Hardness of Approximating the Shortest Vector Problem in Lattices

SUBHASH KHOT

Georgia Institute of Technology, Atlanta, Georgia

Abstract. Let $p > 1$ be any fixed real. We show that assuming NP $\not\subseteq$ RP, there is no polynomial time algorithm that approximates the Shortest Vector Problem (SVP) in ℓ_p norm within a constant factor. Under the stronger assumption NP $\not\subseteq$ RTIME($2^{poly(\log n)}$), we show that there is no polynomial-time algorithm with approximation ratio $2^{(\log n)^{1/2-\epsilon}}$ where n is the dimension of the lattice and $\epsilon > 0$ is an arbitrarily small constant.

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 $2^{(\log n)^{1/2-\epsilon}}$.

Categories and Subject Descriptors: F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Approximation algorithms, cryptography, hardness of approximation, lattices, shortest vector problem

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 nonzero vector

A preliminary version of this article appeared in *Proceedings of the IEEE Symposium on Foundations of Computer Science* (Rome, Italy), IEEE Computer Society Press, Los Alamitos, CA, 2004.

This work was done while S. Khot was the Institute for Advanced Study, Princeton, NJ 08544.

This material is based upon work supported by the National Science Foundation (NSF) under agreement no. DMS-0111298. Any opinions, findings and conclusions or recommendations expresses in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

Author's address: Georgia Institute of Technology, College of Computing, Atlanta, GA 30332, e-mail: khot@cc.gatech.edu.

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 a book by Micciancio and Goldwasser [2002] and an expository article by Kumar and Sivakumar [2001].

The Shortest Vector Problem has been studied since the time of Gauss [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 [1910] deals with the existence of short non-zero vectors in lattices. In a celebrated result, Lenstra et al. [1982] gave a polynomial time algorithm for approximating SVP within factor $2^{n/2}$. This algorithm has numerous applications, for example, factoring rational polynomials [Lenstra et al. 1982], breaking knapsack-based codes [Lagarias and Odlyzko 1985], checking the solvability by radicals [Landau and Miller 1985] and integer programming in a fixed number of variables [Lenstra et al. 1982; Lenstra 1981; Kannan 1983]. Schnorr [1987] 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, e.g., Kannan [1987] and Ajtai et al. [2001]. The latter paper also gives a polynomial-time $2^{n\log\log n/\log n}$ factor approximation, an improvement over Schnorr's algorithm.

van Emde Boas [1981] 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 open problem for a long time. A breakthrough result by Ajtai [1998] finally showed that SVP is NP-hard under randomized reductions. Cai and Nerurkar [1999] improved Ajtai's result to a hardness of approximation result showing a hardness factor of $(1+\frac{1}{n^\epsilon})$. Another breakthrough by Micciancio [2000] 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 exponential approximation factors achieved in Lenstra et al. [1982], Schnorr [1987], and Ajtai et al. [2001].

Showing hardness of approximation results for SVP was greatly motivated by Ajtai's [1996] discovery of worst-case to average-case reduction for SVP and subsequent construction of a lattice-based public key cryptosystem by Ajtai and Dwork [1997]. Ajtai showed that if there is a randomized polynomial time algorithm for solving (exact) 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). In other words, if approximating SVP within factor n^c is hard in the worst case, then solving SVP exactly is hard on average. Based on this reduction, Ajtai and Dwork [1997] constructed a public-key cryptosystem whose security depends on (conjectured) worst-case hardness of approximating SVP (cryptography in general relies on average-case hardness of problems, but for SVP, it is same as worst-case hardness via Ajtai's reduction). The constant c was noted to be 19 in Cai [2003], and brought down to $9+\epsilon$ by Cai and Nerurkar [1997] and then to $4+\epsilon$ by Cai [2003]. Recently, Regev [2003] gave an alternate construction of a public key cryptosystem based on $n^{1.5}$-hardness of SVP (actually all these results assume hardness of a variant called unique-SVP, see Regev [2003] for its definition). Thus, in principle, one could show that approximating SVP within

factor $n^{1.5}$ is NP-hard, and it would imply cryptographic primitives whose security relies on (widely believed) conjecture that P $\neq$ NP, attaining the holy grail of cryptography! Unfortunately, there are barriers to showing such strong hardness results. In fact, showing factor n NP-hardness would imply that NP = coNP [Lagarias et al. 1990; Hastad 1988; Banaszczyk 1993] and showing factor $\sqrt{n/O(\log n)}$ NP-hardness would imply that coNP $\subseteq$ AM [Goldreich and Goldwasser 2000] (and therefore polynomial hierachy would collapse in both cases). Recently, Aharonov and Regev [2004] showed that factor $\sqrt{n}$ NP-hardness for SVP 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. Goldreich et al. [1999] give a Turing reduction from CVP to SVP, showing that any hardness for SVP implies the same hardness for CVP (but not vice-versa). CVP was shown to be NP-hard by van Emde Boas [1981]. Arora et al. [1997] 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 et al. [1998] (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 [2003] showed factor $n^{1/\log\log n}$ NP-hardness for both these problems.

