Improvements on Reductions among Different Variants of SVP and CVP

Gengran Hu(✉) and Yanbin Pan

Key Laboratory of Mathematics Mechanization, NCMIS, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, 100190 Beijing, China
hudiran10@mails.ucas.ac.cn, panyanbin@amss.ac.cn

Abstract. It is well known that Search SVP is equivalent to Optimization SVP. However, the classical reduction from Search SVP to Optimization SVP by Kannan needs polynomial times of calls to the oracle that solves Optimization SVP. In this paper, a new rank-preserving reduction is presented with only one call to the Optimization SVP oracle. The idea also leads to a similar direct reduction from Search CVP to Optimization CVP with only one call to the corresponding oracle. Both of the reductions above can be generalized for l_p norm with $p \in \mathbb{Z}^+$.

On the other hand, whether the search and optimization variants of approximate SVP are computationally equivalent is an outstanding open problem. Recently, Cheng gave a reduction from Search SVP_γ to Optimization $\text{SVP}_{\gamma'}$, where $\gamma' = \gamma^{\frac{1}{n(n-1)\log_2 \gamma n}}$ is much smaller than γ. We slightly improve the reduction by making $\gamma' = \gamma^{\frac{O(\log_2 n)}{n(n-1)\log_2 \gamma n}}$. In addition, a reduction from Search CVP_γ to Optimization $\text{CVP}_{\gamma'}$ with $\gamma' = \gamma^{\frac{1}{n\lceil n/2+\log_2 \gamma \cdot \text{dist}(t,\mathcal{L}(B))\rceil}}$ is also presented.

Keywords: Search SVP · Optimization SVP · GapSVP · Lattice · Reduction

1 Introduction

Lattices have many important applications in cryptographic constructions due to the seminal work of Ajtai [1] in 1996 which first connected the average-case complexity of lattice problems to their complexity in the worst case. Many lattice-based public-key cryptosystems have been proposed since then like the well-known Ajtai-Dwork cryptosystem [2], Regev's LWE-based cryptosystem [18], the GPV system [6] and the famous NTRU [7]. Moreover, a lot of other lattice-based cryptographic primitives have been also presented, such as the hash function [1,12,14,16], the digital signatures schemes NTRUSign [8] and the fully homomorphic encryption [5]. Usually, the securities of these schemes can be

This work was supported in part by the NNSF of China (No.11071285, No.11201458, and No.61121062), in part by 973 Project (No. 2011CB302401) and in part by the National Center for Mathematics and Interdisciplinary Sciences, CAS.

Y. Kim et al. (Eds.): WISA 2013, LNCS 8267, pp. 39–51, 2014.
DOI: 10.1007/978-3-319-05149-9_3,

based on the hardness of some lattice problems, such as SVP and CVP. SVP (the shortest vector problem) and CVP (the closest vector problem) are two of the most famous computational problems of lattice. SVP refers to finding a shortest non-zero vector in a given lattice, whereas CVP asks to find a lattice vector closest to a given target vector.

Depending on whether we have to actually find a shortest vector, find its length, or just decide if it is shorter than some given number, there are three different variants of SVP: Search SVP, Optimization SVP and Decisional SVP (See Sect. 2 for the definitions).

It has been proved that the three problems of SVP are equivalent to each other (see [15]). It is easy to check that Decisional SVP is as hard as Optimization SVP and the optimization variant can be reduced to the search variant.

In 1987, Kannan [11] showed that the search variant can be reduced to the optimization variant. The basic idea of his reduction is to recover the integer coefficients of some shortest vector under the given lattice basis by introducing small errors to the original lattice basis. However, his reduction is a bit complex. It needs to call Optimization SVP oracle polynomial times, since it could not determine the signs of the shortest vector's entries at one time. It also needs an oracle to solve Optimization SVP for some lattices with lower rank along with the same rank as the original lattice.

In this paper, we propose a new rank-preserving reduction which can solve Search SVP with only one call to the given Optimization SVP oracle. It is obvious that there is no reduction with less calls than ours. For the new reduction, we try to construct a new lattice by adding small errors to the original lattice basis such that the integer coefficients of the new lattice's shortest vector under the new basis are the same as the integer coefficients of some shortest vector in the original lattice under the original lattice basis. Moreover, by the Optimization SVP oracle, we can recover the integer coefficients.

A similar direct reduction from Search CVP to Optimization CVP with only one call also holds whereas some popular reductions [15,17] usually take Decisional CVP to bridge Search CVP and Optimization CVP. The former reduction from Decisional CVP to Optimization CVP needs one call to the Optimization CVP oracle, but it needs polynomial times of calls to reduce Search CVP to Decisional CVP.

Both of our two reductions can be generalized to the case for any l_p-norm ($p \in \mathbb{Z}^+$).

Since there exists efficient reduction from Search SVP to Optimization SVP, we want to obtain similar results for the approximate version. In fact, one open problem on the complexity of lattice problems is whether the search and optimization variants of approximate SVP are computationally equivalent. As pointed out in [13], once there exists an efficient reduction from Search SVP_γ to Optimization SVP_γ, almost all the lattice problems used in cryptography, such as uSVP (unique SVP), BDD (Bounded Distance Decoding), SIVP (the shortest independent vector problem), GapSVP (Decisional SVP), SVP, CVP, are equivalent up to polynomial factors.

It seems difficult to generalize our idea above to solve the problem, for our new reduction is sensitive to the error. However, Cheng [9] recently gave a reduction from Search SVP_γ to Optimization $\text{SVP}_{\gamma'}$ with $\gamma' = \gamma^{\frac{1}{n(n-1)\log_2 \gamma n}}$. His reduction uses the framework in [13] but shrinks the factor γ too much.

