector set $\{\vec{g} \mid \vec{g} = \sum\limits_{i=1}^{n} q_i e_i \vec{b}_i$ with $\sum\limits_{i=1}^{n} q_i = p\}$, where e_i is chosen in step 1 of our algorithm 1 or step 5 of our algorithm 2 for all i. The optimization theory ensures that $\vec{c} = \{c_1, c_2, \ldots, c_n\}$, which is the corresponding ratio vector, satisfy $\vec{v}' = \sum\limits_{i=1}^{n} u_i k_i \vec{b}_i$ will perpendicular to E_p for all k and p, where $u_i = c_i / \max\limits_i |c_i|$ (Figure 4).

Remark: There is an exact $p' > 0$ such that $\vec{w} \in E_{p'}$, we denote this plane as E. For example, in algorithm 1, if $B = \{\vec{b}_1, \vec{b}_2, \vec{b}_3, \vec{b}_4\}$ with the shortest vector $\vec{w} = \vec{b}_1 - 3\vec{b}_2 + 4\vec{b}_4$ in $L(B)$, then $\vec{w} \in E_2 = \{\vec{g} \mid \vec{g} = \sum\limits_{i=1}^{4} q_i \vec{b}_i$, where $\sum\limits_{i=1}^{4} q_i = 2\}$.

Theorem 3: Let $z_i \in \mathbb{R}$ for each $i = 1, 2, \ldots, n$ with $\sum\limits_{i=1}^{n} z_i = 0$. Then $\exists z_i' \in \mathbb{Z}$ such that $|z_i - z_i'| < 1$ for each i.

Proof: Let $z_i \in [\alpha_i, \alpha_i + 1)$ where $\alpha_i \in \mathbb{Z}$ for all i. We have

$$\beta = \sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} (\alpha_i - z_i) + \sum_{i=1}^{n} z_i = \sum_{i=1}^{n} (\alpha_i - z_i) \geq -n.$$

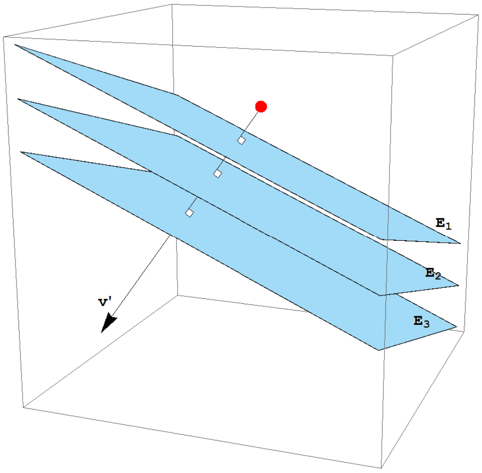

FIGURE 4. **The schematic diagrams of E_p and $\vec{v}'$.**

Choose a set $\Omega = \{i_1, i_2, \ldots, i_{(-\beta)}\} \subseteq \{1, 2, \ldots, n\}$ with $i_s \neq i_t$ $\forall s \neq t$. Finally, set

$$\begin{aligned}
z_i' &= \begin{cases} \alpha_i + 1 & , i \in \Omega \\ \alpha_i & , i \notin \Omega \end{cases} \\
&\Rightarrow \sum_{i=1}^{n} z_i' = \sum_{i \in \Omega} (\alpha_i + 1) + \sum_{i \notin \Omega} \alpha_i \\
&= (\sum_{i=1}^{n} \alpha_i) + (-\beta) = \beta - \beta = 0.
\end{aligned}$$

This concludes the proof. □

If each $|z_i| \leq \eta$, then the proof is done. If $\exists |z_i| > \eta > 1$, we can always find a $z_i' \in \mathbb{Z}$ satisfying $\sum\limits_{i=1}^{n} z_i' = 0$ such that $|z_i - z_i'| \leq 1$ by Theorem 3. Thus,

$$\begin{aligned}
\sum_{i=1}^{n} r_i \vec{b}_i - \sum_{i=1}^{n} f_i \vec{b}_i &= \sum_{i=1}^{n} z_i \vec{b}_i \\
&= \sum_{i=1}^{n} (z_i - z_i') \vec{b}_i + \sum_{i=1}^{n} z_i' \vec{b}_i \ \textit{and}
\end{aligned}$$

$$\sum_{i=1}^{n} r_i \vec{b}_i - \sum_{i=1}^{n} (f_i + z_i') \vec{b}_i = \sum_{i=1}^{n} (z_i - z_i') \vec{b}_i.$$

Note that:

1) Let $\vec{w}' = \sum\limits_{i=1}^{n} (f_i + z_i') \vec{b}_i$ be the output of our algorithm on the plane E since $\sum\limits_{i=1}^{n} (f_i + z_i') = \sum\limits_{i=1}^{n} f_i$.
2) Since $\vec{w}$ and $\vec{w}'$ are on E and $\vec{v}'$ is perpendicular to E, $\|\vec{w}\|_2^2 = \|\vec{v}'\|_2^2 + \|\sum\limits_{i=1}^{n} z_i \vec{b}_i\|^2$ and $\|\vec{w}'\|_2^2 = \|\vec{v}'\|_2^2 + \|\sum\limits_{i=1}^{n} (z_i - z_i') \vec{b}_i\|^2$.

