Hardness of Approximating the Shortest Vector Problem in High L_p Norms

Subhash Khot
email : khot@cs.princeton.edu

April 6, 2003

Abstract

We show that for every $\epsilon > 0$, there is a constant $p(\epsilon)$ such that for all integers $p \geq p(\epsilon)$, it is NP-hard to approximate the Shortest Vector Problem in L_p norm within factor $p^{1-\epsilon}$ under randomized reductions. For large values of p, this improves the factor $2^{1/p} - \delta$ hardness shown by Micciancio [27].

1 Introduction

An n-dimensional lattice $\mathcal{L}$ is a set of vectors $\{\sum_{i=1}^{n} a_i v_i \mid a_i \in \mathbf{Z}\}$ where $v_1, v_2, \ldots, v_n \in \mathbf{R}^m$ is a set of independent vectors called the basis for the lattice. The same lattice could have many bases. Given a basis for an n-dimensional lattice, the Shortest Vector Problem asks for the shortest non-zero vector in the lattice. The length of the vectors can be measured in any L_p norm ($p \geq 1$) and the corresponding problem is denoted by SVP_p. This problem has a beautiful history and we present some of the results below. For a more comprehensive list of references, we refer to Micciancio's PhD thesis [28] and an expository article by Kumar and Sivakumar [20].

The Shortest Vector Problem has been studied since the time of Gauss ([14], 1801) who gave an algorithm for SVP_2 in 2-dimensions. The general problem for arbitrary dimensions was formulated by Dirichlet in 1842. The theory of Geometry of Numbers by Minkowski [29] deals with the existence of shortest non-zero vectors in lattices. In a celebrated result, Lenstra, Lenstra and Lovász [25] gave a polynomial time algorithm for approximating SVP_2 within factor $2^{n/2}$. This algorithm has numerous applications, e.g. factoring rational polynomials [25], breaking knapsack-based codes [22], checking the solvability by radicals [23] and integer programming in a fixed number of variables ([25], [24], [18]). Schnorr [32] improved the approximation factor to $2^{\delta n}$ for any $\delta > 0$. Since all L_p norms are within factor $\sqrt{n}$ from the L_2 norm, these algorithms give similar approximations for SVP_p for any p. It is a major open problem whether SVP has polynomial factor approximations that run in polynomial time. Exact computation of SVP_2 in exponential time is also investigated, for instance [19], [4].

In 1981, van Emde Boas [33] proved that SVP_∞ is NP-hard and conjectured that the same is true for any L_p norm. However proving NP-hardness for any finite p (in particular $p = 2$) was an embarrassing open problem for long time. A breakthrough result by Ajtai [1] in 1998 finally showed that SVP_2 is NP-hard under randomized reductions. Cai and Nerurkar [10] improved Ajtai's result to a hardness of approximation result showing a hardness factor of $(1 + \frac{1}{n^\delta})$. Another breakthrough by Micciancio [27] showed that SVP_p is hard to approximate within some constant factor, specifically factor $2^{1/p} - \delta$ for every $\delta > 0$.

Showing hardness of approximation results for SVP was greatly motivated by Ajtai's discovery [2] of worst-case to average-case hardness and subsequent construction of lattice-based public key cryptosystem by Ajtai and Dwork [3]. Ajtai showed that if there is a randomized polynomial time algorithm for solving SVP_2 on a non-negligible fraction of lattices from a certain natural class of lattices, then there is a randomized polynomial time algorithm for approximating SVP_2 on *every* instance within some polynomial factor n^c. Ajtai-Dwork's work gave hope, for the first time, that cryptography could be based on the (conjectured) worst-case hardness of a problem. Their work implies that if n^c-approximation to SVP_2 is hard, then one can construct a secure cryptosystem. The constant c was noted to be 19 in [9], and brought down to $9 + \delta$ by Cai and Nerurkar [11] and then to $4 + \delta$ by Cai [9]. Recently, Regev [31] gave an alternate construction of a public key cryptosystem based on $n^{1.5}$-hardness of SVP_2 (technically speaking a variant called unique-SVP_2) ! Unfortunately, there are barriers to showing such strong hardness results. In fact, showing factor n NP-hardness would imply that NP = coNP [21] and showing factor $\sqrt{n/O(\log n)}$ NP-hardness would imply that coNP $\subseteq$ AM [15].

