Unique Shortest Vector Problem for max norm is NP-hard

Than Quang Khoat and Nguyen Hong Tan

Faculty of Information Technology, Thai Nguyen University,
Thai Nguyen city, Vietnam
{tqkhoat, tannh}@ictu.edu.vn

Abstract. The *unique Shortest vector problem* (uSVP) in lattice theory plays a crucial role in many public-key cryptosystems. The security of those cryptosystems bases on the hardness of uSVP. However, so far there is no proof for the proper hardness of uSVP even in its exact version. In this paper, we show that the exact version of uSVP for ℓ_∞ norm is NP-hard. Furthermore, many other lattice problems including *unique Subspace avoiding problem*, *unique Closest vector problem* and *unique Generalized closest vector problem*, for any ℓ_p norm, are also shown to be NP-hard.

Keyword. Unique shortest vector problem, unique closest vector problem, unique subspace avoiding problem, Lattice, NP-hard, Lattice-based cryptosystems.

1 Introduction

Lattices are useful tools for both mathematicians and computer scientists in solving many problems. Recently, applying results in lattice theory to solving problems has been very active, especially in cryptology. Many public-key cryptosystems based on the hardness of lattice problems were proposed. Opening the door to this line is the one in [1] by Ajtai and Dwork. Next, many improvements and new cryptosystems were proposed, such as [2], [3], [4], [5], [6]. Besides, many other cryptographic schemes were sequentially generated, see [7], [8], [9], [10].

In [1], Ajtai and Dwork proposed a cyptosystem for which the security bases on on the equivalence between the average-case and worst-case of lattice problems. This is the first known cryptosystem that relies on the hardness in the worst case of a computational problem. More specifically, the security of Ajtai-Dwork cryptosystem heavily relies on the hardness in the worst case of n^8-uSVP.[1] Although n^8-uSVP is not known to be hard, but this cryptosystem attracts considerable attentions from cryptologists. Next, Cai and Cusick [3] also proposed a new one related to the hardness of $n^{4+\epsilon}$-uSVP; still, the security has not been proven to be based on the worst-case hardness of the problem. Another cryptosystem related to uSVP is by Regev [5]. In the security aspect, Regev's cryptosystem has more advantages than the one of Ajtai and Dwork [1] because its security bases on the hardness in the worse-case of $n^{1.5}$-uSVP. Nonetheless, its practical implementation does not improve the one of Ajtai-Dwork due to the fact that the key size is quite big, $O(n^4)$, and the length of the ciphertext is much larger than the one of the plaintext.

From the disadvantages of these cryptosystems, many cryptographic schemes have been proposed to improve the key size, implementation time and the ciphertext's length. The scheme proposed by Regev in [6] bases on the hardness of $n^{2.5}$-SVP for quantum algorithms; more concretely, its security relies on the hardness of *Learning with errors* (LWE), which is not easier than $n^{2.5}$-SVP

[1] In $f(n)$-uSVP, given an n dimensional lattice L for which the shortest vector is shorter than any non-parallel lattice vector at least a factor $f(n)$, we are asked to find the shortest vector $\boldsymbol{v} \in \mathsf{L}$.

for quantum algorithms.[2] The key size of this cryptosystem is $O(n^2)$. Recently, Micciancio [10] improved greatly the key size and implementation time to almost linear in n. However, the one-way function in [10] bases on the hardness of $n^{1+\epsilon}$-SIVP on cyclic lattices; we note that, up to now, every problem on cyclic lattices has no proof for its hardness even for randomized reductions or quantum reductions. There are, on the other hand, many other cryptographic schemes related to lattice problems such as [8], [7], [9].

Let us return to our consideration on uSVP. So far, there is no proof for its proper hardness. The first author considered this problem is Ajtai [12]; he showed that the worst-case and the average-case of n^c-uSVP, for some large constant c, are equivalent. Following this breakthrough result, Kumar and Sivakumar [13] established that uSVP is NP-hard for randomized reductions; however, we note that, in their proof, they proposed a randomized reduction from SVP to uSVP, and that SVP for ℓ_2 norm was shown to be NP-hard for randomized reductions. This result is a strong evidence for the hardness of uSVP. On the other hand, Cai [14] made the issue more attractive by showing that $n^{1/4}$-uSVP cannot be NP-hard unless the polynomial hierarchy collapses. In addition, Regev [11] claimed that $\Theta(n^{2.5})$-uSVP cannot be harder than the *Subset sum problem* (SSP) for quantum reductions; that is, there exists a quantum algorithm for $\Theta(n^{2.5})$-uSVP if SSP is solvable by a certain (quantum) algorithm. Also he showed that the *Dihedral coset problem* in quantum theory cannot be easier than $\Theta(\sqrt{n})$-uSVP for quantum algorithms. In a recent experimental result [15], Gama and Nguyen suggest that uSVP may be easier than SVP for ℓ_2 norm.

Almost all results listed above clearly does not imply either the proper hardness or the ease of uSVP. The result of Kumar and Sivakumar in [13] gives a strong evidence for the hardness of uSVP; however, it is not the proof for the proper NP-hardness of the problem. For this reason and the important role of uSVP in cryptography, we have studied the complexity of uSVP and other lattice problems, and obtain the following results.

Our contribution in this paper:

- We prove that uSVP for ℓ_∞ norm is NP-hard,
- SAP, GCVP, SVP$'$ and CVP under the assumption that the number of solutions (regardless of sign) is at most 1, for any ℓ_p norm, $p \geq 1$, are also NP-hard. To our best knowledge, this is the first result on the hardness of these problems with the given assumption.

The technique: To obtain the hardness result for uSVP, we present a deterministic polynomial time reduction from *0-1 Knapsack Optimization Problem* (KOP) to SVP. The reduction has the property that if KOP has unique solution, then SVP has exactly two solutions which are parallel together; if KOP has no solution, it is easy to check this fact from the solution to SVP. Hence we obtain the main result if KOP having at most one solution is NP-hard. Note that the NP-hardness of KOP having at most one solution was clearly proven in [16]. We remark that the reduction works only for ℓ_∞ norm, and results in a quite complicated lattice.