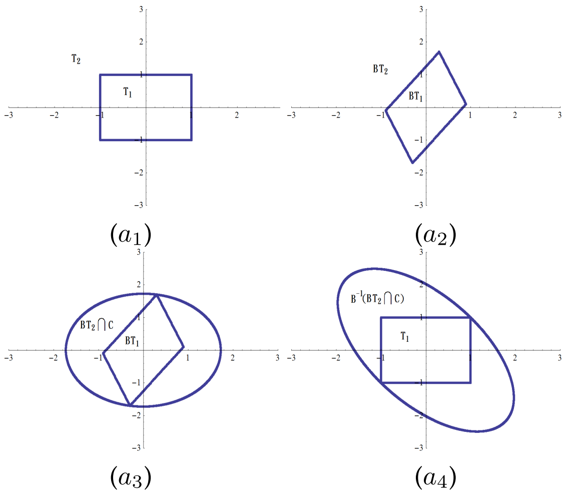

FIGURE 5. The schematic diagrams of P_1 and P_2.

3) Since $\vec{w}$ is the shortest vector.

$$\Rightarrow \|\vec{v}'\|_2^2 + \|\sum_{i=1}^{n} z_i\vec{b}_i\|^2 = \|\vec{w}\|_2^2$$
$$< \|\vec{w}'\|_2^2$$
$$= \|\vec{v}'\|_2^2 + \|\sum_{i=1}^{n}(z_i - z_i')\vec{b}_i\|^2.$$

This means that if $\exists |z_i| > \eta$, then the inequality $\|\sum_{i=1}^{n} z_i\vec{b}_i\|_2 < \|\sum_{i=1}^{n}(z_i - z_i')\vec{b}_i\|_2$ must hold for all $(z_1', z_2', \ldots, z_n')$ chosen from Theorem 3.

Let $P_1 = \sum_{i=1}^{n}(z_i - z_i')\vec{b}_i$, $P_2 = \sum_{i=1}^{n} z_i\vec{b}_i$, $T_1 = \{\vec{y} \mid \|\vec{y}\|_\infty \leq 1, \vec{y} \in \mathbb{R}^n\}$ and $T_2 = \{\vec{y} \mid \|\vec{y}\|_\infty > 1, \vec{y} \in \mathbb{R}^n\}$. The basis $B = \{\vec{b}_1, \vec{b}_2, \ldots, \vec{b}_n\}$ can be viewed as a map $M(B)$. Thus we have $P_1 \in M(B)T_1$ and $P_2 \in M(B)T_2$ (Figure 5). Since each component of an element in T_1 is bounded, we can find a upper bound of $\|P_1\|_2$. That is, we set a sphere C whose center is 0 and radius is $\gamma = \max_{\varphi \in T_1} \|M(B)\varphi\|_2$. It is obvious that if $P_2 \in M(B)T_2 \cap C$, then $\|P_1\|_2$ may be greater than $\|P_2\|_2$. Finally, we use the pseudoinverse map $M(B)^{-1}$ to find the region $T_3 = M(B)^{-1}(M(B)T_2 \cap C)$. We can find that

- If $\vec{z} = (z_1, z_2, \ldots, z_n) \in T_3$, then the inequality $\|\sum_{i=1}^{n} z_i\vec{b}_i\|_2 < \|\sum_{i=1}^{n}(z_i - z_i')\vec{b}_i\|_2$ may hold.
- If $\vec{z} \in T_2 - T_3$, then the inequality doesn't hold. That is, we always can find another lattice point $\vec{w}'$ such that $\|\vec{w}'\|_2 < \|\vec{w}\|_2$, this is a contradiction.