1.1. OUR RESULT. In this article, we improve on all previous hardness results [Micciancio 2000; Khot 2003] for SVP in ℓ_p norms with $1 < p < \infty$ (Khot [2003] shows, for every $\epsilon > 0$, factor $p^{1-\epsilon}$ NP-hardness for SVP in ℓ_p norm for all large enough $p = p(\epsilon)$). We show that

THEOREM 1.1. *Let $p > 1$ be any fixed real. Assuming* NP $\not\subseteq$ RP*, there is no polynomial time algorithm that approximates the Shortest Vector Problem in ℓ_p norm within (any) constant factor. Assuming* NP $\not\subseteq$ RTIME($2^{poly(\log n)}$)*, there is no polynomial time algortihm that approximates SVP 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, that is, $2^{(\log n)^{1-\epsilon}}$. Our result does not work in ℓ_1 norm for which Micciancio [2000] $2-\epsilon$ hardness result remains the best.

1.2. OVERVIEW OF THE ARTICLE. 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, and (iv) Boosting hardness factor via Augmented Tensor Product.

Step (i) appears in Section 4. Steps (ii) and (iii) appear in Section 5.1 and 5.2 respectively. Section 6 defines the augmented tensor product and Section 7 describes the Step (iv), proving Theorem 1.1. We present some conclusions and open problems in Section 8.

1.3. NOTATION AND PROBLEM DEFINITIONS. All vectors are denoted by boldface 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 nonzero 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 article.

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 et al. [1999]. 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. [1999] 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 article is to figure out special properties of lattices that make (a variant of) the tensor product construction go through. As in Micciancio's [2000] paper, 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 by Khot [2003]. 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 without loss of generality that all lattice vectors have integer co-ordinates) :

(1) If the CVP instance is a YES instance, then there is a nonzero $\{0, 1\}$-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 nonzero lattice vector has at least d nonzero 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 $(1/\gamma)^{k/p}$ (indeed just imitate the corresponding tensor product construction for codes as in Dumer et al. [1999]. In the YES case, the product lattice $\mathcal{L}^k$ will have a $\{0, 1\}$-lattice vector with $\gamma^k d^k$ co-ordinates equal to 1 and in the NO case, every nonzero lattice vector will have at least d^k nonzero co-ordinates). Thus, the tensor product would work provided that in the NO case, every nonzero lattice vector is not only *long*, but also has *many* nonzero co-ordinates.

However, we do not know whether such a reduction exists. In this article, we give a reduction that achieves somewhat weaker properties, but is 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 γ such that $1/2 + 1/2^p < \gamma < 1$. 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 nonzero lattice vector* $\mathbf{Bx}$

—Either has at least d nonzero co-ordinates;

—Or has all co-ordinates even and at least $d/2^p$ of them are nonzero;

—Or has at least one co-ordinate that is enormously large.

In particular, every nonzero 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 nonzero lattice vector has either d nonzero co-ordinates or has all co-ordinates even, $d/2^p$ of them nonzero. Then, it can be shown that the standard tensor product is still sufficient to boost hardness. The augmented variant takes care of precisely the remaining third possibility, when a lattice vector has very few nonzero 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,

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

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 Section 4.

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 article. For example, if we instead use $h = d \log N$, the result in our article 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 Figure 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* lattice vectors satisfying Property (1). Call them *good* vectors. We show that the number of good vectors is at least $\#G$.

—$\mathcal{L}_{int}$ *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. We show that the number of annoying vectors is at most $\#A$.

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, that is, $\#G \gg \#A$. 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. The reduction from CVP to the Intermediate Lattice 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 article. The following reduction is due to Arora et al. [1997]. We include a proof for completeness.

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 nonzero 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 nonzero co-ordinates.*

PROOF. 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 Figure 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 incidence matrix 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 nonzero, 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 nonzero multiple of Q. This co-ordinate corresponds to an element that is not covered. □

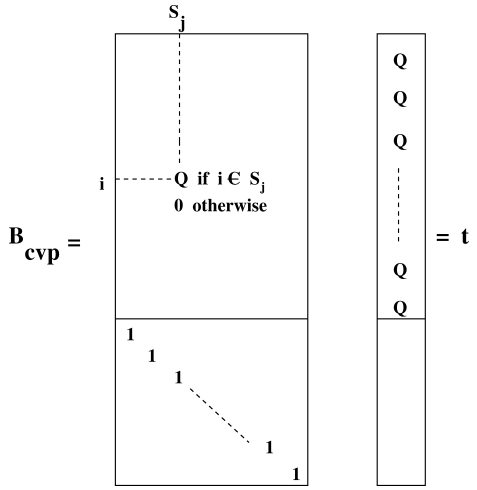

FIG. 1. The CVP instance.

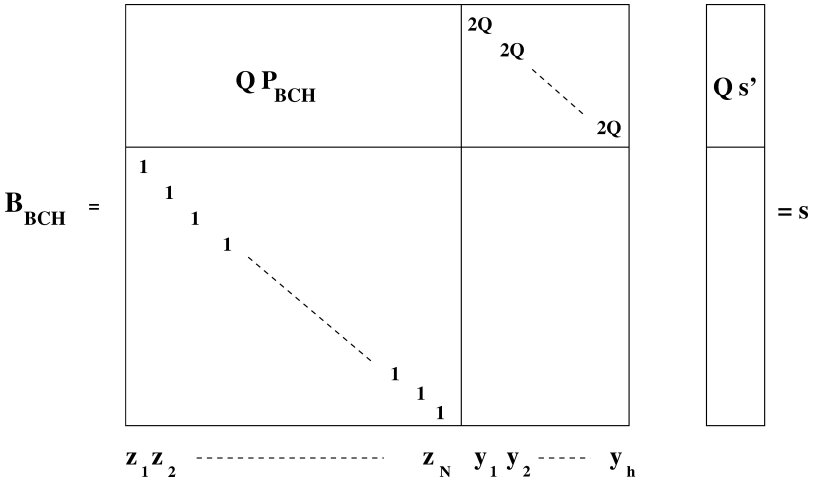

FIG. 2. The BCH lattice.

