Electronic Colloquium on Computational Complexity, Report No. 115 (2013)

Locally Dense Codes

Daniele Micciancio[*]

August 26, 2013

Abstract

The Minimum Distance Problem (MDP), i.e., the computational task of evaluating (exactly or approximately) the minimum distance of a linear code, is a well known NP-hard problem in coding theory. A key element in essentially all known proofs that MDP is NP-hard is the construction of a combinatorial object that we may call a *locally dense code*. This is a linear code with large minimum distance d that admits a ball of smaller radius $r < d$ containing an exponential number of codewords, together with some auxiliary information used to map these codewords.

In this paper we provide a generic method to explicitly construct locally dense binary codes, starting from an arbitrary linear code with sufficiently large minimum distance. Instantiating our construction with well known linear codes (e.g., Reed-Solomon codes concatenated with Hadamard codes) yields a simple proof that MDP is NP-hard to approximate within any constant factor under deterministic polynomial time reductions, simplifying and explaining recent results of Cheng and Wan (STOC 2009 / IEEE Trans. Inf. Theory, 2012) and Khot and Austrin (ICALP 2011).

Our work is motivated by the construction of analogous combinatorial objects over integer lattices, which are used in NP-hardness proofs for the Shortest Vector Problem (SVP). We show that for the *max* norm, locally dense lattices can also be easily constructed. However, all currently known constructions of locally dense lattices in the standard *Euclidean* norm are probabilistic. Finding a deterministic construction of locally dense Euclidean lattices, analogous to the results presented in this paper, would prove the NP-hardness of approximating SVP under deterministic polynomial time reductions, a long standing open problem in the computational complexity of integer lattices.

1 Introduction

A *locally dense code* (or *lattice*) is a linear code (resp. lattice) $\mathcal{L}$ with large minimum distance d admitting a ball $\mathcal{B}(\mathbf{s}, r)$ of smaller[1] radius $r < d$ centered around $\mathbf{s}$ containing a large (exponential in the dimension) number of codewords. Applications of locally dense codes often require not only knowledge of the code, but also the center $\mathbf{s}$, as well as some auxiliary information $\mathbf{T}$ used to index the codewords in $\mathcal{B}(\mathbf{s}, r)$. (See Section 2 for a formal definition.) The construction of locally dense codes is a recurring problem in the computational study of linear codes and point lattices. Beside

[*] University of California, San Diego. Email: daniele@cs.ucsd.edu. This work was supported in part by NSF grant 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.

[1] Clearly, for the ball to contain more than a single codeword, it must be $r \geq d/2$. Here we are interested in balls with radius not much bigger than that, say $r < \gamma \cdot d$ for some constant $1/2 < \gamma < 1$.

ISSN 1433-8092

obvious connections to the problems of designing good spherical codes, and providing explicit upper bounds on the list decoding radius of the code $\mathcal{L}$, locally dense codes play a key role in the computational study of the two most famous problems in coding theory (and their analogous problem on integer lattices) as discussed below.

Lattice and Coding Problems The Nearest Codeword Problem, given a linear code $\mathcal{C}$ over a finite field and a target vector $\mathbf{t}$, asks to find the codeword in $\mathcal{C}$ (approximately) closest to $\mathbf{t}$. The Minimum Distance Problem (MDP), given a linear code $\mathcal{C}$ over a finite field, asks to compute (or approximate) the minimum distance between any two codewords in $\mathcal{C}$, and sometime also a pair of codewords achieving this distance. Equivalently, by linearity, the MDP can be defined as the problem of computing (the length of) a nonzero codeword closest to the origin, and can be regarded as the homogeneous version of NCP. MDP and NCP are directly related to the computational problems of estimating the error correction capability of a code, and recovering from errors, and are arguably the most important computational problems in coding theory. The analogous problems over integer lattices are the famous Shortest Vector Problem (SVP) and the Closest Vector Problem (CVP) arising in many algorithmic applications (e.g., factoring polynomials over the rationals [12] and integer programming with a constant number of variables [13, 9]), as well as the foundation of lattice based cryptography [16, 17].

The homogeneous (MDP, SVP) and inhomogeneous (NCP, CVP) problems are closely related. On the one hand the homogeneous problems can be reduced in polynomial time to the inhomogeneous ones [7], showing that NCP and CVP are at least as hard as MDP and SVP. In the other direction, all NP-hardness proofs for the homogeneous problems [19, 1, 3, 14, 6, 4, 10, 11, 8, 15] work essentially by reduction from (variants of) their inhomogeneous counterparts, which are well known NP-hard problems [18, 2, 5]. In most of these proofs (sometime implicitly) a locally dense code $(\mathcal{L}, \mathbf{s}, \mathbf{T})$ is used to embed an inhomogeneous (NCP or CVP) instance $(\mathcal{C}, \mathbf{t})$ in a higher dimensional space, so to increase the minimum distance of $\mathcal{C}$, while keeping the target $\mathbf{t}$ close to the code. Ultimately, this allows to find the codeword closest to the target $\mathbf{t}$ by looking for a short vector in the code generated by both $\mathcal{C}$ and $\mathbf{t}$, thereby reducing NCP to MDP (and variants of CVP to SVP for lattices.)

