THEORY OF COMPUTING, Volume 8 (2012), pp. 513–531
www.theoryofcomputing.org

Tensor-based Hardness of the Shortest Vector Problem to within Almost Polynomial Factors

Ishay Haviv* Oded Regev†

Received: August 26, 2011; published: September 25, 2012.

Abstract: We show that unless $\mathsf{NP} \subseteq \mathsf{RTIME}(2^{\mathrm{poly}(\log n)})$, there is no polynomial-time algorithm approximating the Shortest Vector Problem (SVP) on n-dimensional lattices in the ℓ_p norm ($1 \le p < \infty$) to within a factor of $2^{(\log n)^{1-\varepsilon}}$ for any $\varepsilon > 0$. This improves the previous best factor of $2^{(\log n)^{1/2-\varepsilon}}$ under the same complexity assumption due to Khot (J. ACM, 2005). Under the stronger assumption $\mathsf{NP} \nsubseteq \mathsf{RSUBEXP}$, we obtain a hardness factor of $n^{c/\log\log n}$ for some $c > 0$.

Our proof starts with Khot's SVP instances that are hard to approximate to within some constant. To boost the hardness factor we simply apply the standard tensor product of lattices. The main novelty is in the analysis, where we show that the lattices of Khot behave nicely under tensorization. At the heart of the analysis is a certain matrix inequality which was first used in the context of lattices by de Shalit and Parzanchevski (2006).

ACM Classification: F.2.2, F.1.3, G.1.6

AMS Classification: 68Q17, 52C07, 11H06, 11H31, 05B40

Key words and phrases: lattices, shortest vector problem, NP-hardness, hardness of approximation

*Supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities. Work done while at Tel Aviv University.

†Supported by an Alon Fellowship, by the Binational Science Foundation, by the Israel Science Foundation, by the European Commission under the Integrated Project QAP funded by the IST directorate as Contract Number 015848, and by the European Research Council (ERC) Starting Grant.

 DOI: 10.4086/toc.2012.v008a023

1 Introduction

A *lattice* is a periodic geometric object defined as the set of all integer combinations of some linearly independent vectors in $\mathbb{R}^n$. The interesting combinatorial structure of lattices has been investigated by mathematicians over the last two centuries, and for at least three decades it has also been studied from an asymptotic algorithmic point of view. Roughly speaking, most fundamental problems on lattices are not known to be efficiently solvable. Moreover, there are hardness results showing that such problems cannot be solved by polynomial-time algorithms unless the polynomial-time hierarchy collapses. One of the main motivations for research on the hardness of lattice problems is their applications in cryptography, as was demonstrated by Ajtai [3], who came up with a construction of cryptographic primitives whose security relies on the worst-case hardness of certain lattice problems.

Two main computational problems associated with lattices are the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). In the former, for a lattice given by *some* basis we are supposed to find (the length of) a shortest nonzero vector in the lattice. The problem CVP is an inhomogeneous variant of SVP, in which given a lattice and some target point one has to find (its distance from) the closest lattice point. The hardness of lattice problems partly comes from the fact that there are many possible bases for the same lattice.

In this paper we improve the best hardness result known for SVP. Before presenting our results let us start with an overview of related work.

1.1 Related work

In the early 1980s, Lenstra, Lenstra, and Lovász (LLL) [20] presented the first polynomial-time approximation algorithm for SVP. Their algorithm achieves an approximation factor of $2^{O(n)}$, where n is the dimension of the lattice. Using their algorithm, Babai [7] gave an approximation algorithm for CVP achieving the same approximation factor. A few years later, improved algorithms were presented for both problems, obtaining a slightly sub-exponential approximation factor, namely $2^{O(n(\log\log n)^2/\log n)}$ [31], and this has since been improved slightly [4, 26]. The best algorithm known for solving SVP *exactly* requires exponential running time in n [18, 4, 26]. All the above results hold with respect to any ℓ_p norm $(1 \le p \le \infty)$.

On the hardness side, it was proven in 1981 by van Emde Boas [32] that it is NP-hard to solve SVP exactly in the ℓ_∞ norm. The question of extending this result to other norms, and in particular to the Euclidean norm ℓ_2, remained open until the breakthrough result by Ajtai [2] showing that exact SVP in the ℓ_2 norm is NP-hard under randomized reductions. Then, Cai and Nerurkar [10] obtained hardness of approximation to within $1+n^{-\varepsilon}$ for any $\varepsilon > 0$. The first inapproximability result of SVP to within a factor bounded away from 1 is that of Micciancio [21], who showed that under randomized reductions SVP in the ℓ_p norm is NP-hard to approximate to within any factor smaller than $\sqrt[p]{2}$. For the ℓ_∞ norm, a considerably stronger result is known: Dinur [14] showed that SVP is NP-hard to approximate in the ℓ_∞ norm to within a factor of $n^{c/\log\log n}$ for some constant $c > 0$.

To date, the strongest hardness result known for SVP in the ℓ_p norm is due to Khot [19] who showed NP-hardness of approximation to within arbitrarily large constants under randomized reductions for any $1 < p < \infty$. Furthermore, under randomized quasipolynomial-time reductions (i. e., reductions that run in time $2^{\mathrm{poly}(\log n)}$), the hardness factor becomes $2^{(\log n)^{1/2-\varepsilon}}$ for any $\varepsilon > 0$. Khot speculated there that it might be possible to improve this to $2^{(\log n)^{1-\varepsilon}}$, as this is the hardness factor known for the analogous problem in linear codes [16].

Khot's proof does not work for the ℓ_1 norm. However, it was shown in [30] that for lattice problems, the ℓ_2 norm is the easiest in the following sense: for any $1 \le p \le \infty$, there exists a randomized reduction from lattice problems such as SVP and CVP in the ℓ_2 norm to the respective problem in the ℓ_p norm with essentially the same approximation factor. In particular, this implies that Khot's results also hold for the ℓ_1 norm.

Finally, we mention that a considerably stronger result is known for CVP, namely that for any $1 \le p \le \infty$, it is NP-hard to approximate CVP in the ℓ_p norm to within $n^{c/\log\log n}$ for some constant $c > 0$ [15]. We also mention that in contrast to the above hardness results, it is known that for any $c > 0$, SVP and CVP are unlikely to be NP-hard to approximate to within a $\sqrt{cn/\log n}$ factor, as this would imply the collapse of the polynomial-time hierarchy [17, 1].

1.2 Our results

The main result of this paper improves the best NP-hardness factor known for SVP under randomized quasipolynomial-time reductions. This and two additional hardness results are stated in the following theorem. Here, RTIME is the randomized one-sided error analogue of DTIME. Namely, for a function f we denote by $\mathsf{RTIME}(f(n))$ the class of problems having a probabilistic algorithm running in time $O(f(n))$ on inputs of size n that accepts YES inputs with probability at least $2/3$, and rejects NO inputs with certainty.

