On the Bounded Integer Programming

THÀN QUANG KHOÁT
Faculty of Information Technology,
Thái Nguyên University,
Thái Nguyên city, Vietnam
Email: quangkhoat@gmail.com

Abstract—The best upper time bound for solving the Bounded Integer Programming (BIP) up to now is \( \text{poly}(\varphi) \cdot n^{2n+o(n)} \), where \( n \) and \( \varphi \) are the dimension and the input size of the problem respectively. In this paper, we show that BIP is solvable in deterministic time \( \text{poly}(\varphi) \cdot n^{n+o(n)} \). Moreover we also show that under some reasonable assumptions, BIP is solvable in probabilistic time \( 2^{O(n)} \).

Index Terms—Bounded Integer programming, Subspace avoiding problem, Closest vector problem, Time bound.

I. INTRODUCTION

Bounded integer programming is a familiar problem with many computer scientists. BIP asks for an integer vector \( x \) satisfying the system \( Ax = b \) of equations and some constraints \( 0 \leq x \leq u \). If there is no upper bound on the variables, then it is called Integer Programming (IP). If BIP has unique solution, then it is called the Bounded Knapsack Problem (BKP). If there is no upper bound on the variables in BKP, then it is called the Knapsack Problem (KP). These problems are surveyed extensively in the literature. However, up to now they still need much time to solve.

In 1983, Lenstra [8] first showed that IP is solvable in polynomial time when the dimension is fixed. After this breakthrough result, researchers continue to improve it and thus, many substantial improvements were proposed. The most remarkable result is from Kannan [5], where he showed that IP is solvable in deterministic time \( \text{poly}(\varphi) \cdot n^{2.5n} \). In his proof of this time bound, lattice problems and approximating subspaces play an important role.

Recently, Khoát [6] showed that BIP is solvable in deterministic time \( \text{poly}(\varphi) \cdot n^{2n+o(n)} \). Moreover, there are some more interesting results for some other problems. For example, IP was shown to be solvable in time \( \text{poly}(\varphi) \cdot n^{n+o(n)} \). He obtained these results by reducing these problems to KP and then reducing KP to SAP, a lattice problem. The reduction from IP to KP is almost efficient in the sense that it preserves the time bound. However, the reduction from BIP to KP is inefficient. The reason is that the number of variables increases doubly after the reduction. Thus, the time bound cannot be preserved.

In this paper, we try to attack BIP and BKP in another way. Instead of reducing to KP, we are going to reduce BKP directly to the Subspace Avoiding Problem (SAP) in lattice theory. The reduction works for both \( \ell_1 \) norm and \( \ell_\infty \) norm, and increases the dimension to \( 2n+2 \), where \( n \) is the dimension (number of variables) of BKP. In [6], the reduction from BKP to SAP uses KP as an intermediate problem, and increases the dimension to \( 4n+2 \). Thus, the reduction here is more efficient.

The reason for our effort in reducing BKP to SAP is that SAP is efficiently reducible to the Closest Vector Problem (CVP) (see [9]), and that CVP can be solved “efficiently”. Indeed, CVP for \( \ell_2 \) norm is solvable in deterministic time \( \text{poly}(\varphi) \cdot n^{n/2+o(n)} \) (see [7]). Furthermore, it is also the time that we need to solve CVP for \( \ell_1 \) norm (see Disc.1 in the Appendix). Therefore, we obtain the new time bound \( \text{poly}(\varphi) \cdot n^{n+o(n)} \) for BKP (and BIP).

Another motivation of this paper is the question: whether integer programming is solvable in time \( 2^{O(n)} \)? This is an interesting question. There are many lattice problems those can be solved within this bound, and the reductions from IP (BIP) to lattice problems give an evidence for the positive answer. However, the right answer may need some complicated tools.

In section III, we study some classes of BIP those are solvable by lattice problems. We refer to [9] for other definitions in lattice theory.

II. REDUCTION FROM BIP TO SAP

This section presents the main result: a reduction from BIP to SAP, and the new time bound for BIP. First of all, we recall some definitions and some conventions.

Subspace Avoiding Problem (SAP):

\( \text{Input:} \) a basis \( B \) of the lattice \( L \), a subspace \( M \subseteq \mathbb{R}^n \).
\( \text{Output:} \) the shortest vector \( v \in L \setminus M \).

Closest Vector Problem (CVP):

\( \text{Input:} \) a basis \( B \) of the lattice \( L \), a vector \( t \subseteq \mathbb{R}^n \).
\( \text{Output:} \) the vector \( v \in L \) closest to \( t \).

