Electronic Colloquium on Computational Complexity, Report No. 20 (2012)

Inapproximability of the Shortest Vector Problem: Toward a Deterministic Reduction

Daniele Micciancio

daniele@cs.ucsd.edu
University of California, San Diego

Abstract

We prove that the Shortest Vector Problem (SVP) on point lattices is NP-hard to approximate for any constant factor under polynomial time reverse unfaithful random reductions. These are probabilistic reductions with one-sided error that produce false negatives with small probability, but are guaranteed not to produce false positives regardless of the value of the randomness used in the reduction process. We also prove inapproximability for quasi-polynomial factors under the same kind of reductions running in subexponential time. Previous hardness results for SVP either incurred 2-sided error, or only proved hardness for some small constant approximation factors. Close similarities between our reduction and recent results on the complexity of analogous problems on linear codes, make our new proof an attractive target for derandomization, paving the road to a possible NP-hardness proof for SVP under deterministic polynomial time reductions.

1 Introduction

Lattices are regular arrangements of points in n-dimensional Euclidean space that arise in several areas of computer science and mathematics. Two central problems in the computational study of point lattices are the Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP). Informally, SVP asks to find the shortest nonzero vector in a point lattice. CVP is the inhomogeneous counterpart of SVP, and asks to find the lattice point closest to a given target. Both SVP and CVP are hard combinatorial problems, and the asymptotically fastest known algorithm to solve them runs in time $2^{O(n)}$ [16].

SVP is the most famous and widely studied problem of the two. It is also the problem for which proving strong intractability results has been most challenging. The NP-hardness of SVP (in the Euclidean norm) was conjectured by van Emde Boas in 1981 [18], but remained an outstanding open problem in computational complexity for almost two decades. In 1998, Ajtai [1] gave a first answer to this problem, proving that solving SVP exactly is NP-hard under *randomized reductions*. This should be contrasted with the inhomogeneous problem, CVP, which admits much simpler NP-hardness proofs [13], has been known to be NP-hard (even under *deterministic* polynomial time reductions) since the early 80s [18], and was proved NP-hard to approximate (again under *deterministic* polynomial time reductions) for factors as large as $n^{1/O(\log\log n)}$ [7]. Proving the NP-hardness of SVP under *deterministic* reductions is still an open problem, even for the exact version of SVP.

Immediately following Ajtai's breakthrough result, the complexity of SVP received renewed attention, leading to several improvements, with the main goal of showing that the problem is hard even in its approximate version. In [1], Ajtai had already observed that hardness for the exact version also implies weak inapproximability results for approximation factors of the form $1 + 1/2^{O(n)}$ that rapidly approach 1 as the lattice dimension n grows. This was slightly improved by Cai and Nerurkar [4] to factors $1 + 1/n^{O(1)}$, still approaching 1 in the limit, though at a lower rate. The first significant inapproximability result for factors bounded away from 1 was shown by Micciancio [14], who proved NP-hardness for any constant factor smaller than $\sqrt{2}$ (independent of the lattice dimension). A nice feature of Micciancio's proof [14] is that it has a

ISSN 1433-8092

very simple and intuitive high level structure. Specifically, in [14] the NP-hardness of SVP is proved by reduction from (a variant of) CVP, through what can be called a "homogenization process" [15]. The idea is roughly the following: if the lattice vector $\mathbf{v} \in \Lambda$ is close to the target $\mathbf{t}$, then $\mathbf{t} - \mathbf{v}$ is a short vector in the lattice generated by Λ and $\mathbf{t}$. So, one can attempt to solve a CVP instance by means of an SVP computation on an augmented lattice. This process, often used as a heuristics in cryptanalysis (e.g., see [12]), does not work in general (see discussion in Section 3). However, [14] showed that a natural geometric gadget (consisting essentially of a lattice coset in Euclidean space with large minimum distance and many short vectors) can be used to turn this simple idea into a formal reduction. The reduction of [14] still admits a nice geometric interpretation (see Sections 3 and 4 for details), and served as a starting point to obtain similar results for the analogous Minimum Distance Problem (MDP) on linear codes. The history of coding problems, MDP and its inhomogeneous counterpart the Nearest Codeword Problem (NCP), closely mirrors that of SVP and CVP. The NP-hardness of the inhomogeneous problem, NCP, was already proved in the late 70s [3] for the exact version of the problem, and improved to NP-hardness of approximation within any constant factor in [2]. Proving the NP-hardness of the homogeneous problem, MDP, took much longer. Hardness for the exact version of MDP was proved by Vardy [19], around the same time as Ajtai's discovery for SVP [1]. However, while Ajtai's reduction was randomized, Vardy [19] could prove NP-hardness of MDP under *deterministic* reductions. Building on [14], Dumer, Micciancio and Sudan [8] proved that MDP is NP-hard even to approximate, for any constant approximation factor. However, [8] inherited from [14] the use of randomization for the construction of the embedding gadget required by the reduction. Finally, in a surprising development, Cheng and Wan [5] showed that the probabilistic construction of the embedding gadget employed in the reduction of [8] can be derandomized, leading to the NP-hardness of approximating MDP within any constant factor under *deterministic* reductions. (The result of Cheng and Wan [5] has also been recently simplified by Khot and Austrin [11].)

Going back to lattices, the strongest inapproximability results for SVP known to date are Khot's proof [10] that SVP is NP-hard to approximate within any constant $O(1)$ factor, and Haviv and Regev's proof [9] that SVP cannot be approximated within some factor $n^{1/O(\log \log n)}$ unless NP is in random subexponential time. However, just like Ajtai's original proof [1], all subsequent inapproximability results for SVP [4, 14, 10, 9] employed randomization, and little progress has been made in proving NP-hardness under deterministic reductions, even for the exact version of SVP. In fact, the most recent and quantitatively strongest results [10, 9] achieve larger inapproximability factors than [14] at the cost of introducing even more randomness: while the randomized reduction of Micciancio [14] had one-sided error, the hardness proofs of Khot [10] and Haviv and Regev [9] incurred 2-sided error.[1] The hardness proofs of [10, 9] depart from the geometrically appealing homogenization framework of [14], and incorporate additional probabilistic techniques (namely, the intersection of lattices with randomly chosen subspaces) that, beside introducing 2-sided errors, also make the high level structure of the reduction more involved and harder to derandomize. In particular, the use of randomization in [10, 9] is not restricted to the construction of a gadget with self-contained description as in [14], but permeates the entire reduction process.

Our results: We present a new, simpler proof that SVP is NP-hard to approximate within any constant factor, which goes back to the geometrically appealing approach of [14], and avoids the introduction of additional probabilistic techniques from [10, 9]. In particular, we prove

- the NP-hardness of SVP for any constant approximation factor as in [10], and
- the hardness of SVP for subpolynomial factors $n^{1/O(\log \log n)}$ under the assumption that NP is not in subexponential time as in [9],

thus matching the strongest known hardness results for SVP, but under probabilistic reductions with one-sided error that may produce false negatives, but are guaranteed not to produce false positives. We regard our

[1] Technically, [14] proved NP-hardness of SVP under reverse unfaithful random reductions. These are reductions that may produce false negatives (i.e., map YES instances to NO instances) with small probability, but are guaranteed not to produce false positives (i.e., map NO instances to YES instances), regardless of the random string or nonuniform advice used by the reduction process. By contrast, the use of randomness in [10, 9] can result in both false negatives an false positives.

results as a partial derandomization of the reductions [10, 9] with two-sided error, and a step toward an NP-hardness proof for SVP under deterministic reductions. Randomness is used within our proof exclusively for the construction of a geometric gadget with similar properties as the one originally introduced by Micciancio in [14]. Beside the technical advantage of resulting in a reduction with one-sided error, we believe this takes us closer to a possible NP-hardness proof for SVP under deterministic reductions for the following reasons:

- In [14], Micciancio showed that a lattice gadget similar to the one used in this paper can be constructed in deterministic polynomial time, under a certain (plausible but unproven) number theoretic conjecture on the distribution of smooth numbers.[2] While proving the number theoretic conjecture of [14] may be difficult, the result in [14] suggests that randomness is not essential for the construction of the lattice gadget used in our proof.
- The probabilistic construction of a similar gadget for linear codes used in [8] to prove the NP-hardness of MDP has been successfully derandomized [5]. This let us hope that a derandomization of the lattice gadget employed in this paper may be within reach using current mathematical knowledge.
- The lattice gadget presented in this paper is constructed using techniques from the theory of linear codes, rather than the number theoretic methods of [14]. So, the techniques in [5, 11] for the derandomization of the coding gadget of [8] may help to derandomize the construction of the lattice gadget described in this paper.

While proving the NP-hardness of (approximating) SVP under deterministic polynomial time reductions is a goal yet to be reached, we believe that our results offer a viable approach to the resolution of this outstanding open problem.

Techniques A standard method to prove hardness results within large approximation factors for lattice and coding problems is to first prove hardness for some fixed small constant factor, and then amplify the constant using some polynomial time (or quasi-polynomial time) transformation. For example, the tensor product of linear codes is used in [8] to amplify the NP-hardness of approximating MDP to arbitrarily large constant factors. This suggests to use the tensor product of lattices to prove the NP-hardness of SVP within large constant factors, starting from the inapproximability result of [14] for factors below $\sqrt{2}$. In fact, using the tensor product is a common theme in the sequence of papers [14, 10, 9] proving hardness of approximation results for SVP. Unfortunately, while the minimum distance of a linear code gets squared when one takes the tensor product of the code with itself, the same is not true for the length of the shortest vector in a lattice. The length of the shortest vector in the tensor product of a lattice with itself can be essentially the same as the length of the shortest vector in the original lattice (e.g., see [9, Lemma 2.4]), and this is why Micciancio [14] could not prove NP-hardness (under randomized reductions with one-sided error) of SVP within any constant factor. Subsequent work [10, 9] went around this obstacle in various ways. Khot [10] introduced a nonstandard notion of "augmented tensor product", and used it to prove NP-hardness results for any constant approximation factors starting from a new hardness result for small constants based on BCH codes. Haviv and Regev [9] were able to prove that the lattices produced by the basic reduction of [10] behave well with respect to the standard tensor product operation, leading to stronger hardness results under superpolynomial time reductions. The proofs in [10, 9] that the (augmented) tensor product does amplify the approximation factor are specific to the basic lattices of [10], and are not immediately applicable to other lattices.

In this paper we revisit the general problem of amplifying the approximation factor for SVP by the standard tensor product operation, and prove that tensoring works when applied to an appropriate variant of SVP. Specifically, we introduce a new method to measure the length of the vectors in a lattice, which can be seen as a hybrid between the Euclidean length typically used for lattices and the Hamming metric used for linear codes. Specifically, the measure associated to an integer vector $\mathbf{v}$ is given by the product of the largest power of 2 that divides $\mathbf{v}$ times the square root of the number of nonzero coordinates in

[2] Namely, [14] conjectured that for every $\epsilon > 0$ there is a $c \geq 1$ such that for all sufficiently large n the interval $[n, n + n^{\epsilon}]$ contains a square-free smooth number, i.e., an integer whose prime factors are all distinct and bounded by $\log^c n$.