We slightly improve this result to $\gamma' = \gamma^{\frac{O(\log_2 n)}{n(n-1)(n+\log_2(\gamma n))}}$, but we have to point out that it is still far away to be useful to give some meaningful result about the complexity of some lattice problems because the approximation factor is still shrunk exponentially.

Finally, enlightened by the idea in the above reduction, we present a new reduction from Search CVP_γ to Optimization $\text{CVP}_{\gamma'}$ where $\gamma' = \gamma^{\frac{1}{n\lceil n/2+\log_2 \gamma \cdot \text{dist}(t,\mathcal{L}(B))\rceil}}$. This is the first reduction from Search CVP_γ to Optimization $\text{CVP}_{\gamma'}$ although γ' is also much smaller than γ.

The remainder of the paper is organized as follows. In Sect. 2, we give some preliminaries needed. In Sect. 3, we describe the new reduction from Search SVP to Optimization SVP. In Sect. 4, an improved reduction from Search SVP_γ to Optimization $\text{SVP}_{\gamma'}$ with $\gamma' = \gamma^{\frac{O(\log_2 n)}{n(n-1)(n+\log_2(\gamma n))}}$ is given. Our reduction from Search CVP_γ to Optimization $\text{CVP}_{\gamma'}$ can be found in Sect. 5. Finally, we give a short conclusion in Sect. 6.

2 Preliminaries

Given a matrix $B = (b_{ij}) \in \mathbb{R}^{m\times n}$ with rank n, the lattice $\mathcal{L}(B)$ spanned by the columns of B is

$$\mathcal{L}(B) = \{Bx = \sum_{i=1}^{n} x_i b_i | x_i \in \mathbb{Z}\},$$

where b_i is the i-th column of B. We call m the dimension of $\mathcal{L}(B)$ and n its rank. The determinant of $\mathcal{L}(B)$, say $\det(\mathcal{L}(B))$, is defined as $\sqrt{\det(B^T B)}$. It is easy to see when B is full-rank ($n = m$), its determinant becomes $|\det(B)|$.

A sublattice of $\mathcal{L}(B)$ is a lattice whose elements are all in $\mathcal{L}(B)$. The space spanned by B is defined as $span(B) = \{By | y \in \mathbb{R}^n\}$. The dual lattice $\mathcal{L}(D)$ of $\mathcal{L}(B)$ is defined as $\mathcal{L}(D) = \{z \in span(B) | \forall y \in \mathcal{L}(B), y^T z \in \mathbb{Z}\}$. Moreover, a basis of $\mathcal{L}(D)$ is given by $B(B^T B)^{-1}$, and $\det(\mathcal{L}(D)) = \det(\mathcal{L}(B))^{-1}$.

The first minima of lattice $\mathcal{L}(B)$ is defined as

$$\lambda_1(\mathcal{L}(B)) = \min_{0\neq v\in\mathcal{L}(B)} \|v\|,$$

where $\|v\|$ is the l_2 norm of vector v. Minkowski's first theorem tells us that for any lattice $\mathcal{L}(B)$ with rank n,

$$\lambda_1(\mathcal{L}(B)) \leq \sqrt{n} \cdot \det(\mathcal{L}(B))^{1/n}.$$

SVP usually refers to finding a vector in $\mathcal{L}(B)$ with length $\lambda_1(\mathcal{L}(B))$. It has the following three variants:

- Search SVP: Given a lattice basis $B \in \mathbb{Z}^{m\times n}$, find $v \in \mathcal{L}(B)$ such that $\|v\| = \lambda_1(\mathcal{L}(B))$.
- Optimization SVP: Given a lattice basis $B \in \mathbb{Z}^{m\times n}$, find $\lambda_1(\mathcal{L}(B))$.
- Decisional SVP: Given a lattice basis $B \in \mathbb{Z}^{m\times n}$ and a rational $r \in \mathbb{Q}$, decide whether $\lambda_1(\mathcal{L}(B)) \leq r$ or not.

Notice that we restrict the lattice basis to be integer vectors instead of arbitrary real vectors. The purpose is to make the input representable in finite bits so we can view it as a standard computation problem.

Since SVP is proved to be NP-hard under randomized reductions (see [3]), its approximate versions are attracting more attention. With approximate factor $\gamma \geq 1$, the corresponding variants of approximate SVP are:

- Search SVP_γ: Given a lattice basis $B \in \mathbb{Z}^{m\times n}$, find $v \in \mathcal{L}(B)$ such that $\|v\| \leq \gamma \cdot \lambda_1(\mathcal{L}(B))$.
- Optimization SVP_γ: Given a lattice basis $B \in \mathbb{Z}^{m\times n}$, find d such that $d \leq \lambda_1(\mathcal{L}(B)) \leq \gamma \cdot d$.
- Decisional SVP_γ: Given a lattice basis $B \in \mathbb{Z}^{m\times n}$ and a rational $r \in \mathbb{Q}$, decide $\lambda_1(\mathcal{L}(B)) \leq r$ or $\lambda_1(\mathcal{L}(B)) > \gamma \cdot r$.

For the Search SVP_γ, the famous LLL algorithm [10] tells us a basis $b_1, b_2, \ldots, b_n$ can be found in polynomial time such that

$$\|b_1\| \leq 2^{(n-1)/2}\lambda_1(\mathcal{L}(B)).$$

The Decisional SVP_γ is usually denoted by GapSVP_γ. This is a promise problem defined by two disjoint sets: the YES instances ($\lambda_1(\mathcal{L}(B)) \leq r$) and the NO instances ($\lambda_1(\mathcal{L}(B)) > \gamma \cdot r$). We have to decide which set the input lattice is taken from.