Theorem 1.1. *For every $1 \le p \le \infty$ the following holds.*

1. *For every constant $c \ge 1$, there is no polynomial-time algorithm that approximates* SVP *in the ℓ_p norm to within a factor of c unless*
$$\mathsf{NP} \subseteq \mathsf{RP} = \bigcup_{c\ge 1} \mathsf{RTIME}(n^c)\,.$$
2. *For every $\varepsilon > 0$, there is no polynomial-time algorithm that approximates* SVP *on n-dimensional lattices in the ℓ_p norm to within a factor of $2^{(\log n)^{1-\varepsilon}}$ unless*
$$\mathsf{NP} \subseteq \mathsf{RTIME}(2^{\mathrm{poly}(\log n)})\,.$$
3. *There exists a $c > 0$ such that there is no polynomial-time algorithm that approximates* SVP *on n-dimensional lattices in the ℓ_p norm to within a factor of $n^{c/\log\log n}$ unless*
$$\mathsf{NP} \subseteq \mathsf{RSUBEXP} = \bigcap_{\delta>0} \mathsf{RTIME}(2^{n^\delta})\,.$$

Theorem 1.1 improves on the best known hardness result for any $p < \infty$. For $p = \infty$, a better hardness result is already known, namely that for some $c > 0$, approximating to within $n^{c/\log\log n}$ is NP-hard [14]. Moreover, item 1 was already proved by Khot [19] and we provide an alternative proof. We remark that all three items follow from a more general statement (see Theorem 3.1).

1.3 Techniques

A standard method to prove hardness of approximation for large constant or super-constant factors is to first prove hardness for some fixed constant factor, and then *amplify* the constant using some polynomial-time (or quasipolynomial-time) transformation. For example, the *tensor product* of linear codes is used to amplify the NP-hardness of approximating the minimum distance in a linear code of block length n to arbitrarily large constants under polynomial-time reductions and to $2^{(\log n)^{1-\varepsilon}}$ (for any $\varepsilon > 0$) under quasipolynomial-time reductions [16]. This example motivates one to use the tensor product of lattices to increase the hardness factor known for approximating SVP. However, whereas the minimum distance of the k-fold tensor product of a code $\mathcal{C}$ is simply the kth power of the minimum distance of $\mathcal{C}$, the behavior of the length of a shortest nonzero vector in a tensor product of lattices is more complicated and not so well understood.

Khot's approach in [19] was to prove a constant hardness factor for SVP instances that have some "code-like" properties. The rationale is that such lattices might behave in a more predictable way under the tensor product. The construction of these "basic" SVP instances is ingenious, and is based on BCH codes as well as a restriction into a random sublattice. However, even for these code-like lattices, the behavior of the tensor product was not clear. To resolve this issue, Khot introduced a variant of the tensor product, which he called *augmented tensor product*, and using it he showed the hardness factor of $2^{(\log n)^{1/2-\varepsilon}}$. This unusual hardness factor can be seen as a result of the augmented tensor product. In more detail, for the augmented tensor product to work, the dimension of Khot's basic SVP instances grows to $n^{\Theta(k)}$, where k denotes the number of times we intend to apply the augmented tensor product. After applying it, the dimension grows to $n^{\Theta(k^2)}$ and the hardness factor becomes $2^{\Theta(k)}$. This limits the hardness factor as a function of the dimension n to $2^{(\log n)^{1/2-\varepsilon}}$.

Our main contribution is showing that Khot's basic SVP instances *do* behave well under the (standard) tensor product. The proof of this fact uses a new method to analyze vectors in the tensor product of lattices, and is related to a technique used by de Shalit and Parzanchevski [13]. Theorem 1.1 now follows easily: we start with (a minor modification of) Khot's basic SVP instances, which are known to be hard to approximate to within some constant. We then apply the k-fold tensor product for appropriately chosen values of k and obtain instances of dimension $n^{O(k)}$ with hardness $2^{\Omega(k)}$.

1.4 Open questions

Some open problems remain. The most obvious is proving that SVP is hard to approximate to within factors greater than $n^{c/\log\log n}$ under some plausible complexity assumption. Such a result, however, is not known for CVP nor for the minimum distance problem in linear codes, and most likely proving it there first would be easier. An alternative goal is to improve on the $O(\sqrt{n/\log n})$ upper bound beyond which SVP is not believed to be NP-hard [17, 1].

A second open question is whether our complexity assumptions can be weakened. For instance, our $n^{c/\log\log n}$ hardness result is based on the assumption that $\mathsf{NP} \not\subseteq \mathsf{RSUBEXP}$. For CVP, such a hardness factor is known based solely on the assumption $\mathsf{P} \neq \mathsf{NP}$ [15]. Showing something similar for SVP would be very interesting. In fact, coming up with a *deterministic* reduction (even for constant approximation factors) already seems very challenging; all known hardness proofs for SVP in ℓ_p norms, $p < \infty$, use randomized reductions. (We note, though, that [21] does describe a deterministic reduction based on a certain number-theoretic conjecture.) Ideas appearing in the recent NP-hardness proofs of the minimum distance problem in linear codes [11, 6] might be useful. Finally, we mention that a significant step towards derandomization was recently made by Micciancio [23]: he strengthened our results by showing reductions with only one-sided error.

1.5 Outline

The rest of the paper is organized as follows. In Section 2 we gather some background on lattices and on the central tool in this paper—the tensor product of lattices. In Section 3 we prove Theorem 1.1. For the sake of completeness, Section 4 provides a summary of Khot's work [19] together with the minor modifications that we need to introduce.

2 Preliminaries

2.1 Lattices

A *lattice* is a discrete additive subgroup of $\mathbb{R}^n$. Equivalently, it is the set of all integer combinations

$$\mathcal{L}(b_1,\dots,b_m) = \left\{ \sum_{i=1}^{m} x_i b_i : x_i \in \mathbb{Z} \text{ for all } 1 \le i \le m \right\}$$

of m linearly independent vectors $b_1,\dots,b_m$ in $\mathbb{R}^n$ $(n \ge m)$. If the rank m equals the dimension n, then we say that the lattice is *full-rank*. The set $\{b_1,\dots,b_m\}$ is called a *basis* of the lattice. Note that a lattice has many possible bases. We often represent a basis by an $n \times m$ matrix B having the basis vectors as columns, and we say that the basis B *generates* the lattice $\mathcal{L}$. In such case we write $\mathcal{L} = \mathcal{L}(B)$. It is well known and easy to verify that two bases B_1 and B_2 generate the same lattice if and only if $B_1 = B_2 U$ for some *unimodular* matrix $U \in \mathbb{Z}^{m\times m}$ (i. e., a matrix whose entries are all integers and whose determinant is ± 1). The *determinant* of a lattice generated by a basis B is $\det(\mathcal{L}(B)) = \sqrt{\det(B^T B)}$. It is easy to show that the determinant of a lattice is independent of the choice of basis and is thus well-defined. A *sublattice* of $\mathcal{L}$ is a lattice $\mathcal{L}(S) \subseteq \mathcal{L}$ generated by some linearly independent lattice vectors $S = \{s_1,\dots,s_r\} \subseteq \mathcal{L}$. It is known that any integer matrix B can be written as $[H\ 0]U$ where H has full column rank and U is unimodular. One way to achieve this is by using the Hermite Normal Form (see, e. g., [12, Page 67]).

For any $1 \le p < \infty$, the ℓ_p norm of a vector $x \in \mathbb{R}^n$ is defined as $\|x\|_p = \sqrt[p]{\sum_i |x_i|^p}$ and its ℓ_∞ norm is $\|x\|_\infty = \max_i |x_i|$. One basic parameter of a lattice $\mathcal{L}$, denoted by $\lambda_1^{(p)}(\mathcal{L})$, is the ℓ_p norm of a shortest nonzero vector in it. Equivalently, $\lambda_1^{(p)}(\mathcal{L})$ is the minimum ℓ_p distance between two distinct points in the lattice $\mathcal{L}$. This definition can be generalized to define the ith *successive minimum* as the smallest r such that $\mathcal{B}_p(r)$ contains i linearly independent lattice points, where $\mathcal{B}_p(r)$ denotes the ℓ_p ball of radius r centered at the origin. More formally, for any $1 \le p \le \infty$, we define

$$\lambda_i^{(p)}(\mathcal{L}) = \min\Big\{ r : \dim\Big(\operatorname{span}\big(\mathcal{L} \cap \mathcal{B}_p(r)\big)\Big) \ge i \Big\}.$$

We often omit the superscript in $\lambda_i^{(p)}$ when $p = 2$.