$\mathbf{v}$. Using this measure, we define a variant of SVP, and prove that it behaves well with respect to the tensor product. Then, we prove that our SVP variant is NP-hard to approximate within some constant factor, under reductions with one-sided error. Tensoring immediately yields inapproximability results for SVP within larger factors, still under reductions with one-sided error. Moreover, our basic NP-hardness proof within small approximation factors is very similar to those in [14, 8], so, as explained in the previous paragraphs, it may be more easily derandomized. We remark that the standard tensor product operation amplifies the approximation factor for any instance of the SVP variant defined in this paper, and not just for the output of our basic reduction. So, the amplification method proposed here is fairly general, and any proof that the SVP variant is NP-hard to approximate within some constant factor (not necessarily obtained by derandomizing the specific reduction given in this paper) would immediately yield similar deterministic NP-hardness results for arbitrarily large constants.

Organization The rest of the paper is organized as follows. In Section 2 we give some standard background about lattices, codes and some useful combinatorial tools. In Section 3 we describe our basic techniques and a construction of very dense lattices with large minimum distance that behaves well with respect to the tensor product operation. In Section 4 we give our main NP-hardness proof for SVP under nonuniform reductions with one-sided error. We chose to first present our result as a nonuniform reduction to make the reduction and analysis as simple as possible and self-contained. However, the non-uniformity of the advice is not used in any essential way in our proof, and in Section 5 we use a combinatorial theorem of Micciancio [14] to replace the nonuniform advice with a uniformly chosen random string, leading to NP-hardness results under randomized reductions with one-sided error.

2 Background

We use $\mathbb{R}$, $\mathbb{Z}$ and 2^A to denote the set of the real numbers, the set of the integers, and the power-set of an arbitrary set A. The m-dimensional Euclidean space is denoted $\mathbb{R}^m$. A *lattice* in $\mathbb{R}^m$ is the set of all integer combinations $\Lambda = \{\sum_{i=1}^n x_i \mathbf{b}_i : x_i \in \mathbb{Z}\}$ of n linearly independent vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ in $\mathbb{R}^m$ ($m \geq n$). The set of vectors $\mathbf{b}_1, \ldots, \mathbf{b}_n$ is called a lattice *basis*, and the integer n is the lattice *rank*. A basis can be compactly represented by the matrix $\mathbf{B} = [\mathbf{b}_1 | \ldots | \mathbf{b}_n] \in \mathbb{R}^{m \times n}$ having the basis vectors as columns. The lattice generated by $\mathbf{B}$ is denoted $\mathcal{L}(\mathbf{B})$. Notice that $\mathcal{L}(\mathbf{B}) = \{\mathbf{Bx} : \mathbf{x} \in \mathbb{Z}^n\}$, where $\mathbf{Bx}$ is the usual matrix-vector multiplication. The determinant of a lattice $\mathcal{L}(\mathbf{B})$ is the volume of the parallelepiped spanned by the basis vectors $\mathbf{B}$ and when $\mathbf{B}$ is a square matrix, it equals the absolute matrix determinant $\det(\mathcal{L}(\mathbf{B})) = |\det(\mathbf{B})|$. (More generally, for non-square bases, $\det(\mathcal{L}(\mathbf{B})) = \sqrt{\det(\mathbf{B}^T\mathbf{B})}$, where $\mathbf{B}^T$ is the matrix transpose of $\mathbf{B}$.)

Lattice problems can be defined with respect to any norm, but the Euclidean norm $\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$ is the most common, and the one we focus on in this paper. We recall that the Euclidean norm is in a technical sense the one for which lattice problems are algorithmically easiest, and hardness results for other norms can be obtained via norm embedding [17]. The *minimum distance* of a lattice, $\lambda(\Lambda)$, is the minimum distance between any two distinct lattice points and equals the length of the shortest nonzero lattice vector:

$$\lambda(\Lambda) = \min\{\|\mathbf{x} - \mathbf{y}\| : \mathbf{x} \neq \mathbf{y} \in \Lambda\} = \min\{\|\mathbf{x}\| : \mathbf{x} \in \Lambda, \mathbf{x} \neq \mathbf{0}\}.$$

For vector $\mathbf{x} \in \mathbb{R}^n$ and real r, let $\mathcal{B}(\mathbf{v}, r) = \{\mathbf{w} \in \mathbb{R}^n : \|\mathbf{v} - \mathbf{w}\| \leq r\}$ be the ball of radius r centered in $\mathbf{v}$. When the ball is centered around the origin $\mathbf{v} = \mathbf{0}$, we simply write $\mathcal{B}(r)$.

When discussing computational issues related to lattices, it is customary to assume that the lattices are represented by a basis matrix $\mathbf{B}$ and that $\mathbf{B}$ has integer entries. We study the decisional (length/distance estimation) variants of SVP and CVP as defined below.

Definition 1 *The promise problem GapSVP$_\gamma$ is defined as follows. Instances are pairs $(\mathbf{B}, d)$, where $\mathbf{B} \in \mathbb{Z}^{n \times k}$ is a lattice basis and d a positive number such that*

1. *$(\mathbf{B}, d)$ is a* YES *instance if $\lambda(\mathcal{L}(\mathbf{B})) \leq d$, i.e., $\|\mathbf{Bz}\| \leq d$ for some $\mathbf{z} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$;*

2. $(\mathbf{B}, d)$ *is a* NO *instance if* $\lambda(\mathcal{L}(\mathbf{B})) > \gamma \cdot d$*, i.e.,* $\|\mathbf{Bz}\| > \gamma \cdot d$ *for all* $\mathbf{z} \in \mathbb{Z}^n \setminus \{\mathbf{0}\}$.

Definition 2 *The promise problem* $GapCVP_\gamma$ *is defined as follows. Instances are triples* $(\mathbf{B}, \mathbf{y}, d)$*, where* $\mathbf{B} \in \mathbb{Z}^{n \times k}$ *is a lattice basis,* $\mathbf{y} \in \mathbb{Z}^n$ *a vector, and* d *a positive number such that*

1. $(\mathbf{B}, \mathbf{y}, d)$ *is a* YES *instance if* $\|\mathbf{Bz} - \mathbf{y}\| \leq d$ *for some* $\mathbf{z} \in \mathbb{Z}^n$*;*
2. $(\mathbf{B}, \mathbf{y}, d)$ *is a* NO *instance if* $\|\mathbf{Bz} - \mathbf{y}\| > \gamma \cdot d$ *for all* $\mathbf{z} \in \mathbb{Z}^n$.

We remark that any algorithm that solves SVP in its standard formulation (given a lattice, find an approximately shortest nonzero lattice vector) can be used to immediately solve GapSVP as well. So, proving hardness results for GapSVP implies hardness of the standard SVP as well. The same observation applies to CVP and GapCVP. However we remark that the converse is not known to be true: giving a reduction from approximate SVP to GapSVP_γ is an important open problem in the complexity of lattice problems.

Some of our constructions rely on techniques from the study of linear codes. For any finite field $\mathbb{F}$, and finite dimensional vector space $\mathbb{F}^n$ over $\mathbb{F}$, a linear code of block length n and dimension k is a k-dimensional linear subspace of $\mathbb{F}^n$. The difference $n - k$ is called the co-dimension of the code. The Hamming weight of a vector $\mathbf{v} \in \mathbb{F}^n$ is the number $\|\mathbf{v}\|_H$ of nonzero coordinates of $\mathbf{v}$. The minimum distance of a linear code $\mathcal{C} \subseteq \mathbb{F}^n$ is the smallest Hamming weight of a nonzero vector in the code $\min\{\|\mathbf{v}\|_H : \mathbf{v} \in \mathcal{C} \setminus \{\mathbf{0}\}\}$. In this paper, we will be primarily interested in binary linear codes, i.e., linear codes over the field $\mathbb{F}_2 = \{0, 1\}$ with two elements. A binary linear code with block length n, dimension k and minimum distance d is usually denoted $\mathcal{C}[n, k, d]_2$. In Section 3 we will build a very dense lattice starting from the family of binary linear codes described in the next lemma.

Lemma 1 *For any* $h = 2^\ell$ *and* $m = 2^\kappa$ *with* $m \geq h^2$*, there is a sequence* $\{0,1\}^m = \mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots \subseteq \mathcal{C}_\ell$ *of binary linear codes* $\mathcal{C}_i[m, k_i, d_i]_2$ *of common block length* m*, minimum distance* $d_i \geq 4^i$ *and co-dimension* $m - k_i \leq \kappa \cdot (4^i/2 - 1) + 1$.

Proof For $i = 0, \ldots, \ell$, let $\mathcal{C}_i[m, k_i, d_i] = \text{EBCH}_{4^i}^m$ be the extended narrow sense primitive binary BCH code of block length m and designed distance $d_i \geq 4^i$. These codes form a chain $\{0,1\}^m = \mathcal{C}_0 \supseteq \mathcal{C}_1 \supseteq \cdots \supseteq \mathcal{C}_\ell$ and have co-dimension $m - k_i \leq \kappa \cdot (4^i/2 - 1) + 1$. (See Appendix C for details.) □