Interestingly, the use of locally dense codes is also what makes most of these proofs *probabilistic*: [1, 3, 14, 6, 10, 8, 15] all provide (sometime implicitly) deterministic constructions of a globally[2] dense code $\mathcal{L}$, and then probabilistically find a dense ball $\mathcal{B}(\mathbf{s}, r)$ (i.e., a ball containing many codewords) by picking its center $\mathbf{s}$ at random. As a result, [1, 3, 14, 6, 10, 8, 15] only show that MDP and SVP are NP-hard *under randomized reductions.*

Recently, for the case of codes over finite fields, Cheng and Wang [4] were able to derandomize the NP-hardness proof of [6], giving an efficient deterministic construction of locally dense codes, and thereby proving that MDP is NP-hard to approximate[3] under the standard notion of deterministic polynomial time reductions. The construction of [4] is by no means simple, making use of advanced mathematical tools like Weil's character sum bound on the affine line. The NP-hardness result from [4] was then extended to asymptotically good codes by Khot and Austrin [11], who also gave a mathematically more elementary proof, but did not provide an explicit construction of

[2] Here, by "globally dense", we mean a linear code with minum distance d such that the number of codewords in a ball $\mathcal{B}(\mathbf{s}, r)$ of radius $r < d$ is high in *expectation*, when the center $\mathbf{s}$ is chosen uniformly at random.

[3] For the exact version of this problem, hardness under deterministic reductions had already been proved by Vardy [19] using a comparatively simpler construction.

locally dense codes.

For the case of lattices, finding deterministic constructions analogous to locally dense codes and proving the NP-hardness of SVP (even in the exact version of the problem) under deterministic reductions is still open. Interestingly, in the most recent work on the NP-hardness of SVP [15], randomness is used exclusively for the construction of locally dense lattices, and moreover these lattices are built from binary linear codes. (As usual, the construction of the lattice $\mathcal{L}$ itself is deterministic, and randomness is used to find a good center $\mathbf{s}$.) This suggests that a better understanding of locally dense code constructions may ultimately lead to deterministic hardness results for SVP.

Our contribution The results presented in this paper started as an attempt to use the methods of [11] to give an explicit construction of locally dense codes, without resorting to the number theoretic techniques of [4]. Perhaps surprisingly, we obtain a construction that is at the same time simple and general: we show how to (deterministically and efficiently) transform any linear code with sufficiently large minimum distance (namely, minimum distance $d > n/\sqrt{6}$) into a locally dense code containing an exponential number of codewords within distance $r < d$ from a known, fixed center $\mathbf{s}$, independent of the original code. Moreover, as the distance of the original code approaches $d \approx n/2$, the radius-to-distance ratio of the resulting locally dense code approaches $(r/d) \approx 2/3$. Our construction comes with the required auxiliary information needed to derandomize the NP-hardness proof in [6], yielding a new, simple proof that MDP is NP-hard to approximate within any constant factor under deterministic polynomial time reductions. While the NP-hardness of MDP was already known [6, 4, 11], we believe that our construction improves our understanding of this important result, and the techniques may find other applications, e.g., in derandomizing similar constructions for integer lattices [14, 15].

Organization The rest of the paper is organized as follows. In Section 2 we provide some background, and a formal definition of locally dense codes as needed in the NP-hardness proof of [6]. In Section 3 we present and analyze our main result on transforming arbitrary binary linear codes with sufficiently large minimum distance into codes that are locally dense around a known center. Section 4 describes known results and open problems about the deterministic construction of locally dense lattices, and possible avenues to prove the NP-hardness of SVP (in the Euclidean norm) under deterministic polynomial time reductions.

2 Background

In this section we introduce the notation used in the rest of the paper, give some background on coding theory and formally define the computational problems we study.

Notation We write $\mathbf{x} = (x_1, \ldots, x_n)$ for the *column* vector with coordinates x_i, $i = 1, \ldots, n$, and $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_n]$ for the matrix with columns $\mathbf{a}_i$, $i = 1, \ldots, n$. The all one vector with n coordinates is denoted $\mathbf{u}_n = (1, \ldots, 1)$, or just $\mathbf{u}$ when n is clear from the context. The transpose of a matrix $\mathbf{A}$ is written $\mathbf{A}^T$.

For any prime power q, let $\mathbb{F}_q$ be the finite field with q elements. The Hamming norm (or weight) of a vector $\mathbf{x} \in \mathbb{F}_q^n$ is the number of nonzero coordinates $\|\mathbf{x}\|_H = |\{i \colon x_i \neq 0\}|$. The Hamming ball of radius r centered around $\mathbf{s} \in \mathbb{F}_q^n$ is $\mathcal{B}_H(\mathbf{s}, r) = \{\mathbf{x} \in \mathbb{F}_q^n \colon \|\mathbf{x} - \mathbf{s}\|_H \leq r\}$. For any two matrices

$\mathbf{A} \in \mathbb{F}_q^{n\times m}$ and $\mathbf{B} = \mathbb{F}_q^{h\times k}$, the tensor product of $\mathbf{A}$ and $\mathbf{B}$ is the matrix $\mathbf{A} \otimes \mathbf{B} \in \mathbb{F}_q^{nh\times mk}$ obtained by replacing each entry $a_{i,j}$ of $\mathbf{A}$ with the matrix $a_{i,j} \cdot \mathbf{B}$. In particular, for any vector $\mathbf{x} \in \mathbb{F}_q^n$, $\mathbf{u}_k \otimes \mathbf{x} \in \mathbb{F}_q^{kn}$ is the vector obtained by concatenating k copies of $\mathbf{x}$ and has norm $\|\mathbf{u}_k \otimes \mathbf{x}\|_H = k \cdot \|\mathbf{x}\|_H$.