Given any $t \in \mathbb{R}^m$, the distance of t to $\mathcal{L}(B)$ is defined as

$$\text{dist}(t, \mathcal{L}(B)) = \min_{v\in\mathcal{L}(B)} \|t - v\|.$$

In the same way, for approximate factor $\gamma \geq 1$, CVP_γ also has three variants:

- Search CVP_γ: Given a lattice basis $B \in \mathbb{Z}^{m\times n}$ and a target $t \in \mathbb{Q}^m$, find $v \in \mathcal{L}(B)$ such that $\|t - v\| \leq \gamma \cdot \text{dist}(t, \mathcal{L}(B))$.
- Optimization CVP_γ: Given a lattice basis $B \in \mathbb{Z}^{m\times n}$ and a target $t \in \mathbb{Q}^m$, find d such that $d \leq \text{dist}(t, \mathcal{L}(B)) \leq \gamma \cdot d$.
- GapCVP_γ: Given a lattice basis $B \in \mathbb{Z}^{m\times n}$, a target $t \in \mathbb{Q}^m$ and a rational $r \in \mathbb{Q}$. In YES instances, $\text{dist}(t, \mathcal{L}(B)) \leq r$. In NO instances, $\text{dist}(t, \mathcal{L}(B)) > \gamma \cdot r$.

For the Search CVP_γ, Babai's Nearest Plane Algorithm [4] says a lattice vector v can be found in polynomial time such that

$$\|t - v\| \leq 2^{(n-1)/2} \cdot \text{dist}(t, \mathcal{L}(B)).$$

Notice that when $\gamma = 1$, these problems will become exact variants of CVP.

3 The New Reduction from Search SVP to Optimization SVP

For simplicity, we just give the new reduction for the full rank lattice, i.e., $n = m$, as in [11], with l_2 norm. It is easy to generalize the new reduction for the lattices with rank $n < m$ and l_p norm ($p \in \mathbb{Z}^+$).

3.1 Some Notations

Given a lattice basis $B = (b_{ij}) \in \mathbb{Z}^{n\times m}$, let $M(B) = \max |b_{ij}|$. For lattice $\mathcal{L}(B)$, we define its SVP solution set S_B as:

$$S_B = \{x \in \mathbb{Z}^n | \|Bx\| = \lambda_1(\mathcal{L}(B))\}.$$

S_B is nonempty and might contain more than one element.

We denote by $poly(n)$ the polynomial in n. More generally, the polynomial in the variables $n_1, n_2, \ldots, n_p$ is denoted by $poly(n_1, n_2, \ldots, n_p)$.

3.2 Some Lemmas and Corollaries

We need some lemmas and corollaries to prove our main theorem.

Lemma 1. *Given a fixed positive integer p, then for every positive integer $n \geq p$, there exist n positive integers $a_1 < a_2 < \cdots < a_n$ s.t. all the $a_{i_1} + \cdots + a_{i_p} (i_1 \leq \ldots \leq i_p)$'s are distinct (up to a permutation) and a_n is bounded by $poly(n)$.*

Proof. We can take

$$a_i = \sum_{k=0}^{p} (p(n+1)^p)^{(p-k)} i^k,$$

for $i = 1, 2, \ldots, n$. Suppose

$$a_{i_1} + a_{i_2} + \cdots + a_{i_p} = a_{j_1} + a_{j_2} + \cdots + a_{j_p},$$

for some $i_1, \ldots, i_p, j_1, \ldots, j_p$.

Let $\sigma_k(i) = \sum_{t=1}^{p} (i_t)^k$ and $\sigma_k(j) = \sum_{t=1}^{p} (j_t)^k$, then the former equality turns to

$$\sum_{k=0}^{p} (p(n+1)^p)^{(p-k)} \sigma_k(i) = \sum_{k=0}^{p} (p(n+1)^p)^{(p-k)} \sigma_k(j).$$

Notice that $\sigma_k(i), \sigma_k(j) < p(n+1)^p$ for $k = 1, 2, \ldots, p$, then by taking both sides modulo $p(n+1)^p$, we get

$$\sigma_p(i) = \sigma_p(j),$$

which leads to

$$\sum_{k=0}^{p-1} (p(n+1)^p)^{(p-k-1)} \sigma_k(i) = \sum_{k=0}^{p-1} (p(n+1)^p)^{(p-k-1)} \sigma_k(j).$$

Again taking both sides modulo $p(n+1)^p$, we get

$$\sigma_{p-1}(i) = \sigma_{p-1}(j).$$

Similarly, we repeat this procedure to obtain

$$\sigma_k(i) = \sigma_k(j),$$

for $k = 1, 2, \ldots, p$. Thus by the property of the symmetric polynomials, we know that $i_1, \ldots, i_p$ and $j_1, \ldots, j_p$ are both exactly all the roots of a same polynomial, which implies $i_1, \ldots, i_p$ and $j_1, \ldots, j_p$ are equal up to a permutation. Hence all the $a_{i_1} + \cdots + a_{i_p} (i_1 \leq \cdots \leq i_p)$'s are distinct. Since p is a fixed positive integer, then by our choice, a_n is bounded by $poly(n)$.

Corollary 1. *For every positive integer $n > 1$, there exist n positive integers $a_1 < a_2 < \cdots < a_n$ s.t. all the $a_i + a_j (i \leq j)$'s are distinct and a_n is bounded by $poly(n)$.*

Lemma 2. *Given a positive odd integer $q > 2$, and any positive integer n, which satisfies $n = \sum_{i=0}^{k} n_i q^i$ where $|n_i| \leq \lfloor q/2 \rfloor$, then we can recover the coefficients n_i's in $\lceil \log_q n \rceil$ steps.*

Proof. We can recover n_0 by computing $a \equiv n \bmod q$ and choose a in the interval from $-\lfloor q/2 \rfloor$ to $\lfloor q/2 \rfloor$. After obtaining n_0, we get another integer $(n - n_0 * q^0)/q$. Recursively in $\lceil \log_q n \rceil$ steps, we can recover all the coefficients.