For any two matrices $\mathbf{B}^{(1)} \in \mathbb{R}^{n_1 \times k_1}$ and $\mathbf{B}^{(2)} \in \mathbb{R}^{n_2 \times k_2}$, define the Kronecker product $\mathbf{B} = \mathbf{B}^{(1)} \otimes \mathbf{B}^{(2)} \in \mathbb{R}^{n_1 n_2 \times k_1 k_2}$ as the matrix with entries $b_{i,j} = b^{(1)}_{i_1,j_1} \cdot b^{(2)}_{i_2,j_2}$ where $i = (i_1 - 1) \cdot n_2 + i_2$ and $j = (j_1 - 1) \cdot k_2 + j_2$ for $i_1 = 1, \ldots, n_1$, $i_2 = 1, \ldots, n_2$, $j_1 = 1, \ldots, k_1$ and $j_2 = 1, \ldots, k_2$. Informally, $\mathbf{B}^{(1)} \otimes \mathbf{B}^{(2)}$ is the block matrix obtained replacing each entry $b^{(1)}_{i_1,j_1}$ of $\mathbf{B}^{(1)}$ with the matrix $b^{(1)}_{i_1,j_1} \cdot \mathbf{B}^{(2)}$. The tensor product of lattices $\mathcal{L}(\mathbf{B}^{(1)})$ and $\mathcal{L}(\mathbf{B}^{(2)})$ is the $n_1 \cdot n_2$ dimensional lattice $\mathcal{L}(\mathbf{B}^{(1)} \otimes \mathbf{B}^{(2)})$ of rank $k_1 k_2$ generated by the Kronecker product of the two basis matrices. Identifying the set $\mathbb{R}^{n_1 n_2}$ of $n_1 n_2$-dimensional vectors with the set $\mathbb{R}^{n_1 \times n_2}$ of $n_1 \times n_2$ matrices in the obvious way, the tensor product of two lattices can be conveniently defined as the set of all matrices

$$\mathcal{L}(\mathbf{B}^{(1)} \otimes \mathbf{B}^{(2)}) = \{\mathbf{B}^{(1)} \mathbf{X} (\mathbf{B}^{(2)})^T \mid \mathbf{X} \in \mathbb{Z}^{k_1 \times k_2}\}$$

where $(\mathbf{B}^{(2)})^T$ is the matrix transpose of $\mathbf{B}^{(2)}$. The tensor product of two linear codes is defined similarly. As mentioned in the introduction, the tensor product operation can be used to amplify hardness results for coding and certain lattice problems to large approximation factors. For example, if $\mathcal{C}$ is a linear code with minimum distance d, then the product code $\mathcal{C} \otimes \mathcal{C}$ has minimum distance d^2. So, if one can approximate the minimum distance of $\mathcal{C} \otimes \mathcal{C}$ within a factor γ^2, then one can also approximate the distance of $\mathcal{C}$ within a factor γ. Similar amplification results are also possible for NCP and CVP. However, this method to amplify the approximation factor of a problem does not work for SVP. It is easy to prove that for any two lattices $\lambda(\Lambda_1 \otimes \Lambda_2) \leq \lambda(\Lambda_1) \cdot \lambda(\Lambda_2)$, and, in particular $\lambda(\Lambda \otimes \Lambda) \leq \lambda(\Lambda)^2$. However, in general $\lambda(\Lambda \otimes \Lambda)$ can be much smaller than $\lambda(\Lambda)^2$. (E.g., see [9, Lemma 2.4].) Lattices Λ_1 for which $\lambda(\Lambda_1 \otimes \Lambda_2) = \lambda(\Lambda_1) \cdot \lambda(\Lambda_2)$ for every lattice Λ_2 are called "E-type" lattices, and are somehow special.

We will prove the NP-hardness of SVP by reduction from the following NP-hard variant of CVP.

Definition 3 (TensorCVP$_\gamma$) *Instances are triples* $(\mathbf{B}, \mathbf{y}, t)$ *where* $\mathbf{B}$ *is an integer lattice,* $\mathbf{y}$ *an integer vector, and* t *a positive number.*

- $(\mathbf{B}, \mathbf{y}, t)$ *is a* YES *instance if* $\exists \mathbf{x} \in \{0,1\}^k$ *such that* $\|\mathbf{y} - \mathbf{B}\mathbf{x}\| \leq t$
- $(\mathbf{B}, \mathbf{y}, t)$ *is a* NO *instance if* $\forall \mathbf{x} \in \mathbb{R}^k$, $\sqrt{\|\mathbf{y} - \mathbf{B}\mathbf{x}\|_H} > \gamma t$.

TensorCVP differs from CVP as follows. In the YES instances, the target is required to be close to a *binary* combination of the basis vectors. In the NO instances, the target is required to be far in *Hamming distance* from the entire *linear space* spanned by the lattice. The above problem is a fairly standard NP-hard variant of CVP, similar to those used in many other previous works in computational complexity. We call this CVP variant TensorCVP because we will use it to prove the NP-hardness of a variant of SVP (TensorSVP, see Definition 5) closely related to the use of the tensor product to amplify the approximation factor. For completeness, we prove the NP-hardness of approximating TensorCVP$_\gamma$ within any constant factor γ in Appendix A.

Our reduction from TensorCVP to SVP uses a combinatorial result, typically referred to as Sauer's lemma. We recall Sauer's lemma and its simple proof in Appendix B. In the context of this paper, it is convenient to reformulate Sauer's lemma in terms of matrices as follows.

Corollary 1 *Let* m *be a positive integer, and* $\mathcal{Z} \subset \{0,1\}^m$ *an arbitrary set of* m*-dimensional binary vectors. If* $|\mathcal{Z}| \geq \sum_{i=0}^{k} \binom{m}{i}$*, then there exists a matrix* $\mathbf{T} \in \{0,1\}^{k \times m}$ *such that* $\{0,1\}^k \subseteq \{\mathbf{T}\mathbf{z} \colon \mathbf{z} \in \mathcal{Z}\}$.

Proof: Let $M = \{1, \ldots, m\}$ and let $\phi_M \colon 2^M \to \{0,1\}^m$ be the bijection that sends each subset of M to its characteristic vector. Define the set $\mathcal{A} = \{A \subseteq M \colon \phi(A) \in \mathcal{Z}\} \subseteq 2^M$. Since $|\mathcal{A}| = |\mathcal{Z}| \geq \sum_{i=0}^{k} \binom{m}{i}$, by Lemma 5 in Appendix B there is a set $T \subseteq M$ such that $\{A \cap T \colon A \in \mathcal{A}\} = 2^T$. Let $\mathbf{T} \in \{0,1\}^{k \times m}$ be the projection matrix such that $\mathbf{T}(\phi_M(A)) = \phi_T(T \cap A)$ for any $A \subseteq M$, where $\phi_T \colon 2^T \to \{0,1\}^k$ is defined similarly to ϕ_M. Then, $\{\mathbf{T}\mathbf{z} \colon \mathbf{z} \in \mathcal{Z}\} = \{\mathbf{T}(\phi_M(A)) \colon A \in \mathcal{A}\} = \{\phi_T(A \cap T) \mid A \in \mathcal{A}\} = \phi_T(2^T) = \{0,1\}^k$. □

3 Techniques

We first recall the framework of [14] to prove hardness results for SVP. Let $(\mathbf{B}, \mathbf{y})$ be a CVP instance. A common heuristic to find the lattice vector $\mathbf{B}\mathbf{x}$ closest to $\mathbf{y}$ is to search for a short vector in the augmented lattice $\mathcal{L}([\mathbf{B}, \mathbf{y}])$. However, this simple heuristic, often used in cryptanalysis, is not guaranteed to work, even if one can solve SVP exactly. There are two different ways in which this approach may fail:

- The shortest nonzero vector in $\mathcal{L}([\mathbf{B}, \mathbf{y}])$ may be of the form $\mathbf{B}\mathbf{x} + c \cdot \mathbf{y}$ with $|c| \geq 2$. This yields a lattice vector $\mathbf{B}\mathbf{x}$ close to a *multiple* of the original target $\mathbf{y}$.
- The shortest nonzero vector in $\mathcal{L}([\mathbf{B}, \mathbf{y}])$ is of the form $\mathbf{B}\mathbf{x}$. This will be the case if the distance of the target $\mathbf{y}$ from the lattice $\mathcal{L}(\mathbf{B})$ is bigger than $\lambda(\mathcal{L}(\mathbf{B}))$.

In the context of proving the NP-hardness of SVP, the first problem is easily solved by reducing from a variant of CVP (like TensorCVP, see Definition 3) where either the target is close to the lattice, or all its nonzero integer multiples are far from it. The second problem is more fundamental, and arises also in the context of proving similar results for linear codes [8]. Building on techniques from [1], Micciancio [14] solved this problem essentially by embedding $\mathbf{B}$ and $\mathbf{y}$ into a higher dimensional space in such a way that

- if $\mathbf{y}$ is close to the lattice $\mathcal{L}(\mathbf{B})$, then after the embedding the target $\mathbf{y}'$ is still close to the lattice $\mathcal{L}(\mathbf{B}')$, and
- the embedding operation increases the minimum distance of the lattice $\mathcal{L}(\mathbf{B})$, so that the distance of $\mathbf{y}'$ from $\mathcal{L}(\mathbf{B}')$ is strictly smaller than $\lambda(\mathcal{L}(\mathbf{B}'))$.

This transformation ensures that the shortest vectors in $\mathcal{L}([\mathbf{B}', \mathbf{y}'])$ are not in $\mathcal{L}(\mathbf{B}')$, and therefore must necessarily make use of the target vector $\mathbf{y}'$. In [14], it is shown that such a transformation can be easily carried out using a geometric gadget consisting of a lattice coset $\mathcal{L}(\mathbf{L}) - \mathbf{s}$ with large minimum distance $\lambda(\mathcal{L}(\mathbf{L}))$ and many short vectors $(\mathcal{L}(\mathbf{L}) - \mathbf{s}) \cap \mathcal{B}(r)$. (Specifically, the length bound r on these vectors should be strictly smaller than the minimum distance of $\mathcal{L}(\mathbf{L})$ by a constant factor.) Moreover, if $\mathcal{L}(\mathbf{L})$ is sufficiently dense (i.e., if its determinant is not too big), then an appropriate coset is guaranteed to exist and can be probabilistically found choosing $\mathbf{s}$ as a random short vector.

In [14] a gadget of this type is constructed using techniques from elementary number theory, which are less "combinatorial" than the coding theory tools used in the NP-hardness proof of [10], and arguably harder to derandomize. In this section we give an alternative and more refined construction of Micciancio's geometric gadget. Similarly to the proofs in [10, 9], we rely on tools from coding theory, namely the construction of BCH codes as those satisfying Lemma 1, rather than number theoretic methods. Beside its potential for easier derandomization, the new construction, which combines lattice and coding elements, has the advantage of behaving well with respect to the tensor product of lattices. Central to our construction and hardness results is new method to measure the length of a vector which is in a sense a hybrid between the Euclidean norm and the Hamming metric. The definition is parametrized by an integer q which we will later set to $q = 2$.

Definition 4 *For any integer vector* $\mathbf{x}$*, let* $\mathrm{pow}_q(\mathbf{x})$ *be the largest power of* q *that evenly divides* $\mathbf{x}$*, and* $\|\mathbf{x}\|_H$ *the Hamming weight of* $\mathbf{x}$*, i.e., the number of nonzero coordinates of* $\mathbf{x}$*. For any integer lattice* Λ*, define the quantity*

$$\tau_q(\Lambda) = \min\{\tau_q(\mathbf{x}) \colon \mathbf{x} \in \mathcal{L}(\Lambda) \setminus \{\mathbf{0}\}\},$$

where $\tau_q(\mathbf{x}) = \mathrm{pow}_q(\mathbf{x}) \cdot \sqrt{\|\mathbf{x}\|_H}$.

Notice that τ_q is not a norm because it satisfies neither the linearity property $\|c \cdot \mathbf{x}\| = c \cdot \|\mathbf{x}\|$, nor the triangle inequality $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ required to be a norm. Still, the quantity $\tau_q(\mathbf{B})$ is useful to study SVP because it gives a lower bound on the norm of integer vectors, and it behaves well with respect to the tensor product of lattices, as shown below.