Coding Theory A q-ary linear code of *length* n, *dimension* k and *minimum distance* (at least) d (written $\mathcal{C}[n,k,d]_q$) is a k-dimensional linear subspace $\mathcal{C} \subseteq \mathbb{F}_q^n$ such that $\|\mathbf{x}\|_H \geq d$ for all nonzero codewords $\mathbf{x} \in \mathcal{C} \setminus \{\mathbf{0}\}$. A linear code $\mathcal{C}[n,k,d]_q$ is usually represented by a generating matrix $\mathbf{C} \in \mathbb{F}_q^{n\times k}$ such that the columns of $\mathbf{C}$ are a *basis* for $\mathcal{C}$ as a vector space over $\mathbb{F}_q$, i.e., the columns of $\mathbf{C}$ are linearly independent and $\mathcal{C} = \mathbf{C}\mathbb{F}_q^k$. A basis $\mathbf{C}$ is in *systematic form* if the first k rows of $\mathbf{C}$ form the identity matrix $\mathbf{I}$, i.e., $\mathbf{C}^T = [\mathbf{I}, \mathbf{C}']$ for some $\mathbf{C}' \in \mathbb{F}_q^{k\times(n-k)}$. Any basis can be put in systematic form by possibly permuting the coordinates (so that the first k rows are linearly independent) and performing elementary column operations.

Let $\mathbf{A}$ and $\mathbf{B}$ be bases for linear codes over $\mathbb{F}_q$ with minimum distance $\mathrm{dist}_H(\mathbf{A})$ and $\mathrm{dist}_H(\mathbf{B})$ respectively. The *tensor product* of the two codes is the linear code generated by the matrix $\mathbf{A} \otimes \mathbf{B}$. It is easy to check that this code does not depend on the choice of the bases $\mathbf{A}, \mathbf{B}$, and it has minimum distance $\mathrm{dist}_H(\mathbf{A} \otimes \mathbf{B}) = \mathrm{dist}_H(\mathbf{A}) \cdot \mathrm{dist}_H(\mathbf{B})$.

In instantiating our results we will make use of two well known linear codes. The *Reed-Solomon* code $\mathcal{RS}[q,k,q-k+1]_q$ is the linear code over $\mathbb{F}_q$ consisting of all codewords $(p(x) \colon x \in \mathbb{F}_q)$ obtained evaluating any polynomial $p(x) \in \mathbb{F}_q[x]$ of degree less than k at every point $x \in \mathbb{F}_q$. The Reed-Solomon code has minimum distance $d = q - k + 1$ because any nonzero polynomial of degree less than k can have at most $k-1$ zeros. We will use Reed-Solomon codes over $\mathbb{F}_q$ for $q = 2^h$ a power of 2. These codes can be turned into binary codes by concatenating them with a Hadamard code. The *Hadamard* code $\mathcal{H}[q^n, n, q^{n-1}(q-1)]_q$ is the linear code consisting of all codewords obtained by evaluating any multilinear polynomial $p(\mathbf{x}) \in \mathbb{F}_q[x_1,\ldots,x_n]$ at all inputs $\mathbf{x} \in \mathbb{F}_q^n$. For any nonzero multilinear polynomial p, the value $p(\mathbf{x})$ is uniformly distributed over $\mathbb{F}_q$ when $\mathbf{x} \in \mathbb{F}_q^n$ is chosen uniformly at random. It follows that all nonzero codewords of the Hadamard code have Hamming norm $q^n(1-\frac{1}{q}) = q^{n-1}(q-1)$. The *concatenation* of $\mathcal{RS}[q^n, k, q^n - k + 1]_{q^h}$ and $\mathcal{H}[q^n, n, q^{n-1}(q-1)]_q$ is the code obtained by viewing the elements of the Reed-Solomon alphabet $\mathbb{F}_{q^n}$ as vectors in $\mathbb{F}_q^n$, and mapping them to corresponding Hadamard codewords. The result of concatenating $\mathcal{RS}[q^n, k, q^n - k + 1]_{q^h}$ and $\mathcal{H}[q^n, n, q^{n-1}(q-1)]_q$ is a linear code with parameters $\mathcal{H} \circ \mathcal{RS}[q^{2n}, kn, (q^n - k + 1)(q-1)q^{n-1}]_q$.

Computational Problems For simplicity in the rest of the paper we focus on binary codes, i.e., linear codes over the alphabet $\mathbb{F}_2$. Approximation versions of the minimum distance and nearest codeword problems are formally defined as follows.

Definition 1 *For any approximation factor $\gamma \geq 1$, the Minimum Distance Problem (MDP$_\gamma$), given a basis matrix $\mathbf{V} \in \mathbb{F}_2^{n\times k}$ and a distance bound d, asks to distinguish between the following two cases:*

1. *There exists a nonzero codeword $\mathbf{V}\mathbf{x}$ ($\mathbf{x} \in \mathbb{F}_2^k \setminus \{\mathbf{0}\}$) of norm $\|\mathbf{V}\mathbf{x}\|_H \leq d$.*

2. *All nonzero codewords $\mathbf{V}\mathbf{x}$ ($\mathbf{x} \in \mathbb{F}_2^k \setminus \{\mathbf{0}\}$) have norm at least $\|\mathbf{V}\mathbf{x}\|_H > \gamma d$.*

Definition 2 *For any approximation factor $\gamma \geq 1$, the Nearest Codeword Problem (NCP$_\gamma$), given a basis $\mathbf{B} \in \mathbb{F}_2^{n\times k}$, a target $\mathbf{y} \in \mathbb{F}_2^n$ and a distance bound t, asks to distinguish between the following two cases:*

1. *There exists a codeword* $\mathbf{Bz}$ *(*$\mathbf{z} \in \mathbb{F}_2^k$*) within distance* $\|\mathbf{Bz} - \mathbf{y}\|_H \leq t$ *from the target.*
2. *All codewords* $\mathbf{Bz}$ *(*$\mathbf{z} \in \mathbb{F}_2^n$*) are at distance at least* $\|\mathbf{Bz} - \mathbf{y}\|_H > \gamma t$ *from the target.*

As shown in [6], the NCP is easily reduced to MDP using, as a gadget, a locally dense code satisfying the following definition.

