Hardness of Approximating the Shortest Vector Problem in Lattices *

Subhash Khot
khot@cc.gatech.edu †

Abstract

Let $p > 1$ be any fixed real. We show that assuming NP $\not\subseteq$ RP, it is hard to approximate the Shortest Vector Problem (SVP) in ℓ_p norm within an arbitrarily large constant factor. Under the stronger assumption NP $\not\subseteq$ RTIME($2^{poly(\log n)}$), we show that the problem is hard to approximate within factor $2^{(\log n)^{1/2-\epsilon}}$ where n is the dimension of the lattice and $\epsilon > 0$ is an arbitrarily small constant. This greatly improves all previous results in ℓ_p norms with $1 < p < \infty$. The best results so far gave only a constant factor hardness, namely, $2^{1/p} - \epsilon$ by Micciancio [27] and $p^{1-\epsilon}$ in high ℓ_p norms by Khot [20].

We first give a new (randomized) reduction from Closest Vector Problem (CVP) to SVP that achieves some constant factor hardness. The reduction is based on BCH Codes. Its advantage is that the SVP instances produced by the reduction behave well under the augmented tensor product, a new variant of tensor product that we introduce. This enables us to boost the hardness factor to $2^{(\log n)^{1/2-\epsilon}}$.

* Full version of the paper is available at http://www.cs.princeton.edu/∼khot/publications.html

† Institute for Advanced Study, Princeton, NJ-08544 and Georgia Institute of Technology, Atlanta, GA-30332. *This material is based upon work supported by the National Science Foundation under agreement No. DMS-0111298. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.*

1. Introduction

An n-dimensional lattice $\mathcal{L}$ is a set of vectors $\{\sum_{i=1}^{n} a_i \mathbf{b}_i \mid a_i \in \mathbb{Z}\}$ where $\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n \in \mathbb{R}^m$ is a set of linearly 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 (SVP) asks for the shortest non-zero vector in the lattice (in ℓ_2 norm unless stated otherwise). This problem has a beautiful history and we present some of the results below. For a more comprehensive list of references, please refer to Micciancio and Goldwasser's book [28] and an expository article by Kumar and Sivakumar [21].

The Shortest Vector Problem has been studied since the time of Gauss ([15], 1801) who gave an algorithm that works for 2-dimensional lattices. 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 short non-zero vectors in lattices. In a celebrated result, Lenstra, Lenstra and Lovász [26] gave a polynomial time algorithm for approximating SVP within factor $2^{n/2}$. This algorithm has numerous applications, e.g. factoring rational polynomials [26], breaking knapsack-based codes [23], checking the solvability by radicals [24] and integer programming in a fixed number of variables ([26], [25], [18]). Schnorr [31] improved the approximation factor to $2^{O(n(\log\log n)^2/\log n)}$. It is a major open problem whether SVP has polynomial factor approximations (that run in polynomial time). Exact computation of SVP in exponential time is also investigated, see for instance [19], [5].

In 1981, van Emde Boas [32] proved that SVP in ℓ_∞ norm is NP-hard and conjectured that the same is true in any ℓ_p norm. However proving NP-hardness in ℓ_2 norm (or in any finite ℓ_p norm for that matter) was an embarrassing open problem for a long time. A breakthrough result by Ajtai [2] in 1998 finally showed that SVP 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 $(1 + \frac{1}{n^\epsilon})$. Another breakthrough by Micciancio

[27] showed that SVP is hard to approximate within some constant factor, specifically any factor less than $\sqrt{2}$. This was the best result known so far leaving a huge gap between the $\sqrt{2}$ hardness factor and the ratio $2^{O(n(\log\log n)^2/\log n)}$ achieved by LLL-Schnorr algorithm.

Showing hardness of approximation results for SVP was greatly motivated by Ajtai's discovery [3] of worst-case to average-case reduction and subsequent construction of a lattice-based public key cryptosystem by Ajtai and Dwork [4]. Ajtai showed that if there is a randomized polynomial time algorithm for solving SVP 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 on *every* instance within some polynomial factor n^c (he also presented a candidate one-way function). This line of work gave hope, for the first time, that cryptography could be based on the (conjectured) worst-case hardness of a problem. These results imply that if n^c-approximation to SVP is hard, then one can construct a secure cryptosystem. The constant c was noted to be 19 in [9], and brought down to $9+\epsilon$ by Cai and Nerurkar [11] and then to $4+\epsilon$ by Cai [9]. Recently, Regev [30] gave an alternate construction of a public key cryptosystem based on $n^{1.5}$-hardness of SVP (technically speaking these results assume hardness of a variant called unique-SVP). Unfortunately, there are barriers to showing such strong hardness results. In fact, showing factor n NP-hardness would imply that NP = coNP ([22], [17], [8]) and showing factor $\sqrt{n/O(\log n)}$ NP-hardness would imply that coNP $\subseteq$ AM [16]. Recently, Aharonov and Regev [1] showed that showing factor $\sqrt{n}$ NP-hardness would imply that NP = coNP.

Another related problem that has received much attention is the Closest Vector Problem (CVP) where given a lattice and a point $\mathbf{y}$, the problem is to find the lattice vector that is closest to $\mathbf{y}$. In spite of the apparent similarity between SVP and CVP, they turn out to be very different problems. Indeed, CVP was shown to be NP-hard by van Emde Boas [32]. Arora, Babai, Sweedyk, and Stern [7] used the PCP machinery to show that approximating CVP within factor $2^{\log^{1-\epsilon} n}$ is hard unless NP $\subseteq$ DTIME($2^{poly(\log n)}$). This was improved to a NP-hardness result by Dinur, Kindler, and Safra [13] (their result gives even a subconstant value of ϵ, i.e. $\epsilon = (\log\log n)^{-c}$ for any $c < \frac{1}{2}$). All the above results for CVP work in all ℓ_p norms. Incidently, SVP in ℓ_∞ norm seems to behave very much like CVP in ℓ_∞ norm and Dinur [12] shows factor $n^{1/\log\log n}$ NP-hardness for both these problems.

Our Result

In this paper, we improve on all previous hardness results ([27], [20]) for SVP in ℓ_p norms with $1 < p < \infty$. We show that

Theorem 1.1 *Let $p > 1$ be any fixed real. Assuming* NP $\nsubseteq$ RP, *it is hard to approximate the Shortest Vector Problem in ℓ_p norm within an arbitrarily large constant factor. Assuming* NP $\nsubseteq$ RTIME($2^{poly(\log n)}$), *the problem is hard to approximate within factor $2^{(\log n)^{1/2-\epsilon}}$ where n is the dimension of the lattice and $\epsilon > 0$ is an arbitrarily small constant.*

Thus we break the barrier of constant factor hardness. It may be possible to extend our result to the *almost polynomial factor* hardness, i.e. $2^{(\log n)^{1-\epsilon}}$. Our result does not work in ℓ_1 norm for which Micciancio's [27] $2-\epsilon$ hardness result remains the best.