In 1896, Hermann Minkowski [28] proved the following classical result, known as Minkowski's First Theorem. We consider here the ℓ_2 norm, although the result has an easy extension to other norms. For a simple proof the reader is referred to [24, Chapter 1, Section 1.3].

Theorem 2.1 (Minkowski's First Theorem)**.** *For any rank-r lattice $\mathcal{L}$,*

$$\det(\mathcal{L}) \geq \left(\frac{\lambda_1(\mathcal{L})}{\sqrt{r}} \right)^r .$$

Our hardness of approximation results will be shown through the promise version GapSVP^p_γ, defined for any $1 \leq p \leq \infty$ and for any approximation factor $\gamma \leq 1$ as follows.

Definition 2.2 (Shortest Vector Problem)**.** An instance of GapSVP^p_γ is a pair (B, s), where B is a lattice basis and s is a number. In YES instances $\lambda_1^{(p)}(\mathcal{L}(B)) \leq \gamma \cdot s$, and in NO instances $\lambda_1^{(p)}(\mathcal{L}(B)) > s$.

2.2 Tensor product of lattices

A central tool in the proof of our results is the *tensor product* of lattices. Let us first recall some basic definitions. For two column vectors u and v of dimensions n_1 and n_2 respectively, we define their tensor product $u \otimes v$ as the $n_1 n_2$-dimensional column vector

$$\begin{pmatrix} u_1 v \\ \vdots \\ u_{n_1} v \end{pmatrix} .$$

If we think of the coordinates of $u \otimes v$ as arranged in an $n_1 \times n_2$ matrix, we obtain the equivalent description of $u \otimes v$ as the matrix $u \cdot v^T$. More generally, any $n_1 n_2$-dimensional vector w can be written as an $n_1 \times n_2$ matrix W. To illustrate the use of this notation, notice that if W is the matrix corresponding to w then

$$\|w\|_2^2 = \operatorname{tr}(W W^T) . \tag{2.1}$$

Finally, for an $n_1 \times m_1$ matrix A and an $n_2 \times m_2$ matrix B, one defines their tensor product $A \otimes B$ as the $n_1 n_2 \times m_1 m_2$ matrix

$$\begin{pmatrix} A_{11} B & \cdots & A_{1 m_1} B \\ \vdots & & \vdots \\ A_{n_1 1} B & \cdots & A_{n_1 m_1} B \end{pmatrix} .$$

Let $\mathcal{L}_1$ be a lattice generated by the $n_1 \times m_1$ matrix B_1 and $\mathcal{L}_2$ be a lattice generated by the $n_2 \times m_2$ matrix B_2. Then the tensor product of $\mathcal{L}_1$ and $\mathcal{L}_2$ is defined as the $n_1 n_2$-dimensional lattice generated by the $n_1 n_2 \times m_1 m_2$ matrix $B_1 \otimes B_2$, and is denoted by $\mathcal{L} = \mathcal{L}_1 \otimes \mathcal{L}_2$. Equivalently, $\mathcal{L}$ is generated by the $m_1 m_2$ vectors obtained by taking the tensor product of two column vectors, one from B_1 and one from B_2. If we think of the vectors in $\mathcal{L}$ as $n_1 \times n_2$ matrices, then we can also define it as

$$\mathcal{L} = \mathcal{L}_1 \otimes \mathcal{L}_2 = \{ B_1 X B_2^T : X \in \mathbb{Z}^{m_1 \times m_2} \} ,$$

with each entry in X corresponding to one of the m_1m_2 generating vectors. We will mainly use this definition in the proof of the main result.

As alluded to before, in the present paper we are interested in the behavior of the shortest nonzero vector in a tensor product of lattices. It is easy to see that for any $1 \le p \le \infty$ and any two lattices $\mathcal{L}_1$ and $\mathcal{L}_2$, we have

$$\lambda_1^{(p)}(\mathcal{L}_1 \otimes \mathcal{L}_2) \le \lambda_1^{(p)}(\mathcal{L}_1) \cdot \lambda_1^{(p)}(\mathcal{L}_2). \tag{2.2}$$

Indeed, any two vectors v_1 and v_2 satisfy $\|v_1 \otimes v_2\|_p = \|v_1\|_p \cdot \|v_2\|_p$. Applying this to shortest nonzero vectors of $\mathcal{L}_1$ and $\mathcal{L}_2$ implies inequality (2.2).

Inequality (2.2) has an analogue for linear codes, with $\lambda_1^{(p)}$ replaced by the minimum distance of the code under the Hamming metric. There, it is not too hard to show that the inequality is in fact an equality: the minimal distance of the tensor product of two linear codes always equals to the product of their minimal distances. However, contrary to what one might expect, there exist lattices for which inequality (2.2) is *strict*. More precisely, for any sufficiently large n there exist n-dimensional lattices $\mathcal{L}_1$ and $\mathcal{L}_2$ satisfying

$$\lambda_1(\mathcal{L}_1 \otimes \mathcal{L}_2) < \lambda_1(\mathcal{L}_1) \cdot \lambda_1(\mathcal{L}_2)\,.$$

The following lemma due to Steinberg shows this fact. Although we do not use this fact later on, the proof is instructive and helps motivate the need for a careful analysis of tensor products. To present this proof we need the notion of a *dual lattice*. For a full-rank lattice $\mathcal{L} \subseteq \mathbb{R}^n$, its *dual* lattice $\mathcal{L}^*$ is defined as

$$\mathcal{L}^* = \left\{x \in \mathbb{R}^n : \langle x, y\rangle \in \mathbb{Z} \text{ for all } y \in \mathcal{L}\right\}.$$

A *self-dual* lattice is one that satisfies $\mathcal{L} = \mathcal{L}^*$. It can be seen that for a full-rank lattice $\mathcal{L}$ generated by a basis B, the basis $(B^{-1})^T$ generates the lattice $\mathcal{L}^*$.

Lemma 2.3 ([27, Page 48])**.** *For any $n \ge 1$ there exists an n-dimensional self-dual lattice $\mathcal{L}$ satisfying $\lambda_1(\mathcal{L} \otimes \mathcal{L}^*) \le \sqrt{n}$ and $\lambda_1(\mathcal{L}) = \lambda_1(\mathcal{L}^*) = \Omega(\sqrt{n})$.*

Proof. We first show that for any full-rank n-dimensional lattice $\mathcal{L}$, $\lambda_1(\mathcal{L} \otimes \mathcal{L}^*) \le \sqrt{n}$. Let $\mathcal{L}$ be a lattice generated by a basis $B = (b_1, \dots, b_n)$. Let $(B^{-1})^T = (\tilde{b_1}, \dots, \tilde{b_n})$ be the basis generating its dual lattice $\mathcal{L}^*$. Now consider the vector $\sum_{i=1}^n b_i \otimes \tilde{b_i} \in \mathcal{L} \otimes \mathcal{L}^*$. Using our matrix notation, this vector can be written as

$$B I_n((B^{-1})^T)^T = BB^{-1} = I_n\,,$$

and clearly has ℓ_2 norm $\sqrt{n}$. To complete the proof, we need to use the (non-trivial) fact that for any $n \ge 1$ there exists a full-rank, n-dimensional and self-dual lattice with shortest nonzero vector of norm $\Omega(\sqrt{n})$. This fact is due to Conway and Thompson; see [27, Page 46] for details. □

3 Proof of results

The following is our main technical result. As we will show later, Theorem 1.1 follows easily by plugging in appropriate values of k.