We refer to [9] for other definitions in lattice theory.

Bounded Integer Programming (BIP):

Find a vector \( x \in \mathbb{Z}^n \) satisfying
\[
\begin{align*}
Ax &= b \\
0 &\leq x \leq u
\end{align*}
\]

Where \( A \) is an \( m \times n \) integral matrix, \( b \) is a vector in \( \mathbb{Z}^m \), \( u \) is a vector in \( \mathbb{Z}_+^n \).

Bounded Knapsack Problem (BKP):
Find a vector \( x \in \mathbb{Z}^n \) satisfying
\[
\begin{cases}
a_1x_1 + \cdots + a_nx_n = b \\ 0 \leq x \leq u
\end{cases}
\]
(2)

Where the coefficients are positive integers.

We denote by \([n]\) the set \( \{1, 2, \ldots, n\} \), by \( \|x\|_p \) the length of vector \( x \) for \( \ell_p \) norm \((p \geq 1)\) and by \( \|x\| \) the length of vector \( x \) for \( \ell_\infty \) norm. If there is no more explanation, \( Bx \) denotes the product of a matrix \( B \) with a column vector \( x \). If \( x \) and \( u \) are vectors in \( \mathbb{R}^n \), then \( \frac{x}{u} \) denotes the vector \( \left( \frac{x_1}{u_1}, \frac{x_2}{u_2}, \ldots, \frac{x_n}{u_n} \right) \).

It is not hard to see that BIP is reducible to BKP (e.g. we can use Kannan’s technique in [4] to aggregate the equations in BIP). Moreover, the reduction from BIP to BKP preserves the dimension (number of variables). Thus, we can say that they have the same properties.

**Theorem 1.** There exists a deterministic polynomial time reduction from BKP to SAP for \( \ell_\infty \) norm. Furthermore, the reduction increases the dimension by \( n + 2 \), where \( n \) is the dimension of BKP.

**Theorem 2.** There exists a deterministic polynomial time reduction from BKP to SAP for \( \ell_1 \) norm. Furthermore, the reduction increases the dimension by \( n + 2 \), where \( n \) is the dimension of BKP.

**Corollary 3.** There exists a deterministic polynomial time reduction from BIP to SAP for \( \ell_1 \) norm. Moreover, the reduction increases the dimension by \( n + 2 \), where \( n \) is the dimension of BIP.

Combining with the result in [9], BIP is reducible to CVP for \( \ell_1 \) norm. Moreover, the reduction increases the dimension to \( 2n + 2 \). Combining with the argument in Disc.1 in the Appendix (CVP for \( \ell_1 \) norm is solvable in deterministic time \( \text{poly}(n) \cdot n^{n/2+o(n)} \)) would lead to the following result.

**Theorem 4.** BIP is solvable in deterministic time \( \text{poly}(n) \cdot n^{n+o(n)} \), where \( n \) and \( \varphi \) are the dimension and the size of the input respectively.

Now we are going to present the reduction from BKP to SAP. The argument is similar with the one in [6]. Assume that we are given BKP (2). Without lost of generality, we assume that \( a_i < b \) for all \( i \in [n] \). Moreover, we assume that \( \sum_{j=1}^n a_j u_j \neq 2b \). This assumption can be guaranteed by some simple modifications to (2) or by reducing it to new BKP, e.g. see Disc 2 in the Appendix. Then we will deal with the subspace \( S = \{ x \in \mathbb{R}^{2n+2} : a_1 u_1 x_1 + \cdots + a_n u_n x_n = 0 \} \).

Consider the lattice \( L_0 \) generated by the basis \( B_0 \), where
\[
B_0 = \begin{pmatrix}
\hat{U}_n & 0 & 0 \\
0 & -s_0 b & 0 \\
0 & 0 & \hat{U}_n
\end{pmatrix}
\]
is a \((2n+2) \times (2n+2)\) matrix; \( \hat{U}_n = \text{diag}(\frac{1}{u_1}, \frac{1}{u_2}, \ldots, \frac{1}{u_n}) \) is the diagonal matrix of size \( n \); \( a = (a_1, \ldots, a_n) \); \( \delta = u_1 \ldots u_n \); \( \delta_i = \delta / u_i \); \( u_{\text{max}} = \max\{u_1, \ldots, u_n\} \); \( s_0, s_1 \) and \( \lambda \) are optional integers satisfying \( s_0 > n, s_1 > n, \lambda > \delta^2 u_{\text{max}} \); \( C = (\delta_1, \delta_2 \lambda, \ldots, \delta_n \lambda^{n-1}) \), and \( \gamma = 1 + \lambda + \cdots + \lambda^n \).