Lemma 2 *For any integer vector* $\mathbf{x} \in \mathbb{Z}^n$*,* $\tau_q(\mathbf{x}) \leq \|\mathbf{x}\|$.

Proof. The vector $\mathbf{x}$ has $\|\mathbf{x}\|_H$ nonzero entries, and each of them is at least $\mathrm{pow}_q(\mathbf{x})$ in absolute value. Therefore $\|\mathbf{x}\| \geq \mathrm{pow}_q(\mathbf{x}) \cdot \sqrt{\|\mathbf{x}\|_H} = \tau_q(\mathbf{x})$. □

Lemma 3 *For any integer lattice* Λ *and (arbitrary) lattice* Λ'*,*

$$\tau_q(\Lambda) \cdot \lambda(\Lambda') \leq \lambda(\Lambda \otimes \Lambda') \leq \lambda(\Lambda) \cdot \lambda(\Lambda').$$

Proof. Let $\Lambda = \mathcal{L}(\mathbf{B})$ and $\Lambda' = \mathcal{L}(\mathbf{B}')$. For the upper bound, simply observe that for any two lattice vectors $\mathbf{Bx}$ and $\mathbf{B}'\mathbf{y}$, the product lattice $\Lambda \otimes \Lambda'$ contains a vector $\mathbf{Bx}(\mathbf{B}'\mathbf{y})^T$ of length $\|\mathbf{Bx}\| \cdot \|\mathbf{B}'\mathbf{y}\|$. Choosing $\mathbf{Bx}$ and $\mathbf{B}'\mathbf{y}$ as the shortest nonzero vectors in Λ and Λ' yields a vector in $\Lambda \otimes \Lambda'$ of length $\lambda(\Lambda) \cdot \lambda(\Lambda')$. In order to prove the lower bound we consider an arbitrary nonzero vector $\mathbf{v} = \mathbf{BX}(\mathbf{B}')^T$ in the tensor product lattice $\Lambda \otimes \Lambda'$, and show that $\|\mathbf{v}\| \geq \tau_q(\Lambda) \cdot \lambda(\Lambda')$. Let h be the number of nonzero rows in $\mathbf{BX}$. Clearly, all columns $\mathbf{c} \in \mathbf{BX}$ have Hamming weight at most $\|\mathbf{c}\|_H \leq h$, and therefore $\tau_q(\mathbf{c}) \leq \mathrm{pow}_q(\mathbf{c}) \cdot \sqrt{h}$. It follows that all nonzero columns $\mathbf{c}$ satisfy $\mathrm{pow}_q(\mathbf{c}) \geq \tau_q(\mathbf{c})/\sqrt{h} \geq \tau_q(\mathbf{B})/\sqrt{h}$. In particular, the largest power q^i that divides the entire matrix $\mathbf{BX}$ satisfies $q^i \geq \tau_q(\Lambda)/\sqrt{h}$. Notice that $\mathbf{v} = (\mathbf{BX}) \cdot (\mathbf{B}')^T$ contains exactly h nonzero rows, and each of them is a nonzero vector in $q^i\Lambda'$. Therefore, $\|\mathbf{v}\| \geq \sqrt{h} q^i \lambda(\Lambda') \geq \tau_q(\Lambda)\lambda(\Lambda')$. □

We use the quantity $\tau_q(\Lambda)$ to define a variant of SVP that behaves well with respect to the tensor product of lattices. Our variant of SVP is defined using the Euclidean norm for the YES instances, and our new measure τ_q for the NO instances.

Definition 5 *TensorSVP$_\gamma$ instances are pairs $(\mathbf{B}, d)$ where $\mathbf{B}$ is an integer lattice basis and d a positive number.*

- *$(\mathbf{B}, d)$ is a* YES *instance if $\lambda(\mathcal{L}(\mathbf{B})) \leq d$.*
- *$(\mathbf{B}, d)$ is a* NO *instance if $\tau_q(\mathcal{L}(\mathbf{B})) > \gamma d$.*

Notice that TensorSVP$_\gamma$ is a special case of the standard GapSVP$_\gamma$ problem because the defining condition for YES instances is the same, and in the NO instances $\tau_q(\mathcal{L}(\mathbf{B}))$ is a lower bound on $\lambda(\mathcal{L}(\mathbf{B}))$. So, in order to establish NP-hardness results for SVP$_\gamma$ it is enough to prove the NP-hardness of TensorSVP$_\gamma$. Moreover, TensorSVP behaves well with respect to the tensor product of lattices, as described in the next theorem.

Theorem 1 *For any positive integer c, the map $(\mathbf{B}, d) \mapsto (\mathbf{B}^{\otimes c}, d^c)$ is a reduction from TensorSVP$_\gamma$ to GapSVP$_{\gamma^c}$, where $\mathbf{B}^{\otimes c}$ denotes the iterated tensor product of c copies of $\mathbf{B}$.*

Proof: Let $(\mathbf{B}, d)$ be an instance of TensorSVP$_\gamma$. If $(\mathbf{B}, d)$ is a YES instance, then $\lambda(\mathcal{L}(\mathbf{B})) \leq d$, and by Lemma 3, $\lambda(\mathcal{L}(\mathbf{B}^{\otimes c})) \leq d^c$. So, $(\mathbf{B}^{\otimes c}, d)$ is a YES instance of GapSVP$_{\gamma^d}$. Conversely, if $(\mathbf{B}, d)$ is a NO instance, then $\lambda(\mathcal{L}(\mathbf{L})) \geq \tau_q(\mathcal{L}(\mathbf{B})) > \gamma d$, and by Lemma 3, $\lambda(\mathcal{L}(\mathbf{B}^{\otimes c})) > d^c$. So, $(\mathbf{B}^{\otimes c}, d)$ is a NO instance of GapSVP$_{\gamma^d}$. □

Notice that for any constant c, the transformation in Theorem 1 runs in polynomial time. So, if TensorSVP$_\gamma$ is NP-hard for *some* constant $\gamma > 1$, then GapSVP$_\gamma$ is NP-hard for *any* constant $\gamma' = \gamma^c > 1$. Similarly, using reductions that run in superpolynomial time, one obtains inapproximability results for even larger factors. (See Corollary 2.)

So, we want to prove the NP-hardness of TensorSVP. We will use the framework of [14] and construct a gadget consisting of a dense lattice $\mathcal{L}(\mathbf{L})$ with large minimum distance, as outlined at the beginning of this section. However, since we want to prove the NP-hardness of TensorSVP (rather than just GapSVP as in [14]), we will need a lattice $\mathcal{L}(\mathbf{L})$ such that not only $\lambda(\mathcal{L}(\mathbf{L}))$, but also $\tau_q(\mathcal{L}(\mathbf{L}))$ is large. Here q can be arbitrary, and for simplicity we fix q to 2. Our methods can be easily adapted (using appropriate q-ary codes,) to any value of q, but this is not needed to prove the hardness of GapSVP. The following theorem gives a construction of dense lattices with large τ_2 minimum based on binary codes. This is essentially "construction D" of [6] instantiated with the binary codes of Lemma 1.

Theorem 2 *For any $m = 2^\kappa$ and $h = 2^\ell$ with $h \leq \sqrt{m}$, there is an m-dimensional full rank integer lattice $\mathbf{L}$ such that $\tau_2(\mathcal{L}(\mathbf{L})) \geq h$ and $\det(\mathcal{L}(\mathbf{L})) < m^{(h^2/1.5)-\ell}$.*

Theorem 2 essentially follows from Lemma 1 and [6, Chapter 8, Theorem 13], the only differences being that here we use a scaled copy of the lattice so that $\mathbf{L}$ is an integer matrix, and we express the bound in terms of $\tau_2(\mathcal{L}(\mathbf{L}))$ rather than $\lambda(\mathcal{L}(\mathbf{B}))$. For completeness, a proof of Theorem 2 is given in Appendix D. Theorem 2 is used in the next section to prove the hardness of TensorSVP. We conclude this section with some remarks and observations.

Remark 1 *The orthogonal lattice $h\mathbb{Z}^m$ satisfies $\tau_2(h\mathbb{Z}^m) = h$ and $\det(h\mathbb{Z}^m) = h^m$. Our lattice achieves the same $\tau_2(\mathcal{L}(\mathbf{L})) \geq h$, but it has much smaller determinant $\det(\mathcal{L}(\mathbf{L})) = h^m/2^{\sum_{i=1}^{\ell} k_i}$, where $k_1, \ldots, k_\ell$ are the dimensions of the codes $\mathbf{C}_i$ used in the construction. The higher the dimension of the codes $\mathbf{C}_i$, the denser the lattice $\mathbf{L}$.*

Remark 2 *Another way to get a sense of how dense lattice $\mathcal{L}(\mathbf{L})$ is, is to compare the lower bound $\lambda(\mathcal{L}(\mathbf{L})) \geq \tau_s(\mathcal{L}(\mathbf{L}))$ on the minimum distance with Minkowski's upper bound $\lambda(\mathcal{L}(\mathbf{L})) \leq O(\sqrt{m}) \cdot \det(\mathcal{L}(\mathbf{L}))^{1/m}$. Consider for example the setting $\kappa = 2\ell + \log \ell$, so that $m = h^2\ell$. Then, Minkowski's bound on the minimum distance of the lattice is $O(\sqrt{m} \det(\mathbf{L})^{1/m}) \leq O(\sqrt{m} \cdot m^{h^2/m}) = O(\sqrt{m} \cdot 2^{(2\ell + \log \ell)/\ell}) = O(\sqrt{m})$. On the other hand, the minimum distance of the lattice is at least $\tau_2(\mathbf{L}) \geq h = O(\sqrt{m/\log m})$. So, the minimum distance is within a polylogarithmic factor $O(\sqrt{\log m})$ from Minkowski's upper bound.*

Remark 3 *We gave a construction of dense lattices with large $\tau_2(\mathcal{L}(\mathbf{L}))$, using certain binary BCH codes from Lemma 1 as a building block. What makes BCH codes useful in this setting is the fact that, for appropriate choice of parameters, they are denser than random codes. As a historical note, the use of BCH codes in the context of proving NP-hardness results for homogeneous lattice and coding problems was first suggested in [8]. More specifically, [8] proves that codes beating the Gilbert-Varshamov bound can be used to build geometric gadgets similar to the one of [14], and mentions Reed-Solomon, Algebraic-Geometry and BCH codes as examples of codes beating this bound. BCH codes where later used by [10, 9] to prove the hardness of SVP for any constant approximation factor and beyond, but in an ad-hoc manner, without connecting them to previous work [14, 8]. Our work explains why BCH codes are useful in proving inapproximability results for SVP for large factors: they have the density properties required by the geometric construction of [14, 8], and as linear codes they behave well with respect to the tensor product operation.*

4 The main reduction

In this section we prove that TensorSVP is NP-hard to approximate within some constant factor under *nonuniform* polynomial time reductions with one-sided error. In Section 5 we show how the nonuniform advise required by our proof can be computed in probabilistic polynomial time. We present our main result as a nonuniform reduction first in order to make the presentation as simple as possible. We remark that the nonuniform reduction presented in this section is just as good a starting point for derandomization as the probabilistic reduction presented in the next section. A randomized uniform reduction is presented in Section 5 mostly to reassure the reader that here we are not using the non-uniformity of the advice in any essential way.