Definition 3 *A locally dense (binary) code with parameters* (m, k, d, r) *is a tuple* $(\mathbf{L}, \mathbf{s}, \mathbf{T})$ *where* $\mathbf{L} \in \mathbb{F}_2^{h \times m}$ *is the basis of an* m *dimensional code with minimum distance (at least)* d*, and* $\mathbf{T} \in \mathbb{F}_2^{k \times h}$ *is a linear transformation such that* $\mathbf{T}(\mathcal{B}_H(\mathbf{s}, r) \cap \mathbf{L}\mathbb{F}_2^m) = \mathbb{F}_2^k$*.*

Notice that, as needed by [6], the above definition requires not only an explicit ball $\mathcal{B}_H(\mathbf{s}, r)$ containing at least 2^k codewords, but also a linear map $\mathbf{T}$ that can be used to index all 2^k vectors $\mathbf{z} \in \mathbb{F}_2^k$ by a codeword in $\mathcal{B}_H(\mathbf{s}, r)$. For completeness, we recall how [6] uses this gadget to reduce NCP to MDP.[4]

Theorem 1 ([6]) *Let* $(\mathbf{L}, \mathbf{s}, \mathbf{T})$ *be a locally dense binary code with parameters* (m, k, d, r) *for some* $d > r$*. Then, for any* $\gamma < \frac{d}{r}$ *and* $\gamma' > (\frac{1}{\gamma} - \frac{r}{d})^{-1}$*, there is a deterministic polynomial time reduction (using the locally dense code as auxiliary input) from* NCP$_{\gamma'}$ *on* k*-dimensional codes to* MDP$_{\gamma}$ *on* $(m+1)$*-dimensional codes.*

Proof Let $(\mathbf{B}, \mathbf{y}, t)$ be an NCP input instance, where $\mathbf{B} \in \mathbb{F}_2^{n \times k}$. The output of the reduction is the basis

$$\mathbf{V} = \left[\begin{array}{cc} \mathbf{u}_a \otimes \mathbf{BTL} & -\mathbf{u}_a \otimes \mathbf{y} \\ \mathbf{u}_b \otimes \mathbf{L} & -\mathbf{u}_b \otimes \mathbf{s} \end{array} \right] \in \mathbb{F}_2^{(an+bh)\times(m+1)}$$

together with the distance bound $d' = at + br$, where a and b are integer scaling factors such that a/b is a rational number in the interval

$$\frac{\gamma r}{(\gamma' - \gamma)t} \leq \frac{a}{b} < \frac{d - r\gamma}{\gamma t}.$$

Notice that this interval is always nonempty under the conditions on γ, γ' stated in the theorem, so the scaling factors a, b can be properly selected. The columns of $\mathbf{V}$ are linearly independent because $\mathbf{L}$ is a basis and $\mathbf{s} \notin \mathbf{L}\mathbb{F}_2^n$. So, $\mathbf{u}_b \otimes [\mathbf{L}, -\mathbf{s}]$ has full column rank and $\mathbf{V}$ is a basis of an $(m+1)$-dimensional code. We need to analyze the minimum distance of the code generated by $\mathbf{V}$.

If $(\mathbf{B}, \mathbf{y}, t)$ is a positive instance of NCP, then there exists a vector $\mathbf{z} \in \{0,1\}^k$ such that $\|\mathbf{Bz} - \mathbf{y}\|_H \leq t$. Moreover, by Definition 3, there is a vector $\mathbf{x}$ such that $\|\mathbf{Lx} - \mathbf{s}\|_H \leq r$ and $\mathbf{TLx} = \mathbf{z}$. It follows that there exists a nonzero codeword $\mathbf{V}(\mathbf{x}, 1)$ of norm at most

$$\|\mathbf{V}(\mathbf{x}, 1)\|_H = \|(\mathbf{u}_a \otimes (\mathbf{Bz} - \mathbf{y}), \mathbf{u}_b(\mathbf{Lx} - \mathbf{s}))\|_H = a \cdot \|\mathbf{Bz} - \mathbf{y}\|_H + b \cdot \|\mathbf{Lx} - \mathbf{s}\|_H \leq at + br = d'.$$

Now assume $(\mathbf{B}, \mathbf{y}, t)$ is a negative NCP instance. Consider any nonzero codeword $\mathbf{V}(\mathbf{x}, c)$ generated by $\mathbf{V}$. We need to prove that $\|\mathbf{V}(\mathbf{x}, c)\|_H > \gamma d'$. We distinguish two cases depending on the value of c. If $c = 0$, then $\mathbf{x} \neq \mathbf{0}$ and

$$\|\mathbf{V}(\mathbf{x}, 0)\|_H = \|(\mathbf{u}_a \otimes \mathbf{BTLx}, \mathbf{u}_b \otimes \mathbf{Lx})\|_H \geq \|\mathbf{u}_b \otimes \mathbf{Lx}\|_H \geq bd > \gamma d'$$

[4] The original paper [6] contains a more general sequence of reductions that proves the hardness of various other intermediate problems and applies to linear codes over possibly larger fields. For simplicity, here we report a stripped down proof specialized to the case of binary codes.