Theorem 3.1. *For any $1 \le p \le \infty$ there exist $c, C > 0$ such that the following holds. There exists a randomized reduction that takes as input a* SAT *instance and an integer $k \ge 1$ and outputs a* GapSVP_γ^p *instance of dimension n^{Ck} with gap $\gamma = 2^{-ck}$, where n denotes the size of the* SAT *instance. The reduction runs in time polynomial in n^{Ck} and has two-sided error, namely, given a* YES *(resp.,* NO*) instance it outputs a* YES *(resp.,* NO*) instance with probability* $9/10$.

In fact, we will only need to prove this theorem for the case $p = 2$ since, as is easy to see, the general case follows from the following theorem (applied with, say, $\varepsilon = 1/2$).[1]

Theorem 3.2 ([30])**.** *For any $\varepsilon > 0$, $\gamma < 1$ and $1 \le p \le \infty$ there exists a randomized polynomial-time reduction from* $\mathsf{GapSVP}_{\gamma'}^2$ *to* GapSVP_γ^p*, where $\gamma' = (1-\varepsilon)\gamma$.*

3.1 Basic SVP

As already mentioned, our reduction is crucially based on a hardness result of a variant of SVP stemming from Khot's work [19]. Instances of this variant have properties that make it possible to amplify the gap using the tensor product. The following theorem summarizes the hardness result on which our proof is based. For a proof the reader is referred to Section 4.

Theorem 3.3 ([19])**.** *There are a constant $\gamma < 1$ and a polynomial-time randomized reduction from* SAT *to* SVP *outputting a lattice basis B, satisfying $\mathcal{L}(B) \subseteq \mathbb{Z}^n$ for some integer n, and an integer d, such that:*

1. *For any* YES *instance of* SAT*, with probability at least* $9/10$, $\lambda_1(\mathcal{L}(B)) \le \gamma \cdot \sqrt{d}$.
2. *For any* NO *instance of* SAT*, with probability at least* $9/10$*, for every nonzero vector $v \in \mathcal{L}(B)$,*
   - *v has at least d nonzero coordinates, or*
   - *all coordinates of v are even and at least $d/4$ of them are nonzero, or*
   - *all coordinates of v are even and $\|v\|_2 \ge d$.*

   In particular, $\lambda_1(\mathcal{L}(B)) \ge \sqrt{d}$.

3.2 Boosting the SVP hardness factor

As mentioned before, we boost the hardness factor using the tensor product of lattices. For a lattice $\mathcal{L}$ we denote by $\mathcal{L}^{\otimes k}$ the k-fold tensor product of $\mathcal{L}$. An immediate corollary of inequality (2.2) is that if (B, d) is a YES instance of the SVP variant in Theorem 3.3, and $\mathcal{L} = \mathcal{L}(B)$, then

$$\lambda_1(\mathcal{L}^{\otimes k}) \le \gamma^k d^{k/2}. \tag{3.1}$$

For the case in which (B, d) is a NO instance we will show that any nonzero vector of $\mathcal{L}^{\otimes k}$ has norm at least $d^{k/2}$, i. e.,

$$\lambda_1(\mathcal{L}^{\otimes k}) \ge d^{k/2}. \tag{3.2}$$

This yields a gap of γ^k between the two cases. Inequality (3.2) easily follows by induction from the central lemma below, which shows that NO instances "tensor nicely."

[1] We note that our results can be shown directly for any $1 < p < \infty$ without using Theorem 3.2 by essentially the same proof.

Lemma 3.4. *Let (B,d) be a* NO *instance of the* SVP *variant given in Theorem 3.3, and denote by $\mathcal{L}_1$ the lattice generated by the basis B. Then for any lattice $\mathcal{L}_2$,*

$$\lambda_1(\mathcal{L}_1 \otimes \mathcal{L}_2) \geq \sqrt{d} \cdot \lambda_1(\mathcal{L}_2).$$

The proof of this lemma is based on some properties of sublattices of NO instances which are established in the following claim.

Claim 3.5. *Let (B,d) be a* NO *instance of the* SVP *variant given in Theorem 3.3, and let $\mathcal{L} \subseteq \mathcal{L}(B)$ be a sublattice of rank $r > 0$. Then at least one of the following properties holds:*

1. *Every basis matrix of $\mathcal{L}$ has at least d nonzero rows (i. e., rows that are not all zero).*
2. *Every basis matrix of $\mathcal{L}$ contains only even entries and has at least $d/4$ nonzero rows.*
3. $\det(\mathcal{L}) \geq d^{r/2}$.

Proof. Assume that $\mathcal{L}$ does not have either of the first two properties. Our goal is to show that the third property holds. Since the first property does not hold, we have $r < d$ and also that any vector in $\mathcal{L}$ has fewer than d nonzero coordinates. By Theorem 3.3, this implies that $\mathcal{L} \subseteq 2 \cdot \mathbb{Z}^n$. By the assumption that the second property does not hold, there must exist a basis of $\mathcal{L}$ that has fewer than $d/4$ nonzero rows. Therefore, all nonzero vectors in $\mathcal{L}$ have fewer than $d/4$ nonzero coordinates, and hence have norm at least d, again by Theorem 3.3. We conclude that $\lambda_1(\mathcal{L}) \geq d$, and by Minkowski's First Theorem (Theorem 2.1) and $r < d$ we have

$$\det(\mathcal{L}) \geq \left(\frac{\lambda_1(\mathcal{L})}{\sqrt{r}}\right)^r \geq d^{r/2}. \qquad \square$$

Proof of Lemma 3.4. Let v be an arbitrary nonzero vector in $\mathcal{L}_1 \otimes \mathcal{L}_2$. Our goal is to show that $\|v\|_2 \geq \sqrt{d} \cdot \lambda_1(\mathcal{L}_2)$. We can write v in matrix notation as $B_1 X {B_2}^T$, where the integer matrix B_1 is a basis of $\mathcal{L}_1$, B_2 is a basis of $\mathcal{L}_2$, and X is an integer matrix of coefficients. Let U be a unimodular matrix for which $X = [H\ 0]U$, where H is a matrix with full column rank. Thus, the vector v can be written as $B_1[H\ 0](B_2U^T)^T$. Since U^T is also unimodular, the matrices B_2 and B_2U^T generate the same lattice. Now remove from B_2U^T the columns corresponding to the zero columns in $[H\ 0]$ and denote the resulting matrix by B'_2. Furthermore, denote the matrix B_1H by B'_1. Observe that both of the matrices B'_1 and B'_2 are bases of the lattices they generate, i. e., they have full column rank. The vector v equals $B'_1{B'_2}^T$, where $\mathcal{L}'_1 := \mathcal{L}(B'_1) \subseteq \mathcal{L}_1$ and $\mathcal{L}'_2 := \mathcal{L}(B'_2) \subseteq \mathcal{L}_2$.

Claim 3.5 guarantees that the lattice $\mathcal{L}'_1$ defined above has at least one of the three properties mentioned in the claim. We show that $\|v\|_2 \geq \sqrt{d} \cdot \lambda_1(\mathcal{L}'_2)$ in each of these three cases. Then, by the fact that $\lambda_1(\mathcal{L}'_2) \geq \lambda_1(\mathcal{L}_2)$, the lemma will follow.