Theorem 3 *For any $\gamma < \lambda < \sqrt{3/2}$ and $\tilde{\gamma} = \gamma\sqrt{1 + 4/((\lambda/\gamma)^2 - 1)}$ there is a nonuniform reduction from TensorCVP$_{\tilde{\gamma}}$ to TensorSVP$_\gamma$. The (nonuniform) advice required by the reduction on input a TensorCVP instance of rank k is a tuple $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ where*

- *$\mathbf{L} \in \mathbb{Z}^{m\times l}$ is a lattice basis with $\tau(\mathbf{L}) \geq \lambda \cdot r$*
- *$\mathbf{T} \in \mathbb{Z}^{k\times m}$ is a linear transformation such that $\mathbf{T}((\mathcal{L}(\mathbf{L}) - \mathbf{s}) \cap \mathcal{B}(r)) \supseteq \{0, 1\}^k$*

The reduction is reverse unfaithful*, i.e., it has one-sided error and always maps* NO *instances to* NO *instances regardless of the value of the (nonuniform) advice.*

Proof. Let $(\mathbf{B}, \mathbf{y}, t)$ be a TensorCVP$_{\tilde{\gamma}}$ instance with $\mathbf{B} \in \mathbb{Z}^{n\times k}$ and $\mathbf{y} \in \mathbb{Z}^n$, and let $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ be as in the statement of the theorem. We begin by scaling the input $(\mathbf{B}, \mathbf{y}, t)$ and the gadget $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ so that $\epsilon/2 \leq t/r < \epsilon$, where $\epsilon = \sqrt{(\lambda/\gamma)^2 - 1}$. This is easily achieved as follows. If $t/r \geq \epsilon$, then simply multiply $\mathbf{L}$, $\mathbf{s}$ and r by an appropriate power of 2. If $t/r < \epsilon/2$, then replace $(\mathbf{B}, \mathbf{y}, t)$ with $(\mathbf{1}_{c^2} \otimes \mathbf{B}, \mathbf{1}_{c^2} \otimes \mathbf{y}, c \cdot t)$, where $\mathbf{1}_{c^2}$ is the all ones vector in dimension c^2 for an appropriately chosen integer c.

The output of the reduction is $(\mathbf{V}, d)$ where $d = \sqrt{t^2 + r^2}$ and

$$\mathbf{V} = \left[\begin{array}{c|c} \mathbf{BTL} & \mathbf{BTs} + \mathbf{y} \\ \mathbf{L} & \mathbf{s} \end{array}\right].$$

We show that the reduction is correct. First, assume $(\mathbf{B}, \mathbf{y}, t)$ is a NO instance of TensorCVP$_{\tilde{\gamma}}$, and let $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ be arbitrary, subject to the constraint $\epsilon/2 \leq t/r < \epsilon$. Consider any nonzero lattice vector

$$\mathbf{v} = \mathbf{V}\left[\begin{array}{c} \mathbf{z} \\ w \end{array}\right] = \left[\begin{array}{c} \mathbf{BT}(\mathbf{Lz} + w\mathbf{s}) + w\mathbf{y} \\ \mathbf{Lz} + w\mathbf{s} \end{array}\right].$$

If $w \neq 0$, then the vector $\mathbf{v}$ satisfies

$$\begin{aligned} \tau(\mathbf{v})^2 &\geq \|\mathbf{v}\|_H \\ &\geq \|\mathbf{BT}(\mathbf{Lz} + w\mathbf{s}) + w\mathbf{y}\|_H \\ &= \|\mathbf{B}(-\mathbf{T}(\mathbf{Lz}/w + \mathbf{s})) - \mathbf{y}\|_H \\ &> (\tilde{\gamma}t)^2 = \gamma^2(t^2 + (2t/\epsilon)^2) \geq \gamma^2 d^2. \end{aligned}$$

Otherwise, $w = 0$, $\mathbf{z} \neq \mathbf{0}$ and

$$\mathbf{v} = \begin{bmatrix} \mathbf{BT}(\mathbf{Lz}) \\ \mathbf{Lz} \end{bmatrix}$$

is divisible by $\mathrm{pow}(\mathbf{Lz})$. Moreover, $\|\mathbf{v}\|_H \geq \|\mathbf{Lz}\|_H$, and therefore

$$\tau(\mathbf{v})^2 \geq \mathrm{pow}(\mathbf{Lz})^2 \cdot \|\mathbf{Lz}\|_H = \tau(\mathbf{Lz})^2 \geq \tau(\mathbf{L})^2 \geq \lambda^2 r^2 > \gamma^2 \cdot d^2.$$

This proves that $\tau(\mathbf{B}) > \gamma d$, i.e., $(\mathbf{B}, d)$ is a NO instance of $\mathrm{TensorSVP}_\gamma$.

Now assume the gadget $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ satisfies the condition in the theorem, and let $(\mathbf{B}, \mathbf{y}, t)$ be a YES instance of TensorCVP. By definition, there is a $\mathbf{x} \in \{0,1\}^k$ such that $\|\mathbf{Bx} - \mathbf{y}\| \leq t$. By construction, there is an integer vector $\mathbf{z} \in \mathbb{Z}^l$ such that $\mathbf{T}(\mathbf{Lz} - \mathbf{s}) = \mathbf{x}$ and $\|\mathbf{Lz} - \mathbf{s}\| \leq r$. So, the lattice vector

$$\mathbf{v} = \mathbf{V} \begin{bmatrix} \mathbf{z} \\ -1 \end{bmatrix} = \begin{bmatrix} \mathbf{BT}(\mathbf{Lz} - \mathbf{s}) - \mathbf{y} \\ \mathbf{Lz} - \mathbf{s} \end{bmatrix} = \begin{bmatrix} \mathbf{Bx} - \mathbf{y} \\ \mathbf{Lz} - \mathbf{s} \end{bmatrix}$$

has squared norm at most $\|\mathbf{v}\|^2 \leq \|\mathbf{Bx} - \mathbf{y}\|^2 + \|\mathbf{Lz} - \mathbf{s}\|^2 = t^2 + r^2 = d^2$. This proves that if $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ satisfies the conditions in the lemma, then $(\mathbf{B}, d)$ is a YES instance of TensorSVP.

In order to complete the proof we need to show that a nonuniform advise $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ as described in the theorem exists. Define $\delta = 1 - 2\lambda^2/3 > 0$, and let h and $m = h^c$ be sufficiently large powers of 2 to be set later. (We anticipate that we will set c to any constant strictly bigger than $2/\delta$, and h to a power of 2 at least as big as $\lambda\sqrt{k/(\delta - 2/c)}$. We delay fixing c and h to these values in the proof because we will need different settings in the proof of Theorem 5.) Define $r = \sqrt{\lfloor (h/\lambda)^2 \rfloor}$ so that $h \geq \lambda r$, and

$$(r^2)^{r^2} \leq \left(\frac{h}{\lambda}\right)^{2(h/\lambda)^2} < h^{2h^2/\lambda^2}. \tag{1}$$

Let $\mathbf{L} \in \mathbb{Z}^{m \times m}$ be the lattice described in Theorem 2 with $\tau(\mathbf{L}) \geq h \geq \lambda r$ and

$$\det(\mathbf{L}) < m^{h^2/1.5 - \log h} \leq h^{(2c/3)h^2 - \omega(1)}. \tag{2}$$

Let $\mathcal{A}$ be the set of all vectors in $\{0,1\}^m$ of norm r. Notice that r^2 is an integer, and $\mathcal{A}$ equals the set of all binary vectors with precisely r^2 ones. In particular, the size of $\mathcal{A}$ is

$$|\mathcal{A}| = \binom{m}{r^2} \geq \left(\frac{m}{r^2}\right)^{r^2} \geq \frac{m^{r^2}}{h^{2h^2/\lambda^2}} \geq h^{(c-2)h^2/\lambda^2 - O(1)}. \tag{3}$$

Partition $\mathcal{A}$ according to the residue classes modulo $\mathcal{L}(\mathbf{L})$, i.e., into the sets $\mathcal{A} \cap (\mathcal{L}(\mathbf{L}) - \mathbf{s})$ where $\mathbf{s} \in \mathbb{Z}^m$. Notice that $\mathcal{L}(\mathbf{L})$ has precisely $\det(\mathbf{L})$ cosets of the form $\mathcal{L}(\mathbf{L}) - \mathbf{s}$ with $\mathbf{s} \in \mathbb{Z}^m$, so, on average, the size of $\mathcal{A} \cap (\mathcal{L}(\mathbf{L}) - \mathbf{s})$ is

$$\mathop{E}_{\mathbf{s}}[|\mathcal{A} \cap (\mathcal{L}(\mathbf{L}) - \mathbf{s})|] = \frac{|\mathcal{A}|}{\det(\mathcal{L}(\mathbf{L}))} \geq h^{h^2 \cdot (c\delta - 2)/\lambda^2 + \omega(1)}. \tag{4}$$

Therefore, there must exist a vector $\mathbf{s} \in \mathbb{Z}^m$ such that $\mathcal{A} \cap (\mathcal{L}(\mathbf{L}) - \mathbf{s})$ contains at least $h^{h^2 \cdot (c\delta - 2)/\lambda^2}$ elements.

Now fix $c > 2/\delta$ and $h \geq \lambda \cdot \sqrt{k/(\delta - 2/c)}$, so that

$$|(\mathcal{L}(\mathbf{L}) - \mathbf{s}) \cap \mathcal{A}| > h^{h^2(c\delta - 2)/\lambda^2} > m^k > \sum_{i=0}^{k} \binom{m}{i}.$$

Using (4) and Corollary 1, we get a matrix $\mathbf{T} \in \{0,1\}^{k \times m}$ such that $\mathbf{T}((\mathcal{L}(\mathbf{L}) - \mathbf{s}) \cap \mathcal{A}) \supseteq \{0,1\}^k$. □

It easily follows that GapSVP is NP-hard to approximate within any constant approximation factor under polynomial time nonuniform reduction with one-sided error.