Overview of the Paper

We outline the overall proof idea and techniques in Section 2. Our reduction proceeds in four steps (i) Construction of a gadget lattice called BCH Lattice using BCH Codes. (ii) Reduction from CVP to an Intermediate Lattice using the BCH Lattice. (iii) Reduction from Intermediate Lattice to an SVP instance, giving a constant factor hardness. (iv) Boosting hardness factor via Augmented Tensor Product.

Step (i) appears in Appendix B. Due to space limitations, we omit proofs of Steps (ii) and (iii), and state only the end-result, namely Theorem 4.1. Section 5 defines the augmented tensor product and Section 6 describes the Step (iv), proving Theorem 1.1. We present some conclusions and open problems in Section 7.

Notation and Problem Definitions

All vectors are denoted by bold face letters. All vectors are column vectors unless stated otherwise.

A lattice $\mathcal{L}$ together with a basis $\mathbf{B}$ is denoted as $\mathcal{L}(\mathbf{B})$. $\mathbf{B}$ is a $M \times N$ real matrix whose columns are linearly independent (and hence $M \geq N$). The lattice $\mathcal{L}$ in $\mathbb{R}^M$ is given by

$$\mathcal{L} = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^N\}$$

We call $\mathbf{x}$ as the coefficient vector (with respect to the specific basis) and any $\mathbf{z} = \mathbf{Bx}$ as the lattice vector.

Note that the same lattice could have many different bases. The Shortest Vector Problem (SVP) in ℓ_p norm asks for the shortest non-zero vector in a lattice when an explicit basis is given. Formally, given a lattice $\mathcal{L}(\mathbf{B})$

where $\mathbf{B}$ has size $M \times N$,

$$\text{minimize}_{\mathbf{x}\in\mathbb{Z}^N,\ \mathbf{x}\neq\mathbf{0}} \quad \|\mathbf{Bx}\|_p$$

The Closest Vector Problem (CVP) asks for a lattice vector in $\mathcal{L}$ which is closest to the given vector $\mathbf{t} \in \mathbb{R}^M$. Formally, given a pair $(\mathcal{L}(\mathbf{B}), \mathbf{t})$ where $\mathbf{B}$ is a $M \times N$ matrix and $\mathbf{t} \in \mathbb{R}^M$,

$$\text{minimize}_{\mathbf{x}\in\mathbb{Z}^N} \quad \|\mathbf{Bx} - \mathbf{t}\|_p$$

2. Techniques

In this section, we give an outline of our reduction and the techniques that enable us to boost the hardness factor beyond a constant barrier. We hope this will facilitate reading the rest of the paper.

The analogue of SVP for finite fields, called the Minimum Distance of Linear Code problem (MDC) is hard to approximate within almost polynomial factors as shown by Dumer, Micciancio and Sudan [14]. Their reduction first gives some constant factor hardness (say $\beta > 1$) and then it is boosted using the tensor product of codes. If G_1, G_2 are generator matrices of linear codes $\mathcal{C}_1, \mathcal{C}_2$ respectively, then $\mathcal{C}_1 \otimes \mathcal{C}_2$ is the linear code whose generator matrix is $G_1 \otimes G_2$. The crucial property of this operation is that the minimum distance of the product code equals the product of minimum distances of the individual codes. Thus

$$dist(\mathcal{C}_1 \otimes \mathcal{C}_2) = dist(\mathcal{C}_1) \cdot dist(\mathcal{C}_2)$$

Dumer et al. give a reduction from SAT to a gap-instance $\mathcal{C}$ of MDC with gap β. Defining inductively, $\mathcal{C}_1 = \mathcal{C}$ and $\mathcal{C}_j = \mathcal{C} \otimes \mathcal{C}_{j-1}$, it follows that $\mathcal{C}_k$ is a gap-instance with gap β^k. If N is the size of the instance $\mathcal{C}$, then $\mathcal{C}_k$ has size N^k. Thus we get a hardness factor of β^k via a reduction that runs in time N^k. In particular, we get an arbitrarily large constant factor hardness via polynomial time reduction. Taking $k = \text{poly}(\log N)$ gives an almost polynomial factor hardness via a quasi-polynomial time reduction.

Ideally speaking, one would like to use the same tensor product technique to boost hardness of SVP. For two lattices $\mathcal{L}_1(\mathbf{B}_1), \mathcal{L}_2(\mathbf{B}_2)$ with basis matrices $\mathbf{B}_1, \mathbf{B}_2$ respectively, we can define a lattice $\mathcal{L}_1 \otimes \mathcal{L}_2$ to be the lattice whose basis matrix is $\mathbf{B}_1 \otimes \mathbf{B}_2$. If $sh()$ denotes the length of the shortest vector in a lattice, it is easy to see that

$$sh(\mathcal{L}_1 \otimes \mathcal{L}_2) \leq sh(\mathcal{L}_1) \cdot sh(\mathcal{L}_2)$$

However, this is not an equality in general and explicit examples are known where the equality fails to hold. This is a well-known difficulty in using the tensor product to boost hardness of SVP. One of the main ideas in this paper is to figure out special properties of lattices that make (a variant of) the tensor product construction go through. As in Micciancio's paper [27], we use a reduction from a gap-version of CVP to SVP. However we would like to stress that the reduction has only a high level resemblance to Micciancio's reduction. His reduction relies on Schnorr-Adleman Prime Number Lattice and a probabilistic version of Sauer's Lemma. We instead use a new lattice based on the BCH Codes. We also use the technique of adding a single Random Linear Form to get rid of some *annoying* lattice vectors. This simple but very powerful technique was introduced in Khot [20]. Finally, we introduce the Augmented Tensor Product that achieves the boosting. The instances of SVP given by our reduction have special properties that make the new version of tensor product work.

2.1. Wishful Thinking

Let us imagine a hypothetical reduction from CVP to an instance $\mathcal{L}(\mathbf{B})$ of SVP that has the following properties (we assume w.l.o.g. that all lattice vectors have integer co-ordinates) :

1. If the CVP instance is a YES instance, then there is a non-zero $\{0, 1\}$-lattice vector with at most γd co-ordinates equal to 1. Here $\gamma < 1$ is a constant.
2. If the CVP instance is a NO instance, then any non-zero lattice vector has at least d non-zero co-ordinates.