where the last inequality follows from the upper bound on $\frac{a}{b}$. On the other hand, if $c = 1$ then using $\|\mathbf{Bz} - \mathbf{y}\|_H > \gamma' t$ for $\mathbf{z} = \mathbf{TLx}$ we get

$$\|\mathbf{V}(\mathbf{x}, 1)\|_H = \|(\mathbf{u}_a \otimes (\mathbf{BTLx} - \mathbf{y}), \mathbf{u}_b \otimes (\mathbf{Lx} - \mathbf{s}))\|_H \geq a \cdot \|(\mathbf{Bz} - \mathbf{y})\|_H > \gamma' at \geq \gamma d'$$

where the last inequality follows from the lower bound on $\frac{a}{b}$. This shows that all nonzero codewords have norm bigger than $\gamma d'$ and concludes the analysis of the reduction. □

Since NCP_γ is NP-hard for any constant factor $\gamma \geq 1$ [2], Theorem 1 immediately implies that if locally dense codes can be efficiently constructed, then MDP_γ is also NP-hard, at least for factors $\gamma < d/r$. As long as we can build locally dense codes with $r/d < 1 - \epsilon$ for some constant $\epsilon > 0$, Theorem 1 proves the NP-hardness of approximating MDP_γ within *some* constant factor $\gamma > 1$. Inapproximability of MDP_γ for *any* constant factor γ also follows, using the fact that $\mathrm{dist}_H(\mathbf{V} \otimes \mathbf{V}) = \mathrm{dist}_H(\mathbf{V})^2$. (See [6] for details.) In summary, in order to prove that MDP_γ is NP-hard for any constant factor γ it is enough to give an efficient construction of locally dense codes with $r/d < 1 - \epsilon$.

3 Main Result

In this section we describe a general method to construct locally dense binary codes, satisfying Definition 3, starting from arbitrary codes with sufficiently large minimum distance. We remark that for the following theorem to yield useful locally dense codes (with relative radius $r/d < 1$) we need to start from linear codes with relative minimum distance at least $d/n > 1/\sqrt{6}$. In Corollary 1 we show how to instantiate the theorem with codes with relative minimum distance arbitrarily close to $d/n \approx 1/2$, resulting in locally dense codes with relative radius arbitrarily close to $r/d \approx 2/3$.

Theorem 2 *There is a polynomial time algorithm that on input a linear code $\mathcal{C}[n, k, d]_2$ with $d \leq n/2$, outputs a locally dense binary code $(\mathbf{L}, \mathbf{s}, \mathbf{T})$ with parameters $(k(k+1)/2, k, 6d^2, n^2)$.*

More specifically, the construction produces a linear code $\mathcal{L}[4n^2, k(k+1)/2, 6d^2]_2$, a target vector $\mathbf{s} \in \{0,1\}^{4n^2}$ and a set of coordinates T of size k such that for any vector $\mathbf{x} \in \mathbb{F}_2^k$ there is a codeword at distance exactly n^2 from $\mathbf{s}$ whose restriction to T equals $\mathbf{x}$.

Proof Let $\mathbf{C} \in \mathbb{F}_2^{n \times k}$ be a generating matrix for $\mathcal{C}$, and assume $\mathbf{C}$ is in systematic form, i.e., its first k rows equal the identity matrix $\mathbf{I}$. The code $\mathcal{L}$ is the set of all $n \times (4n)$ matrices

$$\mathbf{W} = [\mathbf{Y}, \mathbf{Y} + \mathbf{u}\mathbf{y}^T, \mathbf{Y} + \mathbf{y}\mathbf{u}^T, \mathbf{Y} + \mathbf{u}\mathbf{y}^T + \mathbf{y}\mathbf{u}^T]$$

where $\mathbf{Y} = \mathbf{C}\mathbf{X}\mathbf{C}^T$ for some symmetric matrix $\mathbf{X} = \mathbf{X}^T \in \mathbb{F}_2^{k \times k}$, $\mathbf{y}$ is the diagonal of $\mathbf{Y}$ and $\mathbf{u}$ is the all one vector. The resulting code is linear (and a basis $\mathbf{L}$ can be easily computed using linear algebra) because all entries of the codeword $\mathbf{W}$ are linear functions of the entries of $\mathbf{X}$, and the matrix $\mathbf{X}$ ranges over a linear ($k(k+1)/2$-dimensional) subspace of $\mathbb{F}_2^{k \times k}$. Moreover, since codewords are in one-to-one correspondence with $\mathbf{X}$, the code $\mathcal{L}$ has rank $k(k+1)/2$. The center of the Hamming ball is

$$\mathbf{S} = [\mathbf{O}, \mathbf{O}, \mathbf{O}, \mathbf{u}\mathbf{u}^T]$$

and $\mathbf{T}$ is the projection matrix corresponding to the set of coordinates $T = \{(i,i) \mid 1 \leq i \leq k\}$, i.e., $\mathbf{T}$ maps each codeword $\mathbf{W}$ to the k-dimensional vector obtained by taking the first k coordinates along the diagonal of $\mathbf{Y}$. The locally dense code is given by $(\mathbf{L}, \mathbf{S}, \mathbf{T})$.