Another related problem that has received much attention is the Closest Vector Problem (denoted CVP_p) where given a lattice and a vector y, the problem is to find the lattice vector that is closest to y. In spite of the apparent similarity between SVP and CVP, they turn out to be quite different problems. Indeed, CVP_p was shown to be NP-hard for all $p \geq 1$ by van Emde Boas [33]. Arora, Babai, Swedyk, and Stern [5] used the PCP machinery to show that approximating CVP_p (for all $p \geq 1$) within factor $2^{\log^{1-\delta} n}$ is hard unless NP $\subseteq$ DTIME($n^{poly(\log n)}$). This was improved to a NP-hardness result by Dinur, Kindler, and Safra [13] (their result even gives a subconstant value of δ, i.e. $\delta = (\log \log n)^{-c}$ for any $c < \frac{1}{2}$). Incidently, SVP_∞ seems to behave very much like CVP_∞, Dinur [12] shows factor $n^{1/\log\log n}$ NP-hardness for both these problems. Thus for SVP, the cases $p=\infty$ and $p<\infty$ seem to be qualitatively different.

Result in this paper : In this paper, we obtain an improved hardness result for SVP_p for large (but finite) values of p. Specifically we show that

Theorem 1.1 *For every $\epsilon > 0$, there is a constant $p(\epsilon)$ such that for all integers $p \geq p(\epsilon)$, it is impossible to approximate SVP_p within factor $p^{1-\epsilon}$ in polynomial time unless NP = ZPP.*

This improves the hardness factor $2^{1/p}-\delta$ by Micciancio [27] for all large values of p. The result however is only asymptotic and says nothing about small values of p. The value $p(\epsilon)$ depends on some non-explicit constants in Raz's Parallel Repetition Theorem [30].

Significance of the result : Apart from the improved hardness factor, we believe that the result is significant in the following aspects :

1. Very strong hardness results are known for SVP_∞, so it is reasonable to expect that the hardness result gets better as p grows. Our result ($p^{1-\epsilon}$) supports this intuition, whereas Micciancio's result ($2^{1/p}-\delta$) goes in the other direction.

2. Our result is a direct reduction from 2-Prover Games (or equivalently the Label Cover problem introduced by Arora et al [5]). Micciancio's reduction can be seen as an indirect reduction from 2-Prover Games. Lund and Yannakakis [26] reduce 2-Prover Games to the Set Cover problem. Arora et al [5] reduce Set Cover to CVP_p for every finite value of p. Micciancio reduces CVP_p to SVP_p by constructing a sophisticated lattice as a gadget. Constructing this lattice requires many ideas from Ajtai's reduction. In contrast to our and Micciancio's reductions, Ajtai uses a reduction from a much more elementary problem, namely Subset Sum.

3. Our result holds under the assumption that NP $\not\subseteq$ ZPP. Ajtai's [1] and Micciancio's [27] results require a (stronger) assumption that NP $\not\subseteq$ BPP. Our reduction is considerably simpler than the earlier reductions, and it might be easier (if possible at all) to derandomize.

On the flip-side, our result doesn't apply for any small explicit value of p. Definitely, the most interesting case is L_2 norm, since Ajtai's worst-case to average-case reduction is based on hardness of SVP_2. However, we think that our result helps in a better understanding of the Shortest Vector Problem. Considering that even NP-hardness wasn't known for any finite L_p norm till 1998, it is interesting that this paper gives a simple and straightforward proof.

Techniques and Overview : We prove Theorem 1.1 via a reduction from the Label Cover problem which is defined in Section 2.2. The reduction is quite straightforward and Section 3 gives the basic idea of the reduction. Section 4 gives the full reduction.

One technical ingredient in the reduction is a randomized construction of a set X of m integers with the following properties. Let $\delta > 0$ be given and think of m as growing.

- For any set $Y \subseteq X$, $|Y| = \delta m$, the integers in Y have a non-zero $\{0,1,-1\}$ linear combination that vanishes.
- For any set $Z \subseteq X$, $|Z| < \frac{\delta m}{20\log(1/\delta)}$, a non-zero $\{0,1,-1\}$ linear combination of integers in Z cannot vanish.

Choosing m random integers from the range $[1,2,\ldots,2^{\delta m/2}]$ satisfies these properties with high probability. This is the only place where our reduction is randomized. This "trick" is based on the Pigeon-Hole principle and was mentioned to us by Sanjeev Arora. We use a somewhat stronger construction in Section 4. The basic idea in Section 3 *does not* need this construction.

2 Preliminaries

We first define the Shortest Vector Problem in L_p norm. It is defined in a different (but equivalent) manner, without any reference to lattices.