**Lemma 5.** For any vector \( y = B_0 z \in L_0 \), if \( y \) does not satisfy one of the following conditions
\[
y_{n+1} = y_{2n+2} = 0 \quad (3)
y_{i} + y_{n+1+i} = z_{n+1}, \quad \forall i \in [n] \quad (4)
\]
then \( y \) has at least one coordinate with magnitude not less than \( \Theta(p) \), where \( p = \min\{s_0, s_1, \lambda / (\delta^2 u_{\text{max}})\} \).

**Proof:** We rewrite the vector \( y \in L_0 \) in more details: for all \( i \in [n] \)
\[
y_i = \frac{z_i}{u_i} \quad (5)
y_{n+1} = s_0 \left( \sum_{j=1}^n a_j z_j - b z_{n+1} \right) \quad (6)
y_{n+1+i} = \frac{z_{n+1+i}}{u_i} \quad (7)
y_{2n+2} = s_1 \left( \sum_{j=1}^n \delta_1 \lambda^{j-1} z_j + \sum_{j=1}^n \delta_j \lambda^{j-1} z_{n+1+j} \right)
+ s_1 \delta \lambda^{n+1} z_{n+1} - s_1 \lambda \gamma z_{2n+2} 
= s_1 \left( \sum_{j=1}^n \delta_j \lambda^{j-1} (z_j + z_{n+1+j}) + \delta \lambda^n z_{n+1} - \lambda \gamma z_{2n+2} \right) \quad (8)
\]

It is easy to see that if \( y_{n+1} \neq 0 \) then \( y_{2n+2} = 0 \), or \( y_{n+1+i} \neq 0 \).

Now assume that \( y_{n+1} \neq 0 \) and \( y_{2n+2} = 0 \). From (5) and (7), vector \( z \) holds the same property, \( z_i + z_{n+1+i} \neq u_i z_{n+1+i} \) or \( \delta_i (z_i + z_{n+1+i}) \neq \delta z_{n+1+i} \). By the assumption \( y_{n+1} = y_{2n+2} = 0 \), we have
\[
\sum_{j=1}^{n} a_j z_j = b z_{n+1} \quad (9)
\]
\[
\sum_{j=1}^{n} \delta_j \lambda^{j-1} (z_j + z_{n+1+j}) + \delta \lambda^n z_{n+1} = \gamma z_{2n+2} \quad (10)
\]

Let \( w = (w_1, \ldots, w_{n+1}) \in \mathbb{Z}^{n+1} \) such that \( w_i = \delta_i (z_i + z_{n+1+i}), \forall i \in [n] \), and \( w_{n+1} = \delta z_{n+1} \). Equation (10) can be rewritten by:
\[
\sum_{j=1}^{n} \lambda^{j-1} w_j + \lambda^n w_{n+1} = \gamma z_{2n+2} \quad (11)
\]

We see that there exists \( i, w_i \neq w_{n+1} \) (due to the known property of \( \delta \)). That is, \( w \) is an integral solution to (11) other than \((z_{2n+2}, \ldots, z_{2n+2})\). By lemma A.1 in the Appendix, \( w \) has at least one component whose absolute value is not less than \( \Theta(\lambda) \). From this fact, we can finish the proof by examining the following cases:

**Case 1:** \( |w_{n+1}| \geq \Theta(\lambda) \)

We have immediately \( |z_{n+1}| \geq \Theta(\lambda) \) and thus \( |z_{n+1}| \geq \Theta(\lambda/\delta) \). From the fact that \( a_i < b, \forall i \in [n] \), every integral solution to (9) has at least one component \( z_h (h \in [n]) \) such that \( |z_h| \geq \Theta(\lambda/\delta) \). This implies \( |y_h| \geq \Theta(\lambda/\delta u_i) \geq \Theta(p) \).

The desired property of \( y \) follows.

**Case 2:** \( |w_i| \geq \Theta(\lambda) \), for some \( i \in [n] \)
This case leads to $\delta |z_i + z_{n+1+i}| \geq \delta_i |z_i + z_{n+1+i}| \geq \Theta(\lambda)$. It is not hard to see that either $|z_i| \geq \Theta(\lambda/\delta)$ or $|z_{n+1+i}| \geq \Theta(\lambda/\delta)$. From (5) and (7), we have either $|y_i| \geq \Theta(\lambda/\delta u_i)$ or $|y_{n+1+i}| \geq \Theta(\lambda/\delta u_i)$. The proof ends.