We first show that for any $\mathbf{x} \in \{0,1\}^k$, we can build a codeword $\mathbf{W}$ at distance $r = n^2$ from $\mathbf{S}$, and which equals $\mathbf{x}$ at positions T. Let $\mathbf{X} = \mathbf{x}\mathbf{x}^T$, and consider the associated value of $\mathbf{Y} = \mathbf{C}\mathbf{X}\mathbf{C}^T = (\mathbf{C}\mathbf{x})(\mathbf{C}\mathbf{x})^T$. Notice that if we let $\mathbf{y} = \mathbf{C}\mathbf{x}$, then the diagonal of $\mathbf{Y} = \mathbf{y}\mathbf{y}^T$ equals $\mathbf{y}$, because $y_i^2 = y_i$ in $\mathbb{F}_2$. Moreover, the first k elements of the diagonal $\mathbf{y}$ equal $\mathbf{x}$ because $\mathbf{C}$ is systematic and $\mathbf{y} = \mathbf{C}\mathbf{x}$. So, $\mathbf{T}$ projects $\mathbf{W}$ back to $\mathbf{x}$. We need to compute the distance of $\mathbf{W}$ to the center $\mathbf{S}$ of the ball, i.e., Hamming weight of

$$\begin{aligned}\mathbf{W} - \mathbf{S} &= [\mathbf{y}\mathbf{y}^T, \mathbf{y}\mathbf{y}^T + \mathbf{y}\mathbf{u}^T, \mathbf{y}\mathbf{y}^T + \mathbf{u}\mathbf{y}^T, \mathbf{y}\mathbf{y}^T + \mathbf{y}\mathbf{u}^T + \mathbf{u}\mathbf{y}^T + \mathbf{u}\mathbf{u}^T] \\ &= [\mathbf{y}\mathbf{y}^T, \mathbf{y}(\mathbf{y}+\mathbf{u})^T, (\mathbf{y}+\mathbf{u})\mathbf{y}^T, (\mathbf{y}+\mathbf{u})(\mathbf{y}+\mathbf{u})^T].\end{aligned}$$

At this point we simply observe that for any $1 \le i, j \le n$, exactly one of the four square matrices in the last expression equals 1 at position (i,j). E.g., if $y_i = 0$ and $y_j = 1$, then only $(\mathbf{y}+\mathbf{u})\mathbf{y}^T$ equals 1 at position (i,j). Similarly, for the other values of y_i, y_j. So, the distance of $\mathbf{W}$ to $\mathbf{S}$ is precisely n^2.

It remains to bound the minimum distance of the code $\mathcal{L}$, i.e., the minimum weight of nonzero codewords $\mathbf{W}$. Consider an arbitrary nonzero codeword, defined by $\mathbf{X}, \mathbf{Y}, \mathbf{y}$ as above. We distinguish two cases, depending on whether $\mathbf{y}$ is identically 0.

First assume $\mathbf{y} = \mathbf{0}$. Then $\mathbf{W}$ consists of 4 identical copies of $\mathbf{Y}$, and all we need to do is to show that $\mathbf{Y}$ has weight at least $6d^2/4 = (3/2)d^2$. We will show that $\mathbf{Y}$ has at least $(3/2)d$ nonzero columns. Since the columns of $\mathbf{Y} = \mathbf{C}(\mathbf{X}\mathbf{C}^T)$ are codewords of C, their weight (when nonzero) is at least d, and the bound on the total weight of $\mathbf{Y}$ immediately follows. In order to bound the number of nonzero columns, we notice that $\mathbf{Y}$ must contain at least two distinct nonzero rows, because it is symmetric and has zero diagonal. Let $\mathbf{a}^T$ and $\mathbf{b}^T$ be two such rows, and

- let a be the number of coordinates such that $a_i = 1$ and $b_i = 0$,
- let b be the number of coordinates such that $a_i = 0$ and $b_i = 1$,
- let c be the number of coordinates such that $a_i = 1$ and $b_i = 1$.

The rows of $\mathbf{Y}$ are also codewords in C, so $\mathbf{a}$, $\mathbf{b}$ and their sum $\mathbf{a}+\mathbf{b}$ must have weight at least d. So, $a + c = \|\mathbf{a}\| \ge d$, $b + c = \|\mathbf{b}\| \ge d$ and $a + b = \|\mathbf{a}+\mathbf{b}\| \ge d$. Adding up we get $2a+2b+2c \ge 3d$. But the number of nonzero columns of $\mathbf{Y}$ is at least as large as the number of coordinates such that either $\mathbf{a}$ or $\mathbf{b}$ (or both) are nonzero. So, the number of nonzero columns is at least $a+b+c \ge (3/2)d$.

Now consider any codeword $\mathbf{W}$ such that the diagonal $\mathbf{y}$ is nonzero. For any position (i,j) in $\mathbf{Y}$, the codeword $\mathbf{W}$ has 4 entries that equal $y_{i,j}, y_{i,j}+y_i, y_{i,j}+y_j, y_{i,j}+y_i+y_j$. It is easy to see that unless $(y_i, y_j) = (0,0)$, at least (in fact, exactly) 2 of these four entries are 1. So, the weight of $\mathbf{W}$ is at least $2(n^2 - (n-w)^2) = 2w(2n-w)$, where w is the Hamming weight of $\mathbf{y}$. For any symmetric matrix $\mathbf{X}$ with diagonal $\mathbf{x}$, it can be easily verified that the diagonal of $\mathbf{Y} = \mathbf{C}\mathbf{X}\mathbf{C}^T$ equals $\mathbf{C}\mathbf{x}$. In particular, $\mathbf{y}$ is a nonzero codeword of C, and has weight $w = \|\mathbf{y}\| \ge d$. Since $d \le w \le n$, our lower bound $2w(2n-w)$ on the weight of $\mathbf{W}$ is at least $2d(2n-d)$, which, for $d \le n/2$ is at least $6d^2$. □

Some remarks about the above proof are in order. First, notice that the code in the construction is somehow redundant. For example, the two middle matrices in $\mathbf{W}$ are always zero on the diagonal, so we could reduce the dimension of the code by $2n$. Moreover, the two middle matrices are one the transpose of the other, the first and fourth matrix are symmetric, and they are equal on the diagonal. So, every entry is repeated twice. We could remove the duplication, and reduce the dimension of the code by a factor 2, from $4n^2$ down to $2n^2 - n$. However, the center of the

Hamming ball does not have the property that the first and fourth matrix have the same diagonal, so this slightly complicates the analysis.