4. *The BCH Lattice*

In this section, we present the construction of a lattice based on BCH Codes that is central to this article. The following theorem is wellknown, see, for example, Alon et al. [1991, page 229]. Think of $N = d^{O(d)}$ as will be the case later.

THEOREM 4.1. *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 Figure 2. This is a $(h + N) \times (N + h)$ matrix. Here $Q\,P_{BCH}$ denotes Q-multiple of the matrix P_{BCH} with parameters N, d, h as in Theorem 4.1. 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.

LEMMA 4.2. *Every nonzero 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 nonzero 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 nonzero 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 nonzero co-ordinates among the lower N co-ordinates. □

LEMMA 4.3. *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*

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

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 Figure 2).

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

$$\mathbf{v}_J = \sum_{j \in J} (j\text{th column of } P_{BCH}) \qquad \text{and} \quad \mathbf{v}_J \equiv \mathbf{s}' \text{ over } GF(2).$$

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,

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

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

$$\|\mathbf{z}_J\|_p^p = \sum_{j \in J} |z_j|^p + \sum_{i=1}^{h} |y_i|^p \leq r + h\left(\frac{r}{2}\right)^p \leq d + \frac{d}{2}\log N\left(\frac{d}{2}\right)^p \leq (d \log N)^{2p}.$$

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,

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

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

$$\le 2^h \frac{1}{100 \cdot 2^h} = \frac{1}{100}. \quad \square$$

5. *Basic Reduction*

The following theorem gives our basic reduction from CVP to SVP. It is a formal restatement of Theorem 2.1. The theorem gives a constant factor hardness for SVP which is then boosted in Section 7 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 5.1. *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 $k \le d/100$, 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 $N = d^{42C'k/\eta}$. 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 nonzero coefficient vector $\mathbf{x} \in \mathbb{Z}^N$ such that $\|\mathbf{x}\|_p^p \le d^{10p}$ and $\|\mathbf{Bx}\|_p^p \le \gamma d$.*

(3) *If the CVP instance is a NO instance, then with probability at least $9/10$, any nonzero lattice vector $\mathbf{Bx}$ (all its co-ordinates are integers)*
—either has one co-ordinate with magnitude at least d^{20k}
—or has d nonzero co-ordinates
—or has all co-ordinates even and at least $d/2^p$ of them are nonzero.
In particular, $\|\mathbf{Bx}\|_p^p \ge d$.

We prove Theorem 5.1 in two stages. First, we construct an Intermediate Lattice $\mathcal{L}_{int}(\mathbf{B}_{int})$ that: (a) has *many* lattice vectors $\mathbf{B}_{int}\mathbf{x}$ that satisfy Condition (2) (these will be called *good vectors*) and (b) has only a *few* nonzero lattice vectors $\mathbf{B}_{int}\mathbf{x}$ that do not satisfy Condition (3) (these will be called *annoying vectors*).

The desired lattice $\mathcal{L}(\mathbf{B})$ is essentially a random sub-lattice of $\mathcal{L}_{int}(\mathbf{B}_{int})$. It is obtained by introducing a random homogeneous linear constraint on the co-ordinates

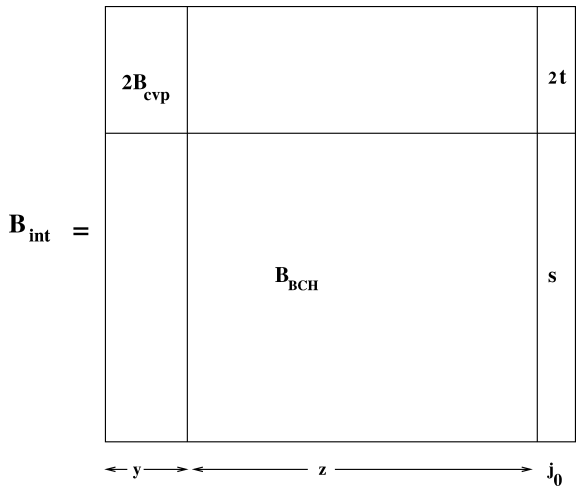

FIG. 3. The intermediate lattice.

of lattice vectors in $\mathcal{L}_{int}(\mathbf{B}_{int})$. With high probability, the sub-lattice $\mathcal{L}$ does not contain any of the annoying vectors in the NO case and contains at least one good vector in the YES case. This proves Theorem 5.1.