**Lemma 6.** Suppose that (2) has solutions. If $y^*$ is a shortest vector in $L_0 \setminus S$ for $\ell_2$ norm, then either $(u_1 y^*_1, ..., u_n y^*_n)$ or $-(u_1 y^*_1, ..., u_n y^*_n)$ is a solution to (2).

Proof: A careful observation would reveal that if $\hat{x}$ is a solution to (2), then $\hat{y} = (\frac{\hat{x}}{\|\hat{x}\|}, 0, e - \frac{\hat{x}}{\|\hat{x}\|} 0)$ is a short vector in $L_0 \setminus S$ and $\|\hat{y}\| \leq 1$ (where $e = (1, 1, ..., 1) \in \mathbb{R}^n$). Therefore, we have $\|y^*\| \leq \|\hat{y}\| \leq 1$ due to the hypothesis of $y^*$. Moreover, from lemma 5, $y^* = B_{\theta n}z$ must satisfy (3) and (4). Otherwise there exists $r$ satisfying $|y^*_r| \geq \Theta(p)$, where $p = \min\{s_0, s_1, \lambda/(\delta n u_{\max})\}$ and thus $\|y^*\| \geq \Theta(p) > 1 \geq \|\hat{y}\|$, 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 similarly.) 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 $z_i = z_{n+1+i} = u_i$, $\forall i \in [n]$. From (12) we have $\sum_{j=1}^n a_j z_j = 2b$, contrary to the assumption $\sum_{j=1}^n a_j u_j = 2b$. Hence, there exists $r \in [n]$ such that either $|y^*_r| > 1$ or $|y^*_{n+1+r}| > 1$. This leads to $\|y^*\| > 1 \geq \|\hat{y}\|$, contrarily.

If $z_{n+1} = 1$, then we have $\sum_{j=1}^n a_j z_j = b$ and $y^*_i + y^*_{n+1+i} = (z_i + z_{n+1+i})/u_i = 1$, $\forall i \in [n]$. That is, $\sum_{j=1}^n a_j z_j = b$ and $z_i + z_{n+1+i} = u_i$, $\forall i \in [n]$. From this fact, the first remark is that if there exists $r \in [n]$ such that $z_r < 0$, then $z_{n+1+r} > u_r$ and thus $y^*_{n+1+r} > 1$. This means $\|y^*\| > 1 \geq \|\hat{y}\|$, contrarily. Another remark is that if there exists $r \in [n]$ such that $z_r > u_r$, then $y^*_r > 1$. This means $\|y^*\| > 1 \geq \|\hat{y}\|$, contrarily. Consequently, $0 \leq z_i \leq u_i$, $\forall i \in [n]$. That is, $(z_1, z_2, ..., z_n)$ is a solution to (2). Equivalently, $(u_1 y^*_1, ..., u_n y^*_n)$ is a solution to (2).

By the same argument as above, if $z_{n+1} = -1$, then $-(u_1 y^*_1, ..., u_n y^*_n)$ is a solution to (2). The lemma follows.

From the arguments in the proof of lemma 6, we remark that if $\|y^*\| = n$. Indeed, if $z_{n+1} = 1$ then $y^*_i \geq 0$, $\forall i$, $y^*_{n+1} = y^*_{2n+2} = 0$, $y^*_i + y^*_{n+1+i} = 1$, $\forall i \in [n]$ and thus $\|y^*\| = \sum_{j=1}^{2n+2} |y^*_j| = \sum_{j=1}^{2n+2} y^*_n$. if $z_{n+1} = -1$, then $y^*_i \leq 0$, $\forall i$, $y^*_{n+1} = y^*_{2n+2} = 0$, $y^*_i + y^*_{n+1+i} = -1$, $\forall i \in [n]$ and thus $\|y^*\| = n$. In short, if $y^*$ is shortest in $L_0 \setminus S$ for $\ell_\infty$ norm, then $\|y^*\| = n$. This observation leads to the hope that if a lattice vector $y \in L_0 \setminus S$ has length not greater than $n$ for $\ell_1$ norm, then it gives a solution to (2). Fortunately, it is the case. The proof is similar with the one of lemma 6 and thus we omit it in this extended abstract.

**Lemma 7.** Suppose that (2) has solutions. If $y^*$ is a shortest vector in $L_0 \setminus S$ for $\ell_1$ norm, then either $(u_1 y^*_1, ..., u_n y^*_n)$ or $-(u_1 y^*_1, ..., u_n y^*_n)$ is a solution to (2).