In particular, this gives a gap-instance of SVP with gap $(1/\gamma)^{1/p}$ in ℓ_p norm. It is not hard to see that if we had such a *magic reduction*, then the k-wise tensor product of the lattice $\mathcal{L}$ would indeed give a gap-instance with gap $(1/\gamma)^{k/p}$. Thus the tensor product would work provided that in the NO case, every non-zero lattice vector is not only *long*, but also has *many* non-zero co-ordinates.

However, we do not know whether such a reduction exists. In this paper, we give a reduction that achieves somewhat weaker properties, but still good enough for our purpose. Then we define a variant of tensor product called *augmented tensor product* and show that the hardness can be boosted via this product.

2.2. Our Basic Reduction

Let us informally state the properties achieved by our basic reduction. This reduction achieves a hardness factor of $(1/\gamma)^{1/p}$ which is then boosted via the augmented tensor product defined next.

Theorem 2.1 *(Informal Statement) Fix any $p > 1$ and $1/2 + 1/2^p < \gamma < 1$. There is a reduction that maps a*

CVP instance to an SVP instance $\mathcal{L}(\mathbf{B})$ with the following properties :

Property 1 : *If the CVP instance is a YES instance, there is a lattice vector $\mathbf{Bx} \neq \mathbf{0}$ such that $\|\mathbf{Bx}\|_p \leq (\gamma d)^{1/p}$ and in addition $\|\mathbf{x}\|_p$ is small.*

Property 2 : *If the CVP instance is a NO instance, then every non-zero lattice vector $\mathbf{Bx}$*

- *Either has at least d non-zero co-ordinates*
- *Or has all co-ordinates even and at least $d/2^p$ of them are non-zero*
- *Or has at least one co-ordinate that is enormously large.*

In particular, every non-zero lattice vector has norm at least $d^{1/p}$.

These properties look strange, but they fit nicely with the *augmented tensor product* construction. The strange even-ness condition comes from the use of binary linear codes (the BCH codes).

2.3. Augmented Tensor Product

The hardness given by Theorem 2.1 is boosted via the augmented tensor product. Suppose that in the NO case in Theorem 2.1, every non-zero lattice vector has either d non-zero co-ordinates or has all co-ordinates even, $d/2^p$ of them non-zero. Then it can be shown that the standard tensor product is sufficient to boost hardness. The augmented variant takes care of precisely the remaining third possibility, when a lattice vector has very few non-zero co-ordinates, but has at least one enormously large co-ordiante.

Recall that the basis matrix for the tensor lattice $\mathcal{L}_1 \otimes \mathcal{L}_2$ is $\mathbf{B}_1 \otimes \mathbf{B}_2$ where $\mathbf{B}_1, \mathbf{B}_2$ are the basis matrices for individual lattices. The augmented tensor product with parameter $\alpha > 0$ is denoted by $\mathcal{L}_1 \otimes_\alpha^A \mathcal{L}_2$. Its basis matrix is given by

$$\mathbf{B} = \mathbf{B}_1 \otimes \left[\frac{\alpha \mathbf{I}}{\mathbf{B}_2}\right] \quad \mathbf{I} = \text{identity matrix}$$

Here is an example that gives some understanding of this construction. Let us say $\mathbf{B}_1\mathbf{x}$ and $\mathbf{B}_2\mathbf{y}$ are *short* lattice vectors and $\mathbf{y}$ is also short. Then we will show that $\mathbf{B}(\mathbf{x} \otimes \mathbf{y})$ is a *short* vector in the product lattice. Indeed,

$$\|\mathbf{B}(\mathbf{x} \otimes \mathbf{y})\|_p^p = \|(\mathbf{B}_1\mathbf{x})\|_p^p \cdot (\alpha^p \|\mathbf{y}\|_p^p + \|\mathbf{B}_2\mathbf{y}\|_p^p)$$

which is small provided $\|\mathbf{B}_1\mathbf{x}\|_p, \|\mathbf{B}_2\mathbf{y}\|_p, \|\mathbf{y}\|_p$ are all small. Thus we need not only that the lattice vector $\mathbf{B}_2\mathbf{y}$ is short, but also that the coefficient vector $\mathbf{y}$ itself is short. This is why we need the coefficient vector to have small norm in Property (1) of Theorem 2.1.

2.4. Using BCH Codes

A crucial component in the proof of Theorem 2.1 is a gadget lattice built from BCH Codes. We call it the BCH Lattice. The YES/NO properties of Theorem 2.1 are essentially inherited from the corresponding properties of the BCH Lattice. The construction appears in Appendix B.

We use the parity check matrix of BCH Codes (denoted P_{BCH}). It is known that columns of this matrix give a family of N vectors in $GF(2)^h$ such that any d of them are linearly independent over $GF(2)$. The crux of our construction is the d-wise independence property and the efficiency of the parameter h. The goal is to minimize h as a function of N, d. Using the columns of P_{BCH}, it is possible to achieve

$$h = \frac{d}{2} \log N$$

The quantitative strength of this result is very important for our paper. For example, if we instead use $h = d \log N$, the result in our paper would fall apart.

Use of linear codes is motivated by the fact that the tensor product works well with codes. We, in some sense, simulate codes by integer lattices. Indeed, modulo some additional tricks, the BCH Lattice is essentially the matrix P_{BCH} appended with $2I$ where I is $h \times h$ identity matrix (see Fig. 2). With lattices, we can only perform integer addition, but the extra columns of the matrix $2I$ *allow* us to perform *mod* 2 operations by suitable addition/subtraction of even integers.

2.5. Intermediate Lattice and Taking a Random Sub-Lattice

Theorem 2.1 is proved in two stages. In the first stage, we reduce CVP instance to an Intermediate Lattice $\mathcal{L}_{int}(\mathbf{B}_{int})$ using the BCH Lattice as a gadget. Roughly speaking,

- $\mathcal{L}_{int}$ satisfies Property (1) of Theorem 2.1. In fact there are *many* such lattice vectors. Call them *good* vectors and let their number be $\geq \#G$.
- $\mathcal{L}_{int}$ *almost* satisfies Property (2) of Theorem 2.1 in the sense that there are only a *few* lattice vectors that do not satisfy Property (2). Call them *annoying* vectors and let their number be $\leq \#A$.

In the second stage, the lattice $\mathcal{L}(\mathbf{B})$ is obtained by taking a random sub-lattice of $\mathcal{L}_{int}(\mathbf{B}_{int})$. Think of this as a filter through which every lattice vector passes with a tiny (carefully chosen) probability. We ensure that there are many more good vectors than annoying vectors, i.e. $\#G \gg \#A$. This ensures that the sub-lattice

$\mathcal{L}(\mathbf{B})$ contains at least one good vector (in the YES case) and has no annoying vector left (in the NO case). Thus the sub-lattice $\mathcal{L}(\mathbf{B})$ satisfies both Properties (1) and (2) of Theorem 2.1 proving the theorem.