5.1. CONSTRUCTING THE INTERMEDIATE LATTICE. Let $(\mathcal{L}_{cvp}(\mathbf{B}_{cvp}), \mathbf{t})$ be the CVP instance given by Theorem 3.1 with parameters (η, C, C', C'', d). Let

$$(\mathcal{L}_{BCH}(\mathbf{B}_{BCH}), \mathbf{s})$$

be the BCH Lattice along with the point $\mathbf{s}$ as in Section 4 and Lemma 4.3. The parameters $N, d, h = \frac{d}{2}\log N$ and $r = (\frac{1}{2} + \frac{1}{2^p} + \eta)d$ are as in Section 4. Let $k \leq d/100$ be a given integer. Let $N = d^{42C'k/\eta}$.

The intermediate lattice $\mathcal{L}_{int}(\mathbf{B}_{int})$ is shown in Figure 3.

The matrix $\mathbf{B}_{int}$ has size $(C'd + h + N) \times (Cd + N + h + 1)$. Let

$$\mathbf{x} = \mathbf{y} \circ \mathbf{z} \circ j_0$$

denote the coefficient vector ($\circ$ denotes concatenation). Thus,

$$\mathbf{B}_{int}\mathbf{x} = \mathbf{B}_{int}(\mathbf{y} \circ \mathbf{z} \circ j_0) = 2\left(\mathbf{B}_{cvp}\mathbf{y} + j_0\mathbf{t}\right) \circ (\mathbf{B}_{BCH}\mathbf{z} + j_0\mathbf{s}).$$

Definition 5.2. Call a nonzero lattice vector $\mathbf{B}_{int}\mathbf{x}$ *good* if

$$\|\mathbf{x}\|_p^p \leq (d\log N)^{2.5p} \quad \text{and} \quad \|\mathbf{B}_{int}\mathbf{x}\|_p^p \leq \gamma d.$$

Definition 5.3. Call a nonzero lattice vector $\mathbf{B}_{int}\mathbf{x}$ *annoying* if it fails to satisfy Condition (3) in Theorem 5.1. In other words, the vector $\mathbf{B}_{int}\mathbf{x}$ is *annoying* if

—All its co-ordinates are bounded by d^{20k} and

—Number of nonzero co-ordinates is fewer than d and

—Either there is an odd co-ordinate or all co-ordinates are even and at most $d/2^p$ of them are nonzero.

LEMMA 5.4. *If the CVP instance is a YES instance, then there are at least* $\frac{1}{100}\binom{N}{r}/2^h$ *good lattice vectors. These good vectors have* $\{0, 1, 2\}$ *co-ordinates and have at least one co-ordinate equal to* 1.

PROOF. Let $\tilde{\mathbf{y}}$ denote the *solution* to the CVP, namely it satisfies Condition (3) of Theorem 3.1. Let $\mathbf{z}$ be any vector that satisfies the hypothesis of Lemma 4.3. As this lemma guarantees, there are at least $\frac{1}{100}\binom{N}{r}/2^h$ such choices for $\mathbf{z}$. Let $j_0 = -1$. For any such $\mathbf{x} = \tilde{\mathbf{y}} \circ \mathbf{z} \circ (-1)$, we have

$$\mathbf{B}_{int}\mathbf{x} = 2(\mathbf{B}_{cvp}\tilde{\mathbf{y}} - \mathbf{t}) \circ (\mathbf{B}_{BCH}\mathbf{z} - \mathbf{s}).$$

Note that $(\mathbf{B}_{cvp}\tilde{\mathbf{y}} - \mathbf{t})$ and $(\mathbf{B}_{BCH}\mathbf{z} - \mathbf{s})$ are $\{0, 1\}$-vectors with exactly ηd and r co-ordinates equal to 1, respectively. Therefore

$$\|\mathbf{B}_{int}\mathbf{x}\|_p^p \le 2^p \eta d + r = 2^p \eta d + \left(\frac{1}{2} + \frac{1}{2^p} + \eta\right) d = \gamma d.$$

Moreover,

$$\|\mathbf{x}\|_p^p \le \|\tilde{\mathbf{y}}\|_p^p + \|\mathbf{z}\|_p^p + 1 \le Cd + (d \log N)^{2p} + 1 \le (d \log N)^{2.5p}.$$

This proves that the lattice vector $\mathbf{B}_{int}\mathbf{x}$ is good. Clearly, its all co-ordinates are $\{0, 1, 2\}$ and exactly r of them are equal to 1. □

LEMMA 5.5. *If the CVP instance is a NO instance, then there are at most* $\binom{N+h}{d/2^p} d^{40C'kd}$ *annoying lattice vectors in the lattice* $\mathcal{L}_{int}$.

PROOF. Let $\mathbf{B}_{int}\mathbf{x}$ be an annoying vector with $\mathbf{x} = \mathbf{y} \circ \mathbf{z} \circ j_0$. We have

$$\mathbf{B}_{int}\mathbf{x} = 2(\mathbf{B}_{cvp}\mathbf{y} + j_0\mathbf{t}) \circ (\mathbf{B}_{BCH}\mathbf{z} + j_0\mathbf{s}).$$

We first prove that $j_0 = 0$. Otherwise, by Condition (4) of Theorem 3.1, the vector $\mathbf{B}_{cvp}\mathbf{y} + j_0\mathbf{t}$ either has one co-ordinate equal to $d^{4d} > d^{20k}$ or has at least d nonzero co-ordinates. In either case, $\mathbf{B}_{int}\mathbf{x}$ will not be an annoying vector.