2.1 Problem Definition

The problem SVP_p is defined as follows : Given a vector $\mathbf{x}$ of integer variables $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ and linear forms $\{\phi_1, \phi_2, \ldots, \phi_m\}$ where

$$\phi_i = \sum_{j=1}^{n} a_{ij} x_j \qquad a_{ij} \in \mathbf{R}$$

The goal is to find a non-zero integer vector $\mathbf{x}$ which minimizes the following objective function :

$$\text{OBJ} = \sum_{i=1}^{m} |\phi_i(\mathbf{x})|^p$$

Remarks : (1) Think of the basis vectors in the lattice as the columns of the matrix $\{a_{ij}\}$ and think of the integers variables x_i's as the (unknown) coefficients when the shortest vector is written as an integer linear combination of the basis vectors. (2) We actually want to minimize $\text{OBJ}^{1/p}$. In order to show a factor k-hardness for SVP_p, it suffices to show a factor k^p-hardness for the above objective function.

The reduction is from a problem called "Label Cover" which we define next. This problem was introduced by Arora et al [5] and can be equivalently viewed as the problem of finding good prover strategies in a 2-prover 1-round game.

2.2 The Label Cover Problem

The Label Cover problem is defined as follows : We are given a bipartite graph $G = (V, W, E)$ with left side vertices V, right side vertices W and a set of edges E. The goal is to assign one "label" to every vertex, where the vertices in W are required to receive a label from set $[R]$ and the vertices in V are required to receive a label from set $[S]$. Thus, a labeling A is just a map $A : W \mapsto [R], A : V \mapsto [S]$. The labeling is supposed to satisfy certain constraints given by maps $\pi_{v,w} : [R] \mapsto [S]$. There is one such map for every edge $(v, w) \in E$. A labeling A "satisfies" an edge (v, w), if

$$\pi_{v,w}(A(w)) = A(v)$$

The optimum OPT of the label cover problem is defined to be the maximum fraction of edges satisfied by any labeling. Let $n = |V|,\ m = |W|$. Let D be the degree of every vertex in V. Think of n, m as growing and $m \gg n$. Think of R, S, D as constants and $R \gg D \gg S$.

The following theorem can be obtained by combining the PCP Theorem ([7], [6]) with Raz's Parallel Repetition Theorem [30]. This theorem is the starting point for most of the recent PCP constructions and hardness results (e.g. [17], [16]).

Theorem 2.1 *There exists an absolute constant $\gamma > 0$ such that for every sufficiently large integer R, it is NP-hard to distinguish whether a Label Cover problem specified as $(G(V, W, E), n, m, \{\pi_{v,w}\}, [R], [S], D)$ has* OPT$= 1$ *or* OPT $\leq \frac{1}{R^\gamma}$.

Remark : It can be ensured that for an integer parameter u, $R = 6^u, S = 3^u, D = 5^u, m = (5/2)^u n$. Also, for every edge (v, w), the map $\pi_{v,w} : [R] \mapsto [S]$ is "regular", meaning for every $j \in [S]$, there are exactly R/S elements in $[R]$ that are mapped to j. We omit how to construct Label Cover instances with these parameters. The precise values of the parameters are not very crucial for this paper.

Remark : The intended hardness factor we wish to achieve will be $k^{1-\epsilon}$. We will have $k = R^{\gamma/20}$ and $p = O(k)$.

3 The Basic Idea in the Reduction

Here we describe the basic idea of the reduction. One needs to fix a lot of technical details later and we present a complete reduction in Section 4.

Theorem 2.1 gives an instance of the Label Cover problem specified as $(G(V, W, E), n, m, \{\pi_{v,w}\}, [R], [S], D)$. The integer variables of the SVP$_p$ will be

$$\{x_{w,i} \mid w \in W, i \in [R]\}$$

For a fixed w, let $B(w)$ be the "block" of variables defined as

$$B(w) = \{x_{w,i} \mid i \in [R]\}$$

For a vertex $v \in V$, let $N(v) \subseteq W$ denote the set of neighbors of v with $|N(v)| = D$. There will be one linear form in SVP$_p$ for every $v \in V$ and every sequence $(w_1, w_2, \ldots, w_S)$ where $w_j \in N(v)$ for $1 \leq j \leq S$. The linear form is sum of R variables, with R/S variables each from the block $B(w_j)$. The linear form is defined to be :

$$\sum_{j=1}^{S} \sum_{i \in [R] : \pi_{v,w_j}(i) = j} x_{w_j,i} \tag{1}$$

Note that the total number of linear forms is nD^S.

3.1 Completeness

For completeness, we will show that if there is a labeling A for the label cover problem satisfying every edge (i.e. OPT = 1), then the problem SVP$_p$ has a solution with the objective function OBJ $= nD^S$. Let $A : W \mapsto [R]$, $A : V \mapsto [S]$ be such a labeling. For every edge (v, w) in the Label Cover instance, we have $\pi_{v,w}(A(w)) = A(v)$.