For SAP, we use the same technique as the one for SVP, with some suitable modifications. However, the lattice of the new SAP is simpler than the one of SVP. To reveal the working of the reduction for any ℓ_p norm, $p \geq 1$, some careful observations on the lattice are necessary. Since SAP is a special case of GCVP, we have the result for GCVP. For SVP$'$ and CVP, we need a result of Micciancio in [17].

[2] Here by saying LWE is not easier than $n^{2.5}$-SVP for quantum algorithms we mean that if there is a (quantum) algorithm for LWE, then $n^{2.5}$-SVP can be solved by a certain quantum algorithm. Sometime we may say that LWE is quantum reducible to $n^{2.5}$-SVP. For more discussion, see [11].

The paper is organized as follows: Section 2 presents some definitions in lattice theory and some problems on which we work. Section 3 is dedicated to presenting the main results. The reduction from KOP to SVP is presented in Section 4. Section 5 presents the reduction from KOP to SAP.

2 Some definitions

Given m linearly independent vectors $\boldsymbol{b}_1, ..., \boldsymbol{b}_m$ in $\mathbb{R}^n$, the lattice L generated by these vectors is the set of all integer linear combinations of them.

$$\mathsf{L} = \{z_1\boldsymbol{b}_1 + \cdots + z_m\boldsymbol{b}_m | z_i \in \mathbb{Z}, i = \overline{1, m}\}$$

$\{\boldsymbol{b}_1, ..., \boldsymbol{b}_m\}$ are said to form a basis of L; in other words, the lattice L is generated by an $n \times m$ basis matrix $\boldsymbol{B} = (\boldsymbol{b}_1, ..., \boldsymbol{b}_m)$. Any vector $\boldsymbol{x} \in \mathsf{L}$ can be expressed as $\boldsymbol{x} = \boldsymbol{Bz}$, where $\boldsymbol{z} \in \mathbb{Z}^m$. If $m = n$, then L is called a full-dimensional lattice. The Euclidean length of a vector $\boldsymbol{x} = (x_1, ..., x_n)$ for ℓ_p norm $(p \geq 1)$ is a real number $\|\boldsymbol{x}\|_p = \sqrt[p]{|x_1|^p + \cdots + |x_n|^p}$. In this paper, we denote by $[n]$ the set $\{1, 2, ..., n\}$. If $\boldsymbol{x}$ is a vector, then the ith component of $\boldsymbol{x}$ is denoted by x_i. The length of a vector or the distance between two vectors in the following problems is for ℓ_p norm.

Definition 1 (SVP$_p$). *Given a lattice L, the* Shortest Vector Problem *is to find the shortest non-zero vector in L.*

Definition 2 (SVP$'_p$). *Given a lattice L generated by a basis $\boldsymbol{B}$, and an index i, we are asked to find the shortest vector among all lattice vectors of the form $\boldsymbol{Bz}$, where $z_i \neq 0$.*

Definition 3 (CVP$_p$). *Given a lattice L and a target vector $\boldsymbol{t}$ in $\mathbb{R}^n$, the* Closest Vector Problem *is to find the closest lattice point to $\boldsymbol{t}$.*

Definition 4 (GCVP$_p$). *Given a lattice L, an affine space A and a target vector $\boldsymbol{t}$ in $\mathbb{R}^n$, the* Generalized Closest Vector Problem *is to find a vector in $\mathsf{L} \backslash \mathsf{A}$ closest to $\boldsymbol{t}$.*

Definition 5 (SAP$_p$). *Given a lattice L and a subspace S in $\mathbb{R}^n$, the* Subspace Avoiding Problem *is to find the shortest vector in $\mathsf{L} \backslash \mathsf{S}$.*

Besides, if there is an additional assumption that the number of solutions regardless of sign is at most one, then we denote SVP$_p$ by uSVP$_p$. We define uSVP$'_p$, uCVP$_p$, uGCVP$_p$, and uSAP$_p$ in the same manner.

Another problem on which we work is the *0-1 Knapsack Optimization Problem* (KOP). If KOP has at most one solution, it is called *Unique 0-1 Knapsack Optimization Problem* (uKOP).

Definition 6 (KOP). *Find a vector $\boldsymbol{x} \in \mathbb{Z}^n$ satisfying*

$$\begin{cases} f(\boldsymbol{x}) = c_1x_1 + \cdots + c_nx_n \to \max \\ a_1x_1 + \cdots + a_nx_n = b \\ \boldsymbol{x} \in \{0; 1\}^n \end{cases} \tag{1}$$

Where the coefficients are positive integers.

3 Main results

The following theorem on uKOP was proven in [16].[3]

Theorem 1. uKOP *is NP-hard.*

In Section 5, we present a reduction from KOP to SAP for any ℓ_p norm, $p \geq 1$. The reduction has an important property that if KOP has solutions, then the number of solutions to SAP_p is as twice as the one to KOP. In particular, if KOP has unique solution, then SAP_p has exactly two parallel solutions; and we can easily check whether KOP has no solution by using the solution to SAP_p. This property implies that we have a reduction from uKOP to uSAP_p. Hence, combining this fact with Theorem 1 yields the following.

Theorem 2. *For any $p \geq 1$,* uSAP_p *is NP-hard.*

It is easy to see that SAP_p is a special case of GCVP_p, where the input of GCVP_p consists of a linear subspace and a target vector $\boldsymbol{t} = 0$. Consequently, any hardness result for SAP_p is applicable to GCVP_p.