Lemma 3. *For bivariate polynomial $f(x, y) = xy$, given any lattice basis matrix $B \in \mathbb{Z}^{n \times n}$, $\lambda_1(\mathcal{L}(B))$ has an upper bound $f(M, n)$, where $M = M(B)$. What's more, for every $x \in S_B$, $|x_i|$ $(i = 1, \ldots, n)$ has an upper bound $f(M^n, n^n)$.*

Proof. The length of any column of B is an upper bound of $\lambda_1(L(B))$, so $\lambda_1(L(B)) \leq n^{1/2} M \leq nM$.

For $x \in S_B$, we let $y = Bx$, then $\|y\| = \lambda_1(L(B)) \leq \sqrt{n} M$. By Cramer's rule, we know that

$$x_i = \frac{\det(B^{(i)})}{\det(B)},$$

where $B^{(i)}$ is formed by replacing the ith column of B by y. By Hadamard's inequality, $|\det(B^{(i)})| \leq n^{n/2} M^n \leq n^n M^n$. We know $|\det(B)| \geq 1$ since $\det(B)$ is a non-zero integer. Hence $|x_i| \leq n^n M^n$.

3.3 The Main Theorem

Theorem 1. *Assume there exists an oracle $\mathcal{O}$ that can solve Optimization SVP for any lattice $L(B')$ with basis $B' \in \mathbb{Z}^{n \times n}$, then there is an algorithm that can solve Search SVP for any lattice $L(B)$ with basis $B \in \mathbb{Z}^{n \times n}$ with only one call to $\mathcal{O}$ in $poly(\log_2 M, n, \log_2 n)$ time, where $M = M(B)$.*

Proof. The main steps of the reduction are as below:
(1) Constructing a new lattice $L = L(B_\epsilon)$.

We construct B_ϵ from the original lattice basis B:

$$B_\epsilon = \epsilon_{n+1} B + \begin{pmatrix} \epsilon_1 & \epsilon_2 & \dots & \epsilon_n \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix},$$

where the ϵ_i will be determined as below.

For any $x \in \mathbb{Z}^n$, we define

$$c(x) = \sum_{i=1}^{n} b_{1i} x_i,$$

for $x \in S_B$. By Lemma 3, $|x_i|$ has an upper bound $f(M^n, n^n)$. Let $M_1 = 2f((M+1)^n, n^n)$. In addition, $\|Bx\| = \lambda_1(L(B))$ is bounded by $f(M, n)$. Let $M_2 = f(M+1, n)$. since $|c(x)| \le \|Bx\|$, $|c(x)|$ is also bounded by M_2 . We let

$$R = 2 * \max\{M_2^2, 2M_1M_2, 2M_1^2\} + 1.$$

By Corollary 1, we can choose $n+1$ positive integers $a_1 < a_2 < \ldots < a_{n+1}$, such that all the $a_i + a_j (i \le j)$'s are distinct where a_{n+1} is bounded by $poly(n)$. Let

$$\epsilon_i = R^{a_i}.$$

We claim that

$$S_{B_\epsilon} \subseteq S_B.$$

Since $S_{B_\epsilon} = S_{\frac{1}{\epsilon_{n+1}} B_\epsilon}$ by scaling, it is enough to prove $S_{\frac{1}{\epsilon_{n+1}} B_\epsilon} \subseteq S_B$.

We first show that $|\det(\frac{1}{\epsilon_{n+1}} B_\epsilon)| \ge \frac{1}{2}$. Notice that

$$\det(\frac{1}{\epsilon_{n+1}} B_\epsilon) = \det(B) + \sum_{i=1}^{n} \alpha_i \frac{\epsilon_i}{\epsilon_{n+1}},$$

where α_i is the cofactor of b_{1i} in B. Since $\frac{\epsilon_i}{\epsilon_{n+1}} \le \frac{1}{R^2}$ and $|\alpha_i| \le M^{n-1}(n-1)^{n-1}$ by Hadamard's inequality, $|\sum_{i=1}^{n} \alpha_i \frac{\epsilon_i}{\epsilon_{n+1}}| \le \frac{1}{R^2} M^{n-1} n^n < \frac{1}{2}$. Notice that $\det(B)$ is a non-zero integer, we get $|\det(\frac{1}{\epsilon_{n+1}} B_\epsilon)| \ge \frac{1}{2}$.

For any $x \in S_{\frac{1}{\epsilon_{n+1}} B_\epsilon}$, by the proof of Lemma 3 and the fact that $|\det(\frac{1}{\epsilon_{n+1}} B_\epsilon)| \ge \frac{1}{2}$, we know that $|x_i| \le M_1$, $|c(x)| \le M_2$. By the choice of R, we have $x_i^2, 2c(x)x_i, 2x_ix_j$ are in the interval $[-\lfloor R/2 \rfloor, \lfloor R/2 \rfloor]$.