Case 1: Assume that at least d of the rows in the basis matrix B'_1 are nonzero. Thus, at least d of the rows of $B'_1{B'_2}^T$ are nonzero lattice points from $\mathcal{L}'_2$, and thus

$$\|v\|_2 \geq \sqrt{d} \cdot \lambda_1(\mathcal{L}'_2).$$

Case 2: Assume that the basis matrix B'_1 contains only even entries and has at least $d/4$ nonzero rows. Hence, at least $d/4$ of the rows of $B'_1 {B'_2}^T$ are even multiples of nonzero lattice vectors from $\mathcal{L}'_2$. Therefore, every such row has ℓ_2 norm at least $2 \cdot \lambda_1(\mathcal{L}'_2)$, and it follows that

$$\|v\|_2 \geq \sqrt{\frac{d}{4}} \cdot 2 \cdot \lambda_1(\mathcal{L}'_2) = \sqrt{d} \cdot \lambda_1(\mathcal{L}'_2)\,.$$

The third case is based on the following central claim, which is similar to Proposition 1.1 in [13]. The proof is based on an elementary matrix inequality relating the trace and the determinant of a symmetric positive semidefinite matrix (see, e. g., [9, Page 47]).

Claim 3.6. *Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two rank-r lattices generated by the bases $U = (u_1, \dots, u_r)$ and $W = (w_1, \dots, w_r)$ respectively. Consider the vector $v = \sum_{i=1}^{r} u_i \otimes w_i$ in $\mathcal{L}_1 \otimes \mathcal{L}_2$, which can be written as $UI_rW^T = UW^T$ in matrix notation. Then,*

$$\|v\|_2 \geq \sqrt{r} \cdot \big(\det(\mathcal{L}_1) \cdot \det(\mathcal{L}_2)\big)^{1/r}\,.$$

Proof. Define the two $r \times r$ symmetric positive definite matrices $G_1 = U^T U$ and $G_2 = W^T W$ (known as the Gram matrices of U and W). By the fact that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ for any matrices A and B and by equation (2.1),

$$\|v\|_2^2 = \mathrm{tr}\big((UW^T)(UW^T)^T\big) = \mathrm{tr}(G_1 G_2) = \mathrm{tr}\big(G_1 G_2^{1/2} G_2^{1/2}\big) = \mathrm{tr}\big(G_2^{1/2} G_1 G_2^{1/2}\big)\,,$$

where $G_2^{1/2}$ is the positive square root of G_2. The matrix $G = G_2^{1/2} G_1 G_2^{1/2}$ is also symmetric and positive definite, and as such it has r real and positive eigenvalues. We can thus apply the inequality of arithmetic and geometric means on these eigenvalues to get

$$\|v\|_2^2 = \mathrm{tr}(G) \geq r \det(G)^{1/r} = r \cdot \big(\det(G_1) \cdot \det(G_2)\big)^{1/r}\,.$$

Taking the square root of both sides of this equation completes the proof. □

Equipped with Claim 3.6 we turn to deal with the third case. In order to bound from below the norm of v, we apply the claim to its matrix form $B'_1 {B'_2}^T$ with the lattices $\mathcal{L}'_1$ and $\mathcal{L}'_2$ as above.

Case 3: Assume that the lattice $\mathcal{L}'_1$ satisfies $\det(\mathcal{L}'_1) \geq d^{r/2}$, where r denotes its rank. Combining Claim 3.6 and Minkowski's First Theorem we have that

$$\|v\|_2 \geq \sqrt{r} \cdot \big(\det(\mathcal{L}'_1) \cdot \det(\mathcal{L}'_2)\big)^{1/r} \geq \sqrt{r} \cdot (d^{r/2})^{1/r} \cdot \frac{\lambda_1(\mathcal{L}'_2)}{\sqrt{r}} = \sqrt{d} \cdot \lambda_1(\mathcal{L}'_2)\,,$$

and this completes the proof of the lemma. □

3.3 Proof of the main theorem

Proof of Theorem 3.1. Recall that it suffices to prove the theorem for $p = 2$. Given a SAT instance of size n, we apply the reduction from Theorem 3.3 and obtain in time $\mathrm{poly}(n)$ a pair (B, d) where B is a basis of a $\mathrm{poly}(n)$-dimensional lattice. We then output $(B^{\otimes k}, d^k)$, where $B^{\otimes k}$ is the k-fold tensor product of B, i. e., a basis of the lattice $\mathcal{L}(B)^{\otimes k}$. The dimension of this lattice is $\mathrm{poly}(n^k)$, and combining inequalities (3.1) and (3.2) we infer a gap of 2^{-ck}. □

Proof of Theorem 1.1. For item 1, choose k to be a sufficiently large constant and apply Theorem 3.1. This shows that any constant factor approximation algorithm to SVP implies a two-sided error algorithm for SAT. Using known self-reducibility properties of SAT (see, e. g., [29, Chapter 11]), this also implies a one-sided error polynomial-time algorithm for SAT. For item 2, apply Theorem 3.1 with $k = (\log n)^{1/\varepsilon}$ (where n is the size of the input SAT instance) and let $N = n^{Ck}$ be the dimension of the output lattice. Since

$$k = \left(\frac{\log N}{C}\right)^{\frac{1}{1+\varepsilon}} > \left(\frac{\log N}{C}\right)^{1-\varepsilon},$$

the gap we obtain as a function of the dimension N is $2^{\Omega((\log N)^{1-\varepsilon})}$. Therefore, an algorithm that approximates SVP better than this gap implies a randomized SAT algorithm running in time $2^{\mathrm{poly}(\log n)}$, and hence the desired containment $\mathsf{NP} \subseteq \mathsf{RTIME}(2^{\mathrm{poly}(\log n)})$. Item 3 follows similarly by applying Theorem 3.1 with $k = n^{\delta}$ for all $\delta > 0$. □

4 Proof of Theorem 3.3

In this section we prove Theorem 3.3. The proof is essentially the same as the one in [19] with minor modifications.

4.1 Comparison with Khot's theorem

For the reader familiar with Khot's proof, we now describe how Theorem 3.3 differs from the one in [19]. First, our theorem is only stated for the ℓ_2 norm (since we use Theorem 3.2 to extend the result to other norms). Second, the YES instances of Khot had another property that we do not need here (namely, that the coefficient vector of the short lattice vector is also short). Third, as a result of the augmented tensor product, Khot's theorem includes an extra parameter k that specifies the number of times the lattice is supposed to be tensored with itself. Since we do not use the augmented tensor product, we simply fix k to be some constant. In more detail, we choose the number of columns in the BCH code to be $d^{O(1)}$, as opposed to $d^{O(k)}$. This eventually leads to our improved hardness factor. Finally, the third possibility in our NO case is different from the one in Khot's theorem (which says that there exists a coordinate with absolute value at least $d^{O(k)}$). We note that coordinates with huge values are used several times in Khot's construction in order to effectively restrict a lattice to a subspace. We instead work directly with the restricted lattice, making the reduction somewhat cleaner.

4.2 The proof

The proof of Theorem 3.3 proceeds in three steps. In the first, a variant of the Exact Set Cover problem, which is known to be NP-hard, is reduced to a gap variant of CVP. In the second step we construct a basis B_{int} of a lattice which, informally, contains many short vectors in the YES case, and few short vectors in the NO case. Finally, in the third step we complete the reduction by taking a random sublattice.