Corollary 1. *For any $p \geq 1$,* uGCVP_p *is NP-hard.*

Recently, Micciancio presents in [17] an efficient reduction from SAP_p to SVP'_p. To establish this reduction, he uses a collection $\mathcal{C}$ of SVP'_p instances. He then solves all these instances to get a collection of vectors, and chooses the shortest one in the collection as the output. Assume that the SAP_p instance in Micciancio's reduction has unique solution. Then it is easy to see that there exists a SVP'_p instance such that its solution is the output of the reduction, and it has unique solution. Furthermore, in his reduction, all members in $\mathcal{C}$ must be solved. These observations implies solving uSVP'_p is at least as hard as solving uSAP_p.

Corollary 2. *For any $p \geq 1$,* uSVP'_p *is NP-hard.*

Now we consider the closest vector problem. It is known that CVP_p is a special case of GCVP_p. However, the hardness of uCVP_p cannot be directly derived from the one of uGCVP_p. To show the result for uCVP_p, we use the reduction from SVP'_p to CVP_p in [17]. The argument is similar to the one above, and hence we obtain

Corollary 3. *For any $p \geq 1$,* uCVP_p *is NP-hard.*

The last problem considered in this paper and behaving hard to be shown its hardness is uSVP. Here we are only able to show its hardness for ℓ_∞ norm by presenting a reduction from KOP to SVP. See the reduction in Section 4.

Theorem 3. uSVP_∞ *is NP-hard.*

[3] In fact, the result is for the Unique Knapsack Minimization Problem. However, we can easily adapt the proof for uKOP. For example, see the Appendix.

4 Reduction from KOP to SVP

In [18], Khoát presents a reduction from the *Bounded integer programming* (BKP) to SAP_∞. The reduction yields a lattice generated by a quite complex basis. Although SVP is a special case of SAP, but we cannot obtain any hardness result for SVP from the one of SAP. The lattice resulted from the reduction in [18] has the properties that a solution for SAP_∞ may not be the shortest vector in the lattice, and the shortest vector of the lattice may not lead to any solution to the original BKP. As a result, with that lattice, we cannot have the reduction from BKP to SVP. Here, we introduce a more complicated lattice.

Overview of the reduction: It is well-known that KOP is easily reducible to the problem of finding a vector $\boldsymbol{x}$ in $\{\boldsymbol{x} \in \{0;1\}^n : \boldsymbol{a} \cdot \boldsymbol{x} = b, \boldsymbol{c} \cdot \boldsymbol{x} \geq \hat{f}\}$, where $\boldsymbol{a} = (a_1, ..., a_n), \boldsymbol{c} = (c_1, ..., c_n)$. So we will solve KOP by the following procedure. Let $f_{\max} > \sum_{j=1}^n c_j$ be the upper bound for $f(\boldsymbol{x})$, $f_0 = 0, \boldsymbol{x}^* = 0$.

Step 1. we use SVP_∞ oracle to find the shortest vector $\hat{\boldsymbol{y}}$ in the lattice L described below for $f_c = \lceil (f_0 + f_{\max})/2 \rceil$.

Step 2. check whether $(|\hat{y}_1|, ..., |\hat{y}_n|)$ is a solution to the problem of finding a vector $\boldsymbol{x}$ in $\{\boldsymbol{x} \in \{0;1\}^n : \boldsymbol{a} \cdot \boldsymbol{x} = b, \boldsymbol{c} \cdot \boldsymbol{x} \geq f_c\}$. If the answer is YES, we set $f_0 := f_c, \boldsymbol{x}^* = (|\hat{y}_1|, ..., |\hat{y}_n|)$. Otherwise, we set $f_{\max} := f_c - 1$.

Step 3. if $f_{\max} > f_0$, then go to *Step 1.* Otherwise, we can have the answer for KOP: if $\boldsymbol{x}^* \neq 0$, then return $\boldsymbol{x}^*$ as the solution of KOP. Otherwise, we conclude that KOP has no solution.

Note that the above procedure bases on binary technique. The upper bound $f_{\max}$ is easily found, and thus the number of calls to SVP_∞ oracle is a polynomial in the length of the input of KOP. This implies KOP is reducible to SVP_∞. However, the question is whether $(|\hat{y}_1|, ..., |\hat{y}_n|)$ is always a solution to ***pKOP***, which denotes the problem in *Step 2*, in case it has solutions. This question will be answered in the next analyzing on L. From these observations, we know that the last problem we must consider is the reduction from pKOP to SVP_∞.

We now illustrate the reduction from pKOP to SVP_∞. Assume that we are given pKOP. Without loss of generality, we assume that $a_i \leq b, \forall i \in [n]$, and $\sum_{i=1}^n a_i \neq 2b$. (The later assumption can be easily guaranteed, for example, see the Appendix). Then the new lattice L is generated by the following basis:

$$\boldsymbol{B} = \begin{pmatrix} \boldsymbol{I}_n & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_0\boldsymbol{a} & -s_0 b & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{I}_n & 0 & 0 & 0 & 0 & 0 \\ s_1\boldsymbol{D} & s_1\lambda^n & s_1\boldsymbol{D} & -s_1\gamma & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{I}_n & 0 & 0 & 0 \\ s_2 s_4 \boldsymbol{I}_n & 0 & -s_2 s_4 \boldsymbol{I}_n & 0 & -s_2 s_4 \boldsymbol{I}_n & s_3 s_4 \boldsymbol{I}_n & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{1}{f_{\max} - f_c} & 0 \\ -s_5 s_6 \boldsymbol{c} & s_5 s_6 f_{\max} & 0 & 0 & 0 & 0 & -s_5 s_6 & s_5 s_7 \end{pmatrix}$$

where $\boldsymbol{I}_n$ is an identity matrix of size n; s_j and λ are optional integers satisfying $s_j > n^2$ for $j \in \{0, 1, 2, 3, 4, 5, 6\}, s_7 \geq f_{\max} n^2 b, \gcd(s_2, s_3) = 1, \gcd(s_6, s_7) = 1$, and $\lambda > n^3$; $\boldsymbol{D} = (1, \lambda, ..., \lambda^{n-1})$, and $\gamma = 1 + \lambda + \cdots + \lambda^n$.

The following lemma presents the first properties of L.

