comput. complex. **14** (2005), 90 – 121
1016-3328/05/020090–32
DOI 10.1007/s00037-005-0193-y

computational complexity

THE COMPLEXITY OF THE COVERING RADIUS PROBLEM

Venkatesan Guruswami, Daniele Micciancio, and Oded Regev

Abstract. We initiate the study of the computational complexity of the covering radius problem for lattices, and approximation versions of the problem for both lattices and linear codes. We also investigate the computational complexity of the shortest linearly independent vectors problem, and its relation to the covering radius problem for lattices.

For the covering radius on n-dimensional lattices, we show that the problem can be approximated within any constant factor $\gamma(n) > 1$ in random exponential time $2^{O(n)}$. We also prove that suitably defined gap versions of the problem lie in AM for $\gamma(n) = 2$, in coAM for $\gamma(n) = \sqrt{n/\log n}$, and in $\mathsf{NP} \cap \mathsf{coNP}$ for $\gamma(n) = \sqrt{n}$.

For the covering radius on n-dimensional linear codes, we show that the problem can be solved in deterministic polynomial time for approximation factor $\gamma(n) = \log n$, but cannot be solved in polynomial time for some $\gamma(n) = \Omega(\log \log n)$ unless NP can be simulated in deterministic $n^{O(\log \log \log n)}$ time. Moreover, we prove that the problem is NP-hard for *any* constant approximation factor, it is Π_2-hard for *some* constant approximation factor, and that it is unlikely to be Π_2-hard for approximation factors larger than 2 (by giving an AM protocol for the appropriate gap problem). This is a natural hardness of approximation result in the polynomial hierarchy.

For the shortest independent vectors problem, we give a coAM protocol achieving approximation factor $\gamma(n) = \sqrt{n/\log n}$, solving an open problem of Blömer and Seifert (STOC'99), and prove that the problem is also in coNP for $\gamma(n) = \sqrt{n}$. Both results are obtained by giving a gap-preserving *nondeterministic* polynomial time reduction to the closest vector problem.

Keywords. Lattices, linear codes, covering radius, approximation algorithms, hardness of approximation, complexity classes, polynomial time hierarchy, interactive proofs.

Subject classification. 68Q17, 68Q25, 11H06, 11H31, 94B05.

1. Introduction

Given a (possibly infinite) set of points P in a metric space $\mathcal{M}$, the covering radius of P in $\mathcal{M}$ is the smallest number r such that spheres of radius r centered at all points in P cover the entire space. Two especially important cases are when P is a point lattice in Euclidean space, and when P is a linear code[1] over a finite field with the Hamming metric. Determining the covering radius of a given lattice or code is a fundamental problem in the study of point lattices and error correcting codes, but it has received very little attention so far from a computational point of view.

In coding theory, the covering radius is a fundamental parameter of a code and good covering codes have a number of applications and interconnections with other areas of mathematics. An entire book has been written on the subject and we refer the reader to (Cohen *et al.* 1997, Chap. 1) for a discussion of various applications of covering codes. Though the minimum distance plays a more central role in uses of codes for error correction, the covering radius is also related to the error correction capability of the code, since if it is less than the distance, the code is maximal and no vector in the Hamming space can be added without worsening the code's distance. Moreover, assuming we perform maximum likelihood decoding of each received word to its nearest codeword, the covering radius measures the largest number of errors in any correctable error pattern (since if the number of errors exceeds the covering radius, the nearest codeword is definitely not the transmitted codeword). For lattices, the covering radius is closely related to the *vector quantization problem*. In this problem, arbitrary real vectors are rounded to nearby points from a discrete set (e.g., a lattice), and the covering radius of the discrete set of points represents the maximum possible error incurred in this process. We refer the reader to Conway & Sloane (1998) for further information about the use of lattices in vector quantization. Besides these applications, attention to the covering radius problem, specifically from a computational complexity point of view, has recently been drawn by Micciancio (2004) who showed that this problem can be used to get tighter connections between the average- and worst-case complexity of lattice problems.

In the covering radius problem, given a lattice (or a code) and some value r, we are supposed to decide if the covering radius is at most r (see Definitions 2.3 and 2.5). Notice that the covering radius problem (CRP) involves a quantifier alternation: determining if the covering radius is at most r is equivalent to

[1] Unless specified otherwise, all codes considered in this paper are linear, and we will refer to them simply as codes.

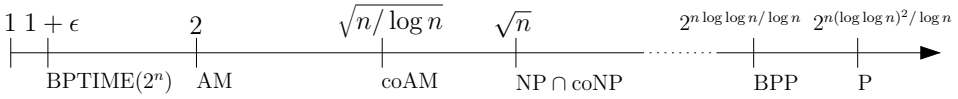

Figure 1.1: The complexity of the covering radius problem for lattices (some constants omitted)

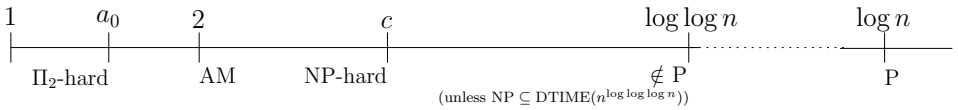

Figure 1.2: The complexity of the covering radius problem for codes (some constants omitted)

proving that *for any* point in space *there exists* a point in the set (lattice or code) within distance at most r. So, the natural algorithms to solve this problem are in Π_2, at the second level of the polynomial hierarchy (Stockmeyer 1977). To date, the problem is not even known to be solvable in nondeterministic polynomial time, and it is conjectured in Micciancio (2004) that the problem is NP-hard. In the case of linear codes, evidence that the problem cannot be efficiently solved was given by McLoughlin (1984), who showed that computing the exact solution of the problem is Π_2-hard, giving one of the most natural complete problems for Π_2.

McLoughlin's result is essentially the only complexity result for the covering radius problem we are aware of. (See Section 3 for a few additional related papers.) We remark that the result of McLoughlin (1984) only applies to linear codes, and to the exact version of the covering radius problem. Many natural questions immediately arise: is the covering radius of a linear code also hard to approximate? What can we say about the covering radius problem for point lattices?

In this paper, we initiate the study of the complexity of the covering radius problem for point lattices, and the complexity of approximating the covering radius in both lattices and linear codes. We also consider the related problem of computing a maximal set of short linearly independent vectors in a lattice (the "shortest independent vectors problem", or SIVP), and its relation to CRP. We present several new results about these problems. Specifically, for the promise problems naturally associated to approximating the covering radius of an n-dimensional point lattice within a factor $\gamma(n)$ we show that (see Definition 2.3 and Figure 1.1)

1. for any constant $\gamma(n) > 1$ the problem can be solved probabilistically in time $2^{O(n)}$;
2. for $\gamma(n) = 2$ the problem is in AM;
3. for $\gamma(n) = \sqrt{n/\log n}$ the problem is in coAM;
4. for $\gamma(n) = \sqrt{n}$ the problem is in $\mathsf{NP} \cap \mathsf{coNP}$;
5. for $\gamma(n) = 2^{\Omega(n \log\log n/\log n)}$ the problem can be solved in random polynomial time;
6. for $\gamma(n) = 2^{\Omega(n(\log\log n)^2/\log n)}$ the problem can be solved in deterministic polynomial time.

The third result shows that approximating the covering radius of a lattice within $\gamma(n) = O(\sqrt{n/\log n})$ is not NP-hard unless the polynomial hierarchy collapses. Moreover, the second result shows that approximating the covering radius within $\gamma(n) = 2$ is unlikely to be Π_2-hard because the problem is also in AM. Whether or not the problem is NP-hard (or Π_2-hard) for smaller approximation factors, or even for the exact version, remains an open problem.[2] Most of our results follow by simple reductions from the covering radius problem to other lattice problems, like the closest vector problem, and the shortest independent vectors problem.

For SIVP, we show that approximating the problem within $\gamma(n) = \sqrt{n/\log n}$ is in coAM and that approximating the problem within $\gamma(n) = \sqrt{n}$ is in coNP. Previously, Blömer & Seifert (1999) proved that approximating SIVP to within $\gamma(n) = n/\sqrt{\log n}$ is in coAM, and suggested that the problem might be NP-hard for approximation factors $\gamma(n) = n^{1-\epsilon}$ for any $\epsilon > 0$. Our results improve their coAM bound and demonstrate that their conjecture is false, unless the polynomial hierarchy collapses.

Now let us move to the problem of approximating the covering radius of a linear code. For this problem we can give a better picture of the situation, and in particular show hardness of approximation results. Specifically, for the promise problem naturally associated to approximating the covering radius of a linear code within a factor $\gamma(n)$ (where n denotes the block length of the code) we show that (see Definition 2.5 and Figure 1.2)

[2] It is mentioned in (Conway & Sloane 1998, Sec. 1.4, p. 40) that the covering radius problem on lattices has been shown to be NP-hard, but the cited reference (van Emde Boas 1981) does not contain any such result.

1. the problem can be solved in polynomial time for $\gamma(n) = 1 + \log_q n$ (for arbitrary linear codes over the finite field with q elements[3]);
2. the problem cannot be solved in polynomial time for $\gamma(n) = c_0 \log\log n$ for some $c_0 > 0$, unless NP can be simulated in deterministic $n^{O(\log\log\log n)}$ time;
3. the problem is NP-hard for $\gamma(n) = c$ for *any* constant $c \geq 1$;
4. the problem is in AM for $\gamma(n) = 2$;
5. the problem is Π_2-hard for $\gamma(n) = a_0$ for *some* constant $a_0 > 1$.

The last result is a very natural hardness of approximation result for a class in the polynomial-time hierarchy beyond NP and coNP. We refer the reader to recent surveys (Schaefer & Umans 2002a,b) on the topic of completeness and hardness of approximation in the polynomial time hierarchy. The fourth result suggests that the problem is Π_2-hard only for small constant approximation factors, and for $\gamma(n) \geq 2$ the problem is unlikely to be Π_2-hard. Another interesting consequence of this is that it is unlikely that there is a polynomial time computable transformation that takes a linear code and outputs another linear code whose covering radius is, say, the square of the original covering radius. Indeed, such a transformation would imply that the covering radius problem is Π_2-hard for $\gamma(n) \geq 2$. This should be contrasted with the fact that for the minimum distance such a construction does exist. These consequences are described in more detail later.

Organization. The rest of the paper is organized as follows. In Section 2 we formally define lattice and coding problems, and basic notation and techniques used throughout the paper. In Section 3 we give a more detailed account of previous work related to this paper. In Section 4 we present our results for lattice problems, including proof systems and algorithms for computing the covering radius, and the improved proof systems for the shortest independent vectors problem. In Section 5 we present our results about the covering radius problem for linear codes. Section 6 concludes with a discussion of open problems.