Define a solution as

$$x_{w,i} = \begin{cases} 1 & \text{if} \quad A(w) = i \\ 0 & \text{otherwise} \end{cases}$$

Note that there are nD^S linear forms of type (1). We will show that each of these linear forms equals 1. Note that out of all the variables in the linear form (1), exactly one equals 1 and the rest are all 0. The non-zero variable is $x_{w_j,i}$ for which $j = A(v), i = A(w_j)$ (Verify ! This is the crux of the reduction).

3.2 Soundness

We wish to show that if OPT $< 1/R^\gamma$ for the Label Cover problem, then for any non-zero integer vector $\mathbf{x} = \{x_{w,i}\}$, the objective function OBJ is at least $\frac{1}{2} nD^{S-k} k^p$. We will show this only for a restricted class of vectors, i.e. vectors $\mathbf{x} = \{x_{w,i}\}$ which "arise" out of labelings $A : W \mapsto [R]$. Eventually we want this to work for **every** non-zero vector $\{x_{w,i}\}$. This involves a lot of technical details and we present the full reduction in Section 4.

So lets assume $A : W \mapsto [R]$ is an assignment and $\{x_{w,i}\}$ is the corresponding vector, i.e.

$$x_{w,i} = \begin{cases} 1 & \text{if} \quad A(w) = i \\ 0 & \text{otherwise} \end{cases}$$

For every vertex $v \in V$, define a set of labels $\Psi(v) \subseteq [S]$ as follows :

$$\Psi(v) = \{\pi_{v,w}(A(w)) \mid w \in N(v) \}$$

Lemma 3.1 *For at least half the vertices in V, $|\Psi(v)| \geq k$ (call such vertices "good").*

Proof: Assume on the contrary that half the vertices in V have $|\Psi(v)| \leq k$. Define label for vertex v to be a random element of $\Psi(v)$. Note that for every $w \in N(v)$, $\pi_{v,w}(A(w)) \in \Psi(v)$. Therefore with probability $1/k$, the edge (v, w) is satisfied by this random labeling to v. Hence there exists a labeling for the Label Cover problem that satisfies at least a fraction $\frac{1}{2k}$ of the edges. This is a contradiction if $\frac{1}{2k} > \frac{1}{R^\gamma}$. ■

For every good vertex v, assume w.l.o.g. that $\{1, 2, \ldots, k\} \subseteq \Psi(v)$. Thus for every $1 \leq j \leq k$, there exists $w_j^* \in N(v)$ such that $\pi_{v,w_j^*}(A(w_j^*)) = j$. Hence for any sequence $(w_1^*, w_2^*, \ldots, w_k^*, w_{k+1}, \ldots, w_S)$ where $w_{k+1}, \ldots, w_S \in N(v)$ are arbitrary, the linear form (1) is at least equal to k. This contributes k^p to the objective function OBJ.

Half the vertices are good, hence OBJ $\geq \frac{n}{2} D^{S-k} k^p$.

3.3 Hardness Factor

Note that the objective function is nD^S in completeness case and at least $\frac{1}{2} nD^{S-k} k^p$ in the soundness case. Choose p such that

$$\frac{1}{2} nD^{S-k} k^p \; > \; nD^S k^{(1-\epsilon)p}$$

Thus there is a penalty of a factor $k^{(1-\epsilon)p}$ in the soundness case. Noting that $D \leq R, k = R^{\gamma/20}$, we see that it suffices to take $p = \frac{20k}{\epsilon\gamma}$. This gives hardness factor of $k^{1-\epsilon}$ or $p^{1-\epsilon'}$ as desired. This completes the basic idea in the reduction. We give the full reduction in the next section.

4 Full Reduction

The set of integer variables is the same, namely

$$\mathbf{x} = \{x_{w,i} \mid w \in W, i \in [R]\}$$

There will be 4 types of linear forms. These forms are supposed to "handle" different types of non-zero vectors $\mathbf{x}$ in the soundness case. The exact role of these linear forms will be clear as we go along.

Type-1 linear forms :

$$\forall\, w \in W,\ \forall i \ \in [R], \qquad x_{w,i}$$

Thus the Type-1 linear forms are just the variables themselves.

Type-2 linear form :

$$\sum_{w \in W, i \in [R]} a_{w,i} x_{w,i}$$

Thus there is only one Type-2 linear form. The coefficients $a_{w,i}$ in this linear form are randomly chosen integers from the range $[1, 2, ..., 2^{m/2}]$.