Lemma 1. *For any vector $\boldsymbol{y} = \boldsymbol{B}\boldsymbol{z} \in L$, if $\boldsymbol{y}$ does not satisfy one of the following conditions*

$$y_{n+1} = y_{2n+2} = y_{3n+2+i} = y_{4n+4} = 0,\ \forall i \in [n] \tag{2}$$

$$y_i + y_{n+1+i} = z_{n+1},\ \forall i \in [n] \tag{3}$$

$$y_i - y_{n+1+i} = y_{2n+2+i},\ \forall i \in [n] \tag{4}$$

$$y_{4n+3} = \left(-\sum_{j=1}^{n} c_j y_j + f_{\max} z_{n+1} \right) \frac{1}{f_{\max} - f_c} \tag{5}$$

then $\boldsymbol{y}$ has at least one component with magnitude not less than $\Theta(p)$, where $p = \min\{s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7/nf_{\max}, \lambda/n\}$.

Proof. The claims (2) and (3) can be proven as the one in Lemma 5 in [18]. The remainder is to prove (4) and (5).

Assuming that $\boldsymbol{y}$ satisfies both (2) and (3), then we have $y_{3n+2+i} = y_{4n+4} = 0, \forall i \in [n]$. This means

$$s_2(y_i - y_{n+1+i} - y_{2n+2+i}) = -s_3 z_{3n+2+i},\ \forall i \in [n].$$

Remembering that $\gcd(s_2, s_3) = 1$, if $y_i - y_{n+1+i} - y_{2n+2+i} \neq 0$, then $y_i - y_{n+1+i} - y_{2n+2+i}$ is a non-zero multiple of s_3; therefore, either $|y_i|$ or $|y_{n+1+i}|$ or $|y_{2n+2+i}|$ is not less than $\Theta(s_3)$. As a consequence, if $\boldsymbol{y}$ does not satisfy (4), then it has at least one component with magnitude at least $\Theta(p)$.

Since $\boldsymbol{y}$ satisfies (2), we have $y_{4n+4} = 0$. This means $s_6\left(-\sum_{j=1}^{n} c_j z_j + f_{\max} z_{n+1} - z_{4n+3}\right) + s_7 z_{4n+4} = 0$, that is,

$$s_6 \left(-\sum_{j=1}^{n} c_j y_j + f_{\max} z_{n+1} - (f_{\max} - f_c) y_{4n+3} \right) + s_7 z_{4n+4} = 0.$$

If $z_{4n+4} = 0$, we have $-\sum_{j=1}^{n} c_j y_j + f_{\max} z_{n+1} - (f_{\max} - f_c) y_{4n+3} = 0$, and thus (5) follows. We now consider the case that $\boldsymbol{y}$ does not satisfy (5). This implies $z_{4n+4} \neq 0$; consequently, $-\sum_{j=1}^{n} c_j y_j + f_{\max} z_{n+1} - (f_{\max} - f_c) y_{4n+3}$ is a non-zero multiple of s_7, due to $\gcd(s_6, s_7) = 1$. In this case, either $\sum_{j=1}^{n} c_j y_j$ or $f_{\max} z_{n+1}$ or $(f_{\max} - f_c) y_{4n+3}$ is of magnitude at least $\Theta(s_7)$.

If $|(f_{\max} - f_c) y_{4n+3}| \geq \Theta(s_7)$, we have $|y_{4n+3}| \geq \Theta(p)$, and therefore we obtain the claim of the lemma. Now assuming that $|f_{\max} z_{n+1}| \geq \Theta(s_7)$, we have $|z_{n+1}| \geq \Theta(s_7/f_{\max})$. Note that $y_{n+1} = 0$, since $\boldsymbol{y}$ satisfies (2). That is, we have the equation

$$\sum_{j=1}^{n} a_j y_j = b z_{n+1}. \tag{6}$$

Since $a_j \leq b, \forall j \in [n]$, it is not hard to see that every solution to (6) has at least one component with magnitude not less than $\Theta(z_{n+1}/n)$. This means $\boldsymbol{y}$ has at least one component with magnitude at least $\Theta(s_7/(nf_{\max}))$.

If $|\sum_{j=1}^{n} c_j y_j| \geq \Theta(s_7)$, we easily derive the claim of the lemma, since $c_j \leq f_{\max}, \forall j \in [n]$. This completes the proof of the lemma. □

Another property of L is presented in the next lemma. This is the key observation for the proof of Theorem 3.

Lemma 2. *Suppose that pKOP has solutions. For every non-zero vector $\boldsymbol{y}^* \in \mathsf{L}$, either $(y_1^*, ..., y_n^*)$ or $-(y_1^*, ..., y_n^*)$ is a solution to pKOP if $\|\boldsymbol{y}^*\|_\infty \leq 1$.*

Proof. Combining the hypothesis $\|\boldsymbol{y}^*\|_\infty \leq 1$ with Lemma 1, $\boldsymbol{y}^* = \boldsymbol{Bz}$ satisfies (2), (3), (4) and (5). Then we immediately have

$$\sum_{j=1}^{n} a_j y_j^* = b z_{n+1}, \tag{7}$$

$$y_i^* + y_{n+1+i}^* = z_{n+1}, \forall i \in [n]. \tag{8}$$

We now prove the lemma by combining the following facts.

Fact 1. if $\sum_{j=1}^n a_j y_j^* \neq 0$, then either $(y_1^*, ..., y_n^*)$ or $-(y_1^*, ..., y_n^*)$ is a solution to pKOP.

Proof. Combining the hypothesis $\sum_{j=1}^n a_j y_j^* \neq 0$ with (7) yields $z_{n+1} \neq 0$. From (8), it is not hard to see that if $|z_{n+1}| > 2$, then either $|y_i^*| > 1$ or $\left|y_{n+1+i}^*\right| > 1$. This means $\|\boldsymbol{y}^*\|_\infty > 1$, contrarily. Consequently, the remainder of the proof is to consider the case $0 < |z_{n+1}| \leq 2$.