2. Preliminaries

For any real x, $\lfloor x \rfloor$ denotes the largest integer not greater than x, and $\lfloor x \rceil = \lfloor x + 1/2 \rfloor$ is the rounding of x to the closest integer. We write log for the

[3] For the case of binary fields, we prove this result for $\gamma(n) = \log_2 n$.

logarithm to base 2, and $\log_q$ when the base q is any number possibly different from 2. Let $\mathbb{R}$ and $\mathbb{Z}$ be the sets of reals and integers, respectively. The m-dimensional Euclidean space is denoted $\mathbb{R}^m$. We use bold lower case letters (e.g., $\mathbf{x}$) to denote vectors, and bold upper case letters (e.g., $\mathbf{M}$) to denote matrices. The ith coordinate of $\mathbf{x}$ is denoted x_i. If $S \subseteq \mathbb{R}^n$ is an arbitrary region of space, and $\mathbf{x} \in \mathbb{R}^n$ is a vector, $S+\mathbf{x} = \{\mathbf{y}+\mathbf{x} : \mathbf{y} \in S\}$ denotes a copy of S shifted by $\mathbf{x}$. The Euclidean norm (also known as the ℓ_2 norm) of a vector $\mathbf{x} \in \mathbb{R}^n$ is $\|\mathbf{x}\| = \sqrt{\sum_i x_i^2}$, and the associated distance is $\operatorname{dist}(\mathbf{x},\mathbf{y}) = \|\mathbf{x}-\mathbf{y}\|$. The distance function is extended to sets in the customary way: $\operatorname{dist}(\mathbf{x},S) = \operatorname{dist}(S,\mathbf{x}) = \min_{\mathbf{y}\in S} \operatorname{dist}(\mathbf{x},\mathbf{y})$ and $\operatorname{dist}(S,Z) = \min_{\mathbf{x}\in S,\,\mathbf{y}\in Z} \operatorname{dist}(\mathbf{x},\mathbf{y})$. We often use matrix notation to denote sets of vectors. For example, a matrix $\mathbf{S} \in \mathbb{R}^{m\times n}$ represents the set $\{\mathbf{s}_1,\ldots,\mathbf{s}_n\}$ of m-dimensional vectors, where $\mathbf{s}_1,\ldots,\mathbf{s}_n$ are the columns of $\mathbf{S}$. The linear space spanned by a set $\mathbf{S}$ of vectors is denoted $\operatorname{span}(\mathbf{S}) = \{\sum_i x_i\mathbf{s}_i : x_i \in \mathbb{R} \text{ for } 1 \le i \le n\}$. For any set $\mathbf{S}$ of linearly independent vectors, we define the centered half-open parallelepiped $\mathcal{P}(\mathbf{S}) = \{\sum_i x_i\mathbf{s}_i : -1/2 \le x_i < 1/2 \text{ for } 1 \le i \le n\}$. For a vector $\mathbf{v} \in \mathbb{R}^n$ and a real r, let $\mathcal{B}(\mathbf{v},r) = \{\mathbf{w} \in \mathbb{R}^n : \operatorname{dist}(\mathbf{v},\mathbf{w}) < r\}$ be the open ball of radius r centered at $\mathbf{v}$, and $\bar{\mathcal{B}}(\mathbf{v},r) = \{\mathbf{w} \in \mathbb{R}^n : \operatorname{dist}(\mathbf{v},\mathbf{w}) \le r\}$ its topological closure.

An m-dimensional *lattice* is the set of all integer combinations

$$\Big\{ \sum_{i=1}^{n} x_i\mathbf{b}_i : x_i \in \mathbb{Z} \text{ for } 1 \le i \le n \Big\}$$

of n linearly independent vectors $\mathbf{b}_1,\ldots,\mathbf{b}_n$ in $\mathbb{R}^m$ ($m \ge n$). The set of vectors $\mathbf{b}_1,\ldots,\mathbf{b}_n$ is called a *basis* for the lattice, and the integer n is called the *rank* of the lattice. If the rank n equals the dimension m, then the lattice is called *full rank* or *full dimensional*. 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.

For any lattice $\mathbf{B}$ and point $\mathbf{x} \in \operatorname{span}(\mathbf{B})$, there exists a unique vector $\mathbf{y} \in \mathcal{P}(\mathbf{B})$ such that $\mathbf{y}-\mathbf{x} \in \mathcal{L}(\mathbf{B})$. This vector is denoted $\mathbf{y} = \mathbf{x} \bmod \mathbf{B}$, and it can be computed in polynomial time given $\mathbf{B}$ and $\mathbf{x}$. The dual of a lattice Λ is the set

$$\Lambda^* = \{\mathbf{x} \in \operatorname{span}(\Lambda) \colon \forall \mathbf{y} \in \Lambda\ \langle \mathbf{x},\mathbf{y}\rangle \in \mathbb{Z}\}$$

of all vectors in the linear span of Λ that have integer scalar product ($\langle \mathbf{x},\mathbf{y}\rangle = \sum_i x_i y_i$) with all lattice vectors. The dual of a lattice is a lattice, and if $\Lambda = \mathcal{L}(\mathbf{B})$ is the lattice generated by the basis $\mathbf{B}$, then $\mathbf{B}^* = \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}$ is a basis for the dual lattice, where $\mathbf{B}^T$ is the transpose of $\mathbf{B}$. (Notice that if $\mathbf{B}$ is a basis, it

has full column rank and the square matrix $\mathbf{B}^T\mathbf{B}$ is invertible.) A sublattice of $\mathcal{L}(\mathbf{B})$ is a lattice $\mathcal{L}(\mathbf{S})$ such that $\mathcal{L}(\mathbf{S}) \subseteq \mathcal{L}(\mathbf{B})$. $\mathcal{L}(\mathbf{S})$ is a *full rank* sublattice of $\mathcal{L}(\mathbf{B})$ if it has the same rank as $\mathcal{L}(\mathbf{B})$. The determinant $\det(\mathcal{L}(\mathbf{B}))$ of a (rank n) lattice is the (n-dimensional) volume of the fundamental parallelepiped $\mathcal{P}(\mathbf{B})$. If $\mathcal{L}(\mathbf{B})$ is full dimensional, then $\det(\mathcal{L}(\mathbf{B}))$ equals the absolute value of the determinant of the $n \times n$ basis matrix, that is, $\det(\mathcal{L}(\mathbf{B})) = |\det(\mathbf{B})|$.

The *minimum distance* of a lattice $\mathcal{L}(\mathbf{B})$, denoted $\lambda_1(\mathbf{B})$, is the minimum distance between any two distinct lattice points, and equals the length of the shortest nonzero lattice vector:

$$\begin{aligned}\lambda_1(\mathbf{B}) &= \min\{\mathrm{dist}(\mathbf{x},\mathbf{y}) : \mathbf{x} \neq \mathbf{y} \in \mathcal{L}(\mathbf{B})\} \\ &= \min\{\|\mathbf{x}\| : \mathbf{x} \in \mathcal{L}(\mathbf{B}) \setminus \{\mathbf{0}\}\}.\end{aligned}$$

This definition can be generalized to define the ith successive minimum as the smallest λ_i such that $\bar{\mathcal{B}}(\mathbf{0},\lambda_i)$ contains i linearly independent lattice points:

$$\lambda_i(\mathbf{B}) = \min\{r : \dim(\mathrm{span}(\mathcal{L}(\mathbf{B}) \cap \bar{\mathcal{B}}(\mathbf{0},r))) \geq i\}.$$

Another important constant associated to a lattice is the *covering radius* $\rho(\mathbf{B})$, which is defined as the maximum distance $\mathrm{dist}(\mathbf{x},\mathcal{L}(\mathbf{B}))$ where $\mathbf{x}$ ranges over the linear span of $\mathbf{B}$:

$$\rho(\mathbf{B}) = \max_{\mathbf{x}\in\mathrm{span}(\mathbf{B})} \{\mathrm{dist}(\mathbf{x},\mathcal{L}(\mathbf{B}))\}.$$

A *deep hole* is a point $\mathbf{x} \in \mathrm{span}(\mathbf{B})$ at distance $\mathrm{dist}(\mathbf{x},\mathcal{L}(\mathbf{B})) = \rho(\mathbf{B})$ from the lattice.

We consider the following lattice problems. For simplicity we define all problems only in their promise version, which is the formulation typically used in computational complexity. (The reader is referred to Micciancio & Goldwasser 2002 for a discussion of different formulations of these lattice problems.) Promise problems are a natural generalization of decision problems: they are defined by a set of YES inputs and a (disjoint) set of NO inputs. However, unlike decision problems, the two sets are not necessarily exhaustive. A machine solves the promise problem if it accepts all the YES inputs and rejects all the NO inputs; no assumption is made on the behavior of the machine on inputs that are in neither of the sets. Let us also mention that the *complement* of a promise problem is obtained by swapping the YES and NO instances; we will denote this by adding the prefix "co" to the problem's name. The following definitions are parameterized by a positive (and typically monotone) real-valued function $\gamma\colon \mathbb{Z}^+ \to \mathbb{R}^+$ of the lattice rank.

DEFINITION 2.1 (Shortest Vector Problem). An input to GapSVP_γ is a pair $(\mathbf{B}, d)$ where $\mathbf{B}$ is a rank n lattice basis and d is a rational number. In YES inputs $\lambda_1(\mathbf{B}) \leq d$ and in NO inputs $\lambda_1(\mathbf{B}) > \gamma(n) \cdot d$.

DEFINITION 2.2 (Shortest Independent Vectors Problem). An input to $\mathsf{GapSIVP}_\gamma$ is a pair $(\mathbf{B}, d)$ where $\mathbf{B}$ is a rank n lattice basis and d is a rational number. In YES inputs $\lambda_n(\mathbf{B}) \leq d$ and in NO inputs $\lambda_n(\mathbf{B}) > \gamma(n) \cdot d$.

DEFINITION 2.3 (Covering Radius Problem). An input to GapCRP_γ is a pair $(\mathbf{B}, d)$ where $\mathbf{B}$ is a rank n lattice basis and d is a rational number. In YES inputs $\rho(\mathbf{B}) \leq d$ and in NO inputs $\rho(\mathbf{B}) > \gamma(n) \cdot d$.

DEFINITION 2.4 (Closest Vector Problem). An input to GapCVP_γ is a triple $(\mathbf{B}, \mathbf{t}, d)$ where $\mathbf{B}$ is a rank n lattice basis, $\mathbf{t}$ is a target vector, and d is a rational number. In YES inputs $\operatorname{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B})) \leq d$ and in NO inputs $\operatorname{dist}(\mathbf{t}, \mathcal{L}(\mathbf{B})) > \gamma(n) \cdot d$.

An error correcting code $\mathcal{A}$ of block length n over a q-ary alphabet Σ is a collection of strings (vectors) from Σ^n, called codewords. For all codes considered in this paper, the alphabet size q is always a prime power and the alphabet $\Sigma = \mathbb{F}_q$ is the finite field with q elements. A code $\mathcal{A} \subseteq \mathbb{F}_q^n$ is *linear* if it is closed under addition and multiplication by a scalar, i.e., $\mathcal{A}$ is a linear subspace of $\mathbb{F}_q^n$ over the base field $\mathbb{F}_q$. For such a code, the information content (i.e., the number $k = \log_q |\mathcal{A}|$ of information symbols that can be encoded with a codeword) is just its dimension as a vector space and the code can be compactly represented by an $n \times k$ generator matrix $\mathbf{A} \in \mathbb{F}_q^{n \times k}$ of rank k such that $\mathcal{A} = \{\mathbf{A}\mathbf{x} \mid \mathbf{x} \in \mathbb{F}_q^k\}$. An alternative representation of linear codes is by their parity check matrix, i.e., an $(n-k) \times n$ matrix $\mathbf{H}$ such that $\mathcal{A} = \{\mathbf{x} \colon \mathbf{H}\mathbf{x} = \mathbf{0}\}$. An important property of a code is its *minimum distance*. For any vector $\mathbf{x} \in \Sigma^n$, the Hamming weight of $\mathbf{x}$ is the number $\|\mathbf{x}\|$ of nonzero coordinates of $\mathbf{x}$. (We use the same notation $\|\cdot\|$, $\operatorname{dist}(\cdot,\cdot)$, ρ etc. for both lattices and codes, as the meaning will always be clear from the context.) The weight function $\|\cdot\|$ is a norm, and the induced metric $\operatorname{dist}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ is called the *Hamming distance*. The (minimum) distance $\operatorname{dist}(\mathcal{A})$ of the code $\mathcal{A}$ is the minimum Hamming distance $\operatorname{dist}(\mathbf{x}, \mathbf{y})$ taken over all pairs of distinct codewords $\mathbf{x}, \mathbf{y} \in \mathcal{A}$. For linear codes it is easy to see that the minimum distance $\operatorname{dist}(\mathcal{A})$ equals the minimum weight $\|\mathbf{x}\|$ of a nonzero codeword $\mathbf{x} \in \mathcal{A} \setminus \{\mathbf{0}\}$. If $\mathcal{A}$ is a linear code with block length n and rank k, then it is customary to say that $\mathcal{A}$ is an $[n, k]$ *linear code*.

The *covering radius* $\rho(\mathcal{A})$ of a code $\mathcal{A} \subseteq \mathbb{F}_q^n$ is defined as the smallest integer r such that for any $\mathbf{x} \in \mathbb{F}_q^n$ there exists a codeword $\mathbf{y} \in \mathcal{A}$ within distance $\mathrm{dist}(\mathbf{x}, \mathbf{y}) \leq r$ from $\mathbf{x}$. A *deep hole* is a point $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathrm{dist}(\mathbf{x}, \mathcal{A}) = \rho(\mathcal{A})$. When a code is represented by a parity check matrix $\mathbf{H}$, we simply write $\rho(\mathbf{H})$ to denote the covering radius of the code defined by $\mathbf{H}$. The covering radius problem for linear codes is defined analogously to the one for lattices.