Corollary 2 *GapSVP$_\gamma$ is NP-hard for any constant factor γ under polynomial time (reverse unfaithful) nonuniform reductions with one-sided error. Moreover, for every $\epsilon > 0$ there is a $\delta > 0$ such that GapSVP$_\gamma$ is NP-hard for $\gamma(n) = n^{\delta/\log\log n}$ under (reverse unfaithful) nonuniform reductions with one-sided error running in subexponential time $2^{O(n^\epsilon)}$.*

Proof. By Theorem 6 in Appendix A, TensorCVP$_{\tilde{\gamma}}$ is NP-hard (under deterministic polynomial time reductions) for any constant factor $\tilde{\gamma}$. If follows from Theorem 3 that TensorSVP$_{\gamma_0}$ is NP-hard for some constant $\gamma_0 > 1$ under reverse unfaithful nonuniform reductions. Finally, for any constant γ, applying Theorem 1 with $c = \lceil \log\gamma / \log\gamma_0 \rceil$, we get that GapSVP$_\gamma$ is NP-hard under the same kind of reductions. (Notice that for any constant γ, c is a constant and the reduction in Theorem 1 runs in polynomial time.)

In general, the reduction runs in time polynomial in $N = n^c$, and produces GapSVP$_\gamma$ instances in dimension N that are hard to approximate within a factor $\gamma = \gamma_0^c$. For any $\epsilon > 0$ let $\delta = \epsilon \cdot \log\gamma_0$ and set $c = n^\epsilon / \log n$, so that $N = n^c = 2^{n^\epsilon}$ and the reduction runs in subexponential time $N^{O(1)} = 2^{O(n^\epsilon)}$. The resulting inapproximability factor is $\gamma(N) = \gamma_0^c = N^{\delta/\log\log N}$. □

5 A probabilistic reduction

The nonuniform reduction presented in Section 4 needs as an advise a tuple $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ such that

1. the lattice coset $\mathcal{L}(\mathbf{L}) - \mathbf{s}$ contains many (in fact, at least 2^k) vectors of norm at most r,
2. the image of this set of short vectors under the linear transformation $\mathbf{T}$ includes all binary strings $\{0,1\}^k$.

In Theorem 3 we proved that an advice $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ with these properties exists, leading to a nonuniform reduction. In this section we show that non-uniformity is not essential, and an advise $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ with the desired properties can be efficiently found in probabilistic polynomial time. The idea is simple, and follows the same path as previous work [1, 14, 8]. First we find a coset $\mathcal{L}(\mathbf{L}) - \mathbf{s}$ containing a lot of short vectors. Since the lattice $\mathcal{L}(\mathbf{L})$ has small determinant, the average number of short vectors in a random coset $\mathcal{L}(\mathbf{L}) - \mathbf{s}$ is large, and choosing $\mathbf{s}$ at random will give with high probability a coset containing many short vectors. More specifically, we use a slight generalization of [8, Lemma 13]. The difference between Lemma 4 below and [8, Lemma 13] (beside minor syntactical differences like our use of additive group notation and lattice cosets) is that [8, Lemma 13] assumes the groups Z, L are finite (as in their contexts they are linear codes) while here we only require the quotient Z/L to be finite, as our groups are lattices and have infinite size.

Lemma 4 (Lemma 13 of [8], variant) *Let $(Z, +)$ be an additive group, $L \subset Z$ a subgroup such that the quotient group Z/L is finite, and $B \subset Z$ an arbitrary subset of Z. Let $\mu = |B|/|Z/L|$ the average number of B elements in a uniformly chosen random coset $L + s$. Then, for any $\epsilon > 0$*

$$\Pr_{s \in B}\{|(L+s) \cap B| \leq \epsilon\mu\} \leq \epsilon.$$

Proof Let $S = \{s \in Z/L : |(L+s) \cap B| \leq \epsilon\mu\}$ be the set of cosets $L + s$ such that $(L + s) \cap B$ is small. Clearly, there are at most $|S| \leq |Z/L|$ such cosets, and each one is selected with probability $\Pr s \in B\{s + L\} = |(L+s) \cap B|/|B| \leq \epsilon\mu/|B| = \epsilon/|Z/L|$. So, the probability of selecting a small coset is at most $\Pr_{s \in B}\{s \in S + L\} \leq \sum_{s \in S} \epsilon/|Z/L| = \epsilon$. □

After using Lemma 4 to find a coset $\mathcal{L}(\mathbf{L}) - \mathbf{s}$ that contains many short vectors, we use the following combinatorial theorem from [14], which can be interpreted as a constructive (probabilistic) variant of Sauer's lemma.

Theorem 4 (Theorem 5.9 of [14]) *Let $\mathcal{Z} \subseteq \{0,1\}^m$ be a set of vectors containing exactly u ones. For any k and $\epsilon > 0$, if $|\mathcal{Z}| \geq u! m^{\frac{4\sqrt{uk}}{\epsilon}}$, and $\mathbf{T} \in \{0,1\}^{k \times m}$ is chosen setting each entry to 1 independently at random with probability $p = \frac{1}{4uk}$, then the probability that all binary vectors $\{0,1\}^k$ are contained in $\mathbf{T}(\mathcal{Z}) = \{\mathbf{Tz} : \mathbf{z} \in \mathcal{Z}\}$ is at least $1 - 6\epsilon$.*

With these tools in our hands, we can prove a probabilistic variant of Theorem 3.

Theorem 5 *For any $\gamma < \lambda < \sqrt{3/2}$ and $\tilde{\gamma} = \gamma\sqrt{1 + 4/((\lambda/\gamma)^2 - 1)}$ there is a probabilistic polynomial time reduction from TensorCVP$_{\tilde{\gamma}}$ to TensorSVP$_\gamma$. The reduction is* reverse unfaithful*, i.e., it has one-sided error and always maps* NO *instances to* NO *instances regardless of the value of the randomness.*

Proof. By Theorem 3, there is a nonuniform reduction satisfying the statement of this theorem. Moreover, all that is needed to turn the nonuniform reduction into a randomized one is a probabilistic construction of an advise string $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ satisfying the properties stated in Theorem 3, namely

- $\mathbf{L} \in \mathbb{Z}^{m \times l}$ is a lattice basis with $\tau(\mathbf{L}) \geq \lambda \cdot r$, and
- $\mathbf{T} \in \mathbb{Z}^{k \times m}$ is a linear transformation such that $\mathbf{T}((\mathcal{L}(\mathbf{L}) - \mathbf{s}) \cap \mathcal{B}(r)) \supseteq \{0,1\}^k$.

We give a probabilistic polynomial time construction of $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ satisfying these properties. The gadget is constructed using the same approach as in the proof of Theorem 3, but for different values of $c > 4/(\lambda^2\delta)$ and $h \geq 4\lambda/(\epsilon(\delta - 4/c)) \cdot k = O(k)$. Let $m = h^c$, $r = \sqrt{\lfloor (h/\lambda)^2 \rfloor}$, $\delta = 1 - 2\lambda^2/3 > 0$, $\mathbf{L} \in \mathbb{Z}^{m \times m}$ and $\mathcal{A} \subseteq \{0,1\}^m$ be as defined in the proof of Theorem 3, so that $h \geq \lambda r$ and (1) and (4) hold true. We recall that $\mathcal{A}$ is the set of all vectors in $\{0,1\}^m$ of norm r, and $\mathbf{L} \in \mathbb{Z}^{m \times m}$ is a basis for the lattice described in Theorem 2 with $\tau(\mathbf{L}) \geq h \geq \lambda r$.

Now choose $\mathbf{s} \in -\mathcal{A}$ at random and let $\mathcal{Z} = (\mathcal{L}(\mathbf{L}) - \mathbf{s}) \cap \mathcal{A}$. By (4) and Lemma 4, we have

$$|\mathcal{Z}| \geq h^{h^2(c\delta - 2)/\lambda^2} \tag{5}$$

except with negligible probability $h^{-\omega(1)} < \epsilon$. By (1) and our choice of $k \leq h\epsilon(\delta - 4/c)/(4\lambda)$, we have

$$(r^2)! m^{4rk/\epsilon} \leq (r^2)^{r^2} h^{4chk/(\lambda\epsilon)} \leq h^{2h^2/\lambda^2} \cdot h^{h^2(c\delta - 4)/\lambda^2} = h^{h^2(c\delta - 2)/\lambda^2} \leq |\mathcal{Z}|.$$

So, a matrix $\mathbf{T}$ chosen at random as in Theorem 4 (with $u = r^2$) satisfies $\mathbf{T}(\mathcal{Z}) \supseteq \{0,1\}^k$ with probability at least $1 - 6\epsilon$. So, by union bound, the probabilistic construction produces a gadget $(\mathbf{L}, \mathbf{s}, \mathbf{T}, r)$ satisfying all required properties except with probability at most 7ϵ. □

As in the previous section, the inapproximability factor can be amplified using the tensor product. The proof is identical to that of Corollary 2.

Corollary 3 *GapSVP$_\gamma$ is NP-hard for any constant factor γ under polynomial time (reverse unfaithful) nonuniform reductions with one-sided error. Moreover, for every $\epsilon > 0$ there is a $\delta > 0$ such that GapSVP$_\gamma$ is NP-hard for $\gamma(n) = n^{\delta/\log\log n}$ under (reverse unfaithful) nonuniform reductions with one-sided error running in subexponential time $2^{O(n^\epsilon)}$.*

6 Conclusion

We proved hardness of approximation results for the Shortest Vector Problem with approximation factors matching the best currently known results [10, 9], but under probabilistic reduction with one-sided error. In particular, our reductions make more restricted use of randomness than [10, 9] and may be easier to derandomize. Randomness in our reduction is used only to produce a lattice coset $\mathcal{L}(\mathbf{L}) - \mathbf{s}$ with large minimum (τ) distance and still containing a large number of short vectors, which maps via an integer linear transformation $\mathbf{T}$ onto the set of all binary vectors $\{0,1\}^k$. We gave a deterministic polynomial time construction of the lattice $\mathcal{L}(\mathbf{L})$, and randomness is used only for the selection of $\mathbf{s}$ and $\mathbf{T}$. In fact, matrix $\mathbf{T}$ is chosen at random mostly as a byproduct of the fact that the selection of $\mathbf{s}$ is probabilistic: intuitively, no matrix $\mathbf{T}$ is good for every $\mathbf{s}$, so if $\mathbf{s}$ is chosen at random, then $\mathbf{T}$ must be chosen at random as well. We believe that all that is needed in order to derandomize our proof is an explicit description of a vector $\mathbf{s}$ such that $\mathcal{L}(\mathbf{L}) - \mathbf{s}$ contains many short vectors. With such a vector $\mathbf{s}$ (and a proof that $\mathbf{s}$ is good), finding a matrix $\mathbf{T}$ that maps all short vectors in $\mathcal{L}(\mathbf{L}) - \mathbf{s}$ to $\{0,1\}^k$ is likely to be easy.

7 Acknowledgments

This work was supported in part by NSF grants CNS-1117936. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation.

References

[1] M. Ajtai. The shortest vector problem in L2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of STOC '98*, pages 10–19. ACM, May 1998.