Thus, assume $j_0 = 0$. Therefore,

$$\mathbf{B}_{int}\mathbf{x} = 2\,\mathbf{B}_{cvp}\mathbf{y} \circ \mathbf{B}_{BCH}\mathbf{z}.$$

We will bound the number of choices for $\mathbf{B}_{cvp}\mathbf{y}$ by $(2d^{20k})^{C'd}$ and the number of choices for $\mathbf{B}_{BCH}\mathbf{z}$ by $\binom{N+h}{d/2^p}(d^{20k})^{d/2^p}$. The product of these two bounds gives the desired bound on the number of annoying lattice vectors.

Since $\mathbf{B}_{int}\mathbf{x}$ is annoying, all co-ordinates of $\mathbf{B}_{cvp}\mathbf{y}$ are bounded by d^{20k}. Since $\mathbf{B}_{cvp}\mathbf{y}$ has only $C'd$ co-ordinates, the number of choices for $\mathbf{B}_{cvp}\mathbf{y}$ is at most $(2d^{20k})^{C'd}$.

The vector $\mathbf{B}_{BCH}\mathbf{z}$ is a lattice vector in $\mathcal{L}_{BCH}$. Now combine Lemma 4.2 and the assumption that $\mathbf{B}_{int}\mathbf{x}$ is annoying. This implies that all the co-ordinates of $\mathbf{B}_{BCH}\mathbf{z}$ are even, bounded by d^{20k} and at most $d/2^p$ of them are non-zero. The vector $\mathbf{B}_{BCH}\mathbf{z}$ has $N + h$ co-ordinates and therefore the number of choices for $\mathbf{B}_{BCH}\mathbf{z}$ is bounded by $\binom{N+h}{d/2^p}(d^{20k})^{d/2^p}$. □

5.2. FINAL CONSTRUCTION AND PROVING THEOREM 5.1. Now we will construct the lattice $\mathcal{L}(\mathbf{B})$ as claimed in Theorem 5.1. The first step is to show that the number of good vectors (in YES case) is much larger than the number of annoying vectors (in NO case).

By Lemma 5.4, the number of good vectors in YES case is at least

$$\frac{1}{100}\binom{N}{r}/2^h = \frac{1}{100}\binom{N}{(1/2 + 1/2^p + \eta)d}/2^{d/2\log N} \ge \frac{N^{(1/2+1/2^p+\eta)d}}{d^d \cdot N^{d/2}}$$
$$= N^{(1/2^p+\eta)d}/d^d = \#G.$$

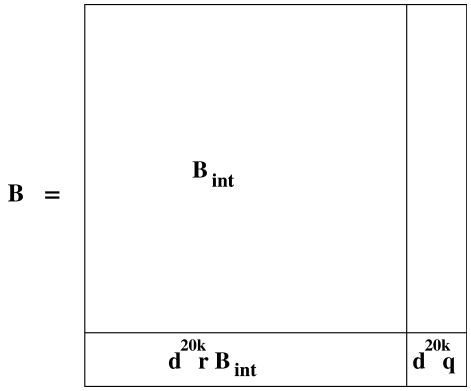

FIG. 4. The final lattice.

By Lemma 5.5, the number of annoying vectors in the NO case is at most

$$\binom{N+h}{d/2^p} d^{40C'kd} \leq (2N)^{d/2^p} d^{40C'kd} \leq N^{d/2^p} d^{41C'kd} = \#A.$$

Since $N = d^{42C'k/\eta}$, we have $\#G \geq 10^5 \#A$.

5.2.1. *Choosing the Prime.* Choose any prime q in the interval $[100\#A, \#G/100]$. This can be done by picking an integer at random in this interval and checking whether it is a prime and repeating the procedure until we find one. By Prime Number Theorem, a fraction $1/O(\log \#G) \geq 1/d^2$ of the integers in this interval are primes and therefore such a prime q can be found with high probability in polynomial time.

5.2.2. *Defining the Lattice* $\mathcal{L}(\mathbf{B})$. Let $\mathcal{L}_{int}(\mathbf{B}_{int})$ be the Intermediate Lattice constructed in Section 5.1. The matrix $\mathbf{B}_{int}$ has size $(C'd + h + N) \times (Cd + N + h + 1)$. The matrix $\mathbf{B}$ for the final lattice $\mathcal{L}$ is defined as in Figure 4. It has one more column and one more row than the matrix $\mathbf{B}_{int}$.

Here $\mathbf{r}$ is a integer row vector with $(C'd + h + N)$ co-ordinates each of which is chosen randomly from the range $[0, q-1]$ and q is the prime picked. Let the coefficient vector for the lattice $\mathcal{L}(\mathbf{B})$ be $\mathbf{x}' = \mathbf{x} \circ l_0$.

The following two lemmas prove Theorem 5.1. The first lemma *filters out* all the annoying vectors.

LEMMA 5.6. *Suppose the CVP instance is a NO instance. Then, with high probability (over the choice of the random row vector* $\mathbf{r}$*), every nonzero lattice vector* $\mathbf{Bx}'$ *in lattice* $\mathcal{L}(\mathbf{B})$

—either has d nonzero co-ordinates

—or has all co-ordinates even and at least $d/2^p$ *of them are nonzero*

—or has one co-ordinate with magnitude at least d^{20k}.