Definition 2.5 (Covering radius for linear codes). An input to $\mathsf{GapCRPcodes}_\gamma$ is a pair $(\mathbf{H}, d)$ where $\mathbf{H}$ is the parity check matrix of a code (with block length n), and d is an integer. In YES inputs $\rho(\mathbf{H}) \leq d$ and in NO inputs $\rho(\mathbf{H}) > \gamma(n) \cdot d$.

3. Previous work

As we already said in the introduction, not much was previously known about the computational complexity of the covering radius problem for codes and lattices. For the case of linear codes, the covering radius was proved Π_2-hard to solve exactly by McLoughlin (1984). The analogous question for lattices was explicitly posed in Micciancio (2004) where the covering radius problem is used to get tighter connections between the worst- and average-case complexity of various lattice problems.

Another relevant result is the transference theorem of Banaszczyk (1993), which shows that for any lattice $\mathcal{L}(\mathbf{B})$ and its dual $\mathcal{L}(\mathbf{B})^*$,

$$1 \leq 2\rho(\mathcal{L}(\mathbf{B})) \cdot \lambda_1(\mathcal{L}(\mathbf{B})^*) \leq n. \tag{3.1}$$

This connection between the covering radius of a lattice and the length of the shortest nonzero vector in the dual allows one to approximately reduce (loosing a factor n, and swapping YES and NO instances) the covering radius problem to the shortest vector problem. For example, by combining (3.1) with known approximation algorithms for the shortest vector problem, we obtain approximation algorithms for the covering radius problem (see Theorem 4.6). By combining (3.1) with other known results, one can deduce that approximating the covering radius of a lattice within $\gamma(n)$ is in coNP for $\gamma(n) = O(n)$, in AM for $\gamma(n) = O(n^{1.5}/\sqrt{\log n})$, and in NP for $\gamma(n) = O(n^{1.5})$. Notice that these bounds are considerably worse than the bounds obtained in this paper. Our improvements are obtained by directly solving the covering radius problem without using (3.1).

Most of our reductions are not gap-preserving, and introduce a small error, ranging from $1 + \epsilon$ up to n. One of our reductions (namely, the one from SIVP

to CVP) preserves the approximation factor exactly. This is similar to the gap-preserving reduction of Goldreich *et al.* (1999) which shows that GapSVP_γ reduces to GapCVP_γ for any approximation factor $\gamma(n)$ (possibly a function of the lattice dimension). However, while the reduction in Goldreich *et al.* (1999) runs in *deterministic* polynomial time, our reduction is *nondeterministic*. Still our nondeterministic reduction suffices to establish coAM and coNP results for GapSIVP.

One last paper related to the covering radius problem for lattices is Kannan (1992), where Kannan considers the problem of computing the covering radius of a lattice with respect to a given input norm (different from the Euclidean one) defined by a convex polytope specified as a system of linear inequalities. In Kannan (1992) it is shown that for any fixed dimension, this problem can be solved in polynomial time.

4. Lattice problems

In this section we present our results for the covering radius problem on point lattices. In Subsections 4.1 and 4.2 we describe proof systems to prove that the covering radius is small or large, and in Subsection 4.3 we give algorithms to approximately compute the covering radius. In Subsection 4.4 we give our nondeterministic reduction from SIVP to CVP, and the corresponding new proof systems for the shortest independent vectors problem.

4.1. Proving that the covering radius is small. In this subsection we give two proof systems to show that the covering radius of a lattice is small. The first is interactive and achieves approximation factor $\gamma(n) = 2$. The second is an NP proof system and achieves factor $\gamma(n) = \sqrt{n}$. The interactive proof system is based on the observation that random points in space are far from the lattice with high probability.

LEMMA 4.1. *For any lattice* $\mathcal{L}(\mathbf{B})$,

$$\Pr_{\mathbf{x}}\big(\operatorname{dist}(\mathbf{x}, \mathcal{L}(\mathbf{B})) \geq \rho(\mathbf{B})/2\big) \geq 1/2$$

where $\mathbf{x}$ *is chosen uniformly at random from* $\mathcal{P}(\mathbf{B})$.

PROOF. Let $\mathbf{v}$ be a deep hole of $\mathcal{L}(\mathbf{B})$ and consider the set $\mathbf{v} + \mathcal{L}(\mathbf{B})$. By the triangle inequality, for any point $\mathbf{x}$, we have $\operatorname{dist}(\mathbf{x}, \mathcal{L}(\mathbf{B})) + \operatorname{dist}(\mathbf{x}, \mathbf{v} + \mathcal{L}(\mathbf{B})) \geq \operatorname{dist}(\mathcal{L}(\mathbf{B}), \mathbf{v} + \mathcal{L}(\mathbf{B})) = \rho(\mathbf{B})$, where the equality follows since $\mathbf{v}$ is a deep hole.

Since $\operatorname{dist}(\mathbf{x}, \mathbf{v} + \mathcal{L}(\mathbf{B})) = \operatorname{dist}((\mathbf{x} - \mathbf{v}) \bmod \mathbf{B}, \mathcal{L}(\mathbf{B}))$, the inequality

$$\operatorname{dist}(\mathbf{x}, \mathcal{L}(\mathbf{B})) + \operatorname{dist}((\mathbf{x} - \mathbf{v}) \bmod \mathbf{B}, \mathcal{L}(\mathbf{B})) \geq \rho(\mathbf{B})$$

holds with probability 1 over the random choice of the vector $\mathbf{x} \in \mathcal{P}(\mathbf{B})$. But we also have

$$\begin{aligned}
\Pr_{\mathbf{x}}\big(\operatorname{dist}(\mathbf{x}, \mathcal{L}(\mathbf{B})) + \operatorname{dist}((\mathbf{x} - \mathbf{v}) \bmod \mathbf{B}, \mathcal{L}(\mathbf{B})) \geq \rho(\mathbf{B})\big) & \\
\leq \Pr_{\mathbf{x}}\big(\operatorname{dist}(\mathbf{x}, \mathcal{L}(\mathbf{B})) \geq \rho(\mathbf{B})/2\big) & \\
+ \Pr_{\mathbf{x}}\big(\operatorname{dist}((\mathbf{x} - \mathbf{v}) \bmod \mathbf{B}, \mathcal{L}(\mathbf{B})) \geq \rho(\mathbf{B})/2\big) & \\
= 2 \Pr_{\mathbf{x}}\big(\operatorname{dist}(\mathbf{x}, \mathcal{L}(\mathbf{B})) \geq \rho(\mathbf{B})/2\big), &
\end{aligned}$$

where $\mathbf{x}$ is chosen uniformly at random from $\mathcal{P}(\mathbf{B})$ and the last equality follows since the distribution of $(\mathbf{x} - \mathbf{v}) \bmod \mathbf{B}$ is also uniform on $\mathcal{P}(\mathbf{B})$. □

Using the lemma, we easily get an interactive proof system for the covering radius problem.

THEOREM 4.2. GapCRP_2 *is in* AM.

PROOF. We present a constant round public coin interactive proof system for GapCRP_2. The idea is to probabilistically reduce the covering radius problem to the closest vector problem by letting the verifier choose the target vector. Details follow.

On input a GapCRP_2 instance $(\mathbf{B}, r)$,

- The verifier picks a uniformly random point $\mathbf{x} \in \mathcal{P}(\mathbf{B})$ and sends it to the prover.
- The prover finds a lattice point $\mathbf{u} \in \mathcal{L}(\mathbf{B})$ and sends it to the verifier.
- The verifier checks that $\mathbf{u} \in \mathcal{L}(\mathbf{B})$ and accepts if and only if $\operatorname{dist}(\mathbf{x}, \mathbf{u}) \leq r$.

We claim that the above protocol has perfect completeness and soundness error 1/2. Completeness is easy, because by definition of covering radius, for any point $\mathbf{x} \in \mathcal{P}(\mathbf{B})$, there exists a lattice point $\mathbf{u} \in \mathcal{L}(\mathbf{B})$ within distance $\rho(\mathbf{B}) \leq r$ from $\mathbf{x}$. Now assume $\rho(\mathbf{B}) > 2r$. According to Lemma 4.1, the point $\mathbf{x}$ chosen by the verifier satisfies $\operatorname{dist}(\mathbf{x}, \mathcal{L}(\mathbf{B})) \geq \rho(\mathbf{B})/2 > r$ with probability at least 1/2. If $\operatorname{dist}(\mathbf{x}, \mathcal{L}(\mathbf{B})) > r$, then the verifier rejects no matter which $\mathbf{u}$ the prover sends. So, the soundness error is at most 1/2. □

The result placing $\mathsf{GapCRP}_{\sqrt{n}}$ in NP is based on the following relation between the covering radius $\rho(\mathbf{B})$ and $\lambda_n(\mathbf{B})$ for any n-dimensional lattice $\mathbf{B}$.

LEMMA 4.3. *For any rank n lattice* $\mathbf{B}$,

$$\lambda_n(\mathcal{L}(\mathbf{B})) \leq 2\rho(\mathcal{L}(\mathbf{B})) \leq \sqrt{n}\lambda_n(\mathcal{L}(\mathbf{B})).$$

The proof of the lemma can be found in (Micciancio & Goldwasser 2002, Theorem 7.9, p. 138). For completeness we repeat the proof below.

PROOF. The upper bound $2\rho \leq \sqrt{n}\lambda_n$ is easy. By definition of λ_n, there exists a set $\mathbf{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_n\} \subset \mathcal{L}(\mathbf{B})$ of n linearly independent lattice vectors such that $\|\mathbf{s}_i\| \leq \lambda_n(\mathbf{B})$ for all $i = 1, \ldots, n$. Let $\mathbf{t} \in \text{span}(\mathbf{B})$ be an arbitrary target point and $\mathbf{s}_1^*, \ldots, \mathbf{s}_n^*$ be the Gram–Schmidt orthogonalization of $\mathbf{S}$. Babai's nearest plane algorithm (Babai 1986) on input $\mathbf{S}$ and $\mathbf{t}$ produces a lattice vector $\mathbf{u} \in \mathcal{L}(\mathbf{S}) \subseteq \mathcal{L}(\mathbf{B})$ such that

$$\text{dist}(\mathbf{t}, \mathbf{u}) \leq \sqrt{\sum_i (\|\mathbf{s}_i^*\|/2)^2} \leq \frac{1}{2}\sqrt{\sum_i \|\mathbf{s}_i\|^2} \leq \frac{1}{2}\sqrt{n}\max_i \|\mathbf{s}_i\| = \sqrt{n}\lambda_n(\mathbf{B})/2.$$

This proves that $\rho(\mathbf{B}) \leq \sqrt{n}\lambda_n(\mathbf{B})/2$.

Now consider the lower bound $\lambda_n \leq 2\rho$. Assume for contradiction $\lambda_n > 2\rho$ and let ϵ be a positive real number such that $\epsilon < \lambda_n - 2\rho$. We iteratively build a set of linearly independent lattice vectors $\mathbf{s}_1, \ldots, \mathbf{s}_n$ as follows. For any $i = 1, \ldots, n$, let $\mathbf{t}_i$ be any vector of length $\rho + \epsilon$ orthogonal to $\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}$, and let $\mathbf{s}_i$ be a lattice point within distance ρ from $\mathbf{t}_i$. Then $\mathbf{s}_i$ is linearly independent of $\mathbf{s}_1, \ldots, \mathbf{s}_{i-1}$, because the distance from $\mathbf{s}_i$ to $\text{span}(\mathbf{s}_1, \ldots, \mathbf{s}_{i-1})$ is at least $\|\mathbf{t}_i\| - \|\mathbf{s}_i - \mathbf{t}_i\| \geq \epsilon$. Moreover, by the triangle inequality, $\|\mathbf{s}_i\| \leq \|\mathbf{t}_i\| + \|\mathbf{s}_i - \mathbf{t}_i\| \leq 2\rho + \epsilon < \lambda_n$. By induction on i, we obtain a set $\mathbf{s}_1, \ldots, \mathbf{s}_n$ of linearly independent lattice vectors of length $\|\mathbf{s}_i\| < \lambda_n$, contradicting the definition of λ_n. □

Notice that the bounds proved in Lemma 4.3 are constructive in the following sense. Given n linearly independent lattice vectors of length at most λ_n and a target vector $\mathbf{t} \in \text{span}(\mathbf{B})$, one can efficiently find a lattice point within distance $(\sqrt{n}/2)\lambda_n \leq \sqrt{n}\rho$ from the target. Similarly, given access to an oracle that on input a lattice $\mathbf{B}$ and a target vector $\mathbf{t} \in \text{span}(\mathbf{B})$ finds a lattice point within distance ρ from the target, one can efficiently compute n linearly independent lattice vectors of length at most $2\rho \leq \sqrt{n}\lambda_n$. We now use the lemma to prove the following:

THEOREM 4.4. $\mathsf{GapCRP}_{\sqrt{n}}$ *is in* NP.