**Proofs of theorem 1 and theorem 2:** Notice that lemma 6 and lemma 7 imply a reduction from BKP to SAP for $\ell_\infty$ norm and a reduction from BKP to SAP for $\ell_1$ norm respectively. Moreover, any parameters including $s_0, s_1, \lambda$ can be chosen to be polynomial time computable. Thus, the reductions are Karp reductions. The dimension of the lattice $L_0$ is $2n + 2$. Hence, it is also the dimension of the new SAP. Consequently, theorem 1 and theorem 2 follow.

### III. BIP UNDER SOME ASSUMPTIONS

We continue to study some classes of BIP, for which there exists algorithms running in probabilistic time $O^{2}(n)$. It is well-known that BKP is NP-hard. Thus, it is likely that there is no polynomial time algorithm for BIP. However, real-life applications expect lower time bound for BIP. This demand motivates us to find more efficient algorithms for it. We believe that BIP is solvable in time $2^{O(n)}$. Nonetheless, here we are only able to show this property for some classes of BIP.

In the previous section, we know that BKP is reducible to SAP for $\ell_\infty$ norm. Thus, the time bound for the new SAP is applicable to the original BKP. It is clear that there are many algorithms for SAP (see [1], [2], [9]). However, the time bound for general SAP is quite large. Fortunately, if SAP has some special properties, then it can be solved in time $2^{O(n)}$ as shown in [2].

**Theorem 8** ([Blömer and Neawe]). Let $L$ be a lattice and $M$ be a subspace of span$(L)$. Assume that there exist absolute constants $c, \varepsilon$ such that the number of $v \in L \setminus M$ satisfying $\|v\| \leq 1 + \varepsilon \lambda_{M}(L)$ is bounded by $2^{c n}$, where $\lambda_{M}(L)$ is the length of the shortest vector in $L \setminus M$. Then there exists an algorithm that solves SAP with probability exponentially close to 1. The running time is $2^{O(n)}$.

This theorem says that if the number of short lattice vectors in $L \setminus M$ is not too large, then we may hope to solve SAP in time $2^{O(n)}$. On the other hand, if the exact solution to SAP is not necessary in some cases, then we can find an approximate solution to SAP in almost same time.

**Theorem 9** ([Blömer and Neawe]). There exists a randomized algorithm that solves SAP with approximation factor $1 + \varepsilon$, $0 < \varepsilon \leq 2$, with probability exponentially close to 1. The running time is $((2 + 1/\varepsilon)^n b)^{O(1)}$, where $b$ is the input size of the problem.

We know that the new SAP from our reduction may not satisfy the assumptions in theorem 8, and that the approximations of it may not give any solution to the original BKP. Thus, we can not expect to improve the time bound for BKP. However, if the original BKP has some special properties, then it can be solved “efficiently”. 
**Theorem 10.** Assume that BKP (2) has a solution \( \hat{x} \) such that \( \frac{\hat{x}}{u_i} \in [\frac{1}{1+\varepsilon}, \frac{1}{1-\varepsilon}] \), \( \forall i \in [n] \), for some absolute constant \( \varepsilon \in (0,1) \). Then there exists a randomized algorithm that solves BKP with probability exponentially close to 1. The running time of the algorithm is \( \text{poly}(\varphi) \cdot 2^{O(n)} \), where \( \varphi \) is the size of the input.

This theorem means that if BKP has a certain solution lying rather closely to the centre of the hyper cube \( [0,u] = \{ x \in \mathbb{R}^n : 0 \leq x \leq u \} \), then it is solvable in probabilistic time \( \text{poly}(\varphi) \cdot 2^{O(n)} \). The same holds for BIP.

**Corollary 11.** Assume that BIP (1) has a solution \( \hat{x} \) such that \( \frac{\hat{x}}{u_i} \in [\frac{1}{1+\varepsilon}, \frac{1}{1-\varepsilon}] \), \( \forall i \in [n] \), for some absolute constant \( \varepsilon \in (0,1) \). Then there exists a randomized algorithm that solves BIP with probability exponentially close to 1. The running time of the algorithm is \( \text{poly}(\varphi) \cdot 2^{O(n)} \), where \( \varphi \) is the size of the input.