Now, we should find the upper bound of vector which in T_3.

$$\max_{\vec{\zeta} \in T_3} \|\vec{\zeta}\|_2 = \max_{\vec{\zeta} \in M(B)^{-1}(M(B)T_2 \cap C)} \|\vec{\zeta}\|_2$$
$$= \max_{\vec{\zeta} \in M(B)^{-1}C} \|\vec{\zeta}\|_2$$
$$= \max_{\vec{\zeta} \in C} \|M(B)^{-1}\vec{\zeta}\|_2$$
$$\leq \max_{\vec{\zeta} \in C} \|M(B)^{-1}\|_2 \|\vec{\zeta}\|_2$$
$$= \max_{\vec{\zeta} \in T_1} \|M(B^{-1})\|_2 \|M(B)\vec{\zeta}\|_2$$
$$\leq \max_{\vec{\zeta} \in T_1} \|M(B)^{-1}\|_2 \|M(B)\|_2 \|\vec{\zeta}\|_2$$
$$\leq \|M(B)^{-1}\|_2 \|M(B)\|_2\| \sqrt{n}$$
$$= \frac{\sigma_1}{\sigma_n}\sqrt{n},$$

where σ_1 and σ_n are the square roots of maximum and minimum eigenvalues of $M(B)^T M(B)$.

Remark:

- The point P_1 is not unique. That is, if P_1 is on the boundary of $M(B)T_1$, then there may exists another point P chosen from Theorem 3 in the interior of $M(B)T_1$, but we can prove it exactly.
- In fact, since $\sum_{i=1}^{n}(z_i - z_i') = 0$, $P_1 \in M(B)T_1 \cap M(B)E'$, where $E' = \{[t_1, t_2, \ldots, t_n] \mid \sum_{i=1}^{n} t_i = 0$, where $t_i \in \mathbb{R}\, \forall i = 1, 2, \ldots, n\}$.

B. QUALITY

In this section, we will show that the Hermite factor of the proposed algorithm is less than $(\sqrt{n}+(\frac{\sigma_1}{\sigma_n})^2\, n)$. First, we have shown that a ratio $(u_1, u_2, \ldots, u_n)$ can be found such that there exists a k satisfying $|ku_i - f_i| < \eta$ for all i, where the shortest vector is $\vec{w} = \sum_{i=1}^{n} f_i\vec{b}_i$.

Let $\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$ be the square roots of the eigenvalue of $M(B)^T M(B)$. The following inequality holds.

$$\sigma_1 \leq \sigma_1(\frac{\sigma_1}{\sigma_n}\frac{\sigma_2}{\sigma_n}\cdots\frac{\sigma_n}{\sigma_n})^{\frac{1}{n}} \leq \frac{\sigma_1}{\sigma_n}det(L(B))^{\frac{1}{n}}$$

Hence, we have

$$\|\vec{w}+\sum_{i=1}^{n}(ku_i - f_i)\vec{b}_i\|_2 \leq \|\vec{w}\|_2 + \|\sum_{i=1}^{n}(ku_i - f_i)\vec{b}_i\|_2$$
$$\leq \sqrt{n}det(L(B))^{\frac{1}{n}} + \|M(B)\|_2\|\sum_{i=1}^{n}(ku_i - f_i)\|_2$$
$$\leq \sqrt{n}det(L(B))^{\frac{1}{n}} + \sigma_1\frac{\sigma_1}{\sigma_n}n$$
$$\leq \sqrt{n}det(L(B))^{\frac{1}{n}} + (\frac{\sigma_1}{\sigma_n})^2\, ndet(L(B))^{\frac{1}{n}}$$
$$= (\sqrt{n} + (\frac{\sigma_1}{\sigma_n})^2\, n)det(L(B))^{\frac{1}{n}}.$$

C. TIME COMPLEXITY

We have shown that the inner product can be applied to construct the matrix M in Section III. On the other hand, it requires $O(n^3)$ to solve the linear system in Step 4. Thus, the algorithm 1 can be performed in $O(n^3)$ obviously.

However, we will change the sign of e_i if $c_ie_i < 0$ and go to step 2 in step 5 in algorithm 2. Now, we will show that the probability of that the "go to Step 2" condition (in Step 5) happens more than twice is negligible with some assumptions. Let $M = [a_{ij}]_{n\times n}$. In order to simplify the analysis, assume that:

1. The components a_{ij}'s of matrix M are independent and $a_{ij} \sim \mu$, where μ is a symmetric distribution.
2. For two linear systems, $M\vec{x} = \vec{e}$ and $(M - [(-1)^k]_{n\times n})\vec{x}' = \vec{e}$ for some $k \in \{0, 1\}$, $x_ix_i' \geq 0$ for all i.

First, we construct a matrix $M = [< \vec{b}_i, \vec{b}_j > +e_ie_j]_{n\times n}$ from a basis $B = \{\vec{b}_1, \vec{b}_2 \ldots, \vec{b}_n\} \in \mathbb{R}^{m\times n}$. By the second assumption, the sign of the solution of $M\vec{x} = \vec{e}$ is the same as the sign of the solution of $M' = [< \vec{b}_i, \vec{b}_j >] = [a_{ij}]_{n\times n}$.