Assume that $z_{n+1} = 2$. (The case $z_{n+1} = -2$ can be dealt with by the same way.) Then we have $y_i^* + y_{n+1+i}^* = 2$, $\forall i \in [n]$. An immediate observation is that if $y_i^* = y_{n+1+i}^* = 1, \forall i \in [n]$, then we have $\sum_{j=1}^n a_j = 2b$, being contrary to the assumption $\sum_{j=1}^n a_j \neq 2b$. Hence, there exists $r \in [n]$ such that either $|y_r^*| > 1$ or $\left|y_{n+1+r}^*\right| > 1$. This leads to $\|\boldsymbol{y}^*\|_\infty > 1$, contrarily. These observations imply $0 < z_{n+1} \leq 1$.

If $z_{n+1} = 1$, then we have $\sum_{j=1}^n a_j y_j^* = b$ and $y_i^* + y_{n+1+i}^* = 1, \forall i \in [n]$. From this fact, we remark that if there exists $r \in [n]$ such that $y_r^* < 0$, then $y_{n+1+r}^* > 1$. This means $\|\boldsymbol{y}^*\|_\infty > 1$, contrarily. Consequently, $0 \leq y_i^* \leq 1, \forall i \in [n]$. That is, $(y_1^*, ..., y_n^*)$ satisfies

$$\sum_{j=1}^{n} a_j y_j^* = b, (y_1^*, ..., y_n^*) \in \{0; 1\}^n. \tag{9}$$

Furthermore, $\boldsymbol{y}^*$ also satisfies $y_{4n+3}^* = \left(-\sum_{j=1}^n c_j y_j^* + f_{\max}\right) \frac{1}{f_{\max} - f_c}$ due to (5). It is easy to see that if $\sum_{j=1}^n c_j y_j^* < f_c$, then $y_{4n+3}^* > 1$; hence, $\|\boldsymbol{y}^*\|_\infty > 1$, contrarily. This suggests that

$$\sum_{j=1}^{n} c_j y_j^* \geq f_c. \tag{10}$$

Note that if $\sum_{j=1}^n c_j y_j^* > f_{\max}$, there exists r such that $y_r^* > 1$. This implies $\|\boldsymbol{y}^*\|_\infty > 1$, contrarily. Thus we must have $\sum_{j=1}^n c_j y_j^* \leq f_{\max}$. Combining this with (9) and (10) leads to the fact that $(y_1^*, ..., y_n^*)$ is a solution to pKOP.

By the same argument we can show that if $z_{n+1} = -1$, then $-(y_1^*, ..., y_n^*)$ is a solution to pKOP. The proof is completed. □

Fact 2. if $\sum_{j=1}^{n} a_j y_j^* = 0$, then $\|\boldsymbol{y}^*\|_\infty > 1$.

Proof. Since $\boldsymbol{y}^*$ satisfies (3), (4) and (7), we have $z_{n+1} = 0$, and thus,

$$y_i^* + y_{n+1+i}^* = 0, \forall i \in [n], \tag{11}$$

$$y_i^* - y_{n+1+i}^* = y_{2n+2+i}^*, \forall i \in [n]. \tag{12}$$

Assume that $y_i^* = 0, \forall i \in [n]$. Then $y_{n+1+i}^* = y_{2n+2+i}^* = 0, \forall i \in [n]$. Moreover, $y_{4n+3}^* = 0$, since $z_{n+1} = 0$. This means $\boldsymbol{y}^* = 0$, contrarily. Consequently, there exists at least an index $i \in [n]$ such that $y_i^* \neq 0$. By (11) we have $y_{n+1+i}^* = -y_i^* \neq 0$. Note that $y_i^* \in \mathbb{Z}$, and thus $|y_{n+1+i}^*| = |y_i^*| \geq 1$. Combining these facts with (12) implies $|y_{2n+2+i}^*| \geq 2$. The claim of Fact 2 is clear. □

Combining Fact 1 with Fact 2 yields the proof of Lemma 2. □

From the property of L revealed in Lemma 2, we remark that the shortest vectors in L for ℓ_∞ norm should have length not greater than 1, provided pKOP has solutions. Indeed, a careful observation reveals that if $\hat{\boldsymbol{x}}$ is a solution to pKOP, then $\hat{\boldsymbol{y}} = (\hat{\boldsymbol{x}}, 0, \boldsymbol{e} - \hat{\boldsymbol{x}}, 0, 2\hat{\boldsymbol{x}} - \boldsymbol{e}, 0.\boldsymbol{e}, \frac{f_{\max} - f(\hat{\boldsymbol{x}})}{f_{\max} - f_c}, 0)$ is a vector in L, and $\|\hat{\boldsymbol{y}}\| \leq 1$.[4] Therefore, $\|\boldsymbol{y}^*\|_\infty \leq \|\hat{\boldsymbol{y}}\|_\infty \leq 1$ if $\boldsymbol{y}^*$ is the shortest vector in L for ℓ_∞ norm. These facts quickly leads to the following corollary.

Corollary 4. *Suppose that pKOP has solutions. If $\boldsymbol{y}^*$ is the shortest vector in L for ℓ_∞ norm, then either $(y_1^*, ..., y_n^*)$ or $-(y_1^*, ..., y_n^*)$ is a solution to pKOP.*

We note that $\hat{\boldsymbol{x}}$ cannot be the zero vector, thus $\|\hat{\boldsymbol{y}}\|_\infty = 1$. It is not hard to see that every shortest vector in L has this length, provided that pKOP has solutions, and that if $\boldsymbol{y}^*$ is the shortest vector in L, then so is $-\boldsymbol{y}^*$. These facts mean the following lemma is straightforward.

Lemma 3. *Suppose that pKOP has solutions. The number of shortest vectors in L is as twice as the number of solutions to pKOP.*

From the above observations, the claim of theorem 3 is easily proven.