**Proof of theorem 10:** It is not hard to see that if \( u_{\text{max}} \) is small, e.g. \( O(1) \), then we can solve BKP by enumerating all possible \( x \in [0,u] \cap \mathbb{Z}^n \). The time needs to do this is \( O(u_{\text{max}}) = 2^{O(n)} \). Thus, without loss of generality, we assume \( u_{\text{max}} \) is not too small. Then the following is straightforward,

\[
1 - \frac{u_{\text{max}} + 1}{u_{\text{max}}(1 + \varepsilon)} < \frac{\varepsilon}{1 + \varepsilon} \tag{14}
\]

On the other hand, due to \( 1 < 1 + 1/u_{\text{max}} \), we have

\[
1 + \varepsilon < \frac{u_{\text{max}} + 1}{u_{\text{max}}(1 + \varepsilon)} \tag{15}
\]

If we combine the assumption of \( \hat{x} \) with (14) and (15), then

\[
1 - \frac{u_{\text{max}} + 1}{u_{\text{max}}(1 + \varepsilon)} < \frac{\hat{x}_i}{u_i} < \frac{u_{\text{max}} + 1}{u_{\text{max}}(1 + \varepsilon)} \quad \forall i \in [n] \tag{16}
\]

Applying the reduction from BKP to SAP as in previous section, we obtain the new SAP. Let the new SAP be the centre of our concentration from now on.

Notice that the vector \( \hat{y} = (\frac{\hat{x}}{u}, 0, e - \frac{\hat{x}}{u}, 0) \) is in \( L_0 \setminus S \), and that \( \|\hat{y}\| = \max(\|\frac{\hat{x}}{u}\|, \|e - \frac{\hat{x}}{u}\|) \). Combining with (16) yields,

\[
\|\hat{y}\| < \frac{u_{\text{max}} + 1}{u_{\text{max}}(1 + \varepsilon)} \tag{17}
\]

Consider any vector \( y = B_0 z \in L_0 \setminus S \) such that \( (u_1 x_1, ..., u_n x_n) \) is not a solution to (2). From the proof of lemma 6, there exists \( r \) such that \( |y_r| > 1 \). If \( y \) does not satisfy (3) and (4) then, due to lemma 5, \( \|y\| \geq 2 > 1 + 1/u_{\text{max}} \). Otherwise, \( r \neq n + 1 \) and \( r \neq 2n + 2 \). This means \( |y_r| = |z_r|/u_{\text{mod}(n+1)} | > 1 \), or equivalently, \( |z_r| > u_{\text{mod}(n+1)} \). From the fact that \( z_r \in \mathbb{Z} \), we have immediately \( |z_r| \geq u_{\text{mod}(n+1)} + 1 \). These observations yield

\[
|y_r| \geq \frac{u_{\text{mod}(n+1)} + 1}{u_{\text{mod}(n+1)}} \geq 1 + \frac{1}{u_{\text{mod}(n+1)}} \geq 1 + \frac{1}{u_{\text{max}}}
\]

Therefore,

\[
\|y\| \geq 1 + \frac{1}{u_{\text{max}}} \tag{18}
\]

\(^1\)by the arguments in the last two paragraphs of the proof of lemma 6, if \( |y_r| \leq 1, \forall r \), then \( y \) gives a solution to (2).

Combining (17) with (18) leads to \( \|\hat{y}\| < \|y\|/(1 + \varepsilon) \), or equivalently, \( \|y\| > (1 + \varepsilon) \|\hat{y}\| \).

Let \( y^* \) be a solution to the new SAP (the shortest vector in \( L_0 \setminus S \)). Then, \( \|y^*\| \leq \|y\| \). That is, \( \|y\| > (1 + \varepsilon) \|y^*\| \), for all \( y \in L_0 \setminus S \) such that \( (u_1 |y_1|, ..., u_n |y_n|) \) is not a solution to (2). This implies that an approximate solution to the new SAP within the factor \( 1 + \varepsilon \) is enough to give a solution to (2). In other word, BKP (2) is reducible to the problem of approximating SAP within a factor of \( 1 + \varepsilon \).

If we know \( \varepsilon \), then the new SAP can be approximated by SAP solver in [2] to find a solution to (2) and the running time is \( \text{poly}(\varphi) \cdot 2^{O(n)} \) (due to theorem 9). Conversely, we can find a solution to (2) as follows: repeat approximating the new SAP (by SAP solver in [2]) correspondingly with approximation factors \( 1 + \frac{1}{\varepsilon}, 1 + \frac{1}{\varepsilon}, ... \) until we find a solution to (2) from approximate solution of the new SAP. It is not hard to see that the number of repeat loops, those we need to solve (2), is at most \( \log_2(1/\varepsilon) \). Thus, the total running time is \( T = T \cdot \log_2(1/\varepsilon) \), where \( T \) is the running time of a repeat loop. Note that \( \varepsilon \) is a certain absolute constant. Hence, we have \( T = O(T) \). The claim of the theorem is clear.