Then we get a matrix $M'' = M'/\max\limits_{i,j}|a_{ij}|$. By the first assumption, each component of $M'' = [a'_{ij}]$ with $a'_{ij} \sim \mu'$, where μ' is a symmetric distribution in $[-1, 1]$. By the Cramer's rule, a solution $\vec{x} = (x_1, x_2, \ldots, x_n)$ can be obtained, where

$$x_j = \frac{1}{\max\limits_{i,j}|a_{ij}|}\frac{\begin{vmatrix} a'_{11} & \cdots & a'_{1j-1} & e_1 & a'_{1j+1} & \cdots & a'_{1n} \\ a'_{21} & \cdots & a'_{2j-1} & e_2 & a'_{2j+1} & \cdots & a'_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a'_{n1} & \cdots & a'_{nj-1} & e_n & a'_{nj+1} & \cdots & a'_{nn} \end{vmatrix}}{\begin{vmatrix} a'_{11} & a'_{12} & a'_{13} & \cdots & a'_{1n-1} & a'_{1n} \\ a'_{21} & a'_{22} & a'_{23} & \cdots & a'_{2n-1} & a'_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a'_{n1} & a'_{n2} & a'_{n3} & \cdots & a'_{nn-1} & a'_{nn} \end{vmatrix}}$$

$$= \frac{1}{\max\limits_{i,j}|a_{ij}|}\frac{\sum_i^n(-1)^{i+j}e_iY_{ji}}{Det(M'')} \quad \text{with}$$

$$Y_{ji} = \begin{vmatrix} a'_{11} & \cdots & a'_{1j-1} & a'_{1j+1} & \cdots & a'_{1n} \\ a'_{21} & \cdots & a'_{2j-1} & a'_{2j+1} & \cdots & a'_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a'_{i-1\,1} & \cdots & a'_{i-1\,j-1} & a'_{i-1\,j+1} & \cdots & a'_{i-1\,n} \\ a'_{i+1\,1} & \cdots & a'_{i+1\,j-1} & a'_{i+1\,j+1} & \cdots & a'_{i+1\,n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ a'_{n1} & \cdots & a'_{nj-1} & a'_{nj+1} & \cdots & a'_{nn}. \end{vmatrix}$$

Let $H = \{1, 2, \ldots, n\}$ and $H'_e = \{i \,|\, e_ix_i < 0$ where $M''x = \vec{e}/\max\limits_{i,j}|a_{ij}|\}$. In Step 5, we will change the sign of e_i if $i \in H'_e$. For x_h, in order to satisfy $x_he_h \geq 0$, we expect that the sign of x_h is fix after changing the sign of e_i for all $i \in H'_e$. WLOG, we assume the each component of $\vec{e}$ is positive and $e_jx_j < 0$ for $j \in H'_e$. Since $\frac{1}{Det(M'')\max\limits_{i,j}|a_{ij}|}$ is a constant, we only consider $\sum_i^n(-1)^{h+i}e_iY_{hi}$. On the other hand, since each component of M'' is a symmetric distribution, Y_{hi} is also a symmetric distribution. Thus, $P(\sum_i^n(-1)^{h+i}e_iY_{hi} > 0) = P(\sum_i^n e_iY_{hi} > 0)$. Now we have probability as follows:

$$\begin{aligned} p_h &= P(\sum_{i\in H-H'_e} Y_{hi} - \sum_{i\in H'_e} Y_{hi} < 0 \,|\, \sum_{i\in H} Y_{hi} < 0) \\ &= \frac{P(\sum_{i\in H-H'_e} Y_{hi} - \sum_{i\in H'_e} Y_{hi} < 0 \wedge \sum_{i\in H} Y_{hi} < 0)}{P(\sum_{i\in H} Y_{hi} < 0)} \\ &= \frac{P(\sum_{i\in H-H'_e} Y_{hi} < -|\sum_{i\in H'_e} Y_{hi}|)}{P(\sum_{i\in H} Y_{hi} < 0)}. \end{aligned}$$