As a last remark, the assumption $d \geq n/2$ was only used at the end to argue that $2d(2n-d) \geq 6d^2$. The construction works also for codes with $d \geq n/2$, in which case the minimum distance of W is $2d(2n-d)$, rather than d^2. However, codes with $d \geq n/2$ necessarily have very low rate.

In the next corollary we instantiate Theorem 2 with binary codes $\mathcal{C}[n,k,d]_2$ with minimum distance d close to $n/2$.

Corollary 1 *For any $\rho < 3/2$, there is an efficient deterministic polynomial time algorithm that on input k outputs a locally dense binary code (satisfying Definition 3) with parameters (m,k,d,r) for some $m = poly(k)$ and $d/r > \rho$.*

Proof The algorithm, on input k, builds a linear code $\mathcal{C}[n,k,d]_2$ with minimum distance $d \approx n/2$ and then invokes Theorem 2 to obtain a locally dense code with parameters (m,k,d,r) where $m = k(k+1)/2$, $r = n^2$ and $d' = 6d^2 > \rho r$. By taking d arbitrarily close to $n/2$, we can obtain any $\rho < 3/2$. The code $\mathcal{C}[n,k,d]_2$ can be constructed as follows. (See Section 2 for a description of Reed-Solomon and Hadamard codes.) Let $q = 2^h$ be the smallest power of 2 such that $q \log_2 q = qh \geq k/\epsilon$. Take a Reed-Solomon code $\mathcal{RS}[q,k',d]_q$ of rate $k' = \lceil k/h \rceil$ and minimum distance $d = q-k'+1 > q-k'$. Concatenate the Reed-Solomon code with an equidistant Hadamard code $\mathcal{H}[q,h,q/2]_2$. The result is a binary code $\mathcal{C}[q^2,k'h,d]_2$ with rate $k'h \geq k$, and minimum distance $d = (q+1-k')q/2 > (q^2/2)(1-k'/q) \geq (q^2/2)(1-\epsilon)$ arbitrarily close to $n/2 = q^2/2$. □

4 Lattices and Open Problems

As described in the introduction, interest in explicit constructions of locally dense codes is motivated by the study of the computational complexity of problems analogous to NCP and MDP, but on integer lattices rather than linear codes over finite fields. For a detailed introduction to lattices and the Shortest Vector Problem the reader is referred to [16, 15] and references therein. An integer lattice is the set $\mathbf{B}\mathbb{Z}^k$ of all integer linear combinations of k linearly independent integer vectors $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_k] \in \mathbb{Z}^{n \times k}$. The Shortest Vector Problem (SVP) is defined analogously to MDP, and asks, given a lattice basis $\mathbf{B}$, to find a nonzero lattice vector $\mathbf{Bx}$ ($\mathbf{x} \in \mathbb{Z}^k \setminus \{\mathbf{0}\}$) of approximately smallest length. SVP and all other lattice problems described in this section can be defined with respect to any norm $\|\cdot\|$, but the Euclidean norm $\|\mathbf{x}\| = \sqrt{\sum_i |x_i|^2}$ is the most important and more commonly used one. NP-hardness of SVP is proved by reduction from (variants of) the Closest Vector Problem (CVP), which is analogous to NCP, but for integer lattices rather than linear codes. In fact, a reduction similar to the one described in Theorem 1, but for integer lattices, was originally given in [14] to prove the inapproximability of SVP. (See also [15] for a more recent, still similar, reduction.) Here we give the formal definition of locally dense lattices, which are the gadgets needed by the reduction in [14] to prove the NP-hardness of SVP.

Definition 4 *A locally dense lattice with parameters (m,k,d,r) is a tuple $(\mathbf{L},\mathbf{s},\mathbf{T})$ where $\mathbf{L} \in \mathbb{Z}^{h \times m}$ is a basis of an m-dimensional integer lattice with minimum distance (at least) d, and $\mathbf{T}\colon (\mathbf{L}\mathbb{Z}^m) \to \mathbb{Z}^k$ is a linear transformation such that $\mathbf{T}(\mathcal{B}_H(\mathbf{s},r) \cap \mathbf{L}\mathbb{Z}^m) \supseteq \{0,1\}^k$.*

Constructions of locally dense lattices satisfying Definition 4 are given in [14, 15], but they are probabilistic, resulting in randomized NP-hardness proofs for SVP. We remark that for the ℓ_∞ (or "max") norm $\|\mathbf{x}\|_\infty = \max_i |x_i|$, locally dense lattices are very easy to build, even deterministically. The locally dense lattice described in the following theorem is implicit in the proof from [18] that SVP in the ℓ_∞ norm is NP-hard to solve exactly. In fact, the construction in Theorem 3 also implies that SVP in the ℓ_∞ norm is NP-hard to *approximate* within any factor $\gamma \leq 2$.