Next, we prove $S_{\frac{1}{\epsilon_{n+1}} B_\epsilon} \subseteq S_B$. Suppose there exists $x \in S_{\frac{1}{\epsilon_{n+1}} B_\epsilon}$ but $x \notin S_B$, then

$$\|Bx\|^2 \ge \lambda_1(L(B))^2 + 1.$$

Taking $y \in S_B$, we get $\frac{1}{\epsilon_{n+1}} B_\epsilon y \in L(\frac{1}{\epsilon_{n+1}} B_\epsilon)$. Noticing $\epsilon_{n+1} > R^2 \epsilon_n$, $\frac{\epsilon_i \epsilon_j}{\epsilon_{n+1}^2} (i \leq j)$'s are different powers of R (by our choice of ϵ_i and Corollary 1), and y_i^2, $2c(y)y_i, 2y_i y_j$ are in the interval $[-\lfloor R/2 \rfloor, \lfloor R/2 \rfloor]$ by the choice of R, we have

$$\begin{aligned}
\|\tfrac{1}{\epsilon_{n+1}} B_\epsilon y\|^2 &= \|By\|^2 + \textstyle\sum_{i=1}^n y_i^2 (\frac{\epsilon_i}{\epsilon_{n+1}})^2 + \sum_{i=1}^n 2c(y) y_i \frac{\epsilon_i}{\epsilon_{n+1}} + \sum_{i<j} 2 y_i y_j \frac{\epsilon_i \epsilon_j}{\epsilon_{n+1}^2} \\
&< \lambda_1(L(B))^2 + (\lfloor R/2 \rfloor + 1) \frac{\epsilon_n}{\epsilon_{n+1}} \\
&\leq \|Bx\|^2 - (1 - (\lfloor R/2 \rfloor + 1) \tfrac{\epsilon_n}{\epsilon_{n+1}}) \\
&< \|Bx\|^2 - (\lfloor R/2 \rfloor + 1) \tfrac{\epsilon_n}{\epsilon_{n+1}} \\
&\leq \|Bx\|^2 + \textstyle\sum_{i=1}^n x_i^2 (\frac{\epsilon_i}{\epsilon_{n+1}})^2 + \sum_{i=1}^n 2c(x) x_i \frac{\epsilon_i}{\epsilon_{n+1}} + \sum_{i<j} 2 x_i x_j \frac{\epsilon_i \epsilon_j}{\epsilon_{n+1}^2} \\
&= \lambda_1(L(\tfrac{1}{\epsilon_{n+1}} B_\epsilon))^2,
\end{aligned}$$

which is a contradiction. Hence $S_{B_\epsilon} \subseteq S_B$.

(2) Querying the oracle $\mathcal{O}$ with B_ϵ once, we get $\lambda_1(\mathcal{L}(B_\epsilon))$.
So there exists $x = (x_1, \ldots, x_n)^{\mathrm{T}} \in S_{B_\epsilon} \subseteq S_B$, such that

$$\|Bx\|^2 \epsilon_{n+1}^2 + \sum_{i=1}^n x_i^2 \epsilon_i^2 + \sum_{i=1}^n 2c(x) x_i \epsilon_{n+1} \epsilon_i + \sum_{i<j} 2 x_i x_j \epsilon_i \epsilon_j = \lambda_1(\mathcal{L}(B_\epsilon))^2.$$

(3) Recovering all the x_i's and output Bx.

Since $x \in S_B$, every coefficient $\|Bx\|^2, x_i^2, 2c(x)x_i, 2x_i x_j$ is in the interval $[-\lfloor R/2 \rfloor, \lfloor R/2 \rfloor]$ and $\epsilon_i \epsilon_j$ $(i \leq j)$'s are different powers of R. Hence, $\log_2(\lambda_1(\mathcal{L}(B_\epsilon)))$ is bounded by $poly(\log_2 M, n, \log_2 n)$. Furthermore, by Lemma 2, we can recover all the coefficients in $poly(\log_2 M, n, \log_2 n)$ time. Especially, we can recover all x_i^2 and $x_i x_j (i \neq j)$. Let $k = \min\{i | x_i \neq 0\}$. We fix $x_k = \sqrt{x_k^2} > 0$, and can recover all the remaining $x_j = sign(x_k x_j)\sqrt{x_j^2}$ according to x_j^2 and $x_k x_j (k \neq j)$.

It is easy to check that the time and space complexity of every step is bounded by $poly(\log_2 M, n, \log_2 n)$.

Remark 1. Notice that the norm in our main theorem is the most common l_2-norm. In fact, our result can be easily generalized to the case for l_p-norm ($p \in \mathbb{Z}^+$) by Lemma 1.

Remark 2. For any Search CVP instance (B, t), given an oracle which can solve the Optimization CVP, we can call the oracle with $(B_\epsilon, \epsilon_{n+1} t)$ only once to solve the Search CVP similarly.

4 Improved Reduction from Search SVP_γ to Optimization $\mathrm{SVP}_{\gamma'}$

In [9], Cheng gave a reduction from Search SVP_γ to Optimization $\mathrm{SVP}_{\gamma'}$ where $\gamma' = \gamma^{\frac{1}{n(n-1)(n+\log_2(\gamma n))}}$. We slightly improve the result to $\gamma' = \gamma^{\frac{O(\log_2 n)}{n(n-1)(n+\log_2(\gamma n))}}$. As in [9] (Theorem 1), the main idea is to obtain lower rank sublattice of $\mathcal{L}(B)$

which still contains an approximate shortest lattice vector of $\mathcal{L}(B)$. After lowering the rank for several $(n-1)$ times, we finally obtain a rank-one sublattice of $\mathcal{L}(B)$ containing a short vector. Since it is easy to find the shortest vector in a lattice with rank one (its basis), we can find an approximate shortest lattice vector of $\mathcal{L}(B)$. Below we will give a self-contained proof.

Theorem 2. *For any $\gamma \geq 1$, Search SVP$_\gamma$ can be polynomially reduced to Optimization SVP$_{\gamma'}$ where $\gamma' = \gamma^{\frac{O(\log_2 n)}{n(n-1)(n+\log_2(\gamma n))}}$.*

Proof. Given the input instance $B = (b_1, b_2, \ldots, b_n)$, we intend to find $v \in \mathcal{L}(B)$ such that $\|v\| \leq \gamma \cdot \lambda_1(\mathcal{L}(B))$.