PROOF. Let $\mathbf{Bx}' = \mathbf{B}(\mathbf{x} \circ l_0)$ be any nonzero lattice vector. Using definition of $\mathbf{B}$, we have

$$\mathbf{Bx}' = \mathbf{B}_{int}\mathbf{x} \circ d^{20k}(\mathbf{rB}_{int}\mathbf{x} + l_0 q).$$

The last co-ordinate is a multiple of d^{20k}. So if it is nonzero, we are done. So we may assume that

$$\mathbf{r}(\mathbf{B}_{int}\mathbf{x}) \equiv 0 \pmod{q}. \tag{1}$$

If the vector $\mathbf{B}_{int}\mathbf{x}$ does not satisfy any of the three conditions in the hypothesis of the lemma, it must be an annoying vector. In particular, it is a nonzero integer vector with all co-ordinates bounded by d^{20k} and hence, it is a nonzero vector $mod\ q$. For a random vector $\mathbf{r}$, the probability that $\mathbf{r}(\mathbf{B}_{int}\mathbf{x}) \equiv 0\ (mod\ q)$ is exactly $1/q$. Since there are at most $\#A$ annoying vectors and $q > 100\#A$, with probability $99/100$, Eq. (1) fails for every annoying vector. Thus, every annoying vector gets *killed*. □

LEMMA 5.7. *Suppose the CVP instance is a YES instance. Then, with high probability (over the choice of the random row vector* $\mathbf{r}$*), there exists a lattice vector* $\mathbf{B}\mathbf{x}'$ *in lattice* $\mathcal{L}(\mathbf{B})$ *such that*

$$\|\mathbf{x}'\|_p^p \le d^{10p}, \qquad \|\mathbf{B}\mathbf{x}'\|_p^p \le \gamma d.$$

PROOF. We know that there are at least $\#G \ge 100q$ good vectors, namely, vectors $\mathbf{B}_{int}\mathbf{x}$ such that

$$\|\mathbf{x}\|_p^p \le (d \log N)^{2.5p}, \quad \|\mathbf{B}_{int}\mathbf{x}\|_p^p \le \gamma d.$$

All good vectors $\mathbf{B}_{int}\mathbf{x}$ have $\{0, 1, 2\}$ co-ordinates with at least one co-ordinate equal to 1. Therefore, any two of them are linearly independent $mod\ q$. Using Lemma 5.8 (see below), with probability $99/100$, there exists a good vector $\mathbf{B}_{int}\mathbf{x}^*$ such that

$$\mathbf{r}(\mathbf{B}_{int}\mathbf{x}^*) \equiv 0 \pmod{q}.$$

Co-ordinates of $\mathbf{r}$ are bounded by q and $\|\mathbf{B}_{int}\mathbf{x}^*\|_1 \le \gamma d$. Hence

$$\mathbf{r}(\mathbf{B}_{int}\mathbf{x}^*) = l_0 q \ \text{ for some } |l_0| \le \gamma d.$$

Define the coefficient vector $\mathbf{x}' = \mathbf{x}^* \circ (-l_0)$. It follows that

$$\mathbf{B}\mathbf{x}' = \mathbf{B}_{int}\mathbf{x}^* \ \circ\ d^{20k}(\mathbf{r}\mathbf{B}_{int}\mathbf{x}^* - l_0 q) = \mathbf{B}_{int}\mathbf{x}^* \circ 0.$$

Therefore, $\|\mathbf{B}\mathbf{x}'\|_p^p = \|\mathbf{B}_{int}\mathbf{x}^*\|_p^p \le \gamma d$ and

$$\|\mathbf{x}'\|_p^p = \|\mathbf{x}^*\|_p^p + |l_0|^p \le (d \log N)^{2.5p} + (\gamma d)^p \le d^{10p}.$$

We used the setting of parameters $N = d^{O(d)}$, so $(d \log N)^{2.5p} \ll d^{10p}$. □

LEMMA 5.8. *Let* q *be a prime and* $\mathcal{S}$ *be a collection of integer vectors such that any two of them are linearly independent mod* q*. Pick a random row vector* $\mathbf{r}$ *mod* q*. Then*

$$\Pr[\exists\, \mathbf{v} \in \mathcal{S} \ \textit{such that}\ \mathbf{r} \cdot \mathbf{v} \equiv 0 \ mod\ q] \ \ge\ 1 - \frac{q}{|\mathcal{S}|}.$$

PROOF. For any fixed vector $\mathbf{v} \in \mathcal{S}$, the probability that $\mathbf{r} \cdot \mathbf{v} \equiv 0\ (mod\ q)$ is exactly $1/q$ and these events are pairwise independent. Now apply Chebyshev's inequality. □

6. *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.

6.1. 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

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

6.2. 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

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

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

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

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

$$\mathbf{B}'' = \mathbf{B} \otimes \begin{bmatrix} \alpha\mathbf{I} \\ \hline \mathbf{B}' \end{bmatrix} \qquad \mathbf{I} = N' \times N' \text{ identity matrix.}$$

Thus,

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

7. *Boosting the SVP Hardness Factor*

In this section, we boost the hardness factor for SVP given by Theorem 5.1. This proves Theorem 1.1.