4.2.1 Step 1

First, consider the following variant of Exact Set Cover. Let $\eta > 0$ be an arbitrarily small constant. An instance of the problem is a pair (S,d), where $S = \{S_1,\dots,S_{n'}\}$ is a collection of subsets of some universe $[n''] = \{1,\dots,n''\}$, and d is a positive integer. In YES instances, there exists $S' \subseteq S$ of size ηd that covers each element of the universe exactly once. In NO instances, there is no $S' \subseteq S$ of size less than d that covers all elements of the universe. This problem is known to be NP-hard for an arbitrarily small $0 < \eta < 1$ and for $n' = O(d)$ [8]. Moreover, it is easy to see that the problem remains NP-hard if we fix η to be any negative power of 2 and restrict d to be a power of 2. Thus, to prove Theorem 3.3, it suffices to reduce from this problem.

In the first step we use a well-known reduction from the above variant of Exact Set Cover to a variant of CVP. For an instance (S,d) we identify S with the $n'' \times n'$ matrix over $\{0,1\}$ whose columns are the characteristic vectors of the sets in S. The reduction outputs an instance (B_{CVP},t), where B_{CVP} is a basis generating the lattice $\{y \in \mathbb{Z}^{n'} : Sy = 0\}$ and t is some integer vector satisfying $St = -(1,1,\dots,1)$. (If no such t exists, the reduction outputs an arbitrary NO instance.) We note that given S the basis B_{CVP} can be constructed in polynomial time (see, e. g., [22, Lemma 3.1]).

Lemma 4.1. *If (S,d) is a* YES *instance of the above variant of Exact Set Cover, then there is a lattice vector $z \in \mathcal{L}(B_{\mathsf{CVP}})$ such that $z-t$ is a $\{0,1\}$ vector and has exactly ηd coordinates equal to* 1. *If (S,d) is a* NO *instance, then for any lattice vector $z \in \mathcal{L}(B_{\mathsf{CVP}})$ and any nonzero integer j_0, the vector $z + j_0 t$ has at least d nonzero coordinates.*

Proof. If (S,d) is a YES instance then there exists a vector $y \in \{0,1\}^{n'}$ with exactly ηd coordinates equal to 1 for which $Sy = (1,1,\dots,1)$. This implies that $S(y+t) = 0$, so $z = y+t$ is the required lattice vector. On the other hand, if (S,d) is a NO instance, then for any $z \in \mathcal{L}(B_{\mathsf{CVP}})$ we have $S(z+j_0t) = -j_0 \cdot (1,1,\dots,1)$. This implies that the nonzero coordinates of $z + j_0 t$ correspond to a cover $S' \subseteq S$ of all elements in $[n'']$, and hence their number must be at least d. □

4.2.2 Step 2

The second step of the reduction is based on BCH codes, as described in the following theorem.

Theorem 4.2 ([5, Page 255])**.** *Let N,d,h be integers satisfying $h = (d/2)\log_2 N$. Then there exists an efficiently constructible matrix P_{BCH} of size $h \times N$ with $\{0,1\}$ entries such that the rows of the matrix are linearly independent over $GF(2)$ and any d columns of the matrix are linearly independent over $GF(2)$.*

Let B_{BCH} be a basis of the lattice $\{y \in \mathbb{Z}^N : (P_{\text{BCH}})y \equiv 0 \pmod 2\}$. Such a basis can be easily constructed in polynomial time by duality (see the preliminaries in [25]). The next lemma states some properties of this lattice.

Lemma 4.3. *Every nonzero vector in $\mathcal{L}(B_{\mathrm{BCH}})$ either has at least d nonzero coordinates or all of its coordinates are even. Also, for any $r \geq 1$ it is possible to find in polynomial time with probability at least $99/100$ a vector $s \in \{0,1\}^N$, such that there are at least*

$$\frac{1}{100 \cdot 2^h}\binom{N}{r}$$

distinct lattice vectors $z \in \mathcal{L}(B_{\mathrm{BCH}})$ satisfying that $z-s$ is a $\{0,1\}$ vector with exactly r coordinates equal to 1.

Proof. Let $y \in \mathcal{L}(B_{\mathrm{BCH}})$ be a nonzero lattice vector. Observe that if y has an odd coordinate then its odd coordinates correspond to column vectors of P_{BCH} that sum to the zero vector over $GF(2)$. Therefore, their number must be at least d. This proves the first statement.

We now prove the second statement. Consider the set $\mathcal{L}(B_{\mathrm{BCH}}) \cap \{0,1\}^N$ whose size is 2^{N-h} (since as a subset of $GF(2)^N$ it is the kernel of P_{BCH} whose dimension is $N-h$). In order to choose s we first uniformly pick a vector in this set and then we uniformly pick r of its coordinates and flip them. For a vector $s \in \{0,1\}^N$ let A_s denote the number of ways in the above process to obtain s among the $2^{N-h} \cdot \binom{N}{r}$ possible ways. The probability that the chosen s satisfies

$$A_s \leq \frac{1}{100 \cdot 2^h}\binom{N}{r} \quad \text{is} \quad \sum_{s \text{ s.t. } A_s \leq \frac{1}{100 \cdot 2^h}\binom{N}{r}} \frac{A_s}{2^{N-h}\binom{N}{r}} \leq 2^N \cdot \frac{\frac{1}{100\cdot 2^h}\binom{N}{r}}{2^{N-h}\binom{N}{r}} \leq \frac{1}{100}\,,$$

thus with probability at least $99/100$ we obtain an s that satisfies

$$A_s > \frac{1}{100 \cdot 2^h}\binom{N}{r}.$$

It remains to notice that such an s also satisfies the requirement in the statement of the lemma (since from each vector in $\{0,1\}^N$ of Hamming distance r from s we can obtain $z \in \mathcal{L}(B_{\mathrm{BCH}})$ as in the statement by simply adding 2 to a subset of its coordinates). $\square$

We now construct the *intermediate lattice* generated by a basis matrix B_{int} (see Figure 1). Let η be a sufficiently small constant, say $1/128$. Let $r = (3/4+\eta)d$ and choose s as in Lemma 4.3. We choose the parameters of B_{BCH} to be $N = d^{2/\eta}$, d, and $h = (d/2)\log_2 N$. Consider a matrix whose upper left block is $2 \cdot B_{\mathsf{CVP}}$, whose lower right block is B_{BCH}, and whose other entries are zeros. Adding to this matrix the column given by the concatenation of $2 \cdot t$ and s, we obtain the basis matrix B_{int} of the intermediate lattice.

The following two lemmas describe the properties of $\mathcal{L}(B_{\mathrm{int}})$. The first one states that if the CVP instance is a YES instance then $\mathcal{L}(B_{\mathrm{int}})$ contains many short vectors. Define $\gamma = (3/4+5\eta)^{1/2} < 1$. A nonzero lattice vector of $\mathcal{L}(B_{\mathrm{int}})$ is called *good* if it has ℓ_2 norm at most $\gamma \cdot \sqrt{d}$, has $\{0,1,2\}$ coordinates, and has at least one coordinate equal to 1.