PROOF. The proof is by reduction to the exact version of SIVP, which is known to be in NP. The reduction is simple: given a $\mathsf{GapCRP}_{\sqrt{n}}$ instance $(\mathbf{B}, r)$, the output is the $\mathsf{GapSIVP}_1$ instance $(\mathbf{B}, 2r)$. Notice that if $(\mathbf{B}, r)$ is a YES instance of $\mathsf{GapCRP}_{\sqrt{n}}$, then $\rho(\mathbf{B}) \leq r$, and by Lemma 4.3, $\lambda_n(\mathbf{B}) \leq 2\rho(\mathbf{B}) \leq 2r$, i.e., $(\mathbf{B}, 2r)$ is a YES instance of $\mathsf{GapSIVP}_1$. Conversely, if $(\mathbf{B}, r)$ is a NO instance of $\mathsf{GapCRP}_{\sqrt{n}}$, then $\rho(\mathbf{B}) > \sqrt{n}r$, and by Lemma 4.3, $\lambda_n(\mathbf{B}) \geq 2\rho(\mathbf{B})/\sqrt{n} > 2r$, i.e., $(\mathbf{B}, 2r)$ is a NO instance of $\mathsf{GapSIVP}_1$.

The NP proof system for $\mathsf{GapCRP}_{\sqrt{n}}$ combines this simple reduction with the obvious NP proof for $\mathsf{GapSIVP}_1$: on input a $\mathsf{GapCRP}_{\sqrt{n}}$ instance $(\mathbf{B}, r)$, the prover sends n linearly independent lattice vectors $\mathbf{s}_1, \ldots, \mathbf{s}_n$ of length $\|\mathbf{s}\| \leq \lambda_n(\mathbf{B})$ to the verifier, and the verifier checks that indeed $\mathbf{s}_1, \ldots, \mathbf{s}_n$ are linearly independent lattice vectors in $\mathcal{L}(\mathbf{B})$ and $\|\mathbf{s}_i\| \leq 2r$ for all $i = 1, \ldots, n$. □

4.2. Proving that the covering radius is large. In this subsection we give proof systems to show that the covering radius is large. The proof systems are based on analogous results for the closest vector problem.

THEOREM 4.5. $\mathsf{GapCRP}_{O(\sqrt{n})}$ *is in* coNP *and* $\mathsf{GapCRP}_{O(\sqrt{n/\log n})}$ *is in* coAM.

PROOF. The idea is to reduce (in nondeterministic polynomial time) the covering radius problem to the closest vector problem, by letting the prover choose the target point.

For the coNP proof system, on input a GapCRP instance $(\mathbf{B}, r)$, we first (nondeterministically) guess a deep hole, i.e., a point $\mathbf{v} \in \mathrm{span}(\mathbf{B})$ at distance $\mathrm{dist}(\mathbf{v}, \mathcal{L}(\mathbf{B})) = \rho(\mathbf{B})$ from the lattice. Then we use the recently discovered proof system of Aharonov & Regev (2004) to show that $(\mathbf{B}, \mathbf{u}, r)$ is a NO instance of $\mathsf{GapCVP}_{O(\sqrt{n})}$. Specifically, Aharonov & Regev (2004) show that $\mathsf{GapCVP}_{O(\sqrt{n})} \in \mathsf{coNP}$ by giving a polynomial time verifier V such that for any $(\mathbf{B}, \mathbf{v}, t)$,

- if $\mathrm{dist}(\mathbf{v}, \mathcal{L}(\mathbf{B})) > \Omega(\sqrt{n}t)$ then there exists a witness $w = w(\mathbf{B}, \mathbf{v}, t)$ such that $V(\mathbf{B}, \mathbf{v}, t, w)$ accepts;
- if $\mathrm{dist}(\mathbf{v}, \mathcal{L}(\mathbf{B})) \leq t$, then for any witness w, $V(\mathbf{B}, \mathbf{v}, t, w)$ rejects.

The verifier for $\mathsf{GapCRP}_{O(\sqrt{n})}$ works in the obvious way. On input a $\mathsf{GapCRP}_{O(\sqrt{n})}$ instance $(\mathbf{B}, r)$ and a witness $(\mathbf{v}, w)$, run $V(\mathbf{B}, \mathbf{v}, r, w)$ and accept if and only if V accepts.

The proof of $\mathsf{GapCRP}_{O(\sqrt{n/\log n})} \in \mathsf{coAM}$ is similar. Here, instead of the coNP proof system of Aharonov & Regev (2004) we use the result of Goldreich & Goldwasser (2000) showing that $\mathsf{GapCVP}_{O(\sqrt{n/\log n})} \in \mathsf{coAM}$, i.e., there exists

a constant round public coin interactive proof system for $\mathsf{coGapCVP}_{O(\sqrt{n/\log n})}$. We construct a constant round interactive proof system for $\mathsf{coGapCRP}_{O(\sqrt{n/\log n})}$ where the prover sends, as its first message, a deep hole $\mathbf{v}$ to the verifier, and then interactively proves that $\mathbf{v}$ is far from the lattice using the constant round public coin protocol from Goldreich & Goldwasser (2000). Therefore $\mathsf{GapCRP}_{O(\sqrt{n/\log n})}$ is in coAM. □

4.3. Algorithms for the covering radius problem. In this subsection we give algorithms to approximate the covering radius of a lattice. The first algorithm immediately follows from well known relations between the covering radius problem and other lattice problems.

THEOREM 4.6. *The problem* $\mathsf{GapCRP}_{\gamma(n)}$ *can be solved in probabilistic polynomial time when* $\gamma(n) = 2^{O(n \log\log n/\log n)}$ *and in deterministic polynomial time when* $\gamma(n) = 2^{O(n(\log\log n)^2/\log n)}$.

PROOF. On input a lattice $\mathbf{B}$, compute the dual lattice $\mathbf{B}^*$, and use the probabilistic algorithm of Ajtai *et al.* (2001) to find a $2^{O(n \log\log n/\log n)}$ approximation to the length of the shortest vector in the dual lattice. It follows from the transference theorem in (3.1) that the inverse of this length is an $n \cdot 2^{O(n \log\log n/\log n)} = 2^{O(n \log\log n/\log n)}$ approximation of the covering radius of the input lattice. The second result follows by a similar argument from the $2^{O(n(\log\log n)^2/\log n)}$ approximation algorithm of Schnorr (1987). □

The proof of the above theorem is based on relations between the covering radius and other lattice quantities that always introduce errors that are polynomial in the rank of the lattice. In particular, even if we solve SVP exactly, the above reduction would only give an $O(n)$ approximation of the covering radius. Below, we give an exponential time algorithm that achieves approximation factors arbitrarily close to 1. The algorithm is based on the following observation.

LEMMA 4.7. *For any lattice basis* $\mathbf{B}$ *and integer* $M > 0$, *there exists a point*

$$\mathbf{v} = a_1 \cdot \mathbf{b}_1 + \cdots + a_n \cdot \mathbf{b}_n$$

such that $a_i \in \{0, 1/M, 2/M, \ldots, (M-1)/M\}$ *for all* i, *and* $\operatorname{dist}(\mathbf{v}, \mathcal{L}(\mathbf{B})) \geq (1 - 1/M)\rho(\mathbf{B})$.

PROOF. Let $\mathbf{v}'$ be a deep hole of the lattice. We would like to round it to a point $\mathbf{v}$ of the above form. First, consider the vector $M \cdot \mathbf{v}'$. Since $\mathcal{L}(\mathbf{B})$

has covering radius $\rho(\mathbf{B})$, there must exist a lattice point $\mathbf{Bu} \in \mathcal{L}(\mathbf{B})$ whose distance from $M \cdot \mathbf{v}'$ is at most $\rho(\mathbf{B})$. This implies that the distance between $\mathbf{v}'$ and $\mathbf{Bu}/M$ is at most $\rho(\mathbf{B})/M$. Moreover, by the triangle inequality,

$$\operatorname{dist}(\mathbf{B}u/M, \mathcal{L}(\mathbf{B})) \geq \operatorname{dist}(\mathbf{v}', \mathcal{L}(\mathbf{B})) - \operatorname{dist}(\mathbf{B}u/M, \mathbf{v}') \geq (1 - 1/M)\rho(\mathbf{B}).$$

Let $\mathbf{a} = (\mathbf{u} \bmod M)/M$ be the vector obtained by reducing all entries of $\mathbf{u}$ modulo M, and dividing the result by M. Clearly, $a_i \in \{0, 1/M, 2/M, \ldots, (M-1)/M\}$ for all i. Moreover, the difference $\mathbf{a} - \mathbf{u}/M$ is an integer vector, so $\mathbf{B}(\mathbf{a} - \mathbf{u}/M)$ belongs to $\mathcal{L}(\mathbf{B})$, and $\operatorname{dist}(\mathbf{Ba}, \mathcal{L}(\mathbf{B})) = \operatorname{dist}(\mathbf{Bu}/M, \mathcal{L}(\mathbf{B})) \geq (1 - 1/M)\rho(\mathbf{B})$. □

THEOREM 4.8. *For any $\epsilon > 0$, there exists an algorithm that runs in time $2^{O(n)}$ and approximates* CRP *to within factor $1 + \epsilon$.*

PROOF. In Ajtai *et al.* (2002), it is shown that for any $\epsilon > 0$ there exists a $1+\epsilon$ approximation algorithm for CVP that runs in time $2^{O(n)}$. Let $M = \lceil 1/\epsilon \rceil$ and let $\mathbf{v}_1, \ldots, \mathbf{v}_n$ generate a lattice $\mathcal{L}(\mathbf{B})$ of covering radius $\rho(\mathbf{B})$. Our CRP algorithm tries all M^n points of the form

$$a_1 \cdot \mathbf{v}_1 + \cdots + a_n \cdot \mathbf{v}_n$$

such that $a_i \in \{0, 1/M, 2/M, \ldots, (M-1)/M\}$. For each such point $\mathbf{u}$, it calls the CVP algorithm in order to obtain a value $d_{\mathbf{u}}$ such that

$$\operatorname{dist}(\mathbf{u}, \mathcal{L}(\mathbf{B})) \leq d_{\mathbf{u}} \leq (1 + \epsilon) \operatorname{dist}(\mathbf{u}, \mathcal{L}(\mathbf{B})).$$

The output of the algorithm is $\rho' = \max d_{\mathbf{u}}/(1-\epsilon)$ over all such $\mathbf{u}$. By Lemma 4.7, for one of the points $\mathbf{u}$, $\operatorname{dist}(\mathbf{u}, \mathcal{L}(\mathbf{B})) \geq (1-\epsilon)\rho(\mathbf{B})$ and hence $\rho' \geq \rho(\mathbf{B})$. Moreover, $\operatorname{dist}(\mathbf{u}, \mathcal{L}(\mathbf{B})) \leq \rho(\mathbf{B})$ for all $\mathbf{u}$ and therefore $\rho' \leq (1+\epsilon)/(1-\epsilon)\rho(\mathbf{B})$. This completes the proof since ϵ is arbitrary. □

4.4. Improved proof systems for the Shortest Independent Vectors Problem. It is known that $\mathsf{GapSIVP}_{O(n)}$ is in coNP; this follows from the transference theorem of Banaszczyk (1993) relating λ_n and λ_1. Another known result is that $\mathsf{GapSIVP}_{O(n/\sqrt{\log n})}$ is in coAM (Blömer & Seifert 1999). We note that the latter result can be easily derived from the results of this paper by combining the relation between λ_n and ρ established in Lemma 4.3 with the coAM proof system for $\mathsf{GapCRP}_{O(\sqrt{n/\log n})}$ of Theorem 4.5. The resulting factor is $\gamma(n) = O(n/\sqrt{\log n})$ because the reduction from SIVP to CRP implicit in Lemma 4.3 introduces a $\sqrt{n}$ error.