Theorem 3 *For any integer $k \geq 1$, there is an efficiently constructible locally dense lattice with respect to the ℓ_∞ norm with parameters $(k, k, 2, 1)$.*

Proof Let $\mathbf{L} = 2\mathbf{I}$ be a basis for $2\mathbb{Z}^k$, $\mathbf{s} = \mathbf{u}_k$ the all one vector, and $\mathbf{T} = \frac{1}{2}\mathbf{I}$ the scalar transformation mapping $2\mathbb{Z}^k$ to $\mathbb{Z}^k$. Clearly, the lattice generated by $\mathbf{L}$ has ℓ_∞ minimum distance 2. Moreover, for any $\mathbf{x} \in \{0,1\}^k$, the lattice vector $\mathbf{Lx} = 2\mathbf{x}$ is at ℓ_∞ distance 1 from $\mathbf{s}$, and it is mapped back to $\mathbf{T}(\mathbf{Lx}) = \mathbf{x}$ by the linear transformation $\mathbf{T}$. □

However, for the standard Euclidean (ℓ_2) norm, no deterministic construction of locally dense lattices is known yet,[5] and the probabilistic constructions of [14, 15] only prove the NP-hardness of SVP (in the ℓ_2 norm) under randomized reductions. Finding a deterministic construction satisfying Definition 4 (as done in Theorems 2 and 3 for the Hamming and ℓ_∞ norms) is an interesting open problem, as it would derandomize the NP-hardness proofs of [14, 15] and show that SVP is NP-hard to approximate (within some constant factor)[6] under *deterministic* polynomial time reductions.

We conjecture that locally dense lattices with respect to the Euclidean norm can be efficiently constructed.

Conjecture 1 *There is a deterministic algorithm that on input an integer $k \geq 1$, outputs (in time polynomial in k) a locally dense lattice $(\mathbf{L}, \mathbf{s}, \mathbf{T})$ with parameters (m, k, d, r) satisfying Definition 4 with respect to the Euclidean norm with $r < d(1 - \epsilon)$ for some constant $\epsilon > 0$.*

We see several ways in which the conjecture may eventually get positively resolved, thereby proving the NP-hardness of approximating SVP under deterministic polynomial time reductions. We remark that a candidate construction of locally dense lattices in the Euclidean norm was given in [14] (see also [16]), and proved correct, conditionally, assuming a certain number theoretic conjecture on the distribution of smooth numbers. Namely, [14] conjectures that for any ϵ there is a c such that for all sufficiently large n, the interval $[n, n + n^\epsilon]$ contains a square free integer with no prime factors larger than $\log^c(n)$. While the conjecture is quite plausible (and may even hold for $\epsilon = O(\log\log n / \log n) = o(1)$), proving it seems very challenging, and other methods to resolve Conjecture 1 may be more successful. Our work suggests two other possible approaches to Conjecture 1.

1. One possibility is to prove a result similar to our Theorem 2, but for lattices. Namely, showing that there is an efficient algorithm to transform an arbitrary input lattice with sufficiently large minimum distance into a locally dense lattice containing exponentially many points nearby a known vector $\mathbf{s}$.

[5] But see [14] for a deterministic construction under a plausible, but unproven, number theoretic conjecture on the distribution of smooth numbers.

[6] Amplifying the inapproximability factor via tensor products, and proving NP-hardness for *any* constant factor is much trickier for lattices than for codes. See [15] for a detailed discussion.

2. Another possibility is to use Theorem 2 as it is, and then use the resulting locally dense binary codes to build a locally dense lattice. In fact, the latest randomized constructions of locally dense lattices [15] use techniques from coding theory, and build the locally dense lattice by combining together several binary codes with large minimum distance. However, [15] first combines the codes with large minimum distance to build a globally dense lattice, and then finds a dense ball $\mathcal{B}(\mathbf{s}, r)$ by choosing its center $\mathbf{s}$ at random. So, one may try to reverse the order of these two steps: first using Theorem 2 to turn each binary code into a locally dense code with known dense center, and then combining the binary codes together into a lattice containing a large number of points nearby a known fixed vector.

As a final note, proving Conjecture 1 would prove that SVP_γ is NP-hard only for *some* constant factor $\gamma < d/r$. Due to the differences between the Hamming and Euclidean norms, and their behavior with respect to the tensor product operation, proving hardness for *any* constant factor in the case of lattice problems is quite a bit trickier than for linear codes, and requires some additional properties. Still, proving NP-hardness of SVP under deterministic reduction for some constant factor would already be a good starting point. We refer the reader to [15, 8] for details on how to use tensor product operation in the computational study of SVP.

References

[1] M. Ajtai. The shortest vector problem in l_2 is NP-hard for randomized reductions (extended abstract). In *Proceedings of STOC*, 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] 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 '98.

[4] Q. Cheng and D. Wan. A deterministic reduction for the gap minimum distance problem. *IEEE Transactions on Information Theory*, 58(11):6935–6941, 2012.

[5] 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 '98.

[6] 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.

[7] O. Goldreich, D. Micciancio, S. Safra, and J.-P. Seifert. Approximating shortest lattice vectors is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55–61, 1999.

[8] 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.

[9] R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of operation research*, 12(3):415–440, Aug. 1987. Preliminary version in STOC '83.

[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 '04.

[11] S. Khot and P. Austrin. A simple deterministic reduction for the gap minimum distance of code problem. In *Proceedings of ICALP*, pages 474–485, 2011.

[12] A. K. Lenstra, H. W. Lenstra, Jr., and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:513–534, 1982.

[13] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, Nov. 1983.

[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 '98.

[15] D. Micciancio. Inapproximability of the shortest vector problem: Toward a deterministic reduction. *Theory of Computing*, 8(1):487–512, 2012.

[16] 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.

[17] D. Micciancio and O. Regev. Lattice-based cryptography. In *Post-quantum cryptography*. Springer, 2008.

[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, University 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.