Since Y_{hi} is a symmetric distribution, $P(\sum_{i\in H} Y_{hi} < 0) = \frac{1}{2}$. Let $Q_1 = \sum_{i\in H-H'_e} Y_{hi}$ and $Q_2 = \sum_{i\in H'_e} Y_{hi}$. To facilitate the analysis, assume that Q_1 and Q_2 are independent. Then we have

$$\begin{aligned} p_h &= 2 * P(Q_1 - Q_2 < 0 \wedge Q_1 + Q_2 < 0) \\ &= 2 * P(Q_1 + |Q_2| < 0). \end{aligned}$$

$$\begin{aligned} &P(Q_1 + |Q_2| < 0) \\ &= \int_0^{-\infty}\int_{-q_1}^{q_1} f_{(Q_1,Q_2)}(q_1, q_2)dq_2\,dq_1 \\ &= \int_0^{-\infty}\int_{-q_1}^{q_1} f_{Q_1}(q_1)f_{Q_2}(q_2)dq_2\,dq_1 \\ &= \int_0^{-\infty} f_{Q_1}(q_1)\int_{-q_1}^{q_1} f_{Q_2}(q_2)dq_2\,dq_1 \\ &= \int_0^{-\infty} f_{Q_1}(q_1)(F_{Q_2}(q_1) - F_{Q_2}(-q_1))dq_1 \\ &= \int_{-\infty}^0 f_{Q_1}(q_1)(F_{Q_2}(-q_1) - F_{Q_2}(q_1))dq_1 \\ &= \int_{-\infty}^0 f_{Q_1}(q_1)(1 - 2F_{Q_2}(q_1))dq_1 \\ &= \int_{-\infty}^0 f_{Q_1}(q_1)dq_1 - 2\int_{-\infty}^0 f_{Q_1}(q_1)F_{Q_2}(q_1)dq_1 \\ &= \frac{1}{2} - 2\int_{-\infty}^0 f_{Q_1}(q_1)F_{Q_2}(q_1)dq_1 \\ &\geq \frac{1}{2} - 2\int_{-\infty}^0 f_{Q_1}(q_1)dq_1\int_{-\infty}^0 F_{Q_2}(q_1)dq_1 \\ &= \frac{1}{2} - \frac{r}{2}\sigma_{Q_2}. \end{aligned}$$

Remark:

1) Since we predict that the mean of covariance of Q_1 and Q_2 is close to 0 under large numbers of independent random variables, we assume that Q_1 and Q_2 are independent.
2) $Card(H'_e)$ is at most $\frac{n}{2}$.
3) Since Q_2 is a symmetric distribution, $F_{Q_2}(x) + F_{Q_2}(-x) = 1$.
4) r is a constant.

TABLE 1. Property comparison.

		Time	Space	Approximation Factor
[6]	Approximation	$O(n^5)$	polynomial	$(\frac{4}{3})^{\frac{n-1}{2}}$
[9]	Exact	$2^{\frac{n}{2e}+O(n)}$	polynomial	1
[19]	Approximation	$O(n^3k^k)$	polynomial	$(\frac{k}{3})^{\frac{n}{k}}$
[10]	Approximation	polynomial	polynomial	1.0109^n [25]
[15]	Exact	$2^{5.9n+O(n)}$ [16]	$2^{2.95n+O(n)}$	$1+\epsilon$
[26]	Approximation	$O(n^3 \log n)$	polynomial	$(\frac{k}{6})^{\frac{n}{k}}$
[18]	Approximation	polynomial	polynomial	$(\gamma_k(1+\varepsilon))^{\frac{n-k}{k-1}}$
[20]-1	Exact	$2^{3.199n+O(n)}$	$2^{1.325n+O(n)}$	$1+\epsilon$
[20]-2	Exact	exponential	$2^{0.41n+O(n)}$	$1+\epsilon$
[23]	Exact	$2^{0.3778n+O(n)}$	$2^{0.2883n+O(n)}$	$1+\epsilon$
[24]	Exact	$2^{0.3717n+O(n)}$	$2^{0.1887n+O(n)}$	$1+\epsilon$
Ours	Approximation	$O(n^3)$	$O(n^2)$	$(\sqrt{n}+(\frac{\sigma_1}{\sigma_n})^2 n)^2$

n is the dimension of the basis. k is the block size. ϵ is a small number.

5) Since each Y_{hi} is the determinant of submatrix M'', $Var(Y_{hi}) \approx Var(a'_{ij})^{n-1}$. Then $Var(Q_2) \approx Card(H'_e)Var(Y_{hi}) \leq \frac{n}{2}Var(a'_{ij})^{n-1}$.

Thus, we find the probability $P(\forall h \in H,\ x_h e_h \geq 0) \geq (1 - r\sigma_{Q_2})^n \geq (1 - r\sqrt{\frac{n}{2}}\sigma_{a'_{ij}}^{n-1})^n$. Let $Z = (1 - r\sqrt{\frac{n}{2}}\sigma_{a'_{ij}}^{n-1})^n$. Now, we will prove that Z is close to 1 as $n \to \infty$. Let $Z_1 = (1 - r\sigma_{a'_{ij}}^{n-1})^n$ and $Z_2 = (1 - \frac{r}{\sqrt{2}}n\sigma_{a'_{ij}}^{n-1})^n$. It is obvious that $Z_1 \geq Z \geq Z_2$.