In this subsection we improve both of these results by a factor of $\sqrt{n}$. Namely, we show that $\mathsf{GapSIVP}_{O(\sqrt{n/\log n})}$ is in coAM and that $\mathsf{GapSIVP}_{O(\sqrt{n})}$ is in coNP. Equivalently, we show that $\mathsf{coGapSIVP}_{O(\sqrt{n/\log n})}$ is in AM and that $\mathsf{coGapSIVP}_{O(\sqrt{n})}$ is in NP. Both results follow from the following theorem:

THEOREM 4.9. *For any $\gamma(n)$, there exists a nondeterministic polynomial time algorithm that on input a $\mathsf{coGapSIVP}_{\gamma(n)}$ instance $(\mathbf{B}, d)$ outputs polynomially many $\mathsf{coGapCVP}_{\gamma(n)}$ instances $(\mathbf{S}_i, \mathbf{t}_i, d)$, $i = 1, \ldots, M$, such that*

- *If $(\mathbf{B}, d)$ is a YES instance, then for some nondeterministic choice of the algorithm, every $(\mathbf{S}_i, \mathbf{t}_i, d)$ is a YES instance.*
- *If $(\mathbf{B}, d)$ is a NO instance, then for every nondeterministic choice of the algorithm, some $(\mathbf{S}_i, \mathbf{t}_i, d)$ is a NO instance.*

PROOF. Let $(\mathbf{B}, d)$ be the input to the algorithm, where $\mathbf{B}$ is a basis of an n-dimensional lattice. For concreteness, we assume that the entries of $\mathbf{B}$ as well as d are all integers; this is no loss of generality since we can always scale rational values without increasing the size of the input by more than a polynomial. The algorithm nondeterministically chooses a basis $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_n]$ of size polynomial in the size of the input. Let M denote some large enough polynomial in the input size to be determined later. First of all, the algorithm checks that $\mathbf{S}$ is a basis of $\mathcal{L}(\mathbf{B})$, i.e., $\mathcal{L}(\mathbf{S}) = \mathcal{L}(\mathbf{B})$. (If not, it outputs M copies of some fixed NO instance of $\mathsf{coGapCVP}_{\gamma(n)}$.) If $\mathbf{S}$ is indeed a basis, then it outputs M $\mathsf{coGapCVP}_{\gamma(n)}$ instances $(\mathbf{S}_i, 2^{i-1}\mathbf{s}_n, d)$, $i = 1, \ldots, M$, where $\mathbf{S}_i := [\mathbf{s}_1, \ldots, \mathbf{s}_{n-1}, 2^i \mathbf{s}_n]$.

We need to prove that if $(\mathbf{B}, d)$ is a YES instance, then there exists a polynomial-sized basis $\mathbf{S}$ such that all $(\mathbf{S}_i, 2^{i-1}\mathbf{s}_n, d)$ are YES instances, while if $(\mathbf{B}, d)$ is a NO instance, then for any basis $\mathbf{S}$ there exists an $i \in \{1, \ldots, M\}$ such that $(\mathbf{S}_i, 2^{i-1}\mathbf{s}_n, d)$ is a NO instance.

Assume $(\mathbf{B}, d)$ is a YES instance of $\mathsf{coGapSIVP}_{\gamma(n)}$, i.e., $\lambda_n(\mathbf{B}) > \gamma(n) \cdot d$, and define a basis $\mathbf{S}$ as follows. Let $\mathbf{s}'_1, \ldots, \mathbf{s}'_k$ be a maximal set of linearly independent lattice vectors of length at most $\gamma(n) \cdot d$, i.e., a set of vectors such that any lattice vector of length at most $\gamma(n) \cdot d$ is in $\mathrm{span}(\mathbf{s}'_1, \ldots, \mathbf{s}'_k)$. Since $\lambda_n(\mathbf{B}) > \gamma(n) \cdot d$, it must be that $k < n$. Let $\mathbf{s}_1, \ldots, \mathbf{s}_k$ be a basis for the sublattice $\mathcal{L}(\mathbf{B}) \cap \mathrm{span}(\mathbf{s}'_1, \ldots, \mathbf{s}'_k)$, and complete it to a basis $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_n]$ for the entire lattice $\mathcal{L}(\mathbf{B})$. We remark that since $\mathbf{s}'_1, \ldots, \mathbf{s}'_k$ are integer vectors of length at most $\gamma(n) \cdot d$, their bit-size is at most $O(n \log(\gamma(n) \cdot d))$ and is therefore polynomial in the input size. Moreover, since $\mathbf{S}$ can be computed in

polynomial time from $\mathbf{B}$ and $\mathbf{s}'_1, \ldots, \mathbf{s}'_k$ using standard techniques, we deduce that the bit-size of the basis $\mathbf{S}$ is also polynomial in the input size.

We want to prove that $(\mathcal{L}(\mathbf{S}_i), 2^{i-1}\mathbf{s}_n, d)$ is a YES instance of $\mathsf{coGapCVP}_{\gamma(n)}$ for all i, i.e., for any integer $i \in \{1, \ldots, M\}$ and lattice vector $\mathbf{y} \in \mathcal{L}(\mathbf{S}_i)$, we have $\|2^{i-1}\mathbf{s}_n - \mathbf{y}\| > \gamma(n) \cdot d$. Fix i and lattice vector $\mathbf{y} \in \mathcal{L}(\mathbf{S}_i)$. Notice that $2^{i-1}\mathbf{s}_n - \mathbf{y} \in \mathcal{L}(\mathbf{S})$, and $2^{i-1}\mathbf{s}_n - \mathbf{y}$ does not belong to $\mathrm{span}(\mathbf{s}_1, \ldots, \mathbf{s}_k) = \mathrm{span}(\mathbf{s}'_1, \ldots, \mathbf{s}'_k)$ because it uses $2^{i-1}\mathbf{s}_n$ an odd number of times and $k < n$. It follows, by the maximality of $\{\mathbf{s}'_1, \ldots, \mathbf{s}'_k\}$, that $\|2^{i-1}\mathbf{s}_n - \mathbf{y}\| > \gamma(n) \cdot d$.

Let us now assume $(\mathbf{B}, d)$ is a NO instance of $\mathsf{coGapSIVP}_{\gamma(n)}$, i.e., $\lambda_n(\mathbf{B}) \leq d$, and let $\mathbf{S}$ be an arbitrary basis for $\mathcal{L}(\mathbf{B})$. We prove that there exists an $i \in \{1, \ldots, M\}$ such that $(\mathcal{L}(\mathbf{S}_i), 2^{i-1}\mathbf{s}_n, d)$ is a NO instance of $\mathsf{coGapCVP}_{\gamma(n)}$, i.e., $\mathrm{dist}(2^{i-1}\mathbf{s}_n, \mathcal{L}(\mathbf{S}_i)) \leq d$. Since $\lambda_n(\mathbf{B}) \leq d$, there exist n linearly independent lattice vectors of length at most d. Since $\mathrm{span}(\mathbf{s}_1, \ldots, \mathbf{s}_{n-1})$ has dimension $n-1$, they cannot all belong to $\mathrm{span}(\mathbf{s}_1, \ldots, \mathbf{s}_{n-1})$. So, let $\mathbf{v} \in \mathcal{L}(\mathbf{B})$ be such that $\|\mathbf{v}\| \leq d$ and $\mathbf{v} \notin \mathrm{span}(\mathbf{s}_1, \ldots, \mathbf{s}_{n-1})$. Write $\mathbf{v} = \sum_{i=1}^{n} a_i \mathbf{s}_i$ and assume without loss of generality that $a_n \geq 0$. (If not, replace $\mathbf{v}$ with $-\mathbf{v}$.) Since $\mathbf{v} \notin \mathrm{span}(\mathbf{s}_1, \ldots, \mathbf{s}_{n-1})$ we see that $a_n \neq 0$. Let i_0 be the maximum $i \geq 1$ such that a_n is divisible by 2^{i-1}, i.e., $a_n/2^{i_0-1}$ is an odd integer. Later, we will see that $i_0 \leq M$. For this i_0 we have $\mathbf{v} + 2^{i_0-1}\mathbf{s}_n \in \mathcal{L}(\mathbf{S}_{i_0})$. Hence, the distance from $2^{i_0-1}\mathbf{s}_n$ to $\mathcal{L}(\mathbf{S}_{i_0})$ is at most $\|\mathbf{v}\| \leq d$.

Finally, in order to choose M, consider the quantity $d/\|\mathbf{s}^*_n\|$ where $\mathbf{s}^*_n$ is the projection of $\mathbf{s}_n$ on the subspace orthogonal to $\mathrm{span}(\mathbf{s}_1, \ldots, \mathbf{s}_{n-1})$. Since this quantity can be computed in polynomial time from the basis $\mathbf{s}_1, \ldots, \mathbf{s}_n$ and d, it follows that its bit-size is polynomial in the size of the input. Moreover, by projecting $\mathbf{v}$ on the same subspace we deduce that $\|\mathbf{v}\| \geq a_n\|\mathbf{s}^*_n\|$. Since $\|\mathbf{v}\| \leq d$ we find that a_n is at most $d/\|\mathbf{s}^*_n\|$ and hence its bit-size is also polynomial. In particular, we can choose M to be a polynomial in the input size such that $a_n \leq 2^{M-1}$. With this choice, it is easy to see that $i_0 \leq M$. □

Combining the nondeterministic algorithm from the above theorem with the coAM and coNP results of Aharonov & Regev (2004) and Goldreich & Goldwasser (2000) we immediately get the following corollary.

COROLLARY 4.10. $\mathsf{GapSIVP}_{\gamma(n)}$ *is in* coAM *for* $\gamma(n) = O(\sqrt{n/\log n})$ *and in* coNP *for* $\gamma(n) = O(\sqrt{n})$.

PROOF. According to Aharonov & Regev (2004), $\mathsf{coGapCVP}_{\gamma(n)}$ is in NP for $\gamma(n)(\sqrt{n})$. So let A be a nondeterministic Turing machine that solves $\mathsf{coGapCVP}_{O(\sqrt{n})}$. Let B a nondeterministic Turing machine as in the above theorem for $\gamma(n) = O(\sqrt{n})$. We construct a nondeterministic Turing machine

C that on input a $\mathsf{coGapSIVP}_{O(\sqrt{n})}$ instance, runs B to generate polynomially many $\mathsf{coGapCVP}_{O(\sqrt{n})}$ instances, applies A to all of them, and outputs the logical AND of the answers output by A.

We claim that C solves $\mathsf{coGapSIVP}_{O(\sqrt{n})}$ and hence $\mathsf{coGapSIVP}_{O(\sqrt{n})} \in \mathsf{NP}$. Indeed, if the coGapSIVP input instance is a YES instance, then for some nondeterministic choice of B, all coGapCVP instances output by B are YES instances. Hence, for each of these instances, and for some nondeterministic choice of algorithm A, the instance is accepted by A. Taking the logical AND of all the answers, C also accepts. Now assume that C is given as input a NO coGapSIVP instance. Then no matter what nondeterministic choices are made by B, one of the coGapCVP instances produced by B must be a NO instance. For this instance, A necessarily rejects no matter what nondeterministic choices it makes. Since C outputs the logical AND of all answers produced by A, machine C rejects too.

A similar argument shows that $\mathsf{coGapSIVP}_{O(\sqrt{n/\log n})}$ is in AM. According to Goldreich & Goldwasser (2000), there exists an AM protocol that solves $\mathsf{coGapCVP}_{\gamma(n)}$ for $\gamma(n)(\sqrt{n/\log n})$. Let A be such a protocol and B as above. Then we can define C as an AM protocol that first applies B to obtain a list of coGapCVP instances, and then, for each of these instances runs the AM protocol given by A. The rest of the argument is essentially identical. □

5. Coding problems

In this section we prove our results concerning the covering radius on linear codes. In Subsection 5.1 we present a simple polynomial time approximation algorithm for linear codes over arbitrary finite fields achieving approximation factor $1 + \log_q n$ where n is the block length of the code and q the alphabet size. Then, in Subsection 5.2, we adapt similar results for lattices to prove that approximating the covering radius of a code within factor 2 is in AM. Finally, in Subsections 5.3 and 5.4 we present our Π_2-hardness and NP-hardness results for approximating the covering radius of a code within constant approximation factors, as well as hardness of approximation within $O(\log \log n)$ factors under the stronger assumption that $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{O(\log \log \log n)})$. All our results hold for linear codes over an arbitrary (fixed) alphabet. In particular, they hold for the special case of binary codes.