[2] S. Arora, L. Babai, J. Stern, and E. Z. Sweedyk. The hardness of approximate optima in lattices, codes, and systems of linear equations. *Journal of Computer and System Sciences*, 54(2):317–331, Apr. 1997. Preliminary version in FOCS'93.

[3] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[4] J.-Y. Cai and A. P. Nerurkar. Approximating the SVP to within a factor $(1 + 1/dim^{\epsilon})$ is NP-hard under randomized reductions. *Journal of Computer and System Sciences*, 59(2):221–239, Oct. 1999. Preliminary version in CCC 1998.

[5] Q. Cheng and D. Wan. A deterministic reduction for the gap minimum distance problem: [extended abstract]. In M. Mitzenmacher, editor, *STOC*, pages 33–38. ACM, 2009.

[6] J. H. Conway and N. J. A. Sloane. *Sphere packings, lattices and groups*. Springer Verlag, 3rd edition, 1998.

[7] I. Dinur, G. Kindler, R. Raz, and S. Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003. Preliminary version in FOCS 1998.

[8] I. Dumer, D. Micciancio, and M. Sudan. Hardness of approximating the minimum distance of a linear code. *IEEE Transactions on Information Theory*, 49(1):22–37, Jan. 2003. Preliminary version in FOCS 1999.

[9] I. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In *Proceedings of STOC*, pages 469–477. ACM, June 2007.

[10] S. Khot. Hardness of approximating the shortest vector problem in lattices. *Journal of the ACM*, 52(5):789–808, Sept. 2005. Preliminary version in FOCS 2004.

[11] S. Khot and P. Austrin. A simple deterministic reduction for the gap minimum distance of code problem. In *ICALP, Proceedings*, 2011. To appear.

[12] J. C. Lagarias and A. M. Odlyzko. Solving low-density subset sum problems. *Journal of the ACM*, 32(1):229–246, Jan. 1985.

[13] D. Micciancio. The hardness of the closest vector problem with preprocessing. *IEEE Transactions on Information Theory*, 47(3):1212–1215, Mar. 2001.

[14] D. Micciancio. The shortest vector problem is NP-hard to approximate to within some constant. *SIAM Journal on Computing*, 30(6):2008–2035, Mar. 2001. Preliminary version in FOCS 1998.

[15] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems: a cryptographic perspective*, volume 671 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, Massachusetts, Mar. 2002.

[16] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In *Proceedings of STOC*, pages 351–358, 2010.

[17] O. Regev and R. Rosen. Lattice problems and norm embeddings. In *Proceedings of STOC*, pages 447–456. ACM, June 2006.

[18] P. van Emde Boas. Another NP-complete problem and the complexity of computing short vectors in a lattice. Technical Report 81-04, Mathematische Instituut, Universiry of Amsterdam, 1981. Available on-line at URL `http://turing.wins.uva.nl/~peter/`.

[19] A. Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. on Information Theory*, 43(6):1757–1766, 1997.

A NP-hardness of TensorCVP

The NP-hardness of TensorCVP is proved by reduction from *exact set cover*. This result is well known, and has been used in many previous works on the complexity of lattice problems. Here we give a slightly simpler proof than the one typically found in the literature, that avoids the introduction of auxiliary variables and large constants.

Remember that an instance of *exact set cover* consists of a collection of sets $S_1, \ldots, S_n \subseteq \{1, \ldots, u\}$ and an integer $t \leq n$. A cover is a subcollection $C \subseteq \{1, \ldots, n\}$ such that $\bigcup_{i \in C} S_i = \{1, \ldots, u\}$. A cover is *exact* if the sets S_i ($i \in C$) in the cover are disjoint, i.e., $\{S_i\}_{i \in C}$ is a *partition* of $\{1, \ldots, u\}$. When reducing set cover problems to lattice problems it is convenient to represent the collection $\{S_1, \ldots, S_n\}$ as a matrix $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_n] \in \{0,1\}^{u \times n}$ where the columns $\mathbf{s}_i$ are the indicator vectors of the sets S_i. Using matrix notation, a cover of size t is represented by a binary vector $\mathbf{c} \in \{0,1\}^n$ with t ones such that $\mathbf{Sc} \geq \mathbf{1}$, where $\mathbf{1}$ is the all-ones vector and the inequality holds component-wise. The cover is exact if $\mathbf{Sc} = \mathbf{1}$.

Definition 6 *For any $\gamma \geq 1$, an instance of the γ-approximate exact set cover problem is a pair $(\mathbf{S}, t)$ where $\mathbf{S} \in \{0,1\}^{u \times n}$ and $t \in \{1, \ldots, n\}$.*

- *$(\mathbf{S}, t)$ is a* YES *instance if there is an exact cover of size at most t, i.e., a binary vector $\mathbf{c} \in \{0,1\}^n$ such that $\|\mathbf{c}\|_H \leq t$ and $\mathbf{Sc} = \mathbf{1}$.*
- *$(\mathbf{S}, t)$ is a* NO *instance if all covers have size bigger than γt, i.e., all binary vectors $\mathbf{c} \in \{0,1\}^n$ such that $\mathbf{Sc} \geq \mathbf{1}$ have Hamming weight $\|\mathbf{c}\|_H > \gamma t$.*

Theorem 6 *For any constant $\gamma \geq 1$, TensorCVP$_\gamma$ is NP-hard.*

Proof. We use the fact that Exact Set Cover is NP-hard for any constant approximation factor γ, and we reduce it to TensorCVP$_{\sqrt{\gamma}}$. On input an exact set cover instance $(\mathbf{S}, t)$, the reduction produces a TensorCVP$_{\sqrt{\gamma}}$ instance $(\mathbf{B}, \mathbf{y}, \sqrt{t})$ where $\mathbf{B} \in \mathbb{Z}^{n \times k}$ and $\mathbf{y} \in \mathbb{Z}^n$. Let $\mathbf{B}$ be a basis for the lattice of all integer vectors $\mathbf{v}$ such that $\mathbf{Sv} = \mathbf{0}$, and let $\mathbf{y}$ be an arbitrary integer solution to $\mathbf{Sy} = \mathbf{1}$. (Both $\mathbf{B}$ and $\mathbf{y}$ can be efficiently computed using linear algebra. If no solution $\mathbf{y}$ exists, then $(\mathbf{S}, t)$ is necessarily a NO instance, and the reduction can output an arbitrary NO instance of TensorCVP.) We need to prove that the reduction is correct.

If $(\mathbf{S}, t)$ is a YES instance, then there is an exact cover of size t, i.e., a vector $\mathbf{c} \in \{0,1\}^n$ with at most t ones such that $\mathbf{Sc} = \mathbf{1}$. It follows that $\mathbf{S}(\mathbf{y} - \mathbf{c}) = \mathbf{0}$, i.e., $\mathbf{y} - \mathbf{c}$ is a lattice vector. Moreover this lattice vector is within distance $\|\mathbf{y} - (\mathbf{y} - \mathbf{c})\| = \|\mathbf{c}\| \leq \sqrt{t}$ from $\mathbf{y}$, proving that $(\mathbf{B}, \mathbf{y}, \sqrt{t})$ is a YES instance of TensorCVP. Now assume $(\mathbf{S}, t)$ is a NO instance. Notice that for any $\mathbf{v}$ in the linear span of $\mathbf{B}$, $\mathbf{S}(\mathbf{y} - \mathbf{v}) = \mathbf{Sy} - \mathbf{Sv} = \mathbf{1}$. So, the nonzero coordinates of $\mathbf{y} - \mathbf{v}$ form a set cover. It follows that $\mathbf{y} - \mathbf{v}$ must have more than γt nonzero coordinates, i.e., $\sqrt{\|\mathbf{y} - \mathbf{v}\|_H} > \sqrt{\gamma t}$. □

B Sauer's lemma

In this section we state and prove Sauer's lemma. The proof is well known and it is reported here just to make the paper self contained.

Lemma 5 (Sauer's Lemma) *Let M be a set of size m, and $\mathcal{A} \subset 2^M$ be an arbitrary collection of subsets of M. For any integer k such that $|\mathcal{A}| \geq \sum_{i=0}^{k} \binom{m}{i}$, there exists a subset $T \subset M$ of size $|T| = k$ which is shattered by $\mathcal{A}$, i.e., $\{A \cap T : A \in \mathcal{A}\} = 2^T$.*

Proof For any m and k, let $[m,k] = \sum_{i=0}^{k} \binom{m}{k}$ be the number of subsets of $M = \{1,\ldots,m\}$ of size at most k. For any $T \subseteq M$, define the restriction of $\mathcal{A}$ to T as $\mathcal{A}|_T = \{A \cap T : A \in \mathcal{A}\}$. We prove, by induction on $m+k$, that

$$(|\mathcal{A}| \geq [m,k]) \Rightarrow \exists T \subseteq M . |T| = k \wedge \mathcal{A}|_T = 2^T.$$

The base case of the induction ($m+k=0$) is trivial. So, consider the inductive step $|\mathcal{A}| \geq [m,k] > 0$. Pick an element a from M and define $M' = M \setminus \{a\}$ and the following two collections of subsets of M':

$$\mathcal{A}_0 = \{A \subseteq M' : A \in \mathcal{A}\}$$

$$\mathcal{A}_1 = \{A \subseteq M' : A \cup \{a\} \in \mathcal{A}\}.$$

If $|\mathcal{A}_0 \cup \mathcal{A}_1| \geq [m',k]$, then by inductive hypothesis there exists a set $T \subseteq M' \subset M$ of size $|T| = k$ such that $(\mathcal{A}_0 \cup \mathcal{A}_1)|_K = 2^T$. Since $a \notin T$, we have $\mathcal{A}|_T = (\mathcal{A}_0 \cup \mathcal{A}_1)|_T = 2^T$ and we are done.