Type-3 linear forms :

$$\forall\, w \in W, \qquad \sum_{i=1}^{R} \pm x_{w,i}$$

Thus there are 2^R Type-3 linear forms for every $w \in W$. There are R variables in every form and there is one linear form for every choice of $+/-$ sign.

Type-4 linear forms :

$$\forall\, v \in V,\ \forall w_1, w_2, \ldots, w_S \in N(v), \qquad \sum_{j=1}^{S} \sum_{i \in [R]:\pi_{v,w_j}(i)=j} \pm x_{w_j,i}$$

There are $2^R D^S$ linear forms for every $v \in V$. These are essentially the linear forms used in Section 3, except that now we take all the $+/-$ combinations. Note that there are R variables in each form.

In the completeness case, Type-2 form will contribute 0. Type-1, Type-3 and Type-4 will contribute (at most) $m, 2^R m$ and $2^R n D^S$ respectively. We multiply the Type-1, Type-3 and Type-4 forms by appropriate quantities C_1, C_3, C_4 so that they contribute equally towards the objective function. More precisely, C_1, C_3, C_4 are chosen so that

$$C_1^p\, m = C_3^p\, 2^R m = C_4^p\, 2^R n D^S$$

In the soundness case, we will show that any non-zero vector $\mathbf{x} = \{x_{w,i}\}$, either produces a non-zero value in Type-2 form or it pays a penalty of factor $k^{(1-\epsilon)p}$ for at least one of the remaining three types.

We multiply Type-2 form by a huge quantity. Thus we incur a huge penalty unless the Type-2 form vanishes (it will vanish in the completeness case, so we are fine).

4.1 Completeness

In the completeness case, let $A : W \mapsto [R], A : V \mapsto [S]$ be a correct labeling. Let

$$x_{w,i} = \begin{cases} 0, 1 \text{ or } -1 & \text{if} \quad A(w) = i \\ 0 \quad \text{otherwise} \end{cases}$$

The choice of $0, 1, -1$ for the variables $\{x_{w,A(w)}\}$ is made such that the Type-2 form vanishes. We use the Pigeon-Hole principle (suggested by Sanjeev Arora). Consider the set of m variables $\{x_{w,A(w)}\}$, and the corresponding coefficients $a_{w,A(w)}$ in the Type-2 form. Consider the 2^m different sums for all subsets of these m coefficients. These sums take integer values in the range $[1, 2, 3, \ldots, m2^{m/2}]$. Hence sums for two distinct subsets must be equal, which gives a vanishing $\{0, 1, -1\}$ linear combination of the integers $a_{w,A(w)}$.

Now we look at the remaining 3 types.

- For Type-1 forms, note that at most m of the variables are ± 1, the rest are 0. So we get contribution of at most m.
- For Type-3 forms, note that for every w, there is at most one variable $x_{w,i}$ that is ± 1. Thus we get contribution of at most $2^R m$.
- Every Type-4 form is ± 1 as seen in Section 3.1 and it might even be 0 since we "turn off" some of the variables $x_{w,A(w)}$ to 0. Thus the Type-4 forms contribute at most $2^R n D^S$.

5 Soundness

The crux of the soundness analysis is as in Section 3.2. However, we have to handle cases when $x_{w,i}$'s are negative, or very few of them are non-zero. We do this in several stages.

For a vector $\mathbf{x} = \{x_{w,i}\}$, let $\#\mathbf{x}$ denote the number of variables (or coordinates) that are non-zero. For a block of variables $B(w)$, let $\#B(w)$ denote the number of non-zero variables in this block.

5.1 Handling $\mathbf{x}$ with $\#\mathbf{x} \leq \frac{m}{20 \log R}$ and $\|\mathbf{x}\|_1 \geq mR$

Such $\mathbf{x}$ are handled by Type-1 forms. Note that when at most $\frac{m}{20 \log R}$ coordinates are non-zero and the L_1 norm is at least mR, then L_p norm is minimized when all the non-zero coordinates are equal to $\frac{mR}{m/(20 \log R)}$. Hence

$$\sum_{w,i} |x_{w,i}|^p \geq (20R \log R)^p \frac{m}{20 \log R} \geq R^{p/2} m \geq k^p m$$

Note that the contribution of Type-1 forms in the completeness case is at most m. Thus we get a penalty of factor k^p for the Type-1 forms.

5.2 Handling $\mathbf{x}$ with $\#\mathbf{x} \leq \frac{m}{20 \log R}$ and $\|\mathbf{x}\|_1 \leq mR$

These are handled by Type-2 linear form. The coefficients in this linear form are randomly chosen integers from the range $[1, 2, 3, \ldots, 2^{m/2}]$. Hence for any non-zero vector $\mathbf{x}$, the probability that the Type-2 form vanishes is at most $\frac{1}{2^{m/2}}$. We count the number of vectors $\mathbf{x}$ such that $\#\mathbf{x} \leq \frac{m}{20 \log R}$ and $\|\mathbf{x}\|_1 \leq mR$. We show that there aren't too many of them and we can take a union bound.

Number of such $\mathbf{x}$'s can be bounded by

$$\binom{mR}{\frac{m}{20 \log R}} 2^{m/(20 \log R)} mR \binom{2mR}{\frac{m}{20 \log R}} \leq 2^{2m/5} \ll 2^{m/2}$$

Where one uses the fact that

$$\binom{M}{\delta M} \leq 2^{H(\delta)M} \leq 2^{2\delta \log(1/\delta)M}$$

5.3 Avoiding the Problem of Negative or Large Values

In general the variables $x_{w,i}$ could be positive or negative and could take large integer values. We will now show that one can as well assume that they take values only 0 or 1. This is done by averaging over all the $+/-$ linear combinations and this is the reason why Type-3 and Type-4 forms appear with all possible $+/-$ combinations.

Lemma 5.1 *Let $x_1, x_2, \ldots, x_k$ be k non-zero integers. Let $a_i \in \{1, -1\}$ be chosen randomly. Assume p to be an even integer. Then*

$$E_{a_1, a_2, \ldots, a_k} \left[|\sum_{i=1}^{k} a_i x_i|^p \right] \geq \max\{k^{p-k}, k^{p/2}\}$$

Proof: We can expand out the product $(\sum_{i=1}^{k} a_i x_i)^p$ and take the expectation of each term separately. For the terms in which some $a_i x_i$ occurs to an odd power, the expectation is zero. For the remaining terms, the expectation is at least 1. Thus the expectation is at least the number of terms such that every $a_i x_i$ occurs to an even power. In other words, we want to count the number of functions $f : [p] \mapsto [k]$ such that every $j \in [k]$ has even number of pre-images. Considering functions where $f(1) = f(2), f(3) = f(4), \ldots, f(p-1) = f(p)$, we get a bound of $k^{p/2}$. Another way is to take an arbitrary function $g : [p-k] \mapsto [k]$ and then "extend" it to a function $\tilde{g} : [p] \mapsto [k]$ where the values $\tilde{g}(p-k+1), \tilde{g}(p-k+2), \ldots, \tilde{g}(p)$ are chosen to make sure that for $\tilde{g}$, every value has an even number of pre-images. This gives a bound of k^{p-k}. ∎

The Lemma 5.1 implies that for any set of R variables, with at least k of them non-zero, when summed over all $+/-$ combinations, the contribution to the objective function OBJ is at least $2^R \max\{k^{p-k}, k^{p/2}\}$.

5.4 Handling One More Annoying Case

One more annoying case is when most of the non-zero variables belong to blocks $B(w)$ such that these blocks themselves have too many non-zero variables. To be precise, we want to avoid the situation where

$$\sum_{B(w):\#B(w)\geq k^3} \#B(w) \;\; \geq \;\; \frac{m}{40\log R} \tag{2}$$

This case is handled by Type-3 linear forms. By Lemma 5.1, for any block $B(w)$, the contribution of Type-3 forms towards the objective function is at least $2^R(\#B(w))^{p/2}$. Hence the contribution of blocks in (2) is at least,

$$\sum_{B(w):\#B(w)\geq k^3} 2^R(\#B(w))^{p/2}$$

which is minimized when all $\#B(w)$ are equal to k^3 and there are $\frac{m}{k^3 40\log R}$ of them. Thus the contribution is at least

$$2^R(k^3)^{p/2}\frac{m}{k^3 40\log R} \;\; \gg \;\; k^p 2^R m$$

Note that the contribution in the completeness case is $2^R m$ and therefore one gets a penalty of factor k^p as desired.

5.5 Finishing the Proof

After handling all the annoying cases, we can now assume that $\#\mathbf{x} \geq \frac{m}{20\log R}$ and that

$$\sum_{B(w):\#B(w)\geq k^3} \#B(w) \;\; \leq \;\; \frac{m}{40\log R}$$

This implies that for at least $\frac{m}{k^3 40\log R} = \delta m$ vertices $w \in W$, the block $B(w)$ contains at least one non-zero variable. Let

$$W' = \{\ w \mid B(w) \text{ contains at least one non-zero variable}\}$$

We have $|W'| \geq \delta|W|$. Fix one non-zero variable in $B(w)$ for every $w \in W'$ and let this variable be $x_{w,A(w)}$ (thus we get an assignment A of labels to vertices in W').

By an averaging argument, for at least $\delta/4$ fraction of the vertices $v \in V$, at least $\delta/4$ fraction of their neighbors are in W'. Call any such vertex v "good". For any good vertex v, let

$$\Psi(v) = \{\ \pi_{v,w}(A(w)) \mid w \in W', w \in N(v)\}$$

Lemma 5.2 *For at most half the good vertices v, $|\Psi(v)| \leq k$.*

Proof: Assume on the contrary that for half of the good vertices v, $|\Psi(v)| \leq k$. Thus for every such vertex v, we can assign at most k labels such that for every neighbor $w \in W'$ of v, the label $\pi_{v,w}(A(w))$ is included. Choosing at random, one of the at most k labels for every such vertex v, gives a labeling to the Label Cover problem that satisfies following fraction of edges :

$$\frac{\delta}{8}\frac{\delta}{4}\frac{1}{k} = \frac{1}{32 \cdot 1600(\log R)^2 k^7} \gg \frac{1}{R^\gamma} \qquad \text{since } k = R^{\gamma/20}$$

This contradicts Theorem 2.1. ∎

Hence we can assume that for at least half of the good vertices v, $|\Psi(v)| \geq k$. For any such vertex v, assume w.l.o.g. that $\{1, 2, \ldots, k\} \subseteq \Psi(v)$. Thus for every $1 \leq j \leq k$, there exists $w_j^* \in N(v)$ such that $\pi_{v,w_j^*}(A(w_j^*)) = j$. Hence for any sequence $(w_1^*, w_2^*, \ldots, w_k^*, w_{k+1}, \ldots, w_S)$ where $w_{k+1}, \ldots, w_S \in N(v)$ are arbitrary, the Type-4 linear form has at least k non-zero variables. Applying Lemma 5.1, we get a contribution of at least $2^R D^{S-k} k^{p-k}$ towards the objective function.

$\delta/4$ fraction of the vertices v are good, hence the objective function is at least

$$(\frac{\delta}{8}n)2^R D^{S-k} k^{p-k}$$

Note that the contribution in the completeness case is at most $2^R n D^S$. We will choose p such that

$$\frac{\delta}{8}n 2^R D^{S-k} k^{p-k} \ > \ 2^R n D^S \cdot k^{(1-\epsilon)p}$$

Thus we get a penalty of factor $k^{(1-\epsilon)p}$ in the soundness case. Noting that $D \leq R, k = R^{\gamma/20}$, we see that it suffices to take $p = \frac{20k}{\epsilon\gamma}$. This gives hardness factor of $k^{1-\epsilon}$ or $p^{1-\epsilon'}$ as desired. This completes the full reduction.

Remark : Note that in the completeness case, irrespective of the choice of the integers $\{a_{w,i}\}$ in the Type-2 linear form, there always exists a short vector. In the soundness case, with high probability over the choice of these integers, every non-zero lattice vector is long. This implies that if there exists a polynomial-time $p^{1-\epsilon}$-approximation for SVP$_p$, then NP $\subseteq$ coRP, and hence NP = ZPP.

6 Acknowledgments

I thank Sanjeev Arora for suggesting the use of Pigeon-Hole principle. Many thanks to Daniele Micciancio for pointing out that the reduction in this paper holds under the assumption NP $\not\subseteq$ ZPP. Thanks also to Oded Regev, Miki Ajtai, Johan Håstad, Amit Chakrabarti, Ravi Kumar and D. Sivakumar for reading a preliminary write-up of this paper and for their excellent comments.

References

[1] M. Ajtai. The shortest vector problem in L_2 is NP-hard for randomized reductions. In Proc. of the 30^{th} Annual ACM Symposium on the Theory of Computing, 1998 pp 10-19.

[2] M. Ajtai. Generating hard instances of lattice problems. In Proc. of the 28^{th} Annual ACM Symposium on the Theory of Computing, 1996, pp 99-108.

[3] M. Ajtai, C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In Proc. of the 29^{th} Annual ACM Symposium on the Theory of Computing, 1997, pp 284-293.

[4] M. Ajtai, R. Kumar, D. Sivakumar. A sieve algorithm for the shortest lattice vector problem. In Proc. of the 33^{rd} Annual ACM Symposium on the Theory of Computing, 2001, pp 601-610.

[5] S. Arora, L. Babai, J. Stern, E.Z. Sweedyk. The hardness of approximate optima in lattices, codes and systems of linear equations. Journal of Computer and Systems Sciences (54), 1997, pp 317-331.

[6] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy. Proof verification and the hardness of approximation problems. Journal of the ACM, 45(3), 1998, pp 501-555.

[7] S. Arora and S. Safra. Probabilistic checking of proofs : A new characterization of NP. Journal of the ACM, 45(1), 1998, pp 70-122.

[8] M. Bellare, S. Goldwasser, C. Lund, A. Russell. Efficient probabilistically checkable poofs and applications to approximation. In Proc. of the 25^{th} Annual ACM Symposium on Theory of Computing, 1993.

[9] J.Y. Cai. Applications of a new transference theorem to Ajtai's connection factor. In Proc. of the 14^{th} Annual IEEE Conference on Computational Complexity, 1999.

[10] J.Y. Cai, A. Nerurkar. Approximating the SVP to within a factor $(1 + 1/\dim^{\epsilon})$ is NP-hard under randomized reductions. In Proc. of the 13^{th} Annual IEEE Conference on Computational Complexity, 1998, pp 151-158.

[11] J.Y. Cai, A. Nerurkar. An improved worst-case to average-case connection for lattice problems. In 38^{th} IEEE Symposium on Foundations of Computer Science, 1997.

[12] I. Dinur. Approximating SVP$_\infty$ to within almost polynomial factors is NP-hard. Proc. of the 4th Italian Conference on Algorithms and Complexity, LNCS, vol 1767, Springer, 2000.

[13] I. Dinur, G. Kindler, S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. In Proc. of the 39^{th} IEEE Symposium on Foundations of Computer Science, 1998.

[14] C.F. Gauss. Disquisitiones arithmeticae. (leipzig 1801), art. 171. Yale Univ. Press, 1966. English translation by A.A. Clarke.

[15] O. Goldreich, S. Goldwasser. On the limits of non-approximability of lattice problems. In Proc. of the 30^{th} Annual ACM Symposium on the Theory of Computing, 1998, pp 1-9.

[16] V. Guruswami, J. Håstad, M. Sudan. Hardness of approximate hypergraph coloring. In Proc. of the 41^{st} IEEE Symposium on Foundations of Computer Science, 2000, pp 149-158.

[17] J. Håstad. Some optimal inapproximability results. In Proc. of the 29^{th} Annual ACM Symposium on Theory of Computing, 1997, pp 1-10.

[18] R. Kannan. Improved algorithms for integer programming and related lattice problems. In Proc. of the 15^{th} Annual ACM Symposium on Theory of Computing, 1983, pp 193-206.

[19] R. Kannan. Minkowski's convex body theorem and integer programming. Mathematics of Operations Research, 12:415-440, 1987.

[20] R. Kumar, D. Sivakumar. Complexity of SVP - A reader's digest. SIGACT News, 32(3), Complexity Theory Column (ed. L. Hemaspaandra), 2001, pp 40-52.

[21] J.C. Lagarias, H.W. Lenstra, C.P. Schnorr. Korkine-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. Combinatorica, vol 10, 1990, pp 333-348.

[22] J.C. Lagarias, A.M. Odlyzko. Solving low-density subset sum problems. Journal of the ACM, 32(1), 1985, pp 229-246

[23] S. Landau, G.L. Miller. Solvability of radicals is in polynomial time. Journal of Computer and Systems Sciences, 30(2), 1985, pp 179-208.

[24] H.W. Lenstra. Integer programming with a fixed number of variables. Tech. Report 81-03, Univ. of Amsterdam, Amstredam, 1981.

[25] A.K. Lenstra, H.W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Ann., 261, 1982, pp 513-534.

[26] C. Lund, M. Yannakakis. On the hardness of approximating minimization problems. Journal of the ACM, 41(5), 1994, pp 960-981.

[27] D. Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. In Proc. of the 39^{th} IEEE Symposium on Foundations of Computer Science, 1998.

[28] D. Micciancio. PhD Thesis, MIT, 1998.

[29] H. Minkowski. Geometrie der zahlen. Leizpig, Tuebner, 1910.

[30] R. Raz. A parallel repetition theorem. SIAM J. of Computing, 27(3), 1998, pp 763-803.

[31] O. Regev, New lattice based cryptographic constructions. To appear in Proc. of the 35^{th} Annual ACM Symposium on the Theory of Computing, 2003.

[32] C.P. Schnorr. A hierarchy of polynomial-time basis reduction algorithms. In Proc. of Conference on Algorithms, Péecs (Hungary), 1985, pp 375-386.

[33] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Tech. Report 81-04, Mathematische Instiut, Univ. of Amsterdam, 1981.