2.6. Quantitative Parameters

Our reduction will have quite a few parameters. Let us point out some high level facts that make the reduction work. Hopefully it will give a quantitative feel for the parameters.

Roughly speaking, the reduction maps a CVP instance of size d to an SVP instance of size N using a family $\mathcal{F}$ of N vectors over $GF(2)^h$ that are d-wise independent. In the YES case, the Intermediate Lattice has at least $\#G$ good lattice vectors. As we will see, every good vector corresponds to a set of γd vectors from the family $\mathcal{F}$ such that the sum (over $GF(2)$) of these γd vectors equals a fixed binary vector $\mathbf{s} \in GF(2)^h$. Since there are $\binom{N}{\gamma d}$ subsets of $\mathcal{F}$ of size γd, and 2^h possible sums, we can pick an $\mathbf{s}$ that occurs at least $\binom{N}{\gamma d}/2^h$ many times as the sum of γd vectors from $\mathcal{F}$. Hence $\#G = \binom{N}{\gamma d}/2^h = \binom{N}{\gamma d}/N^{d/2}$ since $h = \frac{d}{2}\log N$.

On the other hand, in the NO case, we want to bound the number of annoying lattice vectors in $\mathcal{L}_{int}$. A vector is annoying if, for example, it has less than $d/2^p$ co-ordinates that are equal to 2 and the rest are zero. Thus the number of annoying vectors is potentially $\binom{N}{d/2^p} = \#A$. Since we wish to have $\#G \gg \#A$, we need

$$\frac{\binom{N}{\gamma d}}{N^{d/2}} \gg \binom{N}{d/2^p}$$

Thus we need that $1/2 + 1/2^p < \gamma < 1$ and therefore our reduction works only with $p > 1$. Also, we need N to be a large polynomial in d.

A few words about the quantitative aspect of augmented tensor product. Let us say that we want to apply k-wise augmented tensor product to the size N SVP instance. This blows up the size to N^k. However, the way our analysis works, the SVP instance has to be *robust enough* to *sustain* a k-wise product. Its size N needs to be $d^{O(k)}$ to sustain the k-wise product. Therefore, the final size of the product instance is $d^{O(k^2)}$ whereas a k-wise product implies a hardness factor of only $2^{\Omega(k)}$. This is the reason why we can boost the hardness only to $2^{(\log n)^{1/2-\epsilon}}$ and not to $2^{(\log n)^{1-\epsilon}}$ as one would otherwise expect. This is an undesirable feature of our reduction.

3. The CVP Instance

In this section, we define a suitable gap-version of CVP that we use in this paper. The following reduction is due to Arora, Babai, Stern and Sweedyk [7]. We include a proof in Appendix A.

Theorem 3.1 *For any constant $\eta > 0$, there are constants C, C', C'', and a reduction from SAT instance of size n to a CVP instance $(\mathcal{L}_{cvp}(\mathbf{B}_{cvp}), \mathbf{t})$ with the following properties :*

1. *$\mathbf{B}_{cvp}$ is an integer matrix with size $C'd \times Cd$. The vector $\mathbf{t}$ also has integer co-ordinates and it is linearly independent of the columns of matrix $\mathbf{B}_{cvp}$.*
2. *The reduction runs in time $n^{C''}$ and therefore $d \leq n^{C''}$.*
3. *If the SAT instance is a YES instance, then there is a $\{0,1\}$ coefficient vector $\mathbf{y} \in \mathbb{Z}^{Cd}$ such that the vector $\mathbf{B}_{cvp}\mathbf{y} - \mathbf{t}$ is also a $\{0,1\}$-vector and has exactly ηd co-ordinates equal to 1.*
4. *If the SAT instance is a NO instance, then for any coefficient vector $\mathbf{y} \in \mathbb{Z}^{Cd}$, and any non-zero integer j_0, the vector $\mathbf{B}_{cvp}\mathbf{y} - j_0\mathbf{t}$ either has one co-ordinate equal to d^{4d}, or has at least d non-zero co-ordinates.*

4. The Basic Reduction

The following theorem gives our basic reduction from CVP to SVP. It is a formal restatement of Theorem 2.1. We omit its proof for the lack of space. The proof relies on construction of a lattice based on BCH Codes and this construction appears in Appendix B.

The theorem gives a constant factor hardness for SVP which is then boosted in Section 6 via augmented tensor product. The parameter k in this theorem refers to the k-wise tensor product to be applied later. This is an unusual feature of our reduction; the basic reduction itself depends on the extent of tensor product to be applied later. Think of k as constant or poly$(\log d)$.

Theorem 4.1 *Fix any real $p > 1$. Let $\eta > 0$ be such that $\gamma = \frac{1}{2} + \frac{1}{2^p} + (2^p + 1)\eta < 1$. Let $(\mathcal{L}_{cvp}(\mathbf{B}_{cvp}), \mathbf{t})$ be an instance of CVP given by Theorem 3.1 with parameters η, C, C', C'', d. Then for any integer $k \leq d$, there is a randomized reduction from the CVP instance to an SVP instance $\mathcal{L}(\mathbf{B})$ with the following properties :*

1. *$\mathbf{B}$ is a $2N \times N$ integer matrix with $N = d^{14C'k/\eta}$. The reduction runs in time polynomial in N.*
2. *If the CVP instance is a YES instance, then with probability at least $9/10$, there is a non-zero coefficient vector $\mathbf{x} \in \mathbb{Z}^N$ such that $\|\mathbf{x}\|_p^p \leq d^{3p}$ and $\|\mathbf{B}\mathbf{x}\|_p^p \leq \gamma d$.*

3. *If the CVP instance is a NO instance, then with probability at least* $9/10$*, any non-zero lattice vector* $\mathbf{Bx}$ *(its all co-ordinates are integers)*
   - *either has* d *non-zero co-ordinates*
   - *or has all co-ordinates even and at least* $d/2^p$ *of them are non-zero*
   - *or has one co-ordinate with magnitude at least* d^{4k}.

   In particular, $\|\mathbf{Bx}\|_p^p \geq d$.

5. Augmented Tensor Product of Lattices

In this section, we define the tensor product of lattices and its new variant called *augmented tensor product* that we introduce.

5.1. Tensor Product

Let $\mathcal{L}(\mathbf{B}), \mathcal{L}'(\mathbf{B}')$ be lattices where $\mathbf{B}, \mathbf{B}'$ are matrices with size $M \times N$ and $M' \times N'$ respectively. The tensor product $\mathcal{L}'' = \mathcal{L} \otimes \mathcal{L}'$ is a lattice defined as follows. We view the coefficient vector of $\mathcal{L} \otimes \mathcal{L}'$ as a $N \times N'$ matrix $\mathbf{x}$ and define

$$\mathcal{L}'' = \mathcal{L} \otimes \mathcal{L}' = \{\mathbf{B}\mathbf{x}\mathbf{B}'^T \mid \mathbf{x} \in \mathbb{Z}^{NN'}\}$$

Note that the lattice vectors in $\mathcal{L}''$ have MM' co-ordinates.