So, assume $|\mathcal{A}_0 \cup \mathcal{A}_1| < [m',k]$, and notice that

$$|\mathcal{A}_0 \cup \mathcal{A}_1| + |\mathcal{A}_0 \cap \mathcal{A}_1| = |\mathcal{A}_0| + |\mathcal{A}_1| = |\mathcal{A}| \geq [m,k] = [m',k] + [m',k-1].$$

Since $|\mathcal{A}_0 \cup \mathcal{A}_1| < [m',k]$, it must be $|\mathcal{A}_0 \cap \mathcal{A}_1| \geq [m-1,k-1]$, and, by inductive hypothesis, there exists a set $T' \subseteq M' \subset M$ of size $|T'| = k-1$ such that $(\mathcal{A}_0 \cap \mathcal{A}_1)|_{T'} = 2^{T'}$. We show that $\mathcal{A}|_K = 2^K$ where $K = K' \cup \{a\}$ is a set of size $|T| = |T'|+1 = k$. The inclusion $\mathcal{A}|_T \subseteq 2^T$ is obvious. So, let's prove $2^T \subseteq \mathcal{A}|_T$. Notice that for any $A \in 2^T$, the set $A \setminus \{a\}$ belongs to both $\mathcal{A}_0|_{T'}$ and $\mathcal{A}_1|_{T'}$. Therefore $A \setminus \{a\} \in \mathcal{A}_T$ and $A \cup \{a\} \in \mathcal{A}_T$. Since A equals either $A \setminus \{a\}$ or $A \cup \{a\}$, we conclude that $A \in \mathcal{A}_T$. □

C The Extended BCH codes

In this section we briefly recall the construction of *extended primitive narrow sense binary BCH codes*, and their most basic properties as used in Lemma 1. For brevity, we refer to these codes just as "extended BCH codes". Extended BCH codes can be defined for any block length $m = 2^\kappa$ that is a power of 2, and are obtained by appending a parity check bit to a BCH code of block length $n = m-1$. BCH codes are polynomial codes, i.e., they can be described algebraically as the set of all (coefficient vectors of) polynomials of degree less than n that are divisible by a given generating polynomial $g(X) \in \mathbb{F}_2[X]$. The co-dimension of a polynomial code equals the degree $n-k = \deg(g)$ of the generating polynomial. Let α be a generator of the multiplicative group of $\mathbb{F}_{2^m}$, the finite field with 2^m elements. A basic fact in the theory of polynomial codes is that if $g(\alpha^i) = 0$ for t consecutive powers of α, then the polynomial code generated by $g(X)$ has minimum distance at least $t+1$.

For any even $h \leq m$, the BCH code with designed minimum distance $h-1 \leq m-1$ is the polynomial code generated by the least common multiple $g_h(X)$ of the minimal polynomials $p_1(X), p_3(X), \ldots, p_{h-3}(X)$ of the first $h/2-1$ odd powers $\alpha^1, \alpha^3, \ldots, \alpha^{h-3}$ of the primitive element α. Notice that for any even power α^{2j}, $g_h(\alpha^{2j}) = (g_h(\alpha^j))^2$ because squaring is a linear operation over $\mathbb{F}_m$. So, $g_h(\alpha^j) = 0$ for all $j = 1,\ldots,h-2$, and the minimum distance of the BCH code is at least $h-1$. The extended BCH code $\mathrm{EBCH}_h^m[m,k,d]_2$ is obtained by appending a parity check bit to the cyclic code generated by $g_h(X)$. Since $h-1$ is odd, appending a parity check bit increases the (designed) minimum distance of the code to $d \geq h$. The block length and co-dimension also increase by 1, while the dimension of the code remains the same. Since the

degree of each minimal polynomial h_j is at most κ, the degree of g_h is bounded by $\kappa \cdot (h/2-1)$, and the co-dimension of $\mathrm{EBCH}_h^m[m,k,d]$ is at most $m-k \le \kappa(h/2-1)+1$. Notice that for any $h \le h'$, $g_{h'}$ is a multiple of g_h, and therefore $\mathrm{EBCH}_h^m \supseteq \mathrm{EBCH}_{h'}^m$.

In summary, for any even h, the extended BCH code $\mathrm{EBCH}_h^m[m,k,d]$ of block length $m=2^\kappa$ and minimum distance $d \ge h$ has co-dimension $m-k \le \kappa(h/2-1)+1$. Moreover, the codes satisfy $\{0,1\}^m = \mathrm{EBCH}_0^m \supseteq \mathrm{EBCH}_2^m \supseteq \cdots \supseteq \mathrm{EBCH}_m^m$.

D Proof of Theorem 2

In this Section we prove Theorem 2. We need to prove that for any $m=2^\kappa$ and $h=2^\ell$ with $h \le \sqrt{m}$, there is an m-dimensional full rank integer lattice $\mathbf{L}$ such that $\tau_2(\mathcal{L}(\mathbf{L})) \ge h$ and $\det(\mathcal{L}(\mathbf{L})) < m^{(h^2/1.5)-\ell}$.

Let $\mathbf{C}_0 \supseteq \mathbf{C}_1 \supseteq \cdots \supseteq \mathbf{C}_\ell$ be the sequence of binary linear codes $\mathbf{C}_i[m,k_i,d_i]_2$ from Lemma 1. We recall that these are codes of common block length m, minimum distance $d_i \ge 4^i$, and co-dimension $m-k_i \le \kappa(4^i/2-1)+1$, and $\mathbf{C}_0[m,m,1]_2 = \mathbb{Z}_2^m$. We combine these codes into a lattice using "construction D" from [6, Chapter 8]. More specifically, we define the m-dimensional integer lattice $\mathcal{L}(\mathbf{L})$ generated by the columns of $2^{\ell-i}\mathbf{C}_i$ for all $i=0,\ldots,\ell$. Of course, the vectors in $2^\ell\mathbf{C}_0, 2^{\ell-1}\mathbf{C}_1, \ldots, \mathbf{C}_\ell$ are not linearly independent, but a basis for the lattice they generate can be easily obtained as follows. Using the inclusions $\mathbf{C}_0 \supseteq \cdots \supseteq \mathbf{C}_\ell$, we may assume that each generating matrix $\mathbf{C}_i$ equals the last k_i columns of $\mathbf{C}_0$. In other words, $\mathbf{C}_0 = [\mathbf{K}_0, \ldots, \mathbf{K}_\ell]$, and each generating matrix $\mathbf{C}_i = [\mathbf{K}_i, \mathbf{C}_{i+1}]$ is obtained extending the generating matrix of the next code in the sequence $\mathbf{C}_{i+1}$ with $k_i' = k_i - k_{i+1}$ more columns $\mathbf{K}_i$. (For convenience, we also define $k_{\ell+1}=0$ and $k_\ell' = k_\ell - k_{\ell+1} = k_\ell$.) By properly choosing the order of the coordinates, and performing elementary column operations, we may further assume that each $\mathbf{K}_i$ has the form

$$\mathbf{K}_i = \begin{bmatrix} \mathbf{K}_i' \\ \mathbf{I} \\ \mathbf{O} \end{bmatrix}$$

where $\mathbf{K}_i' \in \mathbb{F}_2^{(m-k_i)\times k_i'}$, $\mathbf{I}$ is the $k_i' \times k_i'$ identity matrix, and $\mathbf{O}$ is the $k_{i+1} \times k_i'$ all-zero matrix.

Consider the $m \times m$ integer matrix

$$\mathbf{L} = [2^\ell\mathbf{K}_0, 2^{\ell-1}\mathbf{K}_1, \ldots, \mathbf{K}_\ell].$$

The columns of $\mathbf{L}$ are a subset of $\mathbf{C}_0, 2\mathbf{C}_1, \ldots, 2^\ell\mathbf{C}_\ell$. Moreover, all vectors in $2^\ell\mathbf{C}_0, 2^{\ell-1}\mathbf{C}_1, \ldots, \mathbf{C}_\ell$ can be obtained by multiplying the columns of $\mathbf{L}$ by appropriate powers of 2. Therefore $\mathcal{L}(\mathbf{L})$ is precisely the lattice generated by $2^\ell\mathbf{C}_0, 2^{\ell-1}\mathbf{C}_1, \ldots, \mathbf{C}_\ell$.

Consider an arbitrary nonzero lattice vector $\mathbf{v} = \sum_i (2^i \cdot \mathbf{K}_i)\mathbf{x}_i = \sum_i \mathbf{K}_i\mathbf{y}_i$, where $\mathbf{y}_i = 2^i\mathbf{x}_i$. We want to prove that $\tau_2(\mathbf{v}) \ge h$. Let $2^P = \mathrm{pow}_2(\mathbf{y}_0, \ldots, \mathbf{y}_k)$ be the largest power of 2 that divides all $\mathbf{y}_i$'s. Clearly, 2^P also divides $\mathbf{v}$. If $P \ge \ell$, then we immediately get $\tau_2(\mathbf{v}) \ge \mathrm{pow}_2(\mathbf{v}) \ge 2^\ell = h$. So, assume $P < \ell$ and let $p = \min\{i : \mathbf{y}_i \ne \mathbf{0}, \mathrm{pow}_2(\mathbf{y}_i) = 2^P\}$ be the smallest index such that $\mathbf{y}_p$ is divisible precisely by 2^P. Notice that $P \ge \ell - p$ because $2^P = \mathrm{pow}_2(\mathbf{y}_p) = \mathrm{pow}_2(2^{\ell-p}\mathbf{x}_p) \ge 2^{\ell-p}$. By definition of p and P, all $\mathbf{y}_i/2^P$ are integer vectors, $\mathbf{y}_p/2^P \ne \mathbf{0} \pmod 2$ and $\mathbf{y}_i/2^P = \mathbf{0} \pmod 2$ for all $i < p$. So,

$$\|\mathbf{v}\|_H = \|\mathbf{v}/2^P\|_H \ge \|(\mathbf{v}/2^P) \bmod 2\|_H = \|\sum_{i\ge p} \mathbf{K}_i(\mathbf{y}_i/2^P) \bmod 2\|_H \ge d_p \ge 4^p$$

where we have used the fact that $\sum_{i\ge p} \mathbf{K}_i(\mathbf{y}_i/2^P) \bmod 2$ is a nonzero codeword in $\mathbf{C}_p$. It follows that

$$\tau_2(\mathbf{v}) = \mathrm{pow}_2(\mathbf{v})\sqrt{\|\mathbf{v}\|_H} \ge 2^P \cdot \sqrt{4^p} \ge 2^{\ell-p} \cdot 2^p = 2^\ell.$$

This proves that $\tau_2(\mathbf{L}) \ge 2^\ell = h$.

In order to bound the determinant of the lattice, we notice that, by our choice of $\mathbf{K}_i$, the matrix $\mathbf{C}_0$ is upper triangular. It follows that $\mathbf{L}$ is also a triangular matrix with k_i' diagonal entries equal to $2^{\ell-i}$ for

$i = 0, \ldots, \ell$. So, the determinant satisfies

$$\log_2 \det(\mathbf{L}) = \sum_{i \le \ell} (\ell - i) k'_i = \sum_{i=1}^{\ell} (m - k_i).$$

Finally, using the bound on the co-dimension $m - k_i \le \kappa \cdot (4^i/2 - 1) + 1$ from Lemma 1 we get

$$\sum_{i=1}^{\ell} (m - k_i) \le \sum_{i=1}^{\ell} \left(\kappa \cdot \frac{4^i}{2} - (\kappa - 1) \right) = \kappa \frac{4^\ell - 1}{1.5} - (\kappa - 1)\ell < k \left(\frac{h^2}{1.5} - \ell \right),$$

which, substituted into the expression for the determinant gives $\det(\mathbf{L}) \le m^{\frac{h^2}{1.5} - \ell}$. $\square$

ECCC ISSN 1433-8092

http://eccc.hpi-web.de