Proof (Proof of Theorem 3). Notice that Corollary 4 implies a reduction from pKOP to SVP$_\infty$. Moreover, any parameters in constructing the lattice L can be chosen in time polynomial in the size of the input of pKOP. Thus, the reduction is deterministic polynomial time. Furthermore, if pKOP has unique solution, so is SVP$_\infty$ up to a sign. As a consequence, combining these observations with the procedure at the beginning of this section, we have the reduction from uKOP to uSVP$_\infty$. The proof of Theorem 3 is completed. □

5 Hardness of uSAP for any ℓ_p norm

The reduction from KOP to SAP can be similarly constructed to the one in Section 4. However, some suitable modifications are needed to obtain the desired result. Hence, we will present the reduction in a high level, not be detail in some claims if unnecessary.

[4] Here $\boldsymbol{e} = (1, 1, ..., 1) \in \mathbb{R}^n$.

Assume that we are given a pKOP instance. Then the subspace we deal with is $\mathsf{S} = \{\boldsymbol{y} \in \mathbb{Z}^{2n+4} : a_1y_1 + \cdots + a_ny_n = 0\}$. Consider the lattice L_1 generated by the following basis:

$$\boldsymbol{B}_1 = \begin{pmatrix} \boldsymbol{I}_n & 0 & 0 & 0 & 0 & 0 \\ s_0\boldsymbol{a} & -s_0b & 0 & 0 & 0 & 0 \\ 0 & 0 & \boldsymbol{I}_n & 0 & 0 & 0 \\ s_1\boldsymbol{D} & s_1\lambda^n & s_1\boldsymbol{D} & -s_1\gamma & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{f_{\max}-f_c} & 0 \\ -s_5s_6\boldsymbol{c} & s_5s_6f_{\max} & 0 & 0 & -s_5s_6 & s_5s_7 \end{pmatrix},$$

where the parameters are chosen as in Section 4, except $f_{\max} > n^2\sum_{j=1}^n c_j$.

For this lattice, we have the first properties as follows.

Lemma 4. *For any vector $\boldsymbol{y} = \boldsymbol{B}_1\boldsymbol{z} \in \mathsf{L}_1$, if $\boldsymbol{y}$ does not satisfy one of the following conditions*

$$y_{n+1} = y_{2n+2} = y_{2n+4} = 0, \forall i \in [n] \tag{13}$$

$$y_i + y_{n+1+i} = z_{n+1}, \forall i \in [n] \tag{14}$$

$$y_{2n+3} = \left(-\sum_{j=1}^n c_jy_j + f_{\max}z_{n+1}\right)\frac{1}{f_{\max} - f_c} \tag{15}$$

then $\boldsymbol{y}$ has at least one component with magnitude not less than $\Theta(p)$, where $p = \min\{s_0, s_1, s_5, s_6, s_7/nf_{\max}, \lambda/n\}$.

The proof of the lemma is the same as the one of Lemma 1, thus we omit it here. The next lemma is the key observations to prove Theorem 2.

Lemma 5. *Suppose that pKOP has solutions. If $\boldsymbol{y}^*$ is the shortest vector in $\mathsf{L}_1\backslash\mathsf{S}$ for ℓ_p norm, $p \geq 1$, then either $(y_1^*, ..., y_n^*)$ or $-(y_1^*, ..., y_n^*)$ is a solution to pKOP.*

Proof. A careful observation yields the fact that if $\hat{\boldsymbol{x}}$ is a solution to pKOP, then $\hat{\boldsymbol{y}} = (\hat{\boldsymbol{x}}, 0, \boldsymbol{e} - \hat{\boldsymbol{x}}, 0, \frac{f_{\max}-f(\hat{\boldsymbol{x}})}{f_{\max}-f_c}, 0)$ is a vector in $\mathsf{L}_1\backslash\mathsf{S}$. Since $\hat{\boldsymbol{x}} \in \{0;1\}^n$ and $f_{\max} - f(\hat{\boldsymbol{x}}) \leq f_{\max} - f_c$, we have $\hat{y}_j \in \{0;1\}, \forall j \neq 2n+3$, and $0 \leq \hat{y}_{2n+3} = \frac{f_{\max}-f(\hat{\boldsymbol{x}})}{f_{\max}-f_c} \leq 1$. Therefore, $\|\hat{\boldsymbol{y}}\|_p = \sqrt[p]{n + (\frac{f_{\max}-f(\hat{\boldsymbol{x}})}{f_{\max}-f_c})^p} \leq \sqrt[p]{n+1}$. This suggests that $\|\boldsymbol{y}^*\|_p \leq \|\hat{\boldsymbol{y}}\|_p = \sqrt[p]{n+1}$. Combining this fact with Lemma 4, $\boldsymbol{y}^*$ must satisfy (13), (14) and (15).

Since $\boldsymbol{y}^* \in \mathsf{L}_1$, there is an integer vector $\boldsymbol{z}^*$ such that $\boldsymbol{y}^* = \boldsymbol{B}_1\boldsymbol{z}^*$. It is not hard to see that $y_i^* = z_i^*, y_{n+1+i}^* = z_{n+1+i}^*, \forall i \in [n], y_{2n+3}^* = \frac{z_{2n+3}^*}{f_{\max}-f_c}$. Since $\boldsymbol{y}^*$ satisfies (13), (14) and (15), we have $\sum_{i=1}^{2n+4} y_i^* = \sum_{i=1}^n(y_i^* + y_{n+1+i}^*) + y_{2n+3}^* = nz_{n+1}^* + y_{2n+3}^*$. On the other hand, by the hypothesis $\boldsymbol{y}^* \notin \mathsf{S}$ and $y_{n+1}^* = s_0(\sum_{j=1}^n a_jz_j^* - bz_{n+1}^*) = 0$, z_{n+1}^* must be non-zero. Furthermore, if $|z_{n+1}^*| \geq 2$, then $\|\boldsymbol{y}^*\|_p \geq \sqrt[p]{2n}$; this means $\|\boldsymbol{y}^*\|_p > \|\hat{\boldsymbol{y}}\|_p$, contrarily. Hence, the remainder of the proof is to consider the case $|z_{n+1}^*| = 1$.