$$\begin{aligned}
\lim_{n\to\infty} \ln Z_1 &= \lim_{n\to\infty} n \ln(1 - r\sigma_{a'_{ij}}^{n-1}) \\
&= \lim_{n\to\infty} \frac{\ln(1 - r\sigma_{a'_{ij}}^{n-1})}{\frac{1}{n}} \\
&= \lim_{n\to\infty} \frac{\frac{1}{1-r\sigma_{a'_{ij}}^{n-1}}(-r)\sigma_{a'_{ij}}^{n-1} \ln \sigma_{a'_{ij}}}{\frac{-1}{n^2}} \\
&= \lim_{n\to\infty} \frac{n^2}{\sigma_{a'_{ij}}^{-(n-1)} - r} r \ln \sigma_{a'_{ij}} \\
&= \lim_{n\to\infty} \frac{2n}{-\sigma_{a'_{ij}}^{-(n-1)} \ln \sigma_{a'_{ij}}} r \ln \sigma_{a'_{ij}} \\
&= \lim_{n\to\infty} \frac{2}{\sigma_{a'_{ij}}^{-(n-1)} (\ln \sigma_{a'_{ij}})^2} r \ln \sigma_{a'_{ij}} \\
&= 0 \\
&\Rightarrow \lim_{n\to\infty} Z_1 = 1.
\end{aligned}$$

$$\begin{aligned}
\lim_{n\to\infty} \ln Z_2 &= \lim_{n\to\infty} n \ln(1 - \frac{r}{\sqrt{2}} n\sigma_{a'_{ij}}^{n-1}) \\
&= \lim_{n\to\infty} \frac{\ln(1 - \frac{r}{\sqrt{2}} n\sigma_{a'_{ij}}^{n-1})}{\frac{1}{n}} \\
&= \lim_{n\to\infty} \frac{\frac{1}{1-\frac{r}{\sqrt{2}}n\sigma_{a'_{ij}}^{n-1}}(\frac{-r}{\sqrt{2}}\sigma_{a'_{ij}}^{n-1} + \frac{-r}{\sqrt{2}}n\sigma_{a'_{ij}}^{n-1} \ln \sigma_{a'_{ij}})}{\frac{-1}{n^2}} \\
&= \lim_{n\to\infty} \frac{n^2(1 + n \ln \sigma_{a'_{ij}})}{\sigma_{a'_{ij}}^{-(n-1)} - \frac{r}{\sqrt{2}} n} \frac{r}{\sqrt{2}} \\
&= \lim_{n\to\infty} \frac{2n + 3n^2 \ln \sigma_{a'_{ij}}}{-\sigma_{a'_{ij}}^{-(n-1)} \ln \sigma_{a'_{ij}} - \frac{r}{\sqrt{2}}} \frac{r}{\sqrt{2}} \\
&= \lim_{n\to\infty} \frac{2 + 6n \ln \sigma_{a'_{ij}}}{\sigma_{a'_{ij}}^{-(n-1)} (\ln \sigma_{a'_{ij}})^2} \frac{r}{\sqrt{2}} \\
&= \lim_{n\to\infty} \frac{6 \ln \sigma_{a'_{ij}}}{-\sigma_{a'_{ij}}^{-(n-1)} (\ln \sigma_{a'_{ij}})^3} \frac{r}{\sqrt{2}} \\
&= 0 \\
&\Rightarrow \lim_{n\to\infty} Z_2 = 1.
\end{aligned}$$

By the squeeze theorem, we find that $Z = 1$ as $n \to \infty$. Finally,the expected number of the execution of Step 5 is

$$E(Step\,5) = 1 + \frac{1}{Z} \to 1 as n \to \infty.$$

That is, with overwhelming probability, the "go to Step 2" condition in Step 5 only happens one time.

D. SPACE COMPLEXITY

In this subsection, we show that the space complexity is $O(n^2)$. The proposed algorithm needs n^2 numbers to store the basis as a matrix form $M(B)$ at first. Second, we used n numbers to store the initial value of e_i for $i = 1, 2, \ldots, n$. In Step 2 to Step 4, we stored the result of $\frac{\partial S}{\partial x_i}$ for $i = 1, 2, \ldots, n$ by $M(B)^T M(B)$, which needs n^2 numbers. Third, n numbers are required to store the vector $\vec{u} = [c_1, c_2, \ldots, c_n]/t$, where $t = \max_{1 \leq i \leq n} |c_i|$. Fourth, in Step 6, we need $2n$ numbers to store the vector $\vec{v}_i$ and the minimum vector of $\|\vec{v}\|_2 = \min_{1 \leq i \leq n} \|\vec{v}_i\|_2$, where $\|\vec{v}\|_2$ is the output of the algorithm. Totally, it is required to store $n^2 + n + n^2 + n + 2n = 2n^2 + 4n$ numbers in the proposed algorithm.