5.1. Approximation algorithm for CRP on codes. We begin with the following standard lemma that characterizes the covering radius of linear codes.

LEMMA 5.1. *Let* $\mathbf{C}$ *be an* $[n,k]$ *linear code over a field* $\mathbb{F}$ *defined by an* $(n-k)\times n$ *parity check matrix* $\mathbf{H}$ *so that* $\mathbf{C}=\{\mathbf{c}\in\mathbb{F}^n : \mathbf{Hc}=0\}$. *Then the covering radius of* $\mathbf{C}$ *is the smallest* r *such that for every* $\mathbf{y}\in\mathbb{F}^{n-k}$ *there exists a vector* $\mathbf{z}\in\mathbb{F}^n$ *of weight at most* $\|\mathbf{z}\|\le r$ *for which* $\mathbf{Hz}=\mathbf{y}$.

The lemma is used to obtain a very simple $\log n$ approximation of the covering radius.

THEOREM 5.2. *For any prime power* q, $\mathsf{GapCRPcodes}_{\gamma(n)}$ *over* q*-ary codes can be solved in polynomial time for* $\gamma(n)=\log_q(n(q-1))\le 1+\log_q n$.

PROOF. The parity check matrix of an $[n,k]$ code has column rank exactly $n-k$, and therefore using Lemma 5.1, it follows that the covering radius r of an $[n,k]$ linear code is at most $n-k$. On the other hand, a simple volume bound shows that $q^k\sum_{i=0}^{r}\binom{n}{i}(q-1)^i\ge q^n$. This implies $n^r(q-1)^r\ge q^{n-k}$, or, solving for r,

$$r\ge\frac{n-k}{\log_q(n(q-1))}.$$

Therefore the bound $(n-k)/\log_q(n(q-1))$ is a factor $\log_q(n(q-1))$ approximation to the covering radius. □

We remark that the simple algorithm in the proof of Theorem 5.2 is polynomial in $\log q$. So, the result holds even for codes over alphabets of variable size $q(n)$.

5.2. Proving that the covering radius is small. In this section we adapt the AM proof system for lattices presented in Subsection 4.1 to linear codes. The proof system for codes is virtually identical to the one for lattices, with only syntactical modifications.

LEMMA 5.3. *For every linear code* $\mathcal{A}\subseteq\mathbb{F}_q^n$,

$$\Pr_{\mathbf{x}}\big(\mathrm{dist}(\mathbf{x},\mathcal{A})\ge\rho(\mathcal{A})/2\big)\ge 1/2$$

where $\mathbf{x}$ *is chosen uniformly at random from* $\mathbb{F}_q^n$.

PROOF. Similar to the one for Lemma 4.1. □

The lemma is used to obtain an interactive protocol for $\mathsf{GapCRPcodes}_2$ as in Theorem 4.2.

THEOREM 5.4. *For any prime power q,* $\mathsf{GapCRPcodes}_2$ *on q-ary codes is in* AM.

As before, the proof system is polynomial in $\log q$, so the result holds for codes over variable alphabet size $q(n)$ as well.

5.3. Π_2-hardness of approximating within some constant factor. CRP on linear codes is one of the most natural complete problems for Π_2—this hardness result is due to McLoughlin (1984). For general codes, when the input is a list of codewords, the covering radius problem was shown to be coNP-complete by Frances & Litman (1997); see also (Cohen *et al.* 1997, Chap. 20).[4] These results are for the exact version of the problem. In this subsection and the next, we prove hardness results for *approximating* CRP on linear codes. The first result stated below shows that there is some constant factor up to which approximating the covering radius is Π_2-complete, thereby giving a very natural hardness of approximation result that falls in the second level of the polynomial time hierarchy.

THEOREM 5.5. *There is a constant $c > 1$ such that for any prime power q,* $\mathsf{GapCRPcodes}_c$ *on q-ary codes is Π_2-complete.*

PROOF. $\mathsf{GapCRPcodes}_c$ can be solved in Π_2 even in its exact version, i.e., $c = 1$. We need to prove that for some constant $c > 1$ the problem is also Π_2-hard. For simplicity, we first prove the theorem for binary codes (i.e., for $q = 2$), and then describe how to adapt the proof to codes over arbitrary alphabets.

The proof is by reduction from the problem $\mathsf{Gap}\forall\exists\text{-}3\text{-}\mathsf{SAT}\text{-}\mathsf{B}_g$ defined as follows. An instance of $\mathsf{Gap}\forall\exists\text{-}3\text{-}\mathsf{SAT}\text{-}\mathsf{B}_g$ is defined by a pair (C, t) where t is an integer and C is a set of clauses[5] such that each clause contains at most 3 variables, each variable occurs in at most B clauses, and the variables are partitioned into two sets: a set A of *universal* variables, and a set E of *existential* variables. For notational convenience, in the rest of this proof we represent

[4] We remark that computational problems on linear codes are not a special case of the same problems for arbitrary codes, because in the linear case the code is usually represented as a generator or parity check matrix. This allows a compact representation of an exponentially large set of codewords, and the efficiency of the algorithms is usually expressed as a function of the size of this implicit representation. Efficient algorithms that take arbitrary codes as input (e.g., represented as an explicit list of codewords) can become exponential time when applied to linear codes.

[5] A *clause* is a disjunction $l_1 \vee \cdots \vee l_m$ of m literals, where each literal l_i is either a variable symbol x or its negation $\neg x$.

truth assignments as subsets of $A \cup E$, where a variable is included in the subset if and only if it is assigned the value true. An instance (C, t) is a YES instance if for every Boolean assignment to the universal variables $W \subseteq A$ there exists an assignment to the existential variables $X \subseteq E$ such that $X \cup W$ satisfies at least t of the clauses in C. An instance (C, t) is a NO instance if there exists an assignment to the universal variables $W \subseteq A$ such that for every assignment to the existential variables $X \subseteq E$, $X \cup W$ satisfies at most t/g of the clauses in C. In Ko & Lin (1995) it is proved that there exists a positive integer B and a constant $g > 1$ such that Gap$\forall\exists$-3-SAT-B$_g$ is Π_2-hard. Let (C, t) be an input instance of Gap$\forall\exists$-3-SAT-B$_g$ with $|A \cup E| = n$ variables and $|C| = m$ clauses. The following simple observations show that we can always assume that $t > n/3$.

- We assume that each universal variable occurs both in positive and negated form in C. This assumption is not restrictive because if a universal variable occurs only in positive or negated form, then we can simply remove all its occurrences from C, and obtain a Gap$\forall\exists$-3-SAT-B$_g$ instance equivalent to the original one. If during this process a clause becomes empty, then the clause itself is removed from the formula.

- We assume that each existential variable occurs both in positive and negated form in C. This assumption is also not restrictive because if an existential variable occurs only in positive or negated form, then we can always satisfy all the clauses containing this variable. So, an equivalent Gap$\forall\exists$-3-SAT-B$_g$ instance can be obtained by removing all clauses containing this variable from C, and decreasing t by the number of removed clauses.

- Finally, assuming that each variable occurs in both positive and negated form, we can also assume without loss of generality that $t > n/3$, where n is the number of variables. This is because any assignment $V \subseteq A \cup E$ satisfies at least one occurrence for each variable (for a total of at least n variable occurrences). Since each clause contains at most 3 variables occurrences, the assignment also satisfies at least $n/3$ clauses. So, if $t \leq n/3$, then (C, t) is certainly a YES instance, and the reduction can trivially output a YES instance of the covering radius problem.

Now, assume we are given a Gap$\forall\exists$-3-SAT-B$_g$ instance (C, t) with m clauses and n variables such that $t > n/3$. Notice that $n \leq 3m$ because each clause contains at most 3 variables, and $m \leq Bn$ because each variable occurs at most B times. In particular, $t > n/3 \geq m/3B$. We map C to a linear code such

that if (C,t) is a YES instance then the covering radius of the code is at most $n+\lceil(m-t)/B\rceil$, while if (C,t) is a NO instance then the covering radius is at least $n+\lceil(m-t/g)/B\rceil$. This proves that the covering radius of a linear code is Π_2-hard to approximate within a factor

$$\frac{n+\lceil(m-t/g)/B\rceil}{n+\lceil(m-t)/B\rceil} > \frac{n+(m-t/g)/B}{n+(m-t)/B+1} = 1+\frac{(1-1/g)t-B}{Bn+m-t+B}$$
$$> 1+\frac{(1-1/g)n-3B}{(6B-1)n+3B} > 1+\frac{1-1/g}{6B} > 1,$$

where in the second inequality we used $t > n/3$ and $m \le Bn$, and the third inequality (which holds for all sufficiently large n) follows by taking the limit for n approaching infinity. The linear code is described by a parity check matrix $\mathbf{H}$ with $m+n+|A|$ rows. The rows are divided into three blocks:

- a block of m rows, indexed by the clauses in C,
- a block of n rows, indexed by the variables in $A \cup E$, and
- a block of $|A|$ additional rows, indexed by the universal variables A.

Notice that there are two distinct rows for each universal variable, one in the second and one in the third block. For each subset of clauses $L \subseteq C$, subset of variables $V \subseteq A \cup E$ and subset of universal variables $W \subseteq A$, let $[L,V,W]$ denote the $(m+n+|A|)$-dimensional vector that equals 1 at the positions corresponding to the elements of the sets L, V and W, and 0 everywhere else. Also, for any variable v, let $C[+v]$ (resp., $C[-v]$) be the set of clauses that contain a positive (resp., negative) occurrence of the variable v.

The columns are divided into various groups, and are defined as follows:

- For every $x \in A$, and subset of clauses $L \subseteq C[+x]$, there are two columns $[L,\{x\},\{x\}]$ and $[L,\emptyset,\{x\}]$.
- For every $x \in A$, and subset of clauses $L \subseteq C[-x]$, there is a column $[L,\{x\},\emptyset]$.
- For every $y \in E$, and subset of clauses $L \subseteq C[+y]$, there is a column $[L,\{y\},\emptyset]$.
- For every $y \in E$, and nonempty[6] subset of clauses $L \subseteq C[-y]$, there is a column $[L,\{y\},\emptyset]$.

[6] In this and the next case, we restrict L to be nonempty just to avoid unnecessary (e.g., repeated or identically zero) columns.

- For every nonempty subset of clauses $L \subseteq C$ of size at most $|L| \leq B$, there is a column of the form $[L, \emptyset, \emptyset]$.

This completes the description of the reduction. It remains to prove that the reduction is correct.

Assume first that (C, t) is a YES instance, i.e., for every assignment $W \subseteq A$ there is an assignment $X \subseteq E$ such that $W \cup X$ satisfies at least t clauses in C. We want to bound the covering radius of the code defined by the parity check matrix $\mathbf{H}$. By Lemma 5.1, we need to prove that for any vector $\mathbf{y} = [L, V, W]$ there exists a subset of columns of $\mathbf{H}$ of size at most $n + \lceil (m-t)/B \rceil$ that adds up to $\mathbf{y}$. Fix some $\mathbf{y} = [L, V, W]$ and let $X \subseteq E$ be an assignment such that $W \cup X$ satisfies at least t clauses in C. Order the variables $A \cup E$ in some arbitrary way, and for each variable $v \in A \cup E$, select the column $[L_v, \{v\} \cap V, \{v\} \cap W]$, where

$$L_v = \begin{cases} C[+v] \cap L \setminus \bigcup_{w<v} L_w & \text{if } v \in W \cup X, \\ C[-v] \cap L \setminus \bigcup_{w<v} L_w & \text{otherwise.} \end{cases}$$

Notice that all these columns belong to the matrix $\mathbf{H}$, and their sum is the vector $[C' \cap L, V, W]$, where C' is the set of clauses satisfied by the assignment $W \cup X$. Since $W \cup X$ leaves at most $m - t$ clauses unsatisfied, the size of $L \setminus C'$ is at most $m - t$. So, we can obtain the vector $[L, V, W]$ by adding at most $\lceil (m-t)/B \rceil$ more columns of the form $[L', \emptyset, \emptyset]$, where $|L'| \leq B$. This proves that the covering radius of the code $\mathbf{H}$ is at most $n + \lceil (m-t)/B \rceil$.