First, for $k = O(\log_2 n)$, we consider $2^{k+1} - 1$ sublattices of $\mathcal{L}(B)$ where their respective bases are $B_{i,j} = (2^i b_1 + j b_2, 2^{k-i} b_2, b_3, \ldots, b_m)(i = 1, 2 \ldots, k, 0 \leq j < 2^{k-i})$. Notice for every $B_{i,j}$, $\det(\mathcal{L}(B_{i,j})) = 2^k \det(\mathcal{L}(B))$. We claim that

$$\mathcal{L}(B) = \bigcup_{i,j} \mathcal{L}(B_{i,j}).$$

For any $w = x_1 b_1 + x_2 b_2 + \cdots + x_n b_n$ in $\mathcal{L}(B)$, $x_1 \in \mathbb{Z}$ can be written as $x_1 = 2^r s$, where s is odd. If $r \geq k$, then $w \in \mathcal{L}(B_{k,0}) = \mathcal{L}(2^k b_1, b_2, \ldots b_m)$. Otherwise, we assume $r < k$. There exist integers p, q such that $sp + 2^{k-r} q = 1$ since $(s, 2^{k-r}) = 1$, which implies $spx_2 + 2^{k-r} q x_2 = x_2$. We take $i = r, j = px_2 \mod 2^{k-r}$, then $s(2^i b_1 + j b_2) + (qx_2 + s\frac{px_2 - j}{2^{k-r}})2^{k-r} b_2 = x_1 b_1 + x_2 b_2$. So $w \in \mathcal{L}(B_{i,j})$, thus $\mathcal{L}(B) \subseteq \bigcup_{i,j} \mathcal{L}(B_{i,j})$. On the other hand, since all the $\mathcal{L}(B_{i,j})$'s are sublattices of $\mathcal{L}(B)$, our claim follows.

Secondly, we want to find a good sublattice $\mathcal{L}(B_{i,j})$ of the original lattice $\mathcal{L}(B)$ still containing a short lattice vector. We query the Optimization SVP$_{\gamma'}$ oracle for 2^{k+1} (which is $ploy(n)$ by the choice of k) times with these $B_{i,j}$ and get the output intervals $I_{i,j} = [r_{i,j}, \gamma' \cdot r_{i,j})$ containing $\lambda_1(\mathcal{L}(B_{i,j}))$ respectively. Specially, we can invoke the SVP$_{\gamma'}$ oracle for B to obtain an interval $I = [r, \gamma' \cdot r)$ containing $\lambda_1(\mathcal{L}(B))$. By our claim, a shortest lattice vector in $\mathcal{L}(B)$ must lie in some $\mathcal{L}(B_{i,j})$ which means I must intersect some $I_{i,j}$'s. We take I_{i_0,j_0} that has the smallest left endpoint from these $I_{i,j}$'s. We claim

$$\lambda_1(\mathcal{L}(B_{i_0,j_0})) \leq \gamma' \cdot \lambda_1(\mathcal{L}(B)).$$

Let $I_{i',j'}$ be the interval where a shortest lattice vector in $\mathcal{L}(B)$ lies. Then by the choice of $I_{i,j}$, $\lambda_1(\mathcal{L}(B_{i_0,j_0})) \leq \gamma' \cdot r_{i_0,j_0} \leq \gamma' \cdot r_{i',j'} \leq \gamma' \cdot \lambda_1(\mathcal{L}(B))$.

Thirdly, we repeat this procedure by replacing the input B with the B_{i_0,j_0}. After $t = \frac{n(n+\log_2(\gamma n))}{O(\log_2 n)}$ steps, we obtain a sublattice $\mathcal{L}(B')$ of $\mathcal{L}(B)$ such that

$$\lambda_1(\mathcal{L}(B')) \leq (\gamma')^t \cdot \lambda_1(\mathcal{L}(B)),$$

where $\det(\mathcal{L}(B')) = 2^{kt} \det(\mathcal{L}(B)) \geq 2^{n(n+\log_2 \gamma n)} \det(\mathcal{L}(B))$.

According to Minkowski's bound, we have $\lambda_1(\mathcal{L}(B)) \leq \sqrt{n}\det(\mathcal{L}(B))^{1/n}$. Denote by $u^{'}$ a shortest lattice vector in $\mathcal{L}(B^{'})$, then

$$\|u^{'}\| \leq (\gamma^{'})^t\sqrt{n}\det(\mathcal{L}(B))^{1/n}.$$

Assume $\mathcal{L}(D)$ is the dual lattice of $\mathcal{L}(B^{'})$. Then $\det(\mathcal{L}(D)) \leq 1/(2^{n(n+\log_2\gamma n)}\det(\mathcal{L}(B)))$. By the LLL Algorithm [10], we can find a vector $u \in \mathcal{L}(D)$ such that

$$\begin{aligned}\|u\| < 2^n\sqrt{n}\det(\mathcal{L}(D))^{1/n} &\leq \sqrt{n}2^n/(2^{(n+\log_2\gamma n)}\det(\mathcal{L}(B))^{1/n}) \\ &= 1/(\gamma\sqrt{n}\det(\mathcal{L}(B))^{1/n}).\end{aligned}$$

By Cauchy–Schwarz inequality, we have

$$|\langle u^{'}, u\rangle| \leq \|u^{'}\| \cdot \|u\| < (\gamma^{'})^t/\gamma \leq 1.$$

Since $u^{'} \in \mathcal{L}(B^{'}), u \in \mathcal{L}(D)$, $\langle u^{'}, u\rangle$ is an integer, which means $\langle u^{'}, u\rangle = 0$. Hence $u^{'}$ lies in the sublattice of $\mathcal{L}(B^{'})$ orthogonal to u. Denote this sublattice by $\mathcal{L}(B_1)$ and notice that its rank is $n-1$. Therefore, we can efficiently find a lower rank sublattice $\mathcal{L}(B_1) \subseteq \mathcal{L}(B)$ such that $\lambda_1(\mathcal{L}(B_1)) \leq (\gamma^{'})^t\lambda_1(\mathcal{L}(B))$.