Another property of BKP also allows the possibility of solving it in time \( 2^{O(n)} \).

**Theorem 12.** Assume that there exist absolute constant \( c, \varepsilon \) such that the number of integral vectors \( \hat{x} \), in the hyper plane \( P = \{ x \in \mathbb{R}^n : a_1 x_1 + \cdots + a_n x_n = b \} \), satisfying \( \|\hat{x}/u\| < 1 + \varepsilon \) is bounded by \( 2^{cn} \). Then there exists a randomized algorithm that solves BKP (2) with probability exponentially close to 1. The running time of the algorithm is \( 2^{O(n)} \).

This theorem says that if integral vectors of the hyper plane of BKP spread out sparsely, then we can hope to solve BKP “efficiently”. This is also the case for BIP.

**Corollary 13.** Assume that there exist absolute constant \( c, \varepsilon \) such that the number of integral vectors \( \hat{x} \), in the hyper plane \( P = \{ x \in \mathbb{R}^n : A x = b \} \), satisfying \( \|\hat{x}/u\| < 1 + \varepsilon \) is bounded by \( 2^{cn} \). Then there exists a randomized algorithm that solves BIP (1) with probability exponentially close to 1. The running time of the algorithm is \( 2^{O(n)} \).

**Sketch of the proof of theorem 12:**

If we chose the parameters \( s_0, s_1, \delta \) large enough, then the resulting lattice \( L_0 \) of the reduction in previous section has the following properties.

For each lattice vector \( y \in L_0 \setminus S \) of length at most \( 1 + \varepsilon \), there exists unique integral vector \( \hat{x} \in P \) such that \( \|\hat{x}/u\| < 1 + \varepsilon \) and either \( \hat{x}/u > -\hat{x}/u \) is the head part of \( y \). This property is easily derived from lemma 6. Conversely, for each vector \( \hat{x} \in P \) satisfying \( \|\hat{x}/u\| < 1 + \varepsilon \), there are at most two lattice vectors in \( L_0 \setminus S \) of length at most \( 1 + \varepsilon (\|\hat{x}/u\| < 1 + \varepsilon \) and \( -\|\hat{x}/u\| < 1 + \varepsilon \) ). Moreover, the number of such \( \hat{x} \) is bounded by \( 2^{cn} \) (due to the hypothesis).

Thus, the number of lattice vector in \( L_0 \setminus S \) of length at most \( 1 + \varepsilon \) is bounded by \( 2^{cn+1} \). Remember that the shortest vector in \( L_0 \setminus S \) has length at most 1 (from the proof of lemma 6). From these observations, we may find a shortest vector in \( L_0 \setminus S \) in probabilistic time \( 2^{O(n)} \) (due to theorem 8). That is, BKP is solvable in probabilistic time \( 2^{O(n)} \).
CONCLUSION

We obtain the new time bound for BIP. Nonetheless, the time bound is still large. We believe that it can be improved substantially in both deterministic and probabilistic sense. An open question is that if there is a randomized algorithm for BIP running in time $2^{O(n)}$? Another one is that if there is a deterministic algorithm for BIP running in time $2^{O(n)}$? It seems that the first question is easier than the second one due to the fact that many lattice problems have this time bound, and that our reduction is an evidence.

ACKNOWLEDGMENT

The author would like to sincerely thank Daniele Micciancio for sharing his result in [9] and for many helpful discussions on lattice problems. Thanks are also to Jin-Yi Cai for pointing me to the reference [2]. The author also thank anonymous reviewers for many helpful comments on the paper.

APPENDIX

The following lemma is an observation in number theory and was proven in [6].

Lemma A.1. Let $n, \lambda, t$ be integers such that $n$ is not a constant, $|\lambda| \geq \Theta(n)$, $\gamma = 1 + \lambda + \cdots + \lambda^n$. Then the Diophantine equation

$$x_1 + \lambda x_2 + \cdots + \lambda^n x_{n+1} = \gamma t$$

has a solution of the form $(t, t, \ldots, t)$ and every other integral solution must have at least one component whose absolute value is not less than $\Theta(\lambda)$.

Proof: We can rewrite equation (19) as follow:

$$(x_1 - t) + \lambda (x_2 - t) + \cdots + \lambda^n (x_{n+1} - t) = 0$$

All integral solutions to (20) satisfy:

$$x_1 - t = \lambda t_1$$

$$x_2 - t = \lambda t_2$$

$$\vdots$$

$$x_n - t = \lambda t_n$$

$$x_{n+1} - t = -t$$

where $t_1, \ldots, t_n$ are integers.