Now assume (C, t) is a NO instance, i.e., there exists an assignment $W \subseteq A$ such that for every assignment $X \subseteq E$, the number of clauses satisfied by $W \cup X$ is at most t/g. We want to prove that the covering radius of the code $\mathbf{H}$ is at least $n + \lceil (m - t/g)/B \rceil$. Let $\mathbf{z} = [C, A \cup E, W]$. By Lemma 5.1 it is enough to prove that any subset of columns of $\mathbf{H}$ that add up to $\mathbf{z}$ must have size at least $n + \lceil (m - t/g)/B \rceil$. Let K be a smallest subset of columns of $\mathbf{H}$ that add up to $\mathbf{z}$. Assume that for every $x \in A$, K contains at most one column of the form $[L, V, \{x\}]$. This involves no loss of generality because for any two columns $[L_1, V_1, \{x\}]$ and $[L_2, V_2, \{x\}]$, the sum $[L_1, V_1, \{x\}] + [L_2, V_2, \{x\}]$ is a vector of the form $[L, V, \emptyset]$ where $L \subseteq L_1 \cup L_2 \subseteq C[+x]$ has size at most B, and $V \subseteq V_1 \cup V_2 \subseteq \{x\}$. So, we can replace $[L_1, V_1, \{x\}]$ and $[L_2, V_2, \{x\}]$ with $[L, \emptyset, \emptyset]$ and $[\emptyset, V, \emptyset]$ without changing the sum.

We next associate each variable $v \in A \cup E$ with a set $L_v \subset C$ and a column $\mathbf{c}_v$ (from K) of the form $[L_v, V, W]$ for some $V \cup W \subseteq \{v\}$. First, note that in order to add up to $[C, A \cup E, W]$, for every $v \in W$, K must contain at least one column of the form $\mathbf{c}_v = [L_v, V, \{v\}]$, and this column must necessarily satisfy $L_v \subseteq C[+v]$. Also, for every $v \in A \setminus W$, K cannot contain any column of the

form $[L, V, \{v\}]$ because of our assumption that K contains at most one column of the form $[L, V, \{v\}]$. Hence, for every $v \in A \setminus W$, K must contain a column of the form $\mathbf{c}_v = [L_v, \{v\}, \emptyset]$ and $L_v \subseteq C[-v]$. Similarly, for every $v \in E$, K must contain a column of the form $\mathbf{c}_v = [L_v, \{v\}, \emptyset]$, where $L_v \subseteq C[+v]$ or $L_v \subseteq C[-v]$.

To summarize, our mapping is such that $L_v \subseteq C[+v]$ for all $v \in W$, $L_v \subseteq C[-v]$ for all $v \in A \setminus W$, and L_v is a subset of either $C[-v]$ or $C[+v]$ for all $v \in E$. Let X be the set of all $v \in E$ such that $L_v \subseteq C[+v]$. The assignment $W \cup X$ satisfies all the clauses in $\bigcup_v L_v$. Therefore, the set $\bigcup_v L_v$ has size at most t/g. In particular, the sum of the n columns $\mathbf{c}_v$ (for $v \in A \cup E$) equals 0 in at least $m - t/g$ of the coordinates in the first block. Since any column of $\mathbf{H}$ contains at most B nonzero entries in the first block, K must include at least $\lceil (m - t/g)/B \rceil$ additional columns in order to add up to $[C, A \cup E, W]$. This proves that the covering radius is at least $n + \lceil (m - t/g)/B \rceil$, and concludes the proof of the theorem for the case of binary codes.

The reduction for q-ary codes is similar. Just replace each column $\mathbf{c}$ in the parity check matrix $\mathbf{H}$ with the set of all q-ary columns with exactly the same zero coordinates as $\mathbf{c}$. Since each vector in $\mathbf{H}$ has at most $B + 2$ nonzero coordinates, this increases the size of the matrix by at most a factor $(q-1)^{B+2}$, which is polynomial in n for any polynomially bounded $q(n)$. The rest of the proof is an easy adaptation of the one for the binary alphabet case. We remark that the inapproximability factor $c > 1 + (1 - 1/g)/6B$ is independent of the alphabet size q. □

We observe that although the proof of Theorem 5.5 holds also for codes with variable alphabet size $q(n)$, this time $q(n)$ must be polynomially bounded in order to ensure that the reduction is polynomial time.

REMARK. In light of Theorem 5.4, $\mathsf{GapCRPcodes}_c$ is unlikely to be Π_2-hard for $c \geq 2$. Indeed, if for some $c \geq 2$, $\mathsf{GapCRPcodes}_c$ were Π_2-hard under Karp reductions, then we would have $\mathsf{coNP} \subseteq \Pi_2 \subseteq \mathsf{AM}$. Boppana *et al.* (1987) show that $\mathsf{coNP} \subseteq \mathsf{AM}$ implies a collapse of the polynomial time hierarchy to the second level.

5.4. NP-hardness of approximating within arbitrary constant factors. An immediate consequence of Theorem 5.5 is that $\mathsf{GapCRPcodes}_c$ is also NP-hard for some constant $c > 1$. But what about arbitrarily large constants? A standard method to prove NP-hardness of approximation for arbitrarily large constants is to first prove NP-hardness for some constant factor, and then amplify the constant using some polynomial time computable transfor-

mation. For example, Dumer *et al.* (2003) give a polynomial time computable transformation that on input a code $\mathcal{C}$ with minimum distance $d(\mathcal{C})$, outputs a code $\mathcal{C}'$ with minimum distance $d(\mathcal{C}') = d(\mathcal{C})^2$, and use this transformation to prove that the minimum distance of a linear code is NP-hard (under randomized reductions) to approximate within any constant. The question here is: is there a polynomial time computable transformation that on input a linear code $\mathcal{C}$ with covering radius $\rho(\mathcal{C})$ outputs a new code with covering radius $\rho(\mathcal{C}') = \rho(\mathcal{C})^2$ (or, more generally, $\rho(\mathcal{C}') = f(\rho(\mathcal{C}))$ for some function f such that $\lim_{n\to\infty} f^n(1+\epsilon) = +\infty$ for all $\epsilon > 0$)? Unfortunately, such a transformation is unlikely to exist for the covering radius. Indeed, such a transformation (or even a weaker transformation with $\lim_n f^n(1+\epsilon) > 2$) would also imply the Π_2-hardness of approximating the covering radius within factors greater than 2; by the remark following Theorem 5.5, this would imply the collapse of the polynomial time hierarchy. So, in order to prove that the covering radius problem is NP-hard to approximate within factors higher than 2, a different approach is needed.

In this section, we present a reduction from set cover that shows that approximating the covering radius of a linear code within any constant factor is NP-hard, and approximating it within $\Omega(\log\log n)$ is also hard under the stronger assumption that $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{O(\log\log\log n)})$. Our starting point is the following hardness result for set cover. Recall that a set cover instance consists of a universe and a collection of subsets of the universe, with the goal being to cover the universe using the fewest possible subsets (see, e.g., Garey & Johnson 1979).

THEOREM 5.6. *There exist absolute constants $a, b, c > 0$ such that for every function $B : \mathbb{N} \to \mathbb{N}$ where $B(n) \le n^{\log\log n}$, there is a reduction that maps any instance ϕ of* 3SAT *on n variables into a set cover instance $\mathcal{I}$ consisting of at most N^a sets, each of size at most $B(n)$, over a universe of size $N = n^{b\log\log B(n)}$, and a parameter p, such that*

- *if ϕ is satisfiable, then $\mathcal{I}$ admits a set cover comprising of at most p sets;*
- *if ϕ is not satisfiable, then every set cover of $\mathcal{I}$ requires at least $cp\log B(n)$ sets.*

Furthermore, the reduction runs in $n^{O(\log\log B(n))}$ time.

The above result, in particular the dependence of the gap on the size of the sets in the instance, is not explicitly stated as such in the literature, but it is implicit in previous work. We refer the interested reader to Trevisan (2001) where a statement similar to the one above is explicitly shown for $B(n)$ that is

a constant independent of n, by a suitable choice of parameters in Feige (1998). The goal there was to show a $\ln B - o(\ln B)$ gap, whereas for our application an $O(\log B(n))$ gap suffices. Therefore, one can show the above theorem by using the 2-prover 1-round proof systems that follow from Raz's parallel repetition theorem (Raz 1998) (with $u = O(\log \log B(n))$ parallel repetitions) in the original set cover reduction of Lund & Yannakakis (1994); see the survey by Arora & Lund (1996) for a nice exposition of this reduction.

We now give an approximation preserving reduction from set cover to GapCRPcodes to prove the following:

THEOREM 5.7. *Assume that* NP $\not\subseteq$ DTIME$(n^{O(\log \log \log n)})$. *Then there exists a constant* $c_0 > 0$ *such that for any prime power* q, GapCRPcodes$_{c_0 \log \log n}$ *over* q*-ary codes with block length* n *is not solvable in polynomial time.*

THEOREM 5.8. *For any constant* $c > 1$ *and prime power* q, GapCRPcodes$_c$ *on* q*-ary codes is* NP*-hard.*

The proof of Theorem 5.8 is identical to the proof of Theorem 5.7 given below, with the only difference that we reduce from set cover instances of Theorem 5.6 where $B(n)$ is a large enough constant (independent of n). Since the reduction we describe below runs in time $n^{O(\log \log B(n))} q^{B(n)}$, the reduction for constant $B(n) = O(1)$ is polynomial time and proves the NP-hardness of GapCRPcodes$_c$. We remark that both proofs hold true even for codes over variable alphabet size $q(n)$. However, while in Theorem 5.8, $q(n)$ can be an arbitrary polynomially bounded function of the block length, in Theorem 5.7 it must be at most $O(\log \log n)$, because $B(n) = \log n$ and larger values of $q(n)$ would increase the running time of the reduction above $n^{O(\log \log \log n)}$. For larger (but still polynomially bounded) values of $q(n)$, Theorem 5.7 holds true, but under the stronger assumption that NP $\not\subseteq$ DTIME$(n^{O(\log n)})$.

PROOF OF THEOREM 5.7. For simplicity, we prove the theorem for binary codes, and then describe how to adapt the proof to arbitrary alphabets.

Let $\mathcal{I} = (U; S_1, S_2, \ldots, S_m)$ be an instance of set cover with universe $U = \{1, 2, \ldots, N\}$ and each $S_j \subset U$ of size at most $B(n)$, as produced by Theorem 5.6. In particular, $N = n^{O(\log \log B(n))}$ and $m = N^{O(1)} = n^{O(\log \log B(n))}$. Define a $0, 1$ matrix $\mathbf{H}$ with N rows, one for each element of U, and $\sum_j 2^{|S_j|}$ columns, one for each subset of S_j for every j, $1 \leq j \leq m$. We will index the rows of $\mathbf{H}$ by i, $1 \leq i \leq N$, and the columns by (j, T) where $1 \leq j \leq m$ and $T \subseteq S_j$. The entries of the matrix are defined as follows:

$$\mathbf{H}_{i,(j,T)} = \begin{cases} 1 & \text{if } i \in T, \\ 0 & \text{otherwise.} \end{cases}$$

Consider the binary linear code $\mathcal{C} = \{\mathbf{c} : \mathbf{Hc} = \mathbf{0}\}$ defined by the parity check matrix $\mathbf{H}$. By Lemma 5.1, the covering radius of $\mathcal{C}$ equals k if k is the minimum value for which $\mathbf{Hz} = \mathbf{y}$ has a solution $\mathbf{z}$ of weight at most k for every $\mathbf{y} \in \{0,1\}^N$. We prove that k is in fact exactly the size of the smallest set cover of instance $\mathcal{I}$. The reduction produces a code of block length $N^* \leq m \cdot 2^{B(n)} \leq n^{O(\log\log B(n))} 2^{B(n)}$ and can be carried out in time $n^{O(\log\log B(n))} 2^{B(n)}$. With the choice $B(n) = \log n$ in the hardness result of Theorem 5.6, we deduce that a polynomial time algorithm for approximating the covering radius within factor $O(\log\log n)$ implies that $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{O(\log\log\log n)})$. For this choice of $B(n)$, the block length N^* of C satisfies $N^* = n^{O(\log\log\log n)}$, and therefore $\log\log n = \Theta(\log\log N^*)$. Therefore, assuming that $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{O(\log\log\log n)})$, we conclude that there is no polynomial time $c_0 \log\log N^*$ approximation algorithm for the covering radius problem on linear codes of block length N^*, for some $c_0 > 0$.