Finally, after repeating $n-1$ times of the above procedures, we obtain a sublattice $\mathcal{L}(B_{n-1})$ of rank one with

$$\lambda_1(\mathcal{L}(B_{n-1})) \leq (\gamma^{'})^{(n-1)t}\lambda_1(\mathcal{L}(B)).$$

Since a lattice basis is already the shortest lattice vector in any 1-rank lattice and $(\gamma^{'})^{(n-1)t} = \gamma$, we can find a lattice vector in $\mathcal{L}(B)$ of length $\lambda_1(\mathcal{L}(B_{n-1})) \leq \gamma\lambda_1(\mathcal{L}(B))$. This completes our proof.

Remark 3. The above reduction is for l_2-norm. Using the fact that for any $v \in \mathbb{R}^n$ and any $p \geq 1$, $\|v\|_2/\sqrt{n} \leq \|v\|_p \leq n^{1/p}\|v\|_2$, we can generalize our reduction to the case for any l_p-norm, where $\gamma^{'} = \gamma^{\frac{O(\log_2 n)}{n(n-1)(n+\log_2(\gamma n^{3/2+1/p}))}}$.

5 Our Reduction from Search CVP_γ to Optimization $\mathrm{CVP}_{\gamma'}$

In this section, we present our reduction from Search CVP_γ to Optimization $\mathrm{CVP}_{\gamma'}$ where $\gamma^{'} = \gamma^{\frac{1}{n\lceil n/2+\log_2\gamma\cdot\mathrm{dist}(t,\mathcal{L}(B))\rceil}}$. We have to point out that the relationship between two approximate factors γ and $\gamma^{'}$ is still waiting to be improved.

Theorem 3. *For any $\gamma^{'} \geq 1$ and $n \geq 4$, Search CVP_γ can be solved in polynomial time given an oracle solving Optimization $CVP_{\gamma'}$ where $\gamma^{'} = \gamma^{\frac{1}{n\lceil n/2+\log_2\gamma\cdot dist(t,\mathcal{L}(B))\rceil}}$.*

Proof. Given the input lattice basis $B = (b_1, b_2, \ldots, b_n) \in \mathbb{Z}^{m\times n}$ and a target $t \in \mathbb{Q}^n$, we call the Optimization $\text{CVP}_{\gamma'}$ oracle to obtain an interval $[r, \gamma' \cdot r)$ containing $\text{dist}(t, \mathcal{L}(B)) \triangleq d$. Our goal is to find a $v \in \mathcal{L}(B)$ s.t. $\|v - t\| \leq \gamma \cdot \text{dist}(t, \mathcal{L}(B))$.

Firstly, a sequence of instance $(B_i, t_i)(i = 0, 1, \ldots, k$, where $k = \lceil n/2 + \log_2 \gamma \cdot d\rceil)$ is constructed in the following way.

Let $B_0 = B$, $t_0 = t$ and $B_i = (2^i b_1, b_2, \ldots, b_n)$. We want to construct t_{i+1} from t_i, B_i and B_{i+1}. Given (B_i, t_i), we call the Optimization $\text{CVP}_{\gamma'}$ oracle on the three inputs (B_i, t_i), (B_{i+1}, t_i) and $(B_{i+1}, t_i - 2^i b_1)$ to get three interval $I_0 = [r_0, \gamma \cdot r_0)$, $I_1 = [r_1, \gamma \cdot r_1)$ and $I_2 = [r_2, \gamma \cdot r_2)$ containing $\text{dist}(t_i, \mathcal{L}(B_i)) \triangleq d_0$, $\text{dist}(t_i, \mathcal{L}(B_{i+1})) \triangleq d_1$ and $\text{dist}(t_i - 2^i b_1, \mathcal{L}(B_{i+1})) \triangleq d_2$ respectively. Notice that

$$\mathcal{L}(B_i) = \mathcal{L}(B_{i+1}) \cup (\mathcal{L}(B_{i+1}) + 2^i b_1),$$

meaning $d_1 = d_0$ or $d_2 = d_0$. So I_0 must intersect at least one of I_1 and I_2. Similar to that in the proof of Theorem 2, let I_{i_0} be the interval having the smallest left endpoint in these I_i's that intersect I_0. Then we set t_{i+1}:

$$t_{i+1} = \begin{cases} t_i & (i_0 = 1) \\ t_i - 2^i b_1 & (i_0 = 2). \end{cases} \tag{1}$$

We can also prove that

$$\text{dist}(t_{i+1}, \mathcal{L}(B_{i+1})) \leq \gamma' \cdot \text{dist}(t_i, \mathcal{L}(B_i)).$$

Hence we can find $(B_k = (2^k b_1, b_2, \ldots b_n), t_k)$ such that $\text{dist}(t_k, \mathcal{L}(B_k)) \leq (\gamma')^k \cdot \text{dist}(t, \mathcal{L}(B))$.

Secondly, by repeating this procedure for other lattice basis vector $b_2, \ldots, b_n$, we obtain $(B_{nk} = (2^k b_1, 2^k b_2, \ldots, 2^k b_n), t_{nk})$ s.t.

$$\text{dist}(t_{nk}, \mathcal{L}(B_{nk})) \leq (\gamma')^{nk} \cdot \text{dist}(t, \mathcal{L}(B)) = \gamma \cdot \text{dist}(t, \mathcal{L}(B)) = \gamma \cdot d,$$

where t_{nk} is of the form $t + u$ ($u \in \mathcal{L}(B)$ is known). We denote $\text{dist}(t_{nk}, \mathcal{L}(B_{nk}))$ by d_{nk}.