Lemma 4.4. *If the* CVP *instance is a* YES *instance and the vector s has the property from Lemma 4.3, then there are at least $\frac{1}{100\cdot 2^h}\binom{N}{r}$ good lattice vectors in $\mathcal{L}(B_{\mathrm{int}})$.*

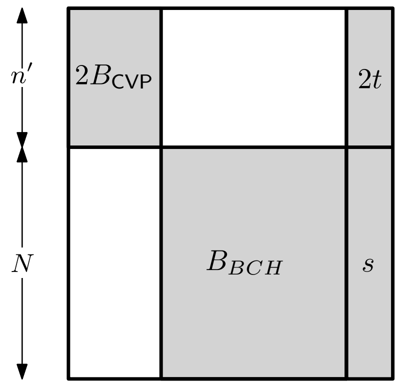

Figure 1: B_{int}

Proof. Assume that the CVP instance is a YES instance. By Lemma 4.1, this implies that there exists y such that $B_{\mathsf{CVP}}y - t$ is a $\{0,1\}$ vector and has exactly ηd coordinates equal to 1. Let s be as in Lemma 4.3, so there are at least $\frac{1}{100\cdot 2^h}\binom{N}{r}$ distinct choices of x for which $(B_{\text{BCH}})x - s$ is a $\{0,1\}$ vector with exactly r coordinates equal to 1. For every such x, the lattice vector[2]

$$B_{\text{int}}(y \circ x \circ (-1)) = \big(2(B_{\mathsf{CVP}}y - t) \circ ((B_{\text{BCH}})x - s)\big)$$

has $\{0,1,2\}$ coordinates, has at least one coordinate equal to 1, and has ℓ_2 norm $\sqrt{4\eta d + r} = \gamma \cdot \sqrt{d}$, as required. □

The second lemma shows that if the CVP instance is a NO instance then $\mathcal{L}(B_{\text{int}})$ contains few vectors that do not have the property from Theorem 3.3, Item 2. We call such vectors *annoying*. In more detail, a lattice vector of $\mathcal{L}(B_{\text{int}})$ is annoying if it satisfies *all* of the following:

- The number of its nonzero coordinates is smaller than d.
- Either it contains an odd coordinate or the number of its nonzero coordinates is smaller than $d/4$.
- Either it contains an odd coordinate or it has norm smaller than d.

Lemma 4.5. *If the* CVP *instance is a* NO *instance, then there are at most* $d^{d/4} \cdot \binom{N+n'}{d/4}$ *annoying lattice vectors in* $\mathcal{L}(B_{\text{int}})$.

Proof. Assume that the CVP instance is a NO instance and let $B_{\text{int}}x$ be an annoying vector with coefficient vector $x = y \circ z \circ (j_0)$. We have

$$B_{\text{int}}x = 2(B_{\mathsf{CVP}}y + j_0 t) \circ (B_{\text{BCH}}z + j_0 s)\,.$$

By Lemma 4.1, if $j_0 \neq 0$ then the vector $B_{\mathsf{CVP}}y + j_0 t$ has at least d nonzero coordinates, so it is not an annoying vector. Thus we can assume that $j_0 = 0$ and therefore $B_{\text{int}}x = 2(B_{\mathsf{CVP}}y) \circ (B_{\text{BCH}}z)$.

[2]We use $\circ$ to denote concatenation of vectors.

Since $B_{\text{int}}x$ is annoying we know that it has fewer than d nonzero coordinates, so by Lemma 4.3 we get that all coordinates of $B_{\text{int}}x$ are even. Again by the definition of an annoying vector, we conclude that fewer than $d/4$ of the coordinates of $B_{\text{int}}x$ are nonzero and all of them have absolute value smaller than d. Thus, we get a bound of $d^{d/4} \cdot \binom{N+n'}{d/4}$ on the number of possible choices for $B_{\text{int}}x$, and this completes the proof of the lemma. □

4.2.3 Step 3

In the third step we construct the final SVP instance as claimed in Theorem 3.3. By Lemma 4.4, the number of good vectors in the YES case is at least

$$\frac{1}{100 \cdot 2^h}\binom{N}{r} = \frac{1}{100 \cdot 2^{(d \log_2 N)/2}}\binom{N}{(3/4+\eta)d} \geq \frac{N^{(3/4+\eta)d}}{100 \cdot d^d \cdot N^{d/2}} = \frac{N^{(1/4+\eta)d}}{100 \cdot d^d} =: G\,.$$

By Lemma 4.5, in the NO case there are at most $A := d^{d/4} \cdot \binom{N+n'}{d/4}$ annoying vectors. By our choice of N and the fact that $n' = O(d)$, for sufficiently large d we have $n' \leq N$ and hence

$$A \leq d^{d/4} \cdot (2N)^{d/4} \leq 10^{-5} \cdot G\,.$$

Choose a prime q in the interval $[100A, G/100]$ and let $w \in \mathbb{Z}^{n'+N}$ be a vector whose coordinates are chosen randomly and uniformly from the range $\{0,\dots,q-1\}$. The final output of the reduction is a basis B of the lattice $\{x \in \mathcal{L}(B_{\text{int}}) : \langle w,x\rangle \equiv 0 \pmod q\}$.

Lemma 4.6. *If the* CVP *instance is a* YES *instance and the vector s has the property from Lemma 4.3, then with probability at least* $99/100$ *over the choice of the vector w, there exists a lattice vector in* $\mathcal{L}(B)$ *with ℓ_2 norm at most $\gamma \cdot \sqrt{d}$.*

Proof. If the CVP instance is a YES instance and s has the property from Lemma 4.3, then by Lemma 4.4 there are at least G good vectors in $\mathcal{L}(B_{\text{int}})$, i.e., vectors with ℓ_2 norm at most $\gamma \cdot \sqrt{d}$, coordinates from $\{0,1,2\}$, and at least one coordinate equal to 1. For each good vector x, consider the event that $\langle w,x\rangle \equiv 0 \pmod q$. Since a good vector is nonzero, we clearly have that each such event occurs with probability $1/q$. Moreover, observe that these vectors are pairwise linearly independent modulo q and therefore these events are pairwise independent. Therefore, using Chebyshev's Inequality, with probability at least $1 - q/G \geq 99/100$, at least one of these events happens, and we are done. □

Lemma 4.7. *If the* CVP *instance is a* NO *instance, then with probability at least* $99/100$ *over the choice of the vector w, for every nonzero lattice vector $v \in \mathcal{L}(B)$,*

- *v has at least d nonzero coordinates, or*
- *all coordinates of v are even and at least $d/4$ of them are nonzero, or*
- *all coordinates of v are even and $\|v\|_2 \geq d$.*

Proof. The probability that a nonzero lattice vector $x \in \mathcal{L}(B_{\text{int}})$ satisfies $\langle w,x\rangle \equiv 0 \pmod q$ is $1/q$. By the union bound, the probability that at least one of the annoying vectors of $\mathcal{L}(B_{\text{int}})$ belongs to $\mathcal{L}(B)$ is at most $A/q \leq 1/100$. Therefore, with probability at least $99/100$, no lattice vector in $\mathcal{L}(B)$ is annoying, and the lemma follows. □

Lemmas 4.6 and 4.7 imply Theorem 3.3. □

Acknowledgements

We thank Daniele Micciancio and Mario Szegedy for useful comments. We also thank an anonymous referee for suggesting to avoid the use of huge coordinates in Khot's proof, which in turn made Claim 3.5 simpler.