There is another equivalent way to look at the tensor product. One can define $\mathcal{L}'' = \mathcal{L} \otimes \mathcal{L}'$ to be the lattice whose basis matrix is $\mathbf{B}'' = \mathbf{B} \otimes \mathbf{B}'$. Thus we have

$$\mathcal{L}'' = \{\mathbf{B}''\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^{NN'}\}$$

5.2. Augmented Tensor Product

The augmented tensor product also has two equivalent ways to look at it and we will use both the viewpoints.

The augmented tensor product $\mathcal{L}'' = \mathcal{L} \otimes_\alpha^A \mathcal{L}'$ with parameter $\alpha > 0$ is defined as follows. The coefficient vector of $\mathcal{L}''$ is a NN' matrix $\mathbf{x}$. Write $\mathbf{x}$ as

$$\mathbf{x} = [\mathbf{x}_1 \mathbf{x}_2 \ldots \mathbf{x}_{N'}]$$

where $\mathbf{x}_i \in \mathbb{Z}^N$ are column vectors. Then

$$\mathcal{L}'' = \{\alpha\mathbf{B}\mathbf{x}_1 \circ \alpha\mathbf{B}\mathbf{x}_2 \circ \ldots \circ \alpha\mathbf{B}\mathbf{x}_{N'} \circ \mathbf{B}\mathbf{x}\mathbf{B}'^T \mid \mathbf{x} \in \mathbb{Z}^{NN'}\}$$

Here, $\circ$ denotes concatenation of vectors. Thus the lattice vectors in the augmented tensor product have $MN' + MM'$ co-ordinates.

Alternatively, $\mathcal{L}'' = \mathcal{L} \otimes_\alpha^A \mathcal{L}'$ can be defined as a lattice whose basis matrix $\mathbf{B}''$ is given by

$$\mathbf{B}'' = \mathbf{B} \otimes \left[\frac{\alpha \mathbf{I}}{\mathbf{B}'}\right] \qquad \mathbf{I} = N' \times N' \text{ identity matrix}$$

Thus

$$\mathcal{L}'' = \{\mathbf{B}''\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^{NN'}\}$$

6. Boosting the SVP Hardness Factor

In this section, we boost the hardness factor for SVP given by Theorem 4.1. This proves Theorem 1.1.

Let $\mathcal{L}(\mathbf{B})$ be the lattice given by reduction of Theorem 4.1 and let $\alpha = 1/d^{3k}$. For $1 \leq j \leq k$, define lattices $\mathcal{L}_j(\mathbf{B}_j)$ inductively as follows.

$$\mathcal{L}_1 = \mathcal{L}, \quad \mathbf{B}_1 = \mathbf{B}$$

$$\mathcal{L}_j = \mathcal{L} \otimes_\alpha^A \mathcal{L}_{j-1}, \quad \mathbf{B}_j = \mathbf{B} \otimes \left[\frac{\alpha \mathbf{I}}{\mathbf{B}_{j-1}}\right] \tag{1}$$

6.1. The YES Case

Assume that the CVP instance is a YES instance. Theorem 4.1 guarantees that there is a non-zero coefficient vector $\mathbf{x}$ such that

$$\|\mathbf{Bx}\|_p^p \leq \gamma d, \qquad \|\mathbf{x}\|_p^p \leq d^{3p}$$

Let $\mathbf{x}^{\otimes j}$ denote the j-wise tensor product of vector $\mathbf{x}$. Note that $\mathbf{x}^{\otimes j}$ serves as a non-zero coefficient vector for the lattice $\mathcal{L}_j(\mathbf{B}_j)$. Clearly,

$$\|\mathbf{x}^{\otimes j}\|_p^p = \|\mathbf{x}\|_p^{pj} \leq d^{3pj}$$

Lemma 6.1 *For* $1 \leq j \leq k$, $\|\mathbf{B}_j \mathbf{x}^{\otimes j}\|_p^p \leq \sum_{l=1}^{j} \gamma^l d^l$. *In particular, the lattice* $\mathcal{L}_k$ *has a non-zero lattice vector of length* $(2\gamma^k d^k)^{1/p}$*, namely the vector* $\mathbf{B}_k \mathbf{x}^{\otimes k}$.

Proof: The proof is by induction. The claim is true for $j = 1$. For $j \geq 2$, using the inductive definition in Equation (1) and noting that $\alpha = 1/d^{3k}$, we have

$$\|\mathbf{B}_j \mathbf{x}^{\otimes j}\|_p^p = \|\mathbf{B}_j(\mathbf{x} \otimes \mathbf{x}^{\otimes(j-1)})\|^p =$$

$$\|\mathbf{Bx}\|_p^p \cdot \left(\alpha^p \|\mathbf{x}^{\otimes(j-1)}\|_p^p + \|\mathbf{B}_{j-1}\mathbf{x}^{\otimes(j-1)}\|_p^p\right)$$

$$\leq \gamma d \left(\alpha^p d^{3p(j-1)} + \sum_{l=1}^{j-1} \gamma^l d^l\right) \leq \gamma d \left(1 + \sum_{l=1}^{j-1} \gamma^l d^l\right)$$

$$= \sum_{l=1}^{j} \gamma^l d^l$$

∎

6.2. The NO Case

Assume that the CVP instance is a NO instance. The vectors in the lattice $\mathcal{L}(\mathbf{B})$ have integer co-ordinates. Theorem 4.1 guarantees that every non-zero vector in $\mathcal{L}(\mathbf{B})$

- Either has d non-zero co-ordinates
- Or has all co-ordinates even and at least $d/2^p$ of them are non-zero
- Or has one co-ordinate with magnitude at least d^{4k}.

In particular, every non-zero vector in $\mathcal{L}(\mathbf{B})$ has length at least $d^{1/p}$.