TABLE 2. The execution number of step 5 in algorithm 2.

Dimension	Execution number	Dimension	Execution number
30	2.8945	1000	4.733
60	3.278	1200	5.433
90	3.452	1400	4.267
120	3.56	1600	4.767
150	3.746	1800	4.912
180	3.937	2000	4.847
210	3.965	2200	4.800
240	4.028	2400	5.503
270	4.101	2600	5.511
300	4.186	2800	5.492
330	4.113	3000	5.504

V. COMPARISON

In this section we compare the proposed algorithm and the existing algorithms in terms of time complexity, space complexity, and approximation factor, where the comparison is shown in Table 1. The approximation factor is used to evaluate the quality of the output of the algorithms. The smaller the factor is, the better the quality is. As we mentioned in Section 1, the existing algorithms can be basically classified into two dual types. One achieves polynomial time/space complexity with exponentially large approximation factor, while the other outputs (almost) the shortest vector with exponential time/space complexity. However, from Table 1, one can observe that our work falls outside the categories of the existing works. Interestingly, our algorithm is the first one achieving polynomially large approximation factor with polynomial time/space complexity.

VI. IMPLEMENTATION

In this section, we will give the experimental data (1000 times in each dimensions) to evidence the number of executions of step 5 in algorithm 2 (Table 2). In low dimension (less than 300), the execution number is in [2.8, 4.2]. Moreover, the execution number is in [4.2, 5.6] in high dimension.

VII. CONCLUSION

In this paper, we have proposed a new SVP approximation algorithm (**algorithm 1**) which is performed in $O(n^3)$ with Hermite factor at most $(\sqrt{n} + (\frac{\sigma_1}{\sigma_n})^2\, n)$ in any case. Our main concept is based on the optimization theory. Through adding one dimension to make interference, we find a non-zero critical point $(c_1, c_2, \ldots, c_n)$ such that the length of vector $\sum_{i=1}^{n} c_i \vec{b}_i$ has the minimum value under $\sum_{i=1}^{n} c_i \approx 1$. Finally, we have found an integer k and computed $\lceil kc_i \rfloor$ for all i to recover the ratio from a rational number to an integer. To the best of our knowledge, it is the first algorithm with polynomial-bounded Hermite factor and polynomial time complexity. Moreover, we have also given an improved algorithm–**Algorithm 2**, which is analysed under some assumptions.

ACKNOWLEDGEMENTS

This work won the 2017 Thesis Award from the Chinese Cryptology and Information Security Association (CCISA).

REFERENCES

[1] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Algorithmic Number Theory*. Berlin, Germany: Springer, 1998, pp. 267–288.

[2] P. Q. Nguyen and J. Stern, "The two faces of lattices in cryptology," in *Cryptography and Lattices*. Springer, 2001, pp. 146–180.

[3] M. Ajtai, "The shortest vector problem in l2 is np-hard for randomized reductions," in *Proc. 13th Annu. ACM Symp. Theory Comput.*, 1998, pp. 10–19.

[4] D. A. Micciancio, "The shortest vector in a lattice is hard to approximate to within some constant," *SIAM J. Comput.*, vol. 30, no. 6, pp. 2008–2035, 2001.

[5] S. Garg, C. Gentry, and S. Halevi, "Candidate multilinear maps from ideal lattices," in *Advances in Cryptology—EUROCRYPT*, vol. 7881. Berlin, Germany: Springer, 2013, pp. 1–17.

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

[7] P. Q. Nguyen and B. Vallée, "The LLL algorithm," in *Information Security and Cryptography*. Berlin, Germany: Springer-Verlag, 2010.

[8] R. Kannan, "Improved algorithms for integer programming and related lattice problems," in *Proc. 15th Annu. ACM Symp. Theory Comput.*, 1983, pp. 193–206.

[9] R. Kannan, "Minkowski's convex body theorem and integer programming," *Math. Oper. Res.*, vol. 12, no. 3, pp. 415–440, 1987.

[10] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," *Math. Program.*, vol. 66, no. 1, pp. 181–199, Aug. 1994.

[11] V. Shoup. (2003). *Number Theory C++ Library (NTL) Version 5.4.1*. [Online]. Available: http://www.shoup.net

[12] N. Gama and P. Q. Nguyen, "Predicting lattice reduction," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 2008, pp. 31–51.

[13] N. Gama, P. Q. Nguyen, and O. Regev, "Lattice enumeration using extreme pruning," in *Advances in Cryptology—EUROCRYPT*. Berlin, Germany: Springer, 2010, pp. 257–278.