Let $\mathcal{L}(\mathbf{B})$ be the lattice given by reduction of Theorem 5.1 and let $\alpha = 1/d^{10k}$. For $1 \le j \le k$, define lattices $\mathcal{L}_j(\mathbf{B}_j)$ inductively as follows:

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

$$\mathcal{L}_j = \mathcal{L} \otimes_\alpha^A \mathcal{L}_{j-1}, \quad \mathbf{B}_j = \mathbf{B} \otimes \begin{bmatrix} \alpha\mathbf{I} \\ \hline \mathbf{B}_{j-1} \end{bmatrix}. \tag{2}$$

7.1. THE YES CASE. Assume that the CVP instance is a YES instance. Theorem 5.1 guarantees that there is a nonzero coefficient vector $\mathbf{x}$ such that

$$\|\mathbf{B}\mathbf{x}\|_p^p \le \gamma d, \qquad \|\mathbf{x}\|_p^p \le d^{10p}.$$

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

$$\|\mathbf{x}^{\otimes j}\|_p^p = \|\mathbf{x}\|_p^{pj} \leq d^{10pj}.$$

LEMMA 7.1. *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 nonzero 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 Eq. (2) and noting that $\alpha = 1/d^{10k}$, we have

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

$$\leq \gamma d \left(\alpha^p d^{10p(j-1)} + \sum_{l=1}^{j-1} \gamma^l d^l\right) \leq \gamma d \left(1 + \sum_{l=1}^{j-1} \gamma^l d^l\right) = \sum_{l=1}^{j} \gamma^l d^l. \quad \square$$

7.2. 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 5.1 guarantees that every nonzero vector in $\mathcal{L}(\mathbf{B})$

—either has d nonzero co-ordinates

—or has all co-ordinates even and at least $d/2^p$ of them are nonzero

—or has one co-ordinate with magnitude at least d^{20k}.

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

LEMMA 7.2. *For $1 \leq j \leq k$, the length of any nonzero 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 nonzero 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

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

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

$$\|\mathbf{B}_j\mathbf{x}\|_p^p = \left\| \alpha\mathbf{B}\mathbf{x}_1 \circ \alpha\mathbf{B}\mathbf{x}_2 \circ \cdots \circ \alpha\mathbf{B}\mathbf{x}_{N'} \circ \mathbf{B}\mathbf{x}\mathbf{B}_{j-1}^T \right\|_p^p \tag{3}$$

We consider three 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^{20k}, then this co-ordinate alone contributes $\alpha^p d^{20kp} \geq d^{kp} \geq d^j$ towards the quantity (3) and we are done. (This is where the augmented tensor product is used. The rest of the proof is similar to the boosting proof via standard tensor product).

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

nonzero. We use the bound

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

We write the right-hand side as

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

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 nonzero co-ordinates. Then, the matrix $\mathbf{X}$ has at least d nonzero rows. Call these rows $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_d$. Then

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

since every vector $\mathbf{B}_{j-1}\mathbf{r}_l^T$ is a nonzero 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 $d/2^p$ nonzero co-ordinates. Therefore, $\mathbf{X}$ has at least $d/2^p$ nonzero even rows. Call these rows $2\mathbf{r}_1, 2\mathbf{r}_2, \ldots, 2\mathbf{r}_{d/2^p}$. Then

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

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

7.3. HARDNESS FACTOR AND PROOF OF THEOREM 1.1. Theorem 1.1 now follows from Lemmas 7.1, 7.2 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 5.1 then reduces the CVP instance to SVP instance of size $2N \times N$ where $N = d^{42C'k/\eta}$. An easy inductive argument shows that the size of the k-wise augmented tensor product lattice $\mathcal{L}_k$ has size at most $N' = N^{2k} \times N^k$. Lemmas 7.1 and 7.2 imply that $\mathcal{L}_k$ is a gap instance of SVP with gap $(\frac{1}{2\gamma^k})^{1/p} = 2^{\Omega(k)}$. Expressing the size of the final SVP instace in terms of initial SAT instance, we get $N' \leq n^{O(C''C'k^2/\eta)}$. Absorbing the constants C'', C', η in the O-notation, we get a hardness factor of $2^{\Omega(k)}$ via a reduction that runs in time $n^{O(k^2)}$.

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 $k = (\log n)^{1/\epsilon}$, we prove a factor $2^{(\log N')^{1/2-\epsilon}}$ hardness where the size N' of the final SVP instance is quasi-polynomial in n.

8. *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 (folklore) 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 [Ajtai 1998] pure NP-hardness reduction is randomized). Our reduction is randomized at two places, namely Lemma 4.3 and Lemma 5.6, and we believe that it would be difficult to derandomize either of them. 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 $p \geq 4$.

ACKNOWLEDGMENT. I would like to thank Noga Alon for his help on BCH Codes. Thanks to Oded Regev, Ravi Kumar, Venkatesan Guruswami and anonymous referees for their valuable comments on the earlier drafts 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

AHARONOV, D., AND REGEV, O. 2004. Lattice problems in np ∩ conp. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA.

AJTAI, M. 1996. Generating hard instances of lattice problems. In *Proceedings of the 28th ACM Symposium on the Theory of Computing*. ACM, New York, 99–108.