Lemma 6.2 *For $1 \leq j \leq k$, the length of any non-zero vector in the lattice $\mathcal{L}_j$ is at least $d^{j/p}$. In particular, the length of the shortest vector in the lattice $\mathcal{L}_k$ is at least $d^{k/p}$.*

Proof: The proof is by induction. The claim is true for $j = 1$. Assume $j \geq 2$. We want to show that for any non-zero coefficient vector $\mathbf{x}$ for the lattice $\mathcal{L}_j$, $\|\mathbf{B}_j\mathbf{x}\|_p^p \geq d^j$. Note that $\mathcal{L}_j = \mathcal{L} \otimes_\alpha^A \mathcal{L}_{j-1}$ and recall the definition of augmented tensor product. We can write $\mathbf{x}$ as

$$\mathbf{x} = [\mathbf{x}_1\mathbf{x}_2 \dots \mathbf{x}_{N'}]$$

where every $\mathbf{x}_i \in \mathbb{Z}^N$ is a column vector and N' is the number of co-ordinates of the coefficient vector for the lattice $\mathcal{L}_{j-1}$. We want to show a lower bound of d^j on

$$\|\mathbf{B}_j\mathbf{x}\|_p^p = \|\ \alpha\mathbf{B}\mathbf{x}_1 \circ \alpha\mathbf{B}\mathbf{x}_2 \dots \circ \mathbf{B}\mathbf{x}_{N'}\ \circ \mathbf{B}\mathbf{x}\mathbf{B}_{j-1}^T\ \|_p^p \tag{2}$$

We consider 3 cases.

Case 1 : Note that the vectors $\mathbf{B}\mathbf{x}_i$ are lattice vectors in $\mathcal{L}$ (so they are all integer vectors). If any of them has a co-ordinate with magnitude at least d^{4k}, then this co-ordinate alone contributes $\alpha^p d^{4kp} = d^{kp} \geq d^j$ towards the quantity (2) and we are done.

So we can now assume that each vector $\mathbf{B}\mathbf{x}_i$ is either identically zero or has at least d non-zero co-ordinates or has all co-ordinates even of which at least $d/2^p$ are non-zero. We use the bound

$$\|\mathbf{B}_j\mathbf{x}\|_p^p \geq \|\mathbf{B}\mathbf{x}\mathbf{B}_{j-1}^T\|_p^p$$

We write the right hand side as

$$\left\|[\mathbf{B}\mathbf{x}_1, \mathbf{B}\mathbf{x}_2, \dots, \mathbf{B}\mathbf{x}_{N'}]\,\mathbf{B}_{j-1}^T\right\|_p^p = \left\|\mathbf{X}\mathbf{B}_{j-1}^T\right\|_p^p$$

where we denoted by $\mathbf{X}$ the integer matrix with columns $\mathbf{B}\mathbf{x}_i, 1 \leq i \leq N'$.

Case 2 : Consider the case when at least one column of $\mathbf{X}$ has at least d non-zero co-ordinates. Then the matrix $\mathbf{X}$ has at least d non-zero rows. Call these rows $\mathbf{r}_1, \mathbf{r}_2, \dots \mathbf{r}_d$. Then

$$\|\mathbf{X}\mathbf{B}_{j-1}^T\|_p^p \geq \sum_{l=1}^{d} \|\mathbf{r}_l\mathbf{B}_{j-1}^T\|_p^p \geq d \cdot d^{j-1}$$

since every vector $\mathbf{B}_{j-1}\mathbf{r}_l^T$ is a non-zero lattice vector in the lattice $\mathcal{L}_{j-1}$ and therefore has norm at least $d^{(j-1)/p}$ by induction hypothesis.

Case 3 : We are left with the case when all entries in the matrix $\mathbf{X}$ are even integers and there is a column with at least $d/2^p$ non-zero co-ordinates. Therefore, $\mathbf{X}$ has at least $d/2^p$ non-zero even rows. Call these rows $2\mathbf{r}_1, 2\mathbf{r}_2, \dots 2\mathbf{r}_{d/2^p}$. Then

$$\|\mathbf{X}\mathbf{B}_{j-1}^T\|_p^p \geq \sum_{l=1}^{d/2^p} 2^p\|\mathbf{r}_l\mathbf{B}_{j-1}^T\|_p^p \geq d/2^p \cdot 2^p \cdot d^{j-1} = d^j$$

where we again used the induction hypothesis that the non-zero lattice vector $\mathbf{B}_{j-1}\mathbf{r}_l^T$ in lattice $\mathcal{L}_{j-1}$ has norm at least $d^{(j-1)/p}$. ∎

6.3. Hardness Factor and Proof of Theorem 1.1

Theorem 1.1 now follows from Lemmas 6.1, 6.2 and a careful examination of the size of the SVP instance produced.

Note that Theorem 3.1 reduces SAT instance of size n to a CVP instance of size d where $d = n^{C''}$ for some constant C''. Theorem 4.1 then reduces the CVP instance to SVP instance of size $2N \times N$ where $N = d^{14C'k/\eta}$. An easy inductive argument shows that the size of the k-wise augmented tensor product lattice $\mathcal{L}_k$ has size at most $N^{2k} \times N^k$. Lemmas 6.1 and 6.2 imply that $\mathcal{L}_k$ is a gap instance of SVP with gap $\left(\frac{1}{2\gamma^k}\right)^{1/p} = 2^{\delta k}$ for some constant δ.

Thus choosing k to be a large enough constant, we prove an arbitrarily large constant factor hardness for SVP via a polynomial time (randomized) reduction. By choosing $k = (\log d)^{O(\frac{C'}{\epsilon\eta\delta})}$, we prove a factor $2^{(\log N')^{1/2-\epsilon}}$ hardness where N' is the size of the final SVP instance (i.e. $\mathcal{L}_k$). N' is quasi-polynomial in n.

Our reduction has both-sided error and therefore the hardness result holds under the assumption NP $\not\subseteq$ BPP (resp. NP $\not\subseteq$ BPTIME($2^{poly(\log n)}$)). Note however that this assumption is equivalent to NP $\not\subseteq$ RP (resp. NP $\not\subseteq$ RTIME($2^{poly(\log n)}$)), implying a reduction with one-sided error.

7. Conclusion

We have shown that a variant of tensor product can be made to work to boost hardness of SVP. However, our reduction has an undesirable property that before we apply k-wise tensor product, the size of the SVP instance we start with must be N^k. This limits the hardness factor to $2^{(\log n)^{1/2-\epsilon}}$. It would be nice to get around this problem and show an almost polynomial factor hardness.

Getting to a polynomial factor hardness (i.e. n^ϵ) however looks impossible with current techniques. Such a result is not known even for CVP. It is however known that CVP is hard to approximate within a polynomial factor under the conjecture that NP has 2-Prover-1-Round proof system with polynomially small error. It would be nice to prove a similar conditional result for SVP.

Finally, our reduction does not work for ℓ_1 norm and we believe that this case is fundamentally different from ℓ_p norms with $p > 1$. The *magic reduction* as described in Section 2.1, if exists, would prove an almost polynomial factor hardness for every $p \geq 1$.

Among other issues, it would be nice to have a reduction to SVP that is deterministic (even Ajtai's [2] pure NP-hardness reduction is randomized). Also, it would be nice to have a reduction from CVP to SVP that increases the size only by a linear (as opposed to a polynomial) factor. We (the author) currently know of a reduction that gives only a linear blow-up and achieves a hardness factor of $2^{1-3/p}$ for all $p \geq 4$.