Assume that $z_{n+1}^* = 1$. Then we immediately have $y_i^* + y_{n+1+i}^* = 1, \forall i \in [n], y_{2n+3}^* = (f_{\max} - \sum_{j=1}^n a_jy_j^*)\frac{1}{f_{\max}-f_c}$ and $\sum_{j=1}^n a_jy_j^* = b$. From this fact, the equation $\sum_{j=1}^{2n+4} y_j^* = n + y_{2n+3}^*$ follows. Noting that $y_i^* = z_i^*, \forall i \in [n]$, every component of $\boldsymbol{y}^*$ is integral, except y_{2n+3}^*.

We now make some more careful observations on $\boldsymbol{y}^*$. If there exists $i \in [n]$ such that $y^*_i < 0$, then $y^*_{n+1+i} \geq 2$. This implies $\sum_{j=1}^{2n+4} |y^*_j| = \sum_{j=1}^{2n+2} |y^*_j| + |y^*_{2n+3}| \geq n+1+|y^*_{2n+3}| > n+1$.[5] As a result, we obtain $\|\boldsymbol{y}^*\|_p > \sqrt[p]{n+1} \geq \|\hat{\boldsymbol{y}}\|_p$, contrarily. Also if $\exists i \in [n], y^*_i > 1$, then $\|\boldsymbol{y}^*\|_p > \|\hat{\boldsymbol{y}}\|_p$, contrarily. In short, $0 \leq y^*_i \leq 1, \forall i \in [n]$; that is, $\boldsymbol{y}^*$ satisfies $(y^*_1, ..., y^*_n) \in \{0;1\}^n$ and $\sum_{j=1}^n a_j y^*_j = b$.

From the above observations, we note that $\|\boldsymbol{y}^*\|_p = \sqrt[p]{n + |y^*_{2n+3}|^p} \leq \sqrt[p]{n+1}$. This implies $|y^*_{2n+3}| \leq 1$; and thus, $f_{\max} - \sum_{j=1}^n c_j y^*_j \leq f_{\max} - f_c$, or equivalently, $f_c \leq \sum_{j=1}^n c_j y^*_j \leq f_{\max}$. As a consequence, $(y^*_1, ..., y^*_n)$ is a solution to pKOP.

For the case $z^*_{n+1} = -1$, we can easily show that $-(y^*_1, ..., y^*_n)$ is a solution to pKOP. This completes the proof of the lemma. □

Proof (Proof of Theorem 2). Notice that Lemma 5 implies a reduction from pKOP to SAP$_p$, $p \geq 1$. Moreover, any parameters in constructing the lattice L_1 can be chosen in time polynomial in the size of the input of pKOP. Thus, the reduction is deterministic polynomial time. Furthermore, if pKOP has unique solution, so is SAP$_p$ up to a sign. As a consequence, combining these observations with the procedure being similar to the one at the beginning of Section 4, we have the reduction from uKOP to uSAP$_p$. The proof of Theorem 2 is completed. □

Conclusion

We have discussed the hardness of uSVP. However, the proof works only for ℓ_∞ norm. For other norm, this technique may not be applied. Still, we believe that uSVP for other norm is hard even in its approximating version. This paper presents the hardness results for various other lattice problems for any ℓ_p norm, such as uCVP, uGCVP, and uSAP. For the approximating versions of these problems, we cannot applied the technique in this paper, and thus the hardness of them is still open. We believe that approximating uSVP, uCVP, uGCVP, and uSAP with any constant factor are hard.

Appendix: the assumption $\sum_{j=1}^n a_j \neq 2b$

Assume now that the original pKOP satisfying $\sum_{j=1}^n a_j = 2b$. Then we reduce it to the new one satisfying our desire. The new pKOP is as follows:

Find a vector $\boldsymbol{x} \in \mathbb{Z}^{n+1}$ *satisfying*

$$\begin{cases} a'_1 x_1 + \cdots + a'_{n+1} x_{n+1} = b' \\ c_1 x_1 + \cdots + c_n x_n \geq f_c \\ \boldsymbol{x} \in \{0;1\}^{n+1} \end{cases} \tag{16}$$

Where $a'_j = a_j$*, for all* $j \leq n, a'_{n+1} = (n+1)b, b' = b + (n+1)b$.

It is not hard to see that solving the new pKOP (16) is equivalent to solving the original pKOP. Indeed, if $(x_1, ..., x_n, 1)$ is a solution to (16), then $a_1 x_1 + \cdots + a_n x_n = b$ and $c_1 x_1 + \cdots + c_n x_n \geq f_c$. This implies that $(x_1, ..., x_n)$ is a solution to the original pKOP. Moreover, $(x_1, ..., x_n, 0)$ cannot be a solution to (16) due to the fact that $a_1 x_1 + \cdots + a_n x_n = b'$ means a contrary to the assumption $a_j < b, \forall j \leq n$. These observations imply that $(x_1, ..., x_n, x_{n+1})$ is a solution to (16) if and only if $(x_1, ..., x_n)$ is a solution the original pKOP.

Note that (16) satisfies $\sum_{j=1}^{n+1} a'_j \neq 2b'$. This means the new pKOP satisfies our desired assumption.

[5] We have the last inequality because of the fact that if $y^*_{2n+3} = 0$, then $f_{\max} = \sum_{j=1}^n c_j y^*_j$, and hence, there exists y^*_j with magnitude at least $\Theta(n^2)$ due to the definition of $f_{\max}$.

Appendix: hardness of uKOP

To make the paper self-contained, we include the proof for the hardness of uKOP. This proof is similar to the one for the hardness of the general Knapsack problem in [16]. Now we consider the search version of the 0-1 Knapsack problem:

Find a vector $\boldsymbol{x} \in \mathbb{Z}^n$ satisfying

$$\begin{cases} \boldsymbol{a} \cdot \boldsymbol{x} = b \\ \boldsymbol{x} \in \{0;1\}^n \end{cases} \tag{17}$$

where the coefficients are positive integers.