[14] M. Schneider and J. A. Buchmann, "Extended lattice reduction experiments using the BKZ algorithm," in *Proc. Sicherheit, Schutz Zuverlässigkeit, Beiträge 5th Jahrestagung Fachbereichs Sicherheit Gesellschaft Informatik e.V. (GI)*, Berlin, Germany, vol. 170, Oct. 2010, pp. 241–252.

[15] M. Ajtai, R. Kumar, and D. Sivakumar, "A sieve algorithm for the shortest lattice vector problem," in *Proc. 33rd Annu. ACM Symp. Theory Comput.*, 2001, pp. 601–610.

[16] P. Q. Nguyen and T. Vidick, "Sieve algorithms for the shortest vector problem are practical," *J. Math. Cryptol.*, vol. 2, no. 2, pp. 181–207, 2008.

[17] Y. Chen and P. Q. Nguyen, "BKZ 2.0: Better lattice security estimates," in *Advances in Cryptology—ASIACRYPT*. Berlin, Germany: Springer, 2011, pp. 1–20.

[18] N. Gama and P. Q. Nguyen, "Finding short lattice vectors within mordell's inequality," in *Proc. 14th Annu. ACM Symp. Theory Comput.*, 2008, pp. 207–216.

[19] C. P. Schnorr, "A hierarchy of polynomial time lattice basis reduction algorithms," *Theor. Comput. Sci.*, vol. 53, nos. 2–3, pp. 201–224, 1987.

[20] D. Micciancio and P. Voulgaris, "Faster exponential time algorithms for the shortest vector problem," in *Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms*. Philadelphia, PA, USA: SIAM, 2010, pp. 1468–1480.

[21] D. Micciancio and P. Voulgaris, "A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations," *SIAM J. Comput.*, vol. 42, no. 3, pp. 1364–1391, 2013.

[22] X. Wang, M. Liu, C. Tian, and J. Bi, "Improved Nguyen–Vidick heuristic sieve algorithm for shortest vector problem," in *Proc. 6th ACM Symp. Inf., Comput. Commun. Secur.*, 2011, pp. 1–9.

[23] F. Zhang, Y. Pan, and G. Hu, "A three-level sieve algorithm for the shortest vector problem," in *Selected Areas in Cryptography—SAC*. Berlin, Germany: Springer, 2013, pp. 29–47.

[24] G. Herold and E. Kirshanova, "Improved algorithms for the approximate k-list problem in Euclidean norm," in *Proc. IACR Int. Workshop Public Key Cryptogr.* Berlin, Germany: Springer, 2017, pp. 16–40.

[25] T. Plantard and W. Susilo, "Recursive lattice reduction," in *Security and Cryptography for Network*. Berlin, Germany: Springer, 2010, pp. 329–344.

[26] H. Koy and C. P. Schnorr, "Segment LLL-reduction of lattice bases," in *Cryptography and Lattices*. Berlin, Germany: Springer, 2001, pp. 67–80.

YU-LUN CHUANG was born in Tainan, Taiwan. He received the M.S. degree in computer science and engineering from National Sun Yat-sen University, Kaohsiung, Taiwan, in 2016. His research interests include numerical analysis, lattice-based cryptography, and number theory.

CHUN-I FAN was born in Tainan, Taiwan. He received the M.S. degree in computer science and information engineering from National Chiao Tung University, Hsinchu, Taiwan, in 1993, and the Ph.D. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1998. He was the Chief Executive Officer of the Aim for the Top University Plan Office, National Sun Yat-sen University. From 1999 to 2003, he was an Associate Researcher and the Project Leader with Telecommunication Laboratories, Chunghwa Telecom Company, Ltd., Taoyuan, Taiwan. In 2003, he joined the Faculty of the Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, where he has been a Full Professor since 2010. He is currently the Chairman of the Chinese Cryptology and Information Security Association. He is also an Outstanding Faculty in academic research with National Sun Yat-sen University. His current research interests include applied cryptology, cryptographic protocols, and information and communication security. He was a recipient of the Best Student Paper Award from the National Conference on Information Security in 1998, the Best Ph.D. Thesis Award from the Institute of Information and Computing Machinery in 1999, the Engineering Professors Award from the Chinese Institute of Engineers-Kaohsiung Chapter in 2016, and the Dragon Ph.D. Thesis Award from the Acer Foundation.

YI-FAN TSENG was born in Kaohsiung, Taiwan. He received the B.S. degree, the M.S. degree, and the Ph.D. degree in computer science and engineering from National Sun Yat-sen University, Taiwan, in 2012, 2014, and 2018, respectively. His research interests include cloud computing and security, network and communication security, information security, cryptographic protocols, and applied cryptography.