It remains to prove that k in fact equals the size of the smallest set cover of $\mathcal{I}$. Let $\mathbf{z}$ be a vector of Hamming weight at most k such that $\mathbf{Hz} = 1^N$ and consider the set of columns for which $\mathbf{z}$ has a nonzero coordinate. These columns add up to 1^N, and since each column corresponds to a subset of some S_j, it follows that the corresponding sets S_j surely cover the universe U. Therefore, the minimum set cover has size at most k. Conversely, let $\mathcal{S} = \langle S_{j_1}, S_{j_2}, \ldots, S_{j_\ell} \rangle$ be a set cover; we will show that $k \leq \ell$. Let $\mathbf{y} \in \{0,1\}^N$ be arbitrary; we will show that there are at most ℓ columns of $\mathbf{H}$, one from each of the "clusters" $\{(j_s, T) : T \subseteq S_{j_s}\}$ for $1 \leq s \leq \ell$, which add up to $\mathbf{y}$. Associate each $i = 1, \ldots, N$ for which $y_i = 1$ with a set S_{j_s} such that $i \in S_{j_s}$. This is possible since $\mathcal{S}$ is a set cover. Then, for each $s = 1, \ldots, \ell$, define $T_s \subseteq S_{j_s}$ as the set of all i's associated to S_{j_s}. It is readily checked that the ℓ columns (j_s, T_s) of $\mathbf{H}$ add up to precisely $\mathbf{y}$. This concludes the proof for the case of binary codes.

The reduction can be easily adapted to q-ary codes by replacing each column $\mathbf{c}$ in the parity check matrix $\mathbf{H}$ with the set of all q-ary columns that have exactly the same zero coordinates as $\mathbf{c}$. This increases the block length N^* of the code (and the running time of the reduction) from $n^{O(\log\log B(n))} 2^{B(n)}$ to $n^{O(\log\log B(n))} q^{B(n)}$. The rest of the proof is a straightforward adaptation of the proof for binary codes. □

REMARK. In the above reduction, the block length of the code is at least $q^{B(n)}$ and the gap in the reduction is $O(\log B(n))$. Hence the best hardness factor as a function of the block length N we can hope to show for covering radius using the above approach is $O(\log\log N)$.

6. Conclusion

There are numerous open questions raised by our work; below we discuss some of them.

- We proved that the covering radius problem for linear codes is Π_2-hard to approximate within some constant factor, and NP-hard for arbitrary constant factors. Does the same hold true for lattices? It is not clear if and how our hardness proofs for linear codes can be adapted to lattices, and we leave proving the hardness of the covering radius problem (in its exact and approximation versions) as an open problem.

- In this paper, we focused on lattice problems in the Euclidean norm ℓ_2, but other norms can be interesting as well. All our results can be immediately adapted to arbitrary ℓ_p norms (for any $p \geq 1$) by using the fact that all ℓ_p norms are within a factor $\sqrt{n}$ from the ℓ_2 one. This introduces a $\sqrt{n}$ loss in the approximation factors. An interesting question is whether our results can be reproduced in any ℓ_p norm without loosing this $\sqrt{n}$ term. Of special interest is the ℓ_∞ norm, since problems in this norm generally appear to be harder than in other ℓ_p norms. (For example, the shortest vector problem was long known to be NP-hard in the ℓ_∞ norm (van Emde Boas 1981), and later proved to be hard to approximate within almost polynomial factors (Dinur 2000), while proving the NP-hardness in the ℓ_2 norm remained an open problem till the work of Ajtai (1998), Micciancio (2001) and Khot (2004), and to date it is known to be NP-hard only for constant approximation factors, and under randomized reductions or unproven number-theoretic conjectures.) Can one show that CRP on lattices in the ℓ_∞ norm is NP-hard or even Π_2-hard in its exact or approximation version?

- Theorem 4.2 shows that the problem of approximating the covering radius of a lattice within a factor 2 is in AM. Can the factor 2 be improved? One possible way to improve this factor is to improve the bound $\rho(\mathbf{B})/2$ in Lemma 4.1. Specifically, is there a constant $c < 2$ such that for any lattice $\mathbf{B}$ the distance of a random point from $\mathbf{B}$ is at least $\rho(\mathbf{B})/c$ with some nonnegligible probability? By considering the lattice $\mathbb{Z}^n$, it can be seen that c has to be at least $\sqrt{3}$. Indeed, for any $c < \sqrt{3}$, the probability that the distance of a random point from the lattice $\mathbb{Z}^n$ is at least $\rho(\mathbb{Z}^n)/c$ is exponentially small in n. The smallest value possible for c is therefore between $\sqrt{3}$ and 2.

We remark that the proof of Lemma 4.1 is valid for any norm, not necessarily the Euclidean one. It turns out that for certain norms, Lemma 4.1 is essentially optimal. Lyubashevsky (2004) has shown that for the ℓ_1 and ℓ_∞ norms, there are lattices such that for any $c < 2$ the probability that a random point is at ℓ_1 (resp. ℓ_∞) distance at least $\rho(\mathbf{B})/c$ from the lattice is exponentially small in the dimension of the lattice. Moreover, for any $p \geq 1$, there is a constant $c_p > 1$ such that the bound in the lemma in the ℓ_p norm cannot be improved below $\rho(\mathbf{B})/c_p$.

- We believe that $O(\log n)$ is the right answer for approximating CRP on linear codes. Can one show that $\mathsf{GapCRPcodes}_{\Omega(\log n)}$ is NP-hard (or quasi-NP-hard)? Our reduction in Subsection 5.4 also raises questions concerning the complexity of the set cover problem itself. In particular, can one show the NP-hardness (as opposed to, say, hardness under the assumption $\mathsf{NP} \not\subseteq \mathsf{DTIME}(n^{O(\log\log n)})$) of approximating set cover with sets of size at most $B(n)$ within an $O(\log B(n))$ factor for the entire range $\Omega(1) \leq B(n) \leq n^{\Omega(1)}$? Currently this seems to be explicitly known only for $B(n)$ that is a constant independent of n.

Acknowledgements

We would like to thank Ravi Kumar for several motivating discussions. VG is supported in part by NSF Career Award CCF-0343672. DM is supported in part by NSF Career Award CCR-0093029 and an Alfred P. Sloan Fellowship. OR is supported by an Alon Fellowship and the Army Research Office grant DAAD19-03-1-0082. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

D. Aharonov & O. Regev (2004). Lattice problems in NP intersect coNP. In *Proc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS)*.

M. Ajtai (1998). The shortest vector problem in L_2 is NP-hard for randomized reductions (extended abstract). In *Proc. 30th Annual ACM Symposium on Theory of Computing—STOC '98* (Dallas, TX), ACM, 10–19.

M. Ajtai, R. Kumar & D. Sivakumar (2001). A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd Annual ACM Symposium on Theory of Computing—STOC 2001* (Heraklion), ACM, 601–610.

M. Ajtai, R. Kumar & D. Sivakumar (2002). Sampling short lattice vectors and the closest lattice vector problem. In *Proc. 17th IEEE Annual Conference on Computational Complexity—CCC '02* (Montreal), IEEE, 53–57.

S. Arora & C. Lund (1996). Hardness of approximation. In *Approximation Algorithms for NP-hard Problems*, D. S. Hochbaum (ed.), PWS, Boston.

L. Babai (1986). On Lovász' lattice reduction and the nearest lattice point problem. *Combinatorica* **6**, 1–13.

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

J. Blömer & J.-P. Seifert (1999). On the complexity of computing short linearly independent vectors and short bases in a lattice. In *Proc. 31st Annual ACM Symposium on Theory of Computing—STOC '99* (Atlanta, GA), ACM, 711–720.

R. Boppana, J. Håstad & S. Zachos (1987). Does co-NP have short interactive proofs? *Inform. Process. Lett.* **25**, 127–132.

G. Cohen, I. Honkala, S. Litsyn & A. Lobstein (1997). *Covering Codes*. North-Holland Math. Library 54, North-Holland, Amsterdam.

J. Conway & N. J. A. Sloane (1998). *Sphere Packings, Lattices and Groups*. 3rd ed., Springer.

I. Dinur (2000). Approximating SVP_∞ to within almost-polynomial factors is NP-hard. In *Proc. 4th Italian Conference on Algorithms and Complexity—CIAC 2000* (Rome), G. Bongiovanni, G. Gambosi & R. Petreschi (eds.), Lecture Notes in Comput. Sci. 1767, Springer, 263–276.

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

I. Dumer, D. Micciancio & M. Sudan (2003). Hardness of approximating the minimum distance of a linear code. *IEEE Trans. Inform. Theory* **49**, 22–37. Preliminary version in FOCS 1999.

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

U. Feige (1998). A threshold of $\ln n$ for approximating set cover. *J. ACM* **45**, 634–652.

M. Frances & A. Litman (1997). On covering problems of codes. *Theory Comput. Systems* **30**, 113–119.

M. Garey & D. Johnson (1979). *Computers and Intractability. A Guide to the Theory of NP-completeness*. W. H. Freeman, San Francisco.

O. Goldreich & S. Goldwasser (2000). On the limits of nonapproximability of lattice problems. *J. Comput. System Sci.* **60**, 540–563. Preliminary version in STOC'98.

O. Goldreich, D. Micciancio, S. Safra & J. Seifert (1999). Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Inform. Process. Lett.* **71**, 55–61.

R. Kannan (1992). Lattice translates of a polytope and the Frobenius problem. *Combinatorica* **12**, 161–177.

S. Khot (2004). Hardness of approximating the shortest vector problem in lattices. In *Proc. 45rd Annual Symposium on Foundations of Computer Science—FOCS 2004* (Rome), IEEE, 126–135.

K.-I Ko & C.-L. Lin (1995). On the longest circuit in an alterable digraph. *J. Global Optim.* **7**, 279–295.

C. Lund & M. Yannakakis (1994). On the hardness of approximating minimization problems. *J. ACM* **41**, 960–981.

V. Lyubashevsky (2004). A note on the tightness of a lemma of Guruswami, Micciancio and Regev. Private communication.

A. McLoughlin (1984). The complexity of computing the covering radius of a code. *IEEE Trans. Inform. Theory* **30**, 800–804.

D. Micciancio (2001). The shortest vector problem is NP-hard to approximate to within some constant. *SIAM J. Comput.* **30**, 2008–2035. Preliminary version in FOCS 1998.

D. Micciancio (2004). Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM J. Comput.* **34**, 118–169. Preliminary version in STOC 2002.

D. Micciancio & S. Goldwasser (2002). *Complexity of Lattice Problems: A Cryptographic Perspective*. Kluwer Internat. Ser Engrg. Comput. Sci. 671, Kluwer, Boston.

R. Raz (1998). A parallel repetition theorem. *SIAM J. Comput.* **27**, 763–803.

M. Schaefer & C. Umans (2002a). Completeness in the polynomial-time hierarchy: a compendium. *SIGACT News*.

M. Schaefer & C. Umans (2002b). Completeness in the polynomial-time hierarchy: part II. *SIGACT News*.

C. Schnorr (1987). A hierarchy of polynomial time lattice basis reduction algorithms. *Theoret. Comput. Sci.* **53**, 201–224.

L. J. Stockmeyer (1977). The polynomial-time hierarchy. *Theoret. Comput. Sci.* **3**, 1–22.

L. Trevisan (2001). Non-approximability results for optimization problems on bounded degree instances. In *Proc. 33rd Annual ACM Symposium on Theory of Computing—STOC 2001* (Heraklion), ACM, 453–461.

Manuscript received 2 October 2004

Venkatesan Guruswami
Department of Computer Science
University of Washington
Seattle, WA 98195, U.S.A.
venkat@cs.washington.edu

Daniele Micciancio
Department of Computer Science
and Engineering
University of California, San Diego
La Jolla, CA 92093, U.S.A.
daniele@cs.ucsd.edu

Oded Regev
Department of Computer Science
Tel-Aviv University
Tel-Aviv 69978, Israel