It is clear that if $t_1 = \cdots = t_n = 0$ then $(t, \ldots, t) \in \mathbb{Z}^{n+1}$ is a solution to (19).

Now, we consider any other integral solution $x$ to (19). There exist integers $t_1, \ldots, t_n$ satisfying (23) and at least one $t_k$ such that $t_k \neq 0$.

Let $t_1$ be the first non-zero element in the sequence $t_1, \ldots, t_n$ and $t_r$ be the last one. To observe the desired property of $x$ easily, we should examine the following cases:

Case 1: $l > 1$

With this assumption, we have immediately $x_1 = t, x_2 = t + \lambda t_2$. It is easy to see that if $|x_1| < \Theta(\lambda)$ then $x_1 = \Theta(\lambda t_1)$; if $|x_1| > \Theta(\lambda)$ then $x_1 = \Theta(\lambda t_1)$ and so $x_1 = \Theta(\lambda t_1)$. In both cases, our claim is true.

Case 2: $r < n$

This case can be proved similarly to Case 1.

Case 3: $l = 1, r = n$

We conclude that $CVP$ for $\ell_2$ norm is solvable in deterministic time $\Theta(n)$.

This means that we can adapt Kannan’s algorithm to solve CVP for $\ell_1$ norm with the caution that the bound should be double (2A instead of A as chosen in [7]). Using the analysis of Hanrot and Stehlé [7], we conclude that CVP for $\ell_1$ norm is solvable in deterministic time $\text{poly}(\psi) \cdot 2^{O(n)} \cdot n^{n/2+o(n)} = \text{poly}(\psi) \cdot n^{n/2+o(n)}$.

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

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

Find a vector \( x \in \mathbb{Z}^{n+1} \) satisfying

\[
\begin{align*}
\sum_{j=1}^{n+1} a'_j x_j &= b' \\
0 &\leq x \leq u \\
0 &\leq x_{n+1} \leq 1
\end{align*}
\] (24)

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

It is not hard to see that solving the new BKP (24) is equivalent to solving (2). Indeed, if \((x_1, \ldots, x_n, 1)\) is a solution to (24), then \(a_1 x_1 + \cdots + a_n x_n = b\) and thus \((x_1, \ldots, x_n)\) is a solution to (2). Moreover, \((x_1, \ldots, x_n, 0)\) cannot be a solution to (24) 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 (24) has a solution \((x_1, \ldots, x_n, x_{n+1})\) if and only if (2) has a solution \((x_1, \ldots, x_n)\).

Note that (24) satisfies \(\sum_{j=1}^{n+1} a'_j u'_j \neq 2b',\) where \(u'_j = u_j, \forall i \leq n\) and \(u'_{n+1} = 1.\) This means the new BKP satisfies our desired assumption.

REFERENCES

[1] J. Blömer, Closest vectors, successive minima and dual HKZ-bases of lattices, In Proc. of the 27th Intern. Colloq. on Automata, Languages and Programming - ICALP 2000, pages 248-259, 2000.

[2] J. Blömer, S. Naewe, Sampling methods for shortest vectors, closest vectors and successive minima, In Proc. of the 34th Intern. Colloq. on Automata, Languages and Programming, LNCS, vol. 4596, pages 65-77, 2007.

[3] F. Eisenbrand, Fast integer programming in fixed dimension, In Proc. of 11th annual European Symposium on Algorithms, Springer-verlag, vol. 2832, pages 196-207, 2003.

[4] R. Kannan, Polynomial-Time Aggregation of Integer Programming Problems, Journal of ACM, vol. 30, pages 133-145, 1983.

[5] R. Kannan, Minkowski’s convex body theorem and integer programming, Mathematics of Operations Research, vol. 12, pages 415-440, 1987.

[6] T. Q. Khoát, Reductions from Integer Programming to Lattice Problems, unpublished.

[7] G. Hanrot, D. Stehlé, Improved Analysis of Kannán’s Shortest Lattice Vector Algorithm, Advanced in Cryptology - CRYPTO07, LNCS, Springer-verlag, vol. 4622, pages 170-186, 2007.

[8] H. W. Lenstra, Integer programming with a fixed number of variables, Mathematics of Operations Research, vol. 8, pages 538-546, 1983.

[9] D. Micciancio, Efficient reductions among lattice problems, In Proc. of the 19th annual ACM-SIAM Symposium on Discrete Algorithms, pages 84-93, 2008.

[10] A. Schrijver, Theory of Linear and Integer Programming, John Wiley, Chichester, 1999.