AJTAI, M. 1998. The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proceedings of the 30th ACM Symposium on the Theory of Computing*. ACM, New York, 10–19.

AJTAI, M., AND DWORK, C. 1997. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the 29th ACM Symposium on the Theory of Computing*. ACM, New York, 284–293.

AJTAI, M., KUMAR, R., AND SIVAKUMAR, D. 2001. A sieve algorithm for the shortest lattice vector problem. In *Proceedings of the 33rd ACM Symposium on the Theory of Computing*. ACM, New York, 601–610.

ALON, N., SPENCER, J., AND ERDOS, P. 1991. *The Probabilistic Method*. Wiley-Interscience Series.

ARORA, S., BABAI, L., STERN, J., AND SWEEDYK, E. 1997. The hardness of approximate optima in lattices, codes and systems of linear equations. *J. Comput. Syst. Sci. 54*, 317–331.

BANASZCZYK, W. 1993. New bounds in some transference theorems in the geometry of numbers. *Math. Ann. 296*, 625–635.

CAI, J. 2003. Applications of a new transference theorem to Ajtai's connection factor. *Discr. Appli. Math. 126*, 1, 9–31.

CAI, J., AND NERURKAR, A. 1997. An improved worst-case to average-case connection for lattice problems. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA.

CAI, J., AND NERURKAR, A. 1999. Approximating the SVP to within a factor $(1 + 1/\dim^\epsilon)$ is NP-hard under randomized reductions. *J. Comput. Syst. Sci. 59*, 2, 221–239.

DINUR, I. 2003. Approximating SVP_∞ to within almost polynomial factors is NP-hard. *Combinatorica 23*, 2, 205–243.

DINUR, I., KINDLER, G., AND SAFRA, S. 1998. Approximating CVP to within almost-polynomial factors is NP-hard. In *Proceedings of the 39th IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA.

DUMER, I., MICCIANCIO, D., AND SUDAN, M. 1999. Hardness of approximating the minimum distance of a linear code. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA.

GAUSS, C. 1801. Disquisitiones arithmetica (Leipzig, 1801: art. 171). Yale Univ. Press. (English translation by A. A. Clarke, 1966.)

GOLDREICH, O., AND GOLDWASSER, S. 2000. On the limits of non-approximability of lattice problems. *J. Comput. Syst. Sci. 60*, 3, 540–563.

GOLDREICH, O., MICCIANCIO, D., SAFRA, S., AND SEIFERT, J. 1999. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inf. Proc. Lett. 71*, 2, 55–61.

HASTAD, J. 1988. Dual vectors and lower bounds for the nearest lattice point problem. *Combinatorica 8*, 75–81.

KANNAN, R. 1983. Improved algorithms for integer programming and related lattice problems. In *Proceedings of the 15th ACM Symposium on Theory of Computing*. ACM, New York, 193–206.

KANNAN, R. 1987. Minkowski's convex body theorem and integer programming. *Math. Oper. Res. 12*, 415–440.

KHOT, S. 2003. Hardness of approximating the shortest vector problem in high L_p norms. In *Proceedings of the 44th IEEE Symposium on Foundations of Computer Science*. IEEE Computer Society Press, Los Alamitos, CA.

KUMAR, R., AND SIVAKUMAR, D. 2001. Complexity of SVP—A reader's digest. Complexity Theory Column, L. Hemaspaandra, Ed. SIGACT News *32*, 3.

LAGARIAS, J., LENSTRA, H., AND SCHNORR, C. 1990. Korkine–Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica 10*, 333–348.

LAGARIAS, J., AND ODLYZKO, A. 1985. Solving low-density subset sum problems. *J. ACM 32*, 1, 229–246.

LANDAU, S., AND MILLER, G. 1985. Solvability of radicals is in polynomial time. *J. Comput. Syst. Sci. 30*, 2, 179–208.

LENSTRA, A., LENSTRA, H., AND LOVÁSZ, L. 1982. Factoring polynomials with rational coefficients. *Math. Ann. 261*, 513–534.

LENSTRA, H. 1981. Integer programming with a fixed number of variables. Tech. Report 81-03. Univ. of Amsterdam, Amsterdam, The Netherland.

MICCIANCIO, D. 2000. The shortest vector problem is NP-hard to approximate to within some constant. *SIAM J. Comput. 30*, 6, 2008–2035.

MICCIANCIO, D., AND GOLDWASSER, S. 2002. *Complexity of Lattice Problems, A Cryptographic Perspective*. Kluwer Academic Publishers.

MINKOWSKI, H. 1910. *Geometrie der zahlen*. Tuebner.

REGEV, O. 2003. New lattice based cryptographic constructions. In *Proceedings of the 35th ACM Symposium on the Theory of Computing*. ACM, New York.

SCHNORR, C. 1987. A hierarchy of polynomial-time basis reduction algorithms. *Theoret. Comput. Sci. 53*, 2-3, 201–224.

VAN EMDE BOAS, P. 1981. Another NP-complete problem and the complexity of computing short vectors in a lattice. Tech. Report 81-04. Mathematische Instiut, Univ. of Amsterdam, Amsterdam, The Netherland.

RECEIVED NOVEMBER 2004; REVISED JUNE 2005; ACCEPTED JUNE 2005