References

[1] DORIT AHARONOV AND ODED REGEV: Lattice problems in NP ∩ coNP. *J. ACM*, 52(5):749–765, 2005. Preliminary version in FOCS'04. [doi:10.1145/1089023.1089025] 515, 516

[2] MIKLÓS AJTAI: The shortest vector problem in L_2 is NP-hard for randomized reductions. In *Proc. 30th STOC*, pp. 10–19. ACM Press, 1998. ECCC. [doi:10.1145/276698.276705] 514

[3] MIKLÓS AJTAI: Generating hard instances of lattice problems. In *Complexity of Computations and Proofs*, volume 13 of *Quad. Mat.*, pp. 1–32. Dept. Math., Seconda Univ. Napoli, Caserta, 2004. Preliminary version in STOC'96. 514

[4] MIKLÓS AJTAI, RAVI KUMAR, AND D. SIVAKUMAR: A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd STOC*, pp. 601–610. ACM Press, 2001. [doi:10.1145/380752.380857] 514

[5] NOGA ALON AND JOEL H. SPENCER: *The Probabilistic Method*. Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience, New York, second edition, 2000. [doi:10.1002/0471722154] 524

[6] PER AUSTRIN AND SUBHASH KHOT: A simple deterministic reduction for the gap minimum distance of code problem. In *Proc. 38th Internat. Colloq. on Automata, Languages and Programming (ICALP'11)*, pp. 474–485. Springer, 2011. [doi:10.1007/978-3-642-22006-7_40] 517

[7] LÁSZLÓ BABAI: On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica*, 6(1):1–13, 1986. Preliminary version in STACS'85. [doi:10.1007/BF02579403] 514

[8] MIHIR BELLARE, SHAFI GOLDWASSER, CARSTEN LUND, AND ALEX RUSSELL: Efficient probabilistically checkable proofs and applications to approximations. In *Proc. 25th STOC*, pp. 294–304. ACM Press, 1993. [doi:10.1145/167088.167174] 524

[9] RAJENDRA BHATIA: *Matrix Analysis*. Springer, 1997. 522

[10] JIN-YI CAI AND AJAY NERURKAR: Approximating the SVP to within a factor $(1+1/\dim^{\varepsilon})$ is NP-hard under randomized reductions. *J. Comput. System Sci.*, 59(2):221–239, 1999. Preliminary version at CCC'98. [doi:10.1006/jcss.1999.1649] 514

[11] QI CHENG AND DAQING WAN: A deterministic reduction for the gap minimum distance problem. In *Proc. 41st STOC*, pp. 33–38. ACM Press, 2009. [doi:10.1145/1536414.1536421] 517

[12] HENRI COHEN: *A Course in Computational Algebraic Number Theory*. Volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993. 517

[13] EHUD DE SHALIT AND ORI PARZANCHEVSKI: On tensor products of semistable lattices, 2006. Preprint available at author's home page. 516, 522

[14] IRIT DINUR: Approximating SVP_∞ to within almost-polynomial factors is NP-hard. *Theoret. Comput. Sci.*, 285(1):55–71, 2002. Preliminary version in CIAC'00. [doi:10.1016/S0304-3975(01)00290-0] 514, 515

[15] IRIT DINUR, GUY KINDLER, RAN RAZ, AND SHMUEL SAFRA: Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003. Preliminary version in FOCS'98. [doi:10.1007/s00493-003-0019-y] 515, 516

[16] ILYA DUMER, DANIELE MICCIANCIO, AND MADHU SUDAN: Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Inform. Theory*, 49(1):22–37, 2003. Preliminary version in FOCS'99. [doi:10.1109/TIT.2002.806118] 515, 516

[17] ODED GOLDREICH AND SHAFI GOLDWASSER: On the limits of nonapproximability of lattice problems. *J. Comput. System Sci.*, 60(3):540–563, 2000. Preliminary version in STOC'98. [doi:10.1006/jcss.1999.1686] 515, 516

[18] RAVI KANNAN: Minkowski's convex body theorem and integer programming. *Math. Oper. Res.*, 12(3):415–440, 1987. [doi:10.1287/moor.12.3.415] 514

[19] SUBHASH KHOT: Hardness of approximating the shortest vector problem in lattices. *J. ACM*, 52(5):789–808, 2005. Preliminary version in FOCS'04. [doi:10.1145/1089023.1089027] 514, 515, 516, 517, 520, 523

[20] ARJEN K. LENSTRA, HENDRIK W. LENSTRA, JR., AND LÁSZLÓ LOVÁSZ: Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982. [doi:10.1007/BF01457454] 514

[21] DANIELE MICCIANCIO: The shortest vector in a lattice is hard to approximate to within some constant. *SIAM J. Comput.*, 30(6):2008–2035, 2001. Preliminary version in FOCS'98. [doi:10.1137/S0097539700373039] 514, 516

[22] DANIELE MICCIANCIO: Efficient reductions among lattice problems. In *Proc. 19th Ann. ACM-SIAM Symp. on Discrete Algorithms (SODA'08)*, pp. 84–93. ACM Press, 2008. [ACM:1347082.1347092] 524

[23] DANIELE MICCIANCIO: Inapproximability of the shortest vector problem: Toward a deterministic reduction. *Theory of Computing*, 8(22):487–512, 2012. ECCC. [doi:10.4086/toc.2012.v008a022] 517

[24] DANIELE MICCIANCIO AND SHAFI GOLDWASSER: *Complexity of Lattice Problems: A Cryptographic Perspective*. Volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, MA, 2002. 518

[25] DANIELE MICCIANCIO AND ODED REGEV: Lattice-based cryptography. In D. J. BERNSTEIN AND J. BUCHMANN, editors, *Post-Quantum Cryptography*, pp. 147–191. Springer, 2009. [doi:10.1007/978-3-540-88702-7_5] 524

[26] DANIELE MICCIANCIO AND PANAGIOTIS VOULGARIS: A deterministic single exponential time algorithm for most lattice problems based on Voronoi cell computations. In *Proc. 42nd STOC*, pp. 351–358. ACM Press, 2010. ECCC. [ACM:1806689.1806738] 514

[27] JOHN W. MILNOR AND DALE HUSEMÖLLER: *Symmetric Bilinear Forms*. Springer, Berlin, 1973. 519

[28] HERMANN MINKOWSKI: *Geometrie der Zahlen. I.* B. G. Teubner, Leipzig, 1896. 518

[29] CHRISTOS H. PAPADIMITRIOU: *Computational Complexity*. Addison Wesley Longman, 1994. 523

[30] ODED REGEV AND RICKY ROSEN: Lattice problems and norm embeddings. In *Proc. 38th STOC*, pp. 447–456. ACM Press, 2006. [doi:10.1145/1132516.1132581] 515, 520

[31] CLAUS-PETER SCHNORR: A hierarchy of polynomial time lattice basis reduction algorithms. *Theoret. Comput. Sci.*, 53(2-3):201–224, 1987. [doi:10.1016/0304-3975(87)90064-8] 514

[32] PETER VAN EMDE BOAS: Another NP-complete partition problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, Math Inst., University of Amsterdam, Amsterdam, 1981. Available at author's home page. 514

AUTHORS

Ishay Haviv
School of Computer Science, The Academic College of Tel Aviv—Yaffo
Tel Aviv, Israel

havivish@tau.ac.il

Oded Regev
Professor
Blavatnik School of Computer Science, Tel Aviv University,

and

CNRS, ENS Paris

regev@di.ens.fr
http://www.cs.tau.ac.il/~odedr

ABOUT THE AUTHORS

ISHAY HAVIV graduated from Tel Aviv University in 2011 under the supervision of Oded Regev. His research interests include computational aspects of lattices, coding theory, and other topics in theoretical computer science.

ODED REGEV graduated from Tel Aviv University in 2001 under the supervision of Yossi Azar. He spent two years as a postdoc at the Institute for Advanced Study, Princeton, and one year at the University of California, Berkeley. He is currently with the cryptography group at the École Normale Supérieure, Paris. His research interests include quantum computation, computational aspects of lattices, and other topics in theoretical computer science. He also enjoys photography, especially of his baby girl.