Notice that the new lattice $\mathcal{L}(B_{nk}) = 2^k \mathcal{L}(B)$ is sparse enough with $\lambda_1(\mathcal{L}(B_{nk})) = 2^k \lambda_1(\mathcal{L}(B)) \geq 2^k \cdot 1 = 2^k$. For the choice of k,

$$\lambda_1(\mathcal{L}(B_{nk})) \geq 2^k \geq 2^{n/2} \gamma d \geq 2^{n/2} d_{nk}.$$

By Babai's Nearest Plane Algorithm [4] on input (B_{nk}, t_{nk}), we can find a lattice vector $v \in \mathcal{L}(B_{nk})$ s.t. $\|v - t_{nk}\| \leq 2^{\frac{n-1}{2}} \cdot d_{nk}$. We claim that v is the lattice vector closest to t_{nk} in $\mathcal{L}(B_{nk})$. Let v' be the lattice vector closest to t_{nk} in $\mathcal{L}(B_{nk})$, then $\|v' - t_{nk}\| = d_{nk}$. We will show $v = v'$. For any $w \neq v' \in \mathcal{L}(B_{nk})$, we have

$$\|w - t_{nk}\| \geq \|w - v'\| - \|v' - t_{nk}\| \geq \lambda_1(\mathcal{L}(B_{nk})) - d_{nk} \geq 2^{n/2} d_{nk} - d_{nk} > 2^{\frac{n-1}{2}} d_{nk},$$

where the last inequality comes from $n \geq 4$. Together with $\|v - t_{nk}\| \leq 2^{\frac{n-1}{2}} \cdot d_{nk}$, we have v is actually the lattice vector closest to t_{nk} in $\mathcal{L}(B_{nk})$. Thus we have

$$\|v - t_{nk}\| = \mathrm{dist}(t_{nk}, \mathcal{L}(B_{nk})) \leq \gamma \cdot \mathrm{dist}(t, \mathcal{L}(B)).$$

Finally, as v is in $\mathcal{L}(B)$, we subtract the known u from v to get our Search CVP_γ solution $v - u$.

Remark 4. The above reduction can also be generalized to the case for any l_p-norm, where $\gamma' = \gamma^{\frac{1}{n\lceil n/2 + \log_2 \gamma n^{1/p} \cdot \mathrm{dist}(t, \mathcal{L}(B))\rceil}}$.

6 Conclusions

In this paper, we give a new reduction from Search SVP to Optimization SVP with only one call, which is the least, to the Optimization SVP oracle. A similar result for CVP also holds. When it goes to approximate version, inspired by the idea in [9], we get an improved result on reduction from Search SVP_γ to Optimization $\mathrm{SVP}_{\gamma'}$ and a reduction from Search CVP_γ to Optimization $\mathrm{CVP}_{\gamma'}$.

Acknowledgements. We thank the anonymous referees for their suggestions on how to improve the presentation of this paper.

References

1. Ajtai, M.: Generating hard instances of lattice problems. In: Annual Symposium on the theory of Computing (STOC), pp. 99–108. ACM Press, New York (1996)
2. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: Annual Symposium on the theory of Computing (STOC) (1997) (An improved version is described in ECCC 2007)
3. Ajtai, M.: The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, pp. 266–275 (1998)
4. Babai, L.: On Lovasz' lattice reduction and the nearest lattice point problem. Combinatorica **6**(1), 1–13 (1986)
5. Gentry, C.: Fully homomorphic encryption using ideal lattices. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing (STOC09), New York, pp. 169–178 (2009)
6. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Annual Symposium on the theory of Computing (STOC), pp. 197–206 (2008)
7. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: a ring-based public key cryptosystem. In: Buhler, J.P. (ed.) ANTS 1998. LNCS, vol. 1423, pp. 267–288. Springer, Heidelberg (1998)
8. Hoffstein, J., Howgrave-Graham, N., Pipher, J., Silverman, J.H., Whyte, W.: NTRUSIGN: Digital signatures using the NTRU lattice. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 122–140. Springer, Heidelberg (2003)

9. Cheng K.: Some complexity results and bit unpredictable for short vector problem. https://eprint.iacr.org/2013/052.pdf
10. Lenstra, A.K., Lenstra Jr, H.W., Lovasz, L.: Factoring polynomials with rational coefficients. Math. Ann. **261**, 513–534 (1982)
11. Kannan, R.: Minkowski's convex body theorem and integer programming. Math. Oper. Res. **12**(3), 415–440 (1987)
12. Lyubashevsky, V., Micciancio, D.: Generalized compact knapsacks are collision resistant. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 144–155. Springer, Heidelberg (2006)
13. Lyubashevsky, V., Micciancio, D.: On bounded distance decoding, unique shortest vectors, and the minimum distance problem. In: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 577–594. Springer, Heidelberg (2009)
14. Lyubashevsky, V., Micciancio, D., Peikert, C., Rosen, A.: Swifft: A modest proposal for fft hashing. Fast Software Encryption (FSE) 2008. LNCS, vol. 5086, pp. 54–72. Springer, Heidelberg (2008)
15. Micciancio, D., Goldwasser, S.: Complexity of Lattice Problems: A Cryptography Perspective. Kluwer Academic Publishers, Boston (2002)
16. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 145–156. Springer, Heidelberg (2006)
17. Regev, O.: Lattices in computer science. Lecture notes of a course given in Tel Aviv University (2004)
18. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Annual Symposium on Theory of Computing (STOC) (2005)