8. Acknowledgments

I would like to thank Noga Alon for his help on BCH Codes. Thanks to Oded Regev, Ravi Kumar and Venkatesan Guruswami for their valuable comments on the earlier draft of the paper. Thanks also to Miki Ajtai, D. Siva Kumar, Daniele Miccianco, Sanjeev Arora, Avi Wigderson, Misha Alekhnovich, Muli Safra and Guy Kindler for helpful discussions at various points in time.

References

[1] D. Aharonov, O. Regev. Lattice problems in NP $\cap$ coNP. Manuscript, 2004.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] N. Alon, J. Spencer, P. Erdos. The probabilistic method. Wiley-Interscience Series.

[7] 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.

[8] W. Banaszczyk. New bounds in some transference theorems in the geometry of numbers. Mathematische Annalen, vol. 296, 1993, pp 625-635.

[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_∞ 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] I. Dumer, D. Micciancio, M. Sudan. Hardness of Approximating the Minimum Distance of a Linear Code. In Proc. of the 40^{th} IEEE Symposium on Foundations of Computer Science, 1999.

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

[16] 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.

[17] J. Håstad. Dual vectors and lower bounds for the nearest lattice point problem. Combinatorica, vol. 8, 1988, pp 75-81.

[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] S. Khot. Hardness of approximating the shortest vector problem in high L_p norms. In Proc. of the 44^{th} IEEE Symposium on Foundations of Computer Science, 2003.

[21] 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.
[22] 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.
[23] J.C. Lagarias, A.M. Odlyzko. Solving low-density subset sum problems. Journal of the ACM, 32(1), 1985, pp 229-246
[24] S. Landau, G.L. Miller. Solvability of radicals is in polynomial time. Journal of Computer and Systems Sciences, 30(2), 1985, pp 179-208.
[25] H.W. Lenstra. Integer programming with a fixed number of variables. Tech. Report 81-03, Univ. of Amsterdam, Amstredam, 1981.
[26] A.K. Lenstra, H.W. Lenstra, L. Lovász. Factoring polynomials with rational coefficients. Mathematische Ann., 261, 1982, pp 513-534.
[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, S. Goldwasser. Complexity of lattice problems, A cryptographic perspective. Kluwer Academic Publishers, 2002.
[29] H. Minkowski. Geometrie der zahlen. Leizpig, Tuebner, 1910.
[30] O. Regev, New lattice based cryptographic constructions. To appear in Proc. of the 35^{th} Annual ACM Symposium on the Theory of Computing, 2003.
[31] C.P. Schnorr. A hierarchy of polynomial-time basis reduction algorithms. In Proc. of Conference on Algorithms, Péecs (Hungary), 1985, pp 375-386.
[32] 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.

A. Proof of Theorem 3.1

In this Section, we prove Theorem 3.1. The reduction is from Exact Set Cover. It is known that for any constant $\eta > 0$, there is a polynomial time reduction from SAT to the Set Cover problem such that : If the SAT instance is a YES instance, then there are ηd sets that cover each element of the universe exactly once. If the SAT instance is a NO instance then there is no set-cover of size d. Let the universe for the set cover instance be $[n']$ and the sets be $S_1, S_2, \ldots, S_{n''}$. It holds that $n' = C_1 d$ and $n'' = Cd$ for some constants C_1, C.

Let the matrix $\mathbf{B}_{cvp}$ and vector $\mathbf{t}$ be as shown in Fig. 1. Here Q is a large integer, say $Q = d^{4d}$. The matrix $\mathbf{B}_{cvp}$ has $n' + n'' = C'd$ rows and n'' columns. $\mathbf{B}_{cvp}$ is Q-multiple of the element-set inclusion matrix

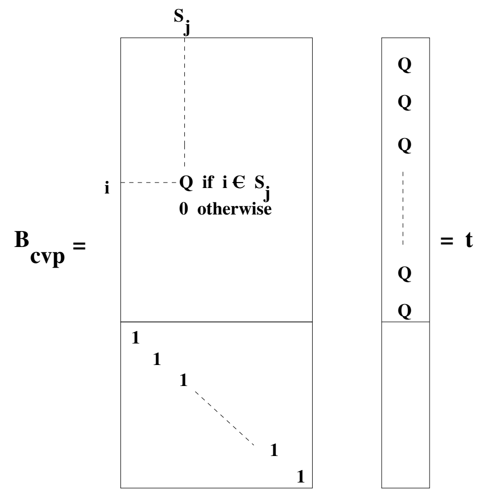

Figure 1. The CVP Instance

appended by an identity matrix. The vector $\mathbf{t}$ has first n' co-ordinates equal to Q and the rest are 0.

Let $\mathbf{y} = (y_1, y_2, \ldots, y_{n''}) \in \mathbb{Z}^{n''}$ be the coefficient vector. If the Set Cover instance has an exact cover consisting of ηd sets, then define $y_j = 1$ if the set S_j is included in the set cover and $y_j = 0$ otherwise. Clearly, $\mathbf{B}_{cvp}\mathbf{y} - \mathbf{t}$ has exactly ηd co-ordinates equal to 1 and the rest are zero.

Now assume there is no set cover of size d. Let $\mathbf{y}$ be an arbitrary coefficient vector and $j_0 \in \mathbb{Z}, j_0 \neq 0$. If at least d of the co-ordinates y_j are non-zero, we are done. Otherwise the family of sets S_j such that $y_j \neq 0$ has fewer than d sets. This family cannot cover the universe and therefore there is a co-ordinate in $\mathbf{B}_{cvp}\mathbf{y} - j_0\mathbf{t}$ that is a non-zero multiple of Q. This co-ordinate corresponds to an element that is not covered.

B. The BCH Lattice

In this section, we present the construction of a lattice based on BCH Codes that is central to this paper. The following theorem is well-known, see e.g. [6, page 229].

Theorem B.1 *Let N, d, h be integers such that $h = \frac{d}{2}\log N$. Then there is a matrix P_{BCH} of size $h \times N$ with $\{0,1\}$-entries such that any d columns of the matrix are linearly independent over $GF(2)$. The parity check matrix for BCH codes has this property and it can be constructed efficiently.*

Let Q be a large integer, say $Q = d^{4d}$. Consider the lattice $\mathcal{L}_{BCH}(\mathbf{B}_{BCH})$ where $\mathbf{B}_{BCH}$ is the matrix as shown in Fig. 2. This is a $(h+N) \times (N+h)$ matrix. Here QP_{BCH} denotes Q-multiple of the matrix P_{BCH}

with parameters N, d, h as in Theorem B.1. The upper right block is a $2Q$-multiple of the $h \times h$ identity matrix. The lower left block is the $N \times N$ identity matrix. The lower right block is zero. Ignore the extra column vector $\mathbf{s}$ for the moment (it is *not* part of matrix $\mathbf{B}_{BCH}$).