It is not hard to see that (17) is NP-hard. We are going to reduce (17) to KOP, which has at most one solution. The new KOP is as follows:

Find a solution $\boldsymbol{x}$ to

$$\begin{gathered} f(\boldsymbol{x}) = d_1x_1 + \cdots + d_nx_n \to \min \\ \begin{cases} \boldsymbol{a} \cdot \boldsymbol{x} = b \\ \boldsymbol{x} \in \{0;1\}^n \end{cases} \end{gathered} \tag{18}$$

where d_i's are optional integers such that:

+ $d_1 > 0$,

+ $d_i > w \sum_{j=1}^{i-1} d_j, \ w = \max_k \{\lceil b/a_k \rceil\}, i > 1$

It is clear that if $\boldsymbol{x}^0$ is a solution to (18), then $\boldsymbol{x}^0$ is also a solution to (17); if (18) has no solution, then neither has (17). Moreover, (18) has at most one solution. Indeed, assume that $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ are different solutions to (18). There exists j, $x_j^1 \neq x_j^2$. Let r be the greatest index such that $x_r^1 \neq x_r^2$. Without loss of generality, we assume that $x_r^1 < x_r^2$. Then we have:

$$f(\boldsymbol{x}^1) \qquad = \boldsymbol{d} \cdot \boldsymbol{x}^1 = \textstyle\sum_{j<r} d_jx_j^1 + d_rx_r^1 + \sum_{k>r} d_kx_k^1,$$

$$\begin{aligned} f(\boldsymbol{x}^2) \qquad &= \boldsymbol{d} \cdot \boldsymbol{x}^2 = \textstyle\sum_{j<r} d_jx_j^2 + d_rx_r^2 + \sum_{k>r} d_kx_k^2 \\ &= f(\boldsymbol{x}^1) + \textstyle\sum_{j<r} d_jx_j^2 + d_r(x_r^2 - x_r^1) - \sum_{j<r} d_jx_j^1 \end{aligned}$$

Note that $0 \leq x_j^1 \leq w, \forall j$. Thus $\sum_{j<r} d_jx_j^1 < w \sum_{j<r} d_j < d_r$ (due to the hypothesis on d_i's). Combining this with the fact that $\boldsymbol{x}^1, \boldsymbol{x}^2 \in \mathbb{Z}_+^n$ would lead to $\sum_{j<r} d_jx_j^1 < d_r(x_r^2 - x_r^1)$. Therefore, from the above representation of $f(\boldsymbol{x}^2)$, we conclude that $f(\boldsymbol{x}^1) < f(\boldsymbol{x}^2)$. That is, (18) has at most one solution.

Theorem 4. *Any 0-1 knapsack problem can be polynomial-computationally reduced to a 0-1 knapsack optimization problem having at most one solution.*

Note that 0-1 KP is NP-complete. Thus we obtain the following.

Corollary 5. *uKOP is NP-hard.*

References

1. Ajtai, M., Dwork, C.: A public-key cryptosystem with worst-case/average-case equivalence. In: Proceedings of the 29th annual ACM symposium on Theory of Computing, ACM (1997) 284–293
2. Micciancio, D.: Improving lattice based cryptosystems using the hermite normal form. In: CaLC'01: Revised Papers from the International Conference on Cryptography and Lattices, London, UK, Springer-Verlag (2001) 126–145

3. Cai, J.Y., Cusick, T.W.: A lattice-based public-key cryptosystem. Information and Computation **151**(1-2) (1999) 17–31
4. Goldreich, O., Goldwasser, S., Halevi, S.: Public-key cryptosystems from lattice reduction problems. In: Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology, London, UK, Springer-Verlag (1997) 112–131
5. Regev, O.: New lattice-based cryptographic constructions. Journal of the ACM **51**(6) (2004) 899–942
6. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: Proceedings of the 37th annual ACM symposium on Theory of computing, ACM (2005) 84–93
7. Peikert, C., Rosen, A.: Efficient collision-resistant hashing from worst-case assumptions on cyclic lattices. In: Theory of Cryptography, Third Theory of Cryptography Conference - TCC, Springer (2006) 145–166
8. Peikert, C., Waters, B.: Lossy trapdoor functions and their applications. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing, ACM (2008) 187–196
9. Gentry, C., Peikert, C., Vaikuntanathan, V.: Trapdoors for hard lattices and new cryptographic constructions. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing. (2008) 197–206
10. Micciancio, D.: Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. Computational Complexity **16**(4) (2007) 365–411
11. Regev, O.: Quantum computation and lattice problems. SIAM J. Computing **33**(3) (2004) 738–760
12. Ajtai, M.: Generating hard instances of lattice problems (extended abstract). In: Proceedings of the 37th annual ACM symposium on Theory of computing. (1996) 99–108
13. Kumar, R., Sivakumar, D.: A note on the shortest lattice vector problem. In: IEEE Conference on Computational Complexity. (1999) 200–204
14. Cai, J.Y.: A relation of primal-dual lattices and the complexity of shortest lattice vector problem. Theorietical Computer Sciences **207**(1) (1998) 105–116
15. Gama, N., Nguyen, P.Q.: Predicting lattice reduction. In: Advances in Cryptology - EUROCRYPT 2008, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques. (2008) 31–51
16. Khoát, T.Q.: Relation between the hardness of a problem and the number of its solutions. To appears in Acta Mathematica Vietnamica (2008)
17. Micciancio, D.: Efficient reductions among lattice problems. In: Proc. of the 19th annual ACM-SIAM Symposium on Discrete Algorithms -SODA. (2008) 84–93
18. Khoát, T.Q.: On the bounded integer programming. In: Proceedings of the 2008 IEEE International Conference on Research, Innovation & Vision for the Future - RIVF, Ho Chi Minh city, Vietnam, IEEE (July 2008) 23–28