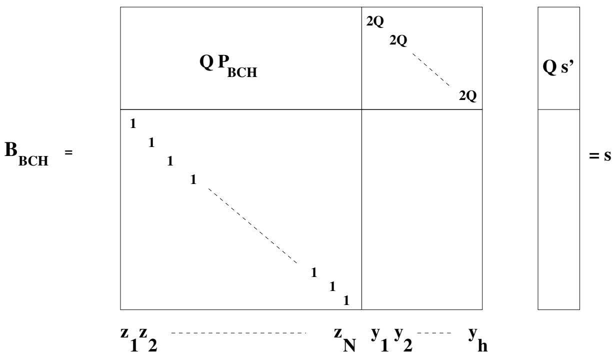

Figure 2. The BCH Lattice

Lemma B.2 *Every non-zero vector in lattice $\mathcal{L}_{BCH}(\mathbf{B}_{BCH})$ has either one co-ordinate with magnitude at least $Q = d^{4d}$, or has at least d non-zero co-ordinates or has all co-ordinates even.*

Proof: Let $z_1, z_2, \ldots, z_N, y_1, y_2, \ldots, y_h$ be the coefficients used to obtain a lattice vector, not all of them zero. If all of $z_1, z_2, \ldots, z_N$ are zero, then clearly, the lattice vector has a co-ordinate that is a non-zero multiple of $2Q$. So assume that not all z_is are zero. If all z_is are even integers, then clearly, the resulting lattice vector has all the co-ordinates even. We are left with the case when there is at least one z_i that is odd. If there are at most d odd z_is, then since the columns of the matrix P_{BCH} are d-wise independent over $GF(2)$, the lattice vector must have a co-ordinate among the top h co-ordinates that is an odd multiple of Q. Lastly, if at least $d+1$ of the z_is are odd, then there are those many non-zero co-ordinates among the lower N co-ordinates. ∎

Lemma B.3 *Let $p > 1$ be fixed and $\eta > 0$ be such that $\frac{1}{2} + \frac{1}{2^p} + \eta < 1$. Let $r = (\frac{1}{2} + \frac{1}{2^p} + \eta)d$. Then it is possible to find a vector $\mathbf{s} \in \mathbb{Z}^{h+N}$ with high probability so that the following holds : there are at least $\frac{1}{100}\binom{N}{r}/2^h$ distinct coefficient vectors $\mathbf{z} \in \mathbb{Z}^{N+h}$ so that*

$$||\mathbf{z}||_p^p \leq (d \log N)^{2p}, \quad ||\mathbf{B}_{BCH}\mathbf{z} - \mathbf{s}||_p^p = r$$

In fact, $\mathbf{B}_{BCH}\mathbf{z} - \mathbf{s}$ is a $\{0,1\}$-vector with exactly r co-ordinates equal to 1.

Proof: First we show the existence of such vector $\mathbf{s}$ and later show that it can be found with high probability. Pick any r columns of the matrix P_{BCH} and sum them up over $GF(2)$. By Piegeon-Hole Principle, there must be a vector $\mathbf{s}' \in \{0,1\}^h$ that occurs at least $\binom{N}{r}/2^h$ times as the sum. Let the desired vector $\mathbf{s}$ be obtained by appending N zero co-ordinates to the vector $Q\mathbf{s}'$ (see Fig. 2).

Let $J \subseteq [N]$ be any set of size r such that

$$\mathbf{v}_J = \sum_{j \in J} (j^{th} \text{ column of } P_{BCH}) \qquad \text{and}$$

$$\mathbf{v}_J \equiv \mathbf{s}' \ \text{ over } \ GF(2)$$

Note that every co-ordinate of $\mathbf{v}_J$ is bounded by r. Define the coefficient vector

$$\mathbf{z}_J = (z_1, z_2, \ldots, z_N, y_1, y_2, \ldots, y_h)$$

as follows : $z_j = 1$ if $j \in J$ and 0 otherwise. Also,

$$y_i = \frac{1}{2}\left((i^{th} \text{ co-ordinate of } \mathbf{s}') - (i^{th} \text{ co-ordinate of } \mathbf{v}_J)\right)$$

Note that y_i are integers because $\mathbf{v}_J \equiv \mathbf{s}'$ over $GF(2)$. It is clear that the vector $\mathbf{B}\mathbf{z}_J - \mathbf{s}$ has exactly r non-zero co-ordinates equal to 1. Also,

$$||\mathbf{z}_J||_p^p = \sum_{j \in J} |z_j|^p + \sum_{i=1}^{h} |y_i|^p$$

$$\leq r + h(r/2)^p \leq d + \frac{d}{2}\log N (d/2)^p \leq (d \log N)^{2p}$$

This proves the existence of the desired vector $\mathbf{s}$. We will show that it can be found with high probability. It suffices to find a vector $\mathbf{s}' \in GF(2)^h$ that occurs at least $\frac{1}{100}\binom{N}{r}/2^h$ times as the sum (over $GF(2)$) of r columns of the matrix P_{BCH}.

Vector $\mathbf{s}' \in GF(2)^h$ is picked as follows. Pick r columns of the matrix P_{BCH} at random and define $\mathbf{s}'$ to be their sum. For any vector $\mathbf{s}' \in GF(2)^h$, let $K_{\mathbf{s}'}$ denote the number of times $\mathbf{s}'$ occurs as the sum of r columns of P_{BCH}. It is clear that the above process picks a particular $\mathbf{s}'$ with probability $K_{\mathbf{s}'}/\binom{N}{r}$. Therefore,

$$\Pr\left[K_{\mathbf{s}'} \leq \frac{1}{100}\binom{N}{r}/2^h\right]$$

$$= \sum_{\mathbf{s}' \in GF(2)^h : K_{\mathbf{s}'} \leq \frac{1}{100}\binom{N}{r}/2^h} \Pr\left[\mathbf{s}' \text{ is picked}\right]$$

$$= \sum_{\mathbf{s}' \in GF(2)^h : K_{\mathbf{s}'} \leq \frac{1}{100}\binom{N}{r}/2^h} \frac{K_{\mathbf{s}'}}{\binom{N}{r}}$$

$$\leq \sum_{\mathbf{s}' \in GF(2)^h : K_{\mathbf{s}'} \leq \frac{1}{100}\binom{N}{r}/2^h} \frac{1}{100 \cdot 2^h} \leq 2^h \frac{1}{100 \cdot 2^h} = \frac{